MINERAÇÃO DE REGRAS PARA CLASSIFICAÇÃO DE ONCOGENES MEDIDOS POR *MICROARRAY* UTILIZANDO ALGORITMOS GENÉTICOS

Por

Laurence Rodrigues do Amaral

DISSERTAÇÃO APRESENTADA À
UNIVERSIDADE FEDERAL DE UBERLÂNDIA, MINAS GERAIS,
COMO PARTE DOS REQUISITOS EXIGIDOS
PARA OBTENÇÃO DO TÍTULO DE MESTRE
EM CIÊNCIA DA COMPUTAÇÃO

AGOSTO DE 2007

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

FACULDADE DE COMPUTAÇÃO

Os abaixo assinados, por meio deste, certificam que leram e recomendam para a Faculdade de Computação a aceitação da dissertação intitulada "**MINERAÇÃO DE REGRAS PARA CLASSIFICAÇÃO DE ONCOGENES MEDIDOS POR** *MICROARRAY* **UTILIZANDO ALGORITMOS GENÉTICOS**" por **Laurence Rodrigues do Amaral** como parte dos requisitos exigidos para a obtenção do título de **Mestre em Ciência da Computação**.

Uberlândia, 13 de Agosto de 2007

Orientadora: _____

Profª. Drª. Gina Maira Barbosa de Oliveira

Universidade Federal de Uberlândia UFU/MG

Banca Examinadora: _____

Profª. Drª. Denise Guliato

Universidade Federal de Uberlândia UFU/MG

_____

Prof. Dr. Alexandre Cláudio Botazzo Delbem

Universidade de São Paulo USP/SP

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Data: Agosto, 2007

Autor: **Laurence Rodrigues do Amaral**

Título: **MINERAÇÃO DE REGRAS PARA CLASSIFICAÇÃO DE ONCOGENES MEDIDOS POR *MICROARRAY* UTILIZANDO ALGORITMOS GENÉTICOS**

Faculdade: **Faculdade de Computação**

Grau: **Mestrado**

_____

Autor

Dedicatória

À minha esposa Kyara, meus pais Ademir e Laurita e a meu irmão Lucas

Agradecimentos

Agradeço primeiramente a Deus por ter me agraciado com a oportunidade de estudar, oportunidade esta, tão rara e difícil nos dias atuais, e saúde para ter completado mais este passo nesta longa caminhada que é o saber.

A minha esposa Kyara, companheira de todas as horas, que soube entender que esta conquista não é apenas minha, mas sim de toda a nossa família e esteve a meu lado nas horas boas e principalmente nas horas ruins.

A toda minha família pelo apoio, incentivo e por terem acreditado em mim.

A minha orientadora Gina, pessoa pelo qual tenho profundo respeito, por ter me ajudado a chegar até aqui, pessoa esta, exemplo de responsabilidade e competência.

A todos os amigos do Unicerp, que direta ou indiretamente, tiveram participação nesta conquista.

A todos vocês, o meu muito obrigado.

Resumo

Técnicas de Inteligência Artificial (IA) têm se tornado cada vez mais importantes na solução de problemas biológicos. Nesta dissertação, utilizamos um Algoritmo Genético (AG) na busca de regras de alto nível do tipo IF-THEN. Este AG foi aplicado na mineração de regras de classificação em uma base de dados de expressão gênica de células cancerígenas (NCI60), advindas de experimentos de *microarray*. O objetivo dessa mineração é descobrir relações entre os níveis de expressões gênicas e os nove tipos de classes de câncer analisados neste trabalho.

Palavras chave: Bioinformática, expressão gênica, algoritmos genéticos, oncogenes, *data mining*.

Abstract

Artificial Intelligence techniques are increasing their role in the solution of biological problems. The present study use a Genetic Algorithm (GA) in the search for high level IF-THEN rules. This GA was applied to miner classification rules from a gene expression database named NCI60. This database was developed using cancer cells measured by microarray. The goal of this mining is the discovery of relations among gene expression level and the nine types of cancer classes analyzed in this work.

Keywords: Bioinformatic, gene expression, genetic algorithms, oncogenes, *data mining*.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Atualmente, a bioinformática é imprescindível para a manipulação dos dados biológicos. Ela pode ser definida como uma modalidade que abrange todos os aspectos de aquisição, processamento, armazenamento, distribuição, análise e interpretação da informação biológica. Através da combinação de procedimentos e técnicas advindos da matemática, da estatística e da ciência da computação, são elaboradas várias ferramentas que auxiliam a compreender o significado biológico representado nos dados genômicos [4]. Uma das áreas em que a aplicação de técnicas computacionais tem se mostrado mais promissora é a Biologia Molecular [5]. O termo expressão gênica refere-se ao processo em que a informação codificada por um determinado gene é decodificada em uma proteína, manisfestando assim, características particulares àquele gene. As células e tecidos têm suas funções normais quando os genes são expressos de forma regulada. A expressão alterada de um gene pode alterar o equilíbrio do organismo, podendo vir a gerar uma doença. Assim, a seleção de genes relevantes a uma determinada doença torna-se uma tarefa importantíssima, podendo num futuro próximo, ser aplicada no diagnóstico médico. Na busca destes pequenos conjuntos de genes preditores, técnicas advindas da Inteligência Artificial (IA), tais como, os algoritmos genéticos e as redes neurais artificiais, são cada vez mais empregados, devido a sua capacidade de aprender automaticamente a partir de grandes volumes de dados e produzir hipóteses úteis [6].

Diferentes técnicas de IA foram aplicadas na análise de dados de expressão gênica, tais como, as redes neurais artificiais [7, 8], as *support vector machines* [9, 10] e os algoritmos

genéticos [11, 3, 12, 13, 14, 15, 16, 17, 18]. Em todos os projetos citados anteriormente, o objetivo é encontrar conjuntos de genes (*clusters*) que possam ser utilizados como classificadores confiáveis, com uma elevada taxa de classificação e um bom desempenho de generalização. Dessa forma, os conjuntos minerados podem auxiliar na classificação de novos casos, facilitando o diagnóstico e o tratamento de doenças. Entretanto, somente em [16, 17, 18], encontramos classificadores baseados em regras de alto nível, por exemplo, regras do tipo IF-THEN. Nos demais, os classificadores obtidos são do tipo caixa-preta, onde a entrada são os dados de expressão de uma determinada amostra de células e a saída é a classe à qual essa amostra provavelmente pertence, podendo esta saída estar associada, por exemplo, a uma classe de doença. Assim, a partir de um conjunto de dados de milhares de genes chega-se a um pequeno conjunto de poucas dezenas de genes que sejam discriminantes para o problema.

1.1 Objetivos

O enfoque desse trabalho foi na busca (mineração) de regras de alto nível, que não só estivessem associadas a cada classe individualmente, reduzindo o problema a poucos genes por classe, mas também associando o nível de expressão gênica a cada gene que compõe a regra. Acreditamos que esse tipo de informação possa ser de grande utilidade aos especialistas que buscam entender o mecanismo responsável pelas alterações nos padrões de expressão gênica associadas ao aparecimento de determinadas doenças. Para tal, elaborou-se um algoritmo genético (AG) para a obtenção de regras do tipo IF-THEN a partir de bases de dados de expressões gênicas. O AG foi fortemente inspirado no modelo proposto por Fidelis e colaboradores [19] para a mineração de regras de classificação. O ambiente evolutivo implementado foi aplicado na classificação de uma base de dados de expressões gênicas de células cancerígenas, advindas de experimentos de *microarray*. Esta base é de domínio público e é conhecida como NCI60 [20]. O principal objetivo do nosso trabalho é a busca das relações entre os níveis de expressões gênicas de nove classes de câncer: mama, sistema nervoso central, cólon, leucemia, melanoma, pulmão, ovário, renal e células reprodutivas. Na base NCI60 [20] foram obtidas expressões de mais de 8.000

genes para 61 amostras de células. Diversos trabalhos aplicaram diferentes técnicas na busca de conjuntos de genes preditores para esta base [21, 3, 13, 22, 23, 24, 25]. Nesse trabalho, como ponto de partida, utilizamos quatro conjuntos reduzidos de genes que foram minerados por Ooi e Tan [3] a partir da NCI60, totalizando 55 genes.

1.2 Organização do Trabalho

Esta dissertação está dividida em 6 capítulos, sendo o primeiro uma introdução sobre o trabalho e os objetivos propostos pelo mesmo.

O segundo capítulo apresenta alguns conceitos sobre biologia molecular, *microarrays* de DNA, bioinformática, trazendo também a descrição de alguns trabalhos aplicados à base NCI60.

O terceiro capítulo apresenta informações a respeito dos algoritmos genéticos, tais como: visão geral do método, representacão do indivíduo, operadores genéticos, seleção de pais, *crossover*, reinserção, dentre outros. Além destes tópicos relatados anteriormente, o capítulo 3 também aborda aplicações de algoritmos genéticos em tarefas de *datamining* e também na análise de dados advindos de expressão gênica.

O quarto capítulo descreve o ambiente evolutivo utilizado neste trabalho, ajustes que foram necessários para se chegar neste ambiente e as bases de dados que foram investigadas.

O quinto capítulo contempla os resultados obtidos no ambiente evolutivo proposto.

O sexto capítulo apresenta as conclusões do trabalho e as propostas de trabalhos futuros.

Capítulo 2

Biologia Molecular e Bioinformática

2.1 Biologia Molecular

Genética é o nome dado ao estudo da hereditariedade, o processo pelo qual as características são passadas dos genitores para a prole de modo que todos os organismos, inclusive os seres humanos, assemelhem-se a seus ancestrais. O conceito central da genética é que a hereditariedade é controlada por um grande número de fatores, os genes, que são pequenas partículas físicas presentes em todos os organismos vivos [10].

Os primeiros geneticistas estavam interessados principalmente em como os genes são transmitidos dos genitores à sua prole durante a reprodução e em características variáveis, tais como altura e cor dos olhos. Durante a década de 1930, a pesquisa tomou novos rumos ao reconhecer que se os genes são entidades físicas, assim como outros componentes da célula, eles devem ser feitos de moléculas e, portanto, deve ser possível estudá-los diretamente por métodos biofísicos e bioquímicos. Isso levou a um novo ramo da genética, chamado Biologia Molecular, que tinha como um de seus objetivos iniciais a identificação da natureza química do gene. Este novo enfoque levou a novos conceitos e os biólogos deixaram de considerar os genes simplesmente como unidades de herança, passando a encará-los como unidades de informação biológica, possuindo a quantidade total de informações necessárias para a construção de um exemplo vivo e funcional daquele organismo [10].

A compreenção científica nos dias de hoje da complexidade e do dinamismo celular

apóia-se nos trabalhos de milhares de cientistas nos últimos 150 anos. Os pesquisadores modernos fundiram conceitos e técnicas experimentais da bioquímica, da genética e da biologia molecular com aqueles da biologia celular clássica para produzirem uma concepção dinâmica da vida celular [26].

Os conhecimentos sobre as células progridem paralelamente ao aperfeiçoamento dos métodos de investigação. Inicialmente, o microscópio óptico possibitou o descobrimento das células e a elaboração da teoria de que todos os seres vivos são constituídos por células. Posteriormente, foram descobertas técnicas citoquímicas que possibitaram a identificação e localização de diversas moléculas constituintes das células. Com o advento dos microscópios eletrônicos, que têm grande poder de resolução, foram observados pormenores da estrutura celular que não poderiam sequer ser imaginados pelos estudos feitos com os microscópios ópticos. Com o uso dos microscópios eletrônicos, foram aperfeiçoados métodos para a separação de organelas celulares e para o estudo *in vitro* de suas moléculas e respectivas funções. A análise de organelas isoladas em grande quantidade, a cultura de células, a possibilidade de manipular o genoma através da adição ou supressão de genes e o aparecimento de numerosas técnicas de uso comum aos diversos ramos da pesquisa biológica levaram ao surgimento da biologia celular e molecular, que é o estudo integrado das células, através de todo o arsenal técnico disponível [27].

Um fragmento de DNA pode conter diversos genes. A propriedade mais importante dos genes está no fato de que eles contêm o código genético para a expressão do mRNA (RNA mensageiro) que será traduzido em proteínas, componentes estes, essenciais a todo ser vivo [28]. As proteínas são polipeptídeos compostos por conjuntos de aminoácidos. Estes aminoácidos são representados por trincas (códons) de nucleotídeos (Adenina - A, Uracila - U, Citosina - C e Guanina - G) no DNA. O processo pelo qual as seqüências de nucleotídeos dos genes são interpretados na produção de proteínas é denominado expressão gênica [28]. Mensurar e analisar informações de expressão gênica é de grande interesse para as Ciências Biológicas. Esse tipo de análise pode fornecer informações importantes sobre as funções de uma célula, uma vez que as mudanças na fisiologia de um organismo são geralmente acompanhadas por mudanças nos padrões de expressão dos genes [29]. Uma das técnicas mais difundidas para esta medição são os *Microarrays* de DNA [30, 31, 32, 33].

2.2 Experimentos de *microarrays* e bases de expressão gênica

O *microarray* de DNA é uma metodologia utilizada para comparar a expressão de um grande número de genes simultaneamente. Essa técnica emprega arranjos (*arrays*), que contêm um grande número de genes distribuídos por um braço robótico de forma ordenada (*spots*) sobre placas de vidro. A quantificação dos níveis de expressão gênica na tecnologia de *microarray* é baseada em experimentos onde os milhares de clones de cDNA [1] são hibridizados [2] com duas sondas marcadas com diferentes fluorecências (geralmente uma emite cor vermelha (Cy5) e outra verde (Cy3)). As sondas podem ser conjuntos de cDNAs gerados a partir de células ou tecidos em duas cituações diferentes, que se deseja comparar. Os resultados são produzidos sob forma de diferentes intensidades de fluorescência que são captadas por microscopia a laser em função dos diferentes níveis de expressão de cada gene. A imagem dos pontos fluorescentes é processada por meio de métodos computacionais com o objetivo de calcular a intensidade obtida para cada mRNA [34]. A Figura 2.1 ilustra todo o processo.

A tecnologia de *microarrays* não fornece apenas informações sobre a função de genes anônimos mas também constitui uma ferramenta indispensável para estudos globais de expressão gênica, com grande aplicabilidade nos estudos de biologia molecular e fisiologia vegetal [34].

Como exemplo do resultado obtido por essa técnica, podemos citar a base NCI60 [20] utilizada em nossa mineração de regras. Essa base de dados faz parte do *NCI60 Cancer Microarray Project*, projeto este, advindo da colaboração entre o laboratório *Brown/Bolstein* do grupo *John Weinstien's* do *Laboratory of Molecular Pharmacology* e do *Laboratory of Developmental Therapeutics*, ambos pertencentes ao *National Cancer Institute*, nos EUA.

Para a construção desta base, foram utilizados *microarrays* de cDNA na busca de

[1] Molécula de DNA produzida a partir de um mRNA e, portanto, sem íntrons [29].

[2] A hibridização de ácidos nucléicos baseia-se na capacidade destas moléculas, quando em cadeias simples, poderem associar com seqüências complementares formando cadeias duplas mais estáveis.

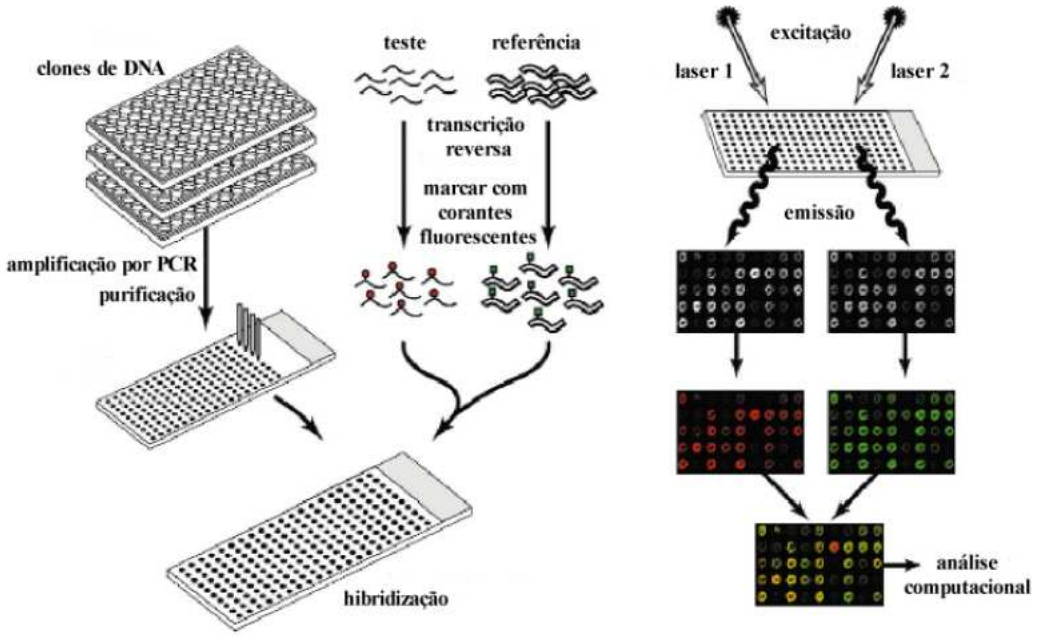

Figura 2.1: Esquema de *microarray* de cDNA [1]

expressões gênicas de aproximadamente 8.000 genes distintos. Estes genes, oriundos de 61 linhagens celulares, foram classificados em 9 (nove) classes de câncer: (1) mama, (2) sistema nervoso central, (3) cólon, (4) leucemia, (5) melanoma, (6) pulmão, (7) ovário, (8) renal e (9) células reprodutivas. Os números entre parênteses referem-se ao número utilizado para representar cada classe na base de dados. O número de ocorrências de cada classe é dado a seguir: mama (7), sistema nervoso central (6), cólom (7), leucemia (6), melanoma (8), pulmão (9), ovário (6), renal (8) e células reprodutivas (4), totalizando 61 amostras. Dentre os 8.000 genes, 3700 foram previamente caracterizados em proteínas humanas, 1900 advindos de genes homólogos de outros organismos e os 2400 restantes foram identificados somente por EST's [3] (*expressed sequences tags*).

No trabalho de Ooi e Tan [3] foi realizado um pré-processamento, no qual foram excluídos genes que estavam em *spots* inválidos, de controle e vazios, levando a 6176 genes. Para cada *array*, a expressão gênica de cada *spot* foi normalizado, subtraindo a média das razões de Cy5/Cy3 dos *spots* de controle e dividindo este resultado pelo desvio padrão da razão Cy5/Cy3 dos *spots* de controle. Finalmente, partindo dos 6176 genes pré-

[3] Os EST's são sequencias parciais de clones de cDNA

Tabela 2.1: Visão geral da base NCI60 reduzida e utilizada nos experimentos de Ooi e Tan [3]

	Expressão	**Expressão**	**Expressão**	...	**Expressão**	**Expressão**	
Amostra	**Gene 1**	**Gene 2**	**Gene 3**		**Gene 999**	**Gene 1000**	**Classificação**
1				...			
2				...			
3				...			
...	...	...	...	...	...	...	...
60				...			
61				...			

processados, Ooi e Tan chegaram a um *dataset* reduzido contendo 1000 genes, os quais, apresentaram os maiores valores de desvio padrão na base NCI60. Estes genes foram indexados de 1 a 1000.

A Tabela 2.1 apresenta uma visão geral da base NCI60 reduzida em [3], composta pela expressão de 1000 genes (colunas), medida para 61 amostras de células (linhas), sendo que cada amostra é classificada em uma das nove classes de câncer citadas anteriormente (última coluna). O apêndice A apresenta um fragmento desta base, com a expressão de 8 genes (os 4 primeiros e os 4 últimos) nas 61 amostras.

O apêndice B apresenta outros quatro fragmentos dessa base, que foram utilizados nos experimentos dessa dissertação. O primeiro, chamado de base B_1, contém a expressão gênica de 13 dos 1000 genes da base NCI60 usados por Ooi e Tan [3]. De forma similar, as bases B_2, B_3 e B_4 apresentam a expressão de 20, 17 e 12 genes, respectivamente, da base NCI60. Na seção 4.1.5, será detalhada a forma como esses fragmentos foram obtidos e como eles foram utilizados neste trabalho.

2.3 Bioinformática

A utilização de técnicas e ferramentas de computação na resolução de problemas da Biologia é chamada de Bioinformática ou Biologia Computacional. Essa área de pesquisa vem se tornando cada vez mais importante [6]. A computação pode ser aplicada na resolução de problemas como comparação de sequências (DNA, RNA e proteínas), montagem de fragmentos, reconhecimento de genes, identificação e análise da expressão de genes e

determinação da estrutura de proteínas [5, 6, 28].

O emprego de métodos computacionais na Biologia iniciou-se na década de 80, quando biólogos experimentais, em conjunto com cientistas da computação, físicos e matemáticos, começaram a aplicar esses métodos na modelagem de sistemas biológicos [28]. Na segunda metade de década de 90, com o surgimento dos seqüenciadores automáticos de DNA, houve uma explosão na quantidade de seqüências a serem armazenadas, exigindo recursos computacionais cada vez mais eficientes. Além do armazenamento ocorria, paralelamente, a necessidade da análise desses dados, o que tornava indispensável a utilização de plataformas computacionais eficientes para a interpretação dos resultados obtidos. Assim, a Bioinformática surgiu para tentar dar significado a essa enorme quantidade de dados [35]. Durante esse período, ferramentas computacionais foram desenvolvidas para análise dos dados, utilizando algoritmos convencionais da Ciência da Computação [28].

Devido à grande quantidade e a complexidade da informação, as ferramentas baseadas na computação convencional têm se mostrado limitadas na abordagem de problemas biológicos complexos. Isto vem ocorrendo, entre outras razões, devido à ausência de uma teoria fundamental em nível molecular. Outra razão para essa dificuldade é a ineficiência das ferramentas convencionais em lidar com grandes quantidades de dados [28]. Técnicas de Inteligência Artificial (IA) [36], tais como, Algoritmos Genéticos, Redes Neurais Artificiais, dentre outros, são assim cada vez mais empregadas para tratar problemas em Biologia Molecular, por sua capacidade de aprender automaticamente a partir de grandes volumes de dados e produzir hipóteses úteis [6]. Um dos principais exemplos de aplicação de técnicas de bioinformática reside na análise de dados de expressão gênica.

2.4 Análise de Expressão Gênica

Devido ao avanço das tecnologias utilizadas na obtenção de dados de expressão gênica, o volume desses dados vem aumentando exponencialmente. Assim, uma das áreas mais proeminentes da Bioinformática nos dias atuais, reside na aplicação de técnicas computacionais para a análise dos dados gerados em experimentos de *microarray*. Diferentes técnicas de Inteligência Artificial foram aplicados na análise de dados de expressão gê-

nica, tais como: Redes Neurais Artificiais em [7, 8], *Support Vector Machines* em [10, 9] e Algoritmos Genéticos em [3, 11, 12, 13, 14, 15].

Um exemplo de aplicação de diferentes técnicas de bioinformática na análise de dados de expressão gênica é a diversidade de trabalhos envolvendo a base NCI60 [20], desde a sua publicação em 2000, descrevendo os experimentos de *microarray*.

Dudoit e colegas (2002) utilizaram a base NCI60 para a comparação de performance entre diferentes métodos de classificação [21]. Os métodos avaliados incluem os classificadores baseados no vizinho mais próximo (*nearest-neighbor*), análise de discriminante linear (*linear discriminant*) e árvores de decisão. Neste trabalho, das 9 classes existentes na base NCI60 foram utilizadas 8, não inserindo na análise a classe 9 (células reprodutivas). Foram obtidos conjuntos preditores formados por 30 genes. Os resultados encontrados para estes conjuntos foram validados utilizando 1/3 das amostras (21), isto é, os classificadores foram treinados utilizando 2/3 de todas as amostras da base (40). Os três ambientes foram executados 200 vezes e os resultados apresentam a média do número de erros encontrados nestas 200 execuções. Para o método que utilizou análise de discriminante linear foram encontrados 9 erros em 21 amostras, isto é, ele classificou corretamente 12 amostras (57,14%). No método baseado em árvores de decisão, foram encontrados 10 erros em 21 amostras, totalizando 52,38% de acertos e os classificadores baseados no vizinho mais próximo erraram 8 amostras em 21 possíveis, totalizando 61,9% de acerto.

Deb e Reddy (2003) buscaram identificar pequenos conjuntos de genes a partir de amostras de câncer que possuem duas ou mais classes [12]. Na busca destes conjuntos, o método NSGA-II (*Nondominated Sorting Genetic Algorithm II*) foi aplicado na otimização de classificadores baseados no método WN/OVA (*weigthed voting/one-versus-all binary pair-wise*). Neste trabalho, a base NCI60, composta por 61 amostras, foi dividida em dois conjuntos, treinamento, contendo 41 amostras, e teste, contendo 20 amostras. Foram encontrados conjuntos formados por 12 genes que obtiveram 92,68% de acurácia em treinamento e 90% em teste.

Ooi e Tan (2003) também identificaram conjutos de genes preditivos a partir da base NCI60, utilizando para isso um AG e um classificador MLHD [3]. Na busca deste conjunto, os pesquisadores partiram de um fragmento da NCI60 formada por 61 amostras de 1000

genes. Estas bases foram divididas em dois conjuntos, treinamento e teste, tendo 2/3 e 1/3 das 61 amostras, respectivamente. Neste trabalho foi obtido um conjunto preditivo com 13 genes com taxa de erro de 14,63% (6 erros em treinamento) no método *leave-one-out cross validation* (LOOCV) e 5% (1 erro em teste) utilizando um conjunto de teste independente. Uma outra análise foi feita neste trabalho retirando-se da função de avaliação a segunda taxa de erro. Para este ambiente, foi encontrado um conjunto preditivo com 12 genes com taxa de erro de 9,76% (4 erros em treinamento) no método *cross validation* e 20% (4 erros em teste) utilizando um conjunto de teste independente.

Liu e colaboradores (2005) [13], utilizaram algoritmos genéticos (AG) combinado a *support vector machines* (SVM) na busca de pequenos conjuntos de genes que fossem classificadores confiáveis em bases multiclasses. O AG foi usado como seletor de genes e a SVM na categorização das classes analisadas. Foi utilizado o método *leave-one-out cross-validations* (LOOCV) na validação dos resultados, obtendo 88,52% de acertos, considerando a base completa (61 amostras), com um conjunto preditor composto por 40 genes para a base NCI60.

Em [22], Umpai e Aitken (2005) encontraram conjuntos de genes preditores utilizados na classificação da base NCI60. Antes de executar a classificação, foi feita uma seleção de genes utilizando o software *RankGene* [37], onde foram selecionados os *top* 100 genes. O ambiente utilizado na busca destes conjuntos é formado por um AG padrão combinado a um classificador *k nearest neighbour* (KNN) [38]. Devido ao baixo número de amostras da base NCI60, os autores não consideraram adequado dividí-la em treinamento e teste. A avaliação deste conjunto preditivo foi feita utilizando LOOCV aplicado à base inteira. O melhor resultado encontrado para a base NCI60 foi 76,23% de acertos e um conjunto preditor de 30 genes.

No trabalho de Uriarte e Andrés (2006) [23], buscou-se a construção de pequenos conjuntos preditores de genes eficazes na classificação multiclasse. Para tal, eles buscaram identificar conjuntos com o menor número de genes possível e bons níveis de predição. Neste trabalho foi investigado o uso de algoritmos de *random forest* [39] na classificação de dados multiclasse advindos de experimentos de *microarray*. Este método é formado por conjuntos de árvores de decisão [40, 41, 42], que segundo os autores, possuem um

bom poder de predição em dados com ruído. Neste trabalho, a base NCI60 foi dividida em dois conjuntos chamados de treinamento e teste. As taxas de erros obtidas utilizando o método .632+ *bootstrap* [43] foi de 25,2%. Esse valor foi comparado com o obtido por outros métodos [44, 38, 42, 45, 46], também utilizando o método de avaliação *bootstrap*, nos quais resultados similares foram obtidos.

No trabalho de Lin e seus colaboradores (2006) [24] foi utilizado um algoritmo genético combinado com uma função discriminante *silhouete statistics* [47] (GASS) para seleção gênica e reconhecimento de padrões. Este AG é utilizado na identificação de um conjunto de características correlatas e então evolui-se este conjunto utilizando *silhouette statistics* com distâncias métricas distintas para filtrar as características chaves para a classificação. Na pré-seleção dos genes que seriam analisados, usou-se o método BSS/WSS [21] para rankear os genes que são fortemente correlacionados à uma determinada classe e que não estão correlacionados às outras analisadas. Bons resultados foram encontrados. Para a base NCI60 foram obtidos 87,8% de acertos em treinamento e 85% em teste.

Capítulo 3

Algoritmos Genéticos (AGs)

3.1 Visão Geral do Método

Constantemente, o homem tem se servido das características e princípios existentes na natureza para a criação de máquinas, métodos e técnicas. Alguns exemplos típicos desta inspiração foram as seguintes invenções: aviões baseados nas características de pássaros, submarinos com sistemas de imersão semelhantes ao dos peixes, sonares baseados nos morcegos, dentre vários outros [48]. Em meados do século XIX, surgiu um dos mais importantes princípios no campo da evolução da vida, a Teoria da Evolução de Darwin, que defende a idéia de que na natureza, os seres vivos com as melhores características tendem a sobreviver frente aos demais. Baseada nesta teoria, a medicina e suas ciências afins buscam mapear toda a informação genética humana, relacionando deste modo, cada gene de cada cromossomo às características que eles representam nos indivíduos: hereditárias, físicas e funcionais [48]. Busca-se assim, elucidar quais genes e características promovem a disparidade entre os indíviduos. A ciência da computação inspirou-se também nestes princípios para a resolução de outros problemas. Surgiu então, a técnica de inteligência artificial conhecida por Algoritmo Genético [49, 50, 51], que teve seu marco inicial no trabalho de John Holland, na década de 60 [52].

Algoritmos Genéticos são métodos computacionais de busca baseados nos mecanismos da evolução natural e na genética, simulando a teoria da seleção natural de Darwin [50].

Os AGs fazem parte da Computação Evolutiva, área da Inteligência Artificial pro-

veniente da interseção entre a Biologia Evolutiva e a Ciência da Computação, sendo constituída de procedimentos de busca e otimização, em que o espaço de busca das soluções de um problema é explorado a partir de uma amostragem aleatória de seus pontos, utilizando um mecanismo inspirado na evolução biológica. Estes pontos sofrem operações, análogas às operações genéticas, de forma a guiar a busca para regiões mais promissoras desse espaço de soluções. A Figura 3.1 ilustra a relação entre essas três áreas.

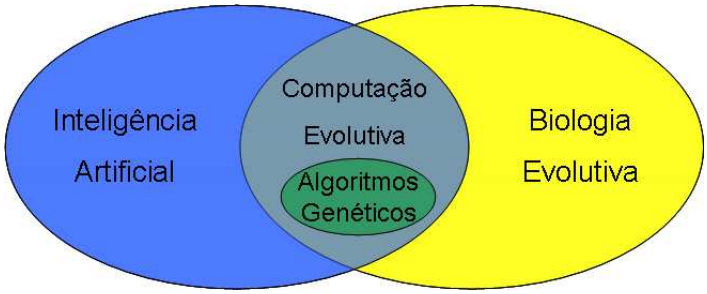

Figura 3.1: Computação Evolutiva: interseção entre a Inteligência Artificial e a Biologia Evolutiva

AGs são métodos computacionais de busca, baseados nos mecanismos da evolução natural e na genética natural. Eles combinam a sobrevivência do melhor adaptado dentre estruturas formadas por sequências de *bits*, com uma troca de informação aleatória e estruturada para formar um algoritmo computacional com algum faro inovador da busca humana. Apesar de não serem determinísticos, os Algoritmos Genéticos não são uma simples caminhada aleatória. Eles exploram eficientemente informações históricas para especular novos pontos de busca com um aumento esperado de performance [50].

O AG é um algoritmo que manipula, em paralelo, um conjunto de indivíduos (chamado de população), tipicamente constituído por cadeias de símbolos de tamanho fixo, que representam os cromossomos. A cada indivíduo é associada uma avaliação. O AG transforma a população corrente em uma nova população usando operações de reprodução e sobrevivência, segundo critérios baseados em uma determinada função de avaliação [53].

Em AGs, uma população de possíveis soluções para o problema em questão evolui de acordo com operadores probabilísticos concebidos a partir de metáforas biológicas, de modo que haja uma tendência de que, na média, os indivíduos representem soluções cada vez melhores à medida que o processo evolutivo continua [54]. O ciclo básico de execução de um AG é ilustrado na Figura 3.2 [55].

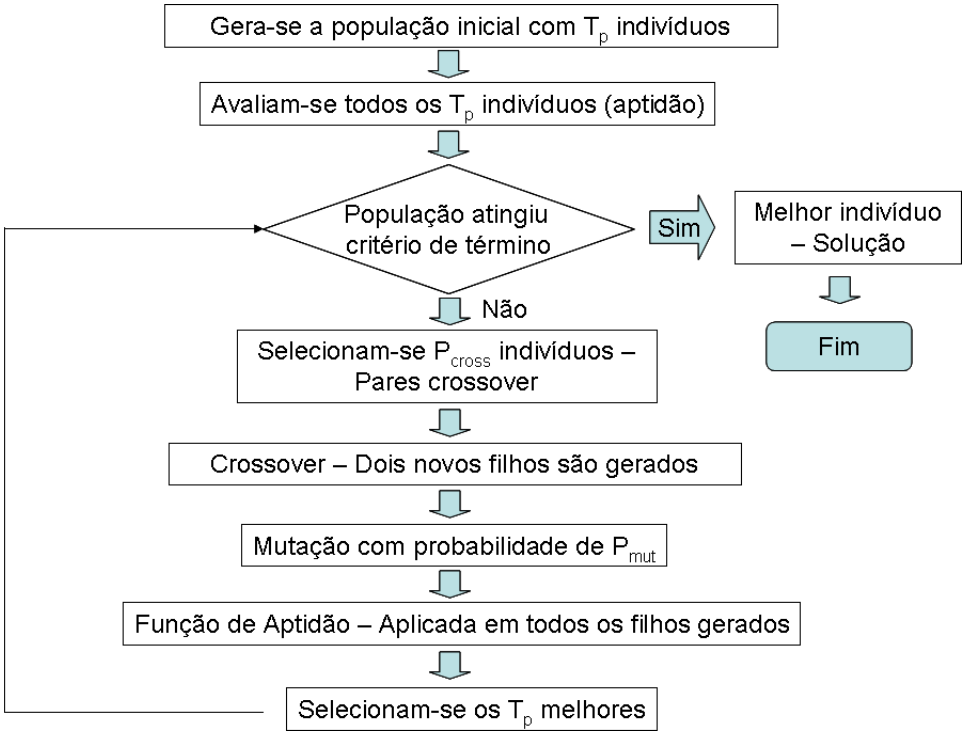

Figura 3.2: Ciclo de Execução Básico de um AG

Como é possível perceber na Figura 3.2, os AGs manipulam uma população de indíviduos, sendo que cada indivíduo na população representa uma possível solução para um dado problema. A cada indivíduo é associado um valor de adaptabilidade, chamado de **aptidão**. A tarefa do AG é procurar uma solução ótima para o problema ou uma solução que satisfaça um determinado critério de qualidade. A cada iteração do AG uma nova geração de indivíduos é criada, usando os princípios Darwianos de reprodução e sobrevivência dos mais aptos, através da aplicação de operações genéticas tais como **recombinação** (*crossover*) e **mutação** [48].

Vários aspectos do projeto devem ser cuidadosamente analisados e especificados para que se possa trabalhar com AGs eficientemente. Dentre esses aspectos podemos citar como principais: [50]:

1. Representação do indivíduo (ou codificação do cromossomo).
2. Definição de uma estratégica para a geração da população inicial.
3. Definição da função de avaliação ou aptidão (*fitness function*).

4. Especificação dos operadores genéticos:

    - Operadores de seleção de indivíduos que serão utilizados na reprodução (pais);
    - Operadores de cruzamento ou *crossover*;
    - Operadores de mutação;
    - Operadores de reinserção da população ao final de cada geração.

5. Definição de um critério de parada.

6. Especificação dos parâmetros genéticos:

    - Tamanho da população (T_p);
    - Taxa de *crossover* (T_c);
    - Taxa de mutação (T_m);
    - Número de gerações (N_{ger}).

Nas seções a seguir, detalharemos alguns desses aspectos.

3.1.1 Representação do Indivíduo e Geração da População Individual

Os AGs manipulam simultaneamente um conjunto de soluções chamado de população. Cada elemento desse conjunto de soluções, ou cada ponto no espaço de busca, é denominado indivíduo ou cromossomo. Cada indivíduo representa uma possível solução do problema que se deseja resolver. Um indivíduo é normalmente representado por uma cadeia de símbolos, podendo esta cadeia ser estática ou dinâmica. As cadeias estáticas podem ser representadas por um vetor (ou por um conjunto de vetores), cujos elementos podem ser binários, inteiros ou reais. As cadeias dinâmicas são geralmente representadas por vetores dinâmicos ou árvores. As cadeias dinâmicas podem, ao longo da execução do AG, diminuir ou aumentar de tamanho. O mesmo não ocorre com as cadeias estáticas, onde o tamanho é fixado no início da execução do AG.

Os AGs iniciam a busca da melhor solução a partir de um conjunto inicial de soluções. Na maioria das aplicações, a geração da população inicial é feita de forma aleatória. Entretanto, em problemas de difícil convergência, a geração da população pode ser feita de forma tendenciosa, utilizando-se algum conhecimento prévio do problema nesta escolha.

3.1.2 Função de Avaliação ou Aptidão (FA)

A Aptidão refere-se ao grau de contribuição de uma determinada solução candidata para a convergência do AG na busca da melhor solução dentro do espaço de busca. Para mensurar esta grandeza utiliza-se uma Função de Avaliação ou Aptidão (*Fitness Function*), cujo objetivo é estabelecer uma medida de qualidade para cada indivíduo da população. Por isso, a definição dessa função decorre diretamente da modelagem do problema onde se deseja utilizar o AG.

Segundo estimativas, o cálculo da função de avaliação consome a maior parte do tempo de processamento de um AG, podendo chegar a até 95% deste tempo de processamento [56]. Devido a este fato, a definição da função de avaliação torna-se um fator crítico e um dos pontos mais importantes no projeto dos AGs.

3.1.3 Operadores Genéticos

O princípio básico dos operadores genéticos é transformar a população (conjunto de soluções candidatas) através de sucessivas gerações, realizando a busca pela melhor solução até que seja alcançado um resultado satisfatório. Os operadores genéticos são necessários para que a população se diversifique mas que também mantenha as boas características de adaptação adquiridas pelas gerações anteriores. Os principais operadores genéticos são: seleção dos pais para a reprodução, cruzamento ou recombinação (*crossover*), mutação e reinserção da população. A seleção seleciona quais serão os pais que passarão seu material genético para a próxima geração. O cruzamento ou *crossover* cria novos indivíduos que possuem em sua carga genética genes vindos dos pais selecionados. A mutação altera um indivíduo para produzir uma nova solução, um pouco diferente de outra já existente na população. A reinserção seleciona quais indivíduos, entre pais e filhos, farão parte da

próxima geração.

Seleção dos Pais

De acordo com a teoria de Darwin, o princípio da seleção natural privilegia os indivíduos mais aptos e com maior longevidade e, portanto, com maior probabilidade de reprodução. Indivíduos com mais descendentes têm mais chance de perpetuarem seus códigos genéticos nas próximas gerações. A maioria dos métodos de seleção de pais são projetados para escolher preferencialmente indivíduos com maiores valores de aptidão, embora não exclusivamente, a fim de manter a diversidade da população. Com base na teoria Darwiniana, foram construídos vários métodos de seleção, dentre os quais podemos citar: truncamento (*Truncation Selection*), ranking (*Rank Based Fitness Assigment*), roleta (*Roulete Wheel Selection*), amostragem estocástica (*Stochastic Universal Sampling*), torneio simples (*Simple Tournament*) e torneio estocástico (*Stochastic Tournament*). Detalharemos os métodos conhecidos por roleta e torneio estocástico por serem os métodos investigados neste trabalho.

O método de seleção de pais mais clássico, proposto no trabalho pioneiro de Holland [52], é conhecido por método da roleta, onde os indivíduos de uma geração são escolhidos para fazer parte da próxima geração, através de um sorteio de roleta. Neste método, cada indivíduo da população é representado na roleta proporcionalmente ao seu índice de aptidão. Assim, aos indivíduos com alta aptidão é dada uma porção maior da roleta, enquanto aos de aptidão mais baixa é dada uma porção relativamente menor. Finalmente, a roleta é girada um determinado número de vezes, dependendo do número de pais que serão selecionados para o *crossover*. Os indivíduos selecionados pela roleta fornecem material genético para a construção de novos indivíduos, chamados filhos. A Figura 3.3 apresenta um exemplo de construção da roleta para seleção dos pais, indicando a distribuição das aptidões relativas para uma população fictícia de 4 indivíduos. A aptidão do indíviduo 1 é igual a 3 e representa 18,75% da soma de todas as aptidões da população. A aptidão do indivíduo 2 é igual a 7 (43,75%) e as aptidões dos indivíduos 3 e 4 iguais a 2 (12,5%) e 4 (25%), respectivamente. Assim, estes percentuais definem as probabilidades de cada indivíduo da população ser sorteado para a formacão dos pais para o *crossover*.

Por exemplo, ao sortearmos um pai para realizar o *crossover*, qualquer um dos quatro indivíduos pode ser sorteado, mas o indivíduo 2, que é o melhor da população, tem uma probabilidade acima de 40% de ser sorteado. Por outro lado, o indivíduo 3, que é o pior, tem uma probabilidade de sorteio abaixo de 15%.

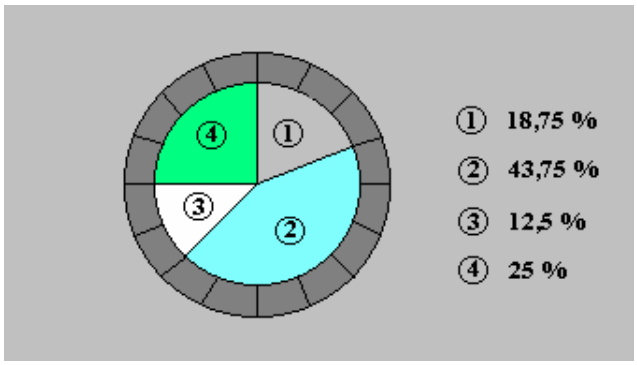

Figura 3.3: Roleta

No método de seleção torneio estocástico [50], n indivíduos que irão participar do torneio são selecionados utilizando uma roleta, elaborada da mesma forma que a explicada anteriormente. A esse número n damos o nome de *tour*. Para que se possa montar um torneio estocástico com *tour* de tamanho 3 (três), por exemplo, teremos que rodar a roleta três vezes, e o vencedor do torneio é aquele indivíduo que tiver a maior aptidão entre os três competidores. Por exemplo, suponha a mesma população de 4 indivíduos cujas avaliações são retratadas na roleta da Figura 3.3. A Figura 3.4 ilustra 2 torneios entre os quatro indivíduos. No primeiro torneio, a roleta foi girada 3 vezes e ocorre a disputa entre os indivíduos 1, 2 e 3. Ao final, temos a vitória do indivíduo 2 por possuir maior valor de aptidão (igual a 7). No segundo torneio, concorrem os indivíduos 1 e 4 sendo que o indivíduo 1 foi sorteado duas vezes. Nesse caso, o indivíduo 4 é o vencedor com uma avaliação de 4.

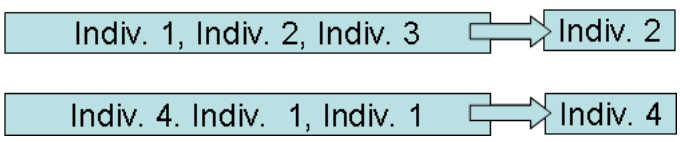

Figura 3.4: Torneio Estocástico de tamanho 3, empregando a roleta da Figura 3.3

Comparando-se os dois métodos, é possível perceber que o torneio estocástico é bem

mais seletivo do que a roleta. Embora todos os indivíduos possam ser sorteados, como pode ser observado no exemplo da população na Figura 3.3, a probabilidade do indivíduo 3 (o pior da população) ser sorteado é bem menor no torneio estocástico. Na roleta, basta um único sorteio com probabilidade de 12,5% (2 casas em 16 possíveis) para que ele seja sorteado. No torneio estocástico, a única forma do indivíduo 3 ser vencedor é se a roleta for girada 3 vezes e nas três ele for sorteado. Caso contrário, qualquer outro indivíduo sorteado será o vencedor em relação ao 3. Assim, a probabilidade do indivíduo 3 ser sorteado por torneio estocástico cai de 12,5% para 0,195%.

Cruzamento ou *Crossover*

Os indivíduos sorteados pelo método de seleção dos pais são recombinados através do operador genético *crossover*. O operador de *crossover* é considerado a característica fundamental dos AGs [57], simulando a reprodução sexuada na natureza.

Este operador gera novas soluções (filhos) a partir de soluções escolhidas da lista de soluções já existentes (pais). O operador de crossover possui diferentes variações, muitas delas específicas a um determinado problema. Alguns exemplos de métodos de crossover são: o *crossover* simples, o *crossover* múltiplo e o *crossover* uniforme.

No *crossover* simples, ocorre o sorteio de um único ponto de corte no cromossomo. Dois filhos são gerados, cada um formado com uma parte do material genético de cada progenitor. O primeiro filho repete os genes do cromossomo do primeiro pai até o ponto de *crossover*. A partir deste ponto, ele repete os genes do segundo pai. O segundo filho repete os genes do segundo pai até o ponto de *crossover* e a partir deste ponto, ele repete os genes do primeiro pai. A Figura 3.5 ilustra como é feita a troca de carga genética em indivíduos binários, através do *crossover* simples.

O *crossover* múltiplo segue a mesma idéia do *crossover* simples. A diferença está no número de pontos de *crossover* sorteados. Enquanto que no *crossover* simples há apenas um sorteio, no *crossover* múltiplo há ao menos dois sorteios. A Figura 3.6 ilustra a troca de carga genética com dois pontos de *crossover*.

O *crossover* uniforme é um tipo de *crossover* múltiplo levado ao extremo, onde ao invés de serem sorteados pontos de *crossover*, sorteia-se uma máscara que possui o mesmo

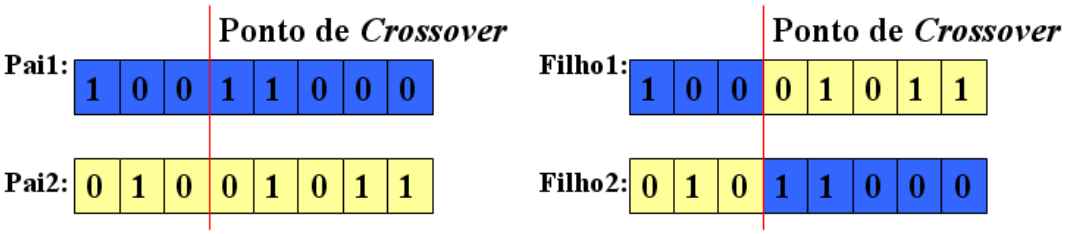

Figura 3.5: *Crossover* Simples

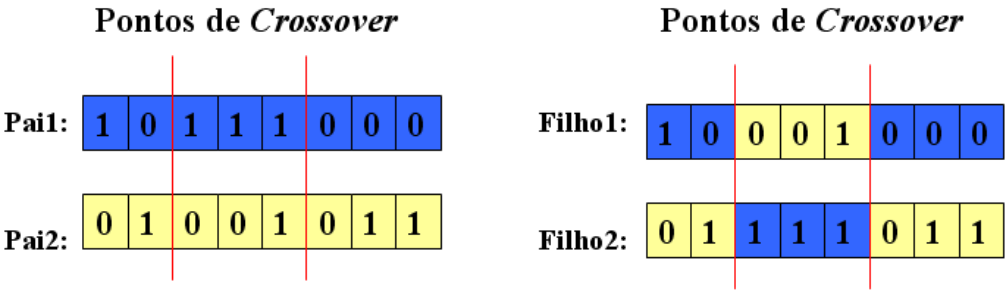

Figura 3.6: *Crossover* Múltiplo

tamanho do cromossomo, que indica qual cromossomo pai fornecerá cada gene do primeiro filho. O segundo filho é gerado pelo complemento desta máscara. O *crossover* uniforme é exemplificado na Figura 3.7.

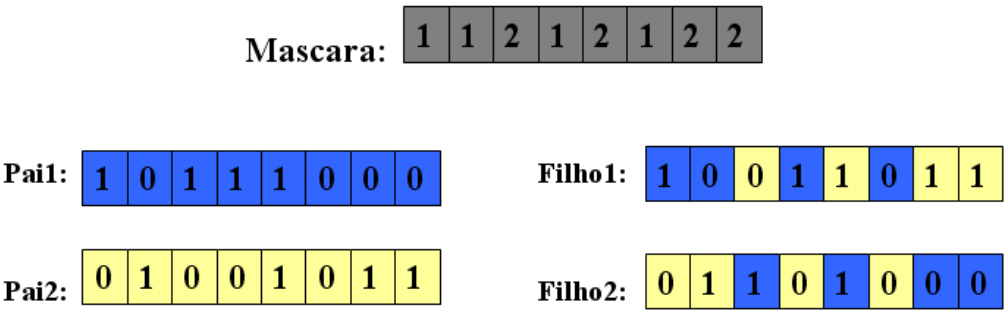

Figura 3.7: *Crossover* Uniforme

Mutação

O operador genético de mutação é aplicado para que seja feita a manutenção da diversidade genética da população, alterando-se arbitrariamente um ou mais genes do cromos-

somo. Dessa forma, a mutação fornece meios para a introdução de novos indivíduos na população assegurando que existe a possibilidade de se chegar a qualquer ponto do espaço de busca. Além disso, ele pode contornar o problema de ótimos locais, alterando levemente a direção da busca.

A operação de mutação muda aleatoriamente a descendência criada pelo *crossover*. Este operador é aplicado aos indivíduos com uma probabilidade dada pela taxa de mutação T_m, fornecida como parâmetro de entrada do AG. Esta taxa de mutação pode ser dada por indivíduo ou por gene.

Os tipos de mutação são diretamente influenciados pela estrutura do indivíduo. Os tipos mais comuns de mutações são: mutação binária, mutação real e permutação.

A mutação binária é aplicada a cromossomos binários. Neste operador troca-se um ou mais bits do cromossomo, modificando-o(s) pelo seu complemento binário. A Figura 3.8 ilustra este operador.

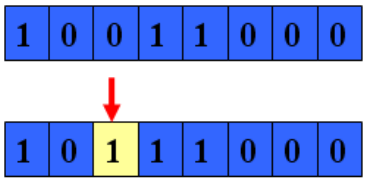

Figura 3.8: Mutação Binária

A mutação real altera o valor original contido no gene através do sorteio de um pequeno valor de incremento ou decremento. Após este sorteio, este valor é incrementado ou decrementado ao valor original. A Figura 3.9 ilustra este tipo de mutação. Neste exemplo foi sorteado um decremento de 0,7.

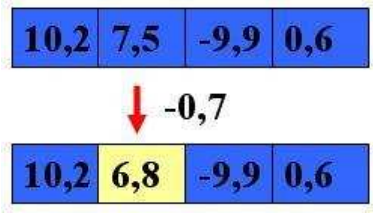

Figura 3.9: Mutação Real

Na permutação, ocorre a troca de lugar entre dois genes ou mais genes. A Figura 3.10

ilustra esta operação.

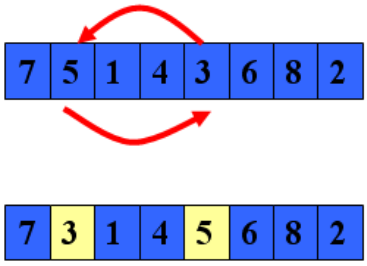

Figura 3.10: Mutação Permutação

Reinserção

O operador genético de reinserção é responsável pela seleção dos indivíduos que farão parte da população de pais para a próxima geração. Os principais métodos de reinserção são: reinserção pura, reinserção uniforme, elitismo e melhores pais e filhos.

No método de reinserção pura, ocorre a substituição de toda a população antiga pela nova população gerada (filhos). Na reinserção uniforme, a seleção dos indivíduos é feita utilizando-se algum método de sorteio, como a roleta e o torneio estocástico, aplicado à união da população de pais e filhos. No método melhores pais e filhos, todos os pais e filhos são colocados numa mesma população e os T_p melhores indivíduos são selecionados para a próxima geração. A escolha destes T_p melhores indivíduos é feita exclusivamente baseada nas suas aptidões. O operador de elitismo garante que os n (fornecido como parâmetro de entrada do AG) melhores indivíduos encontrados na geração são passados para a nova população, de forma que as melhores soluções possam sobreviver às sucessivas gerações.

3.1.4 Critério de Parada e Parâmetros Genéticos

Dependendo das características de cada projeto, os critérios de parada adotados podem variar. Eles podem estar correlacionados a um determinado número de gerações, se o AG encontrou ou não a solução ótima (isso se a mesma for conhecida), perda de diversidade

das soluções ou pode estar correlacionda à convergência nas últimas k gerações, isto é, quando não ocorre melhoria na aptidão média e máxima.

Os parâmetros genéticos influenciam diretamente no comportamento dos AGs. Devido a este fato, devemos estabelecê-los conforme as necessidades do problema em questão e dos recursos disponíveis. Os principais parâmetros genéticos que devemos ajustar são: tamanho da população, taxa de *crossover*, taxa de mutação e o número de gerações.

O tamanho da população afeta diretamente o desempenho global e a eficiência dos AGs. Com uma população pequena o desempenho pode cair, pois deste modo, a população fornece uma pequena cobertura do espaço de busca do problema. Por outro lado, uma grande população fornece uma cobertura representativa do domínio do problema, além de prevenir convergências prematuras para soluções locais ao invés de globais. No entanto, para se trabalhar com grandes populações, são necessários maiores recursos computacionais ou que o AG trabalhe por um período de tempo muito maior.

A taxa de *crossover* representa o número de pais presentes na população atual que serão selecionados para a geração dos indivíduos que irão compor uma nova população. Quanto maior for esta taxa, mais rapidamente novos indivíduos são introduzidos na população, mas também maior é o custo computacional.

A taxa de mutação representa a probabilidade de cada gene do indivíduo ter o seu valor alterado por outro valor válido. A taxa de mutação deve ser o suficiente para assegurar a diversidade dos cromossomos na população. Uma taxa de mutação baixa previne que uma dada população fique estagnada em um valor, além de possibilitar que se chegue a qualquer ponto do espaço de busca. Por outro lado, com uma taxa muito alta, a busca se torna essencialmente aleatória.

O número de gerações corresponde ao número de iterações completas que o AG deverá executar. O número de gerações deve ser analisado cuidadosamente para que se tenha um melhor aproveitamento das execuções.

3.2 Variações do AG Padrão

O modelo de AG discutido na seção 3.1 é conhecido por AG Padrão. Esse modelo é fortemente baseado no modelo original de Holland [52] e foi extensamente difundido e investigado nas décadas de 80 e 90. Recentemente, novos modelos surgiram incorporando características que melhor se adaptavam a algumas classes de problemas. Podemos citar como exemplos desses novos modelos os AGs Coevolutivos [58] e os AGs Multi-Objetivos [59]. O apêndice C apresenta uma visão geral dos AGs Multi-Objetivos.

3.3 Aplicações de Algoritmos Genéticos em *Data Mining*

Data Mining é um conjunto de técnicas e ferramentas aplicado para a descoberta do conhecimento em bases de dados. A tarefa de classificação é uma das várias estudadas em *data mining*. Em essência, o problema consiste em atribuir valores para os registros pertencentes a um pequeno conjunto de classes, e assim, descobrir algum relacionamento entre estes atributos. Cada registro é composto de um conjunto de atributos preditivos e um atributo objetivo [60, 61]. Um algoritmo de *data mining* é aplicado ao conjunto de treinamento, contrapondo-os a uma classe conhecida, na busca de algumas relações entre os atributos preditivos e o atributo objetivo. Estes relacionamentos são então usados para predizer a classe (o valor do atributo objetivo) de amostras cuja classe é desconhecida [19].

O conhecimento descoberto pode ser representado na forma de regras de classificação do tipo IF-THEN. Este tipo de regra se destaca devido ao seu alto nível de entendimento e pela representação do conhecimento simbólico, contribuindo para compreensibilidade das informações descobertas. As regras descobertas podem ser construídas de acordo com vários critérios, tais como: grau de confiança da predição, taxa de acerto da classificação para amostras de classes desconhecidas, compreensibilidade, dentre outros [19].

Como exemplo de aplicação de AGs em *data mining*, podemos citar o trabalho de Fidelis e colaboradores [19], no qual um modelo de AG foi elaborado para a obtenção de regras de classificação em bases de dados clínicos. Esse AG foi implementado através do GALOPPS 3.2 [62], ferramenta esta de domínio público que incorpora várias características propostas pelos AGs. Deste modelo, várias características são importantes elucidar,

tais como: codificação do indivíduo, operadores genéticos e função de avaliação.

O indivíduo é composto por n genes e cada gene é dividido em três partes, peso, operador e valor. Cada gene corresponde a uma condição da parte IF da regra, e o indivíduo corresponde à toda parte conseqüente da regra. A parte ENTÃO é omitida no indivíduo. A cada execução do AG, todas as regras são evoluídas para uma mesma parte conseqüente. Assim, por exemplo, se o atributo objetivo possui 5 valores C_1, C_2, C_3, C_4 e C_5, o AG deve ser executado pelo menos 5 vezes: a primeira execução para minerar as regras com conseqüente atributo-objetivo = C_1, a segunda para atributo objetivo = C_2 e assim sucessivamente.

Os operadores genéticos de seleção, *crossover* e reinserção aplicados em [19] foram os tradicionais: torneio estocástico, *crossover* ponto-simples e elitismo respectivamente. Foram desenvolvidos três operadores de mutação, específicos para os campos de peso, operador e valor.

A função de avaliação avalia a qualidade de cada regra ou indivíduo. Foi utilizado neste trabalho a função de avaliação empregada em [63], combinando indicadores comumente utilizados em domínios médicos, chamados de sensibilidade e especificidade.

Dados sobre domínios dermatológicos e de câncer de mama compunham as bases de dados, extraídas do UCI *Machine Learning Repository* (*University of California at Irvine*) que podem ser obtidas em *www.ics.uci.edu*. Os resultados obtidos neste trabalho foram satisfatórios. Para a base dermatológica foram encontrados regras simples, variando de duas a seis condições na parte IF, resultando em aptidões de treinamento variando entre 85,5% a 100% e aptidões de teste de 78,3% a 100%. Os resultados obtidos para a base de câncer de mama foram um pouco piores. Foram obtidas regras com três condições na parte IF com aptidões de treinamento variando de 49,7% a 56,4% e aptidões de teste de 36,5% a 39,3%.

Além deste trabalho [19], vários outros foram desenvolvidos utilizando-se AGs na solução de tarefas de classificação [64, 65, 19, 66, 67, 68, 69, 12, 3, 70, 71, 72, 73, 13, 15]. Os AGs também foram utilizados em outras tarefas de *datamining*, tais como: associação [74, 75], modelo de dependência [76], clusterização [11, 14], dentre outros.

Existem duas abordagens na aplicação de AGs, para a obtenção de regras de classifica-

ção chamadas Michigan e Pittsburgh. A abordagem Michigan, proveniente dos trabalhos de Holland e Reitman, na década de 70, na Universidade de Michigan, emprega uma forma de representação que ficou conhecida como Michigan em referência ao nome da universidade de origem. Nessa abordagem, a população como um todo é a solução para o problema, isto é uma parte da solução candidata é composta por todas as regras (população) [77]. A abordagem Pittsburgh é proveniente dos trabalhos desenvolvidos por De Jong e Smith na Universidade de Pittsburgh. Esta abordagem emprega outra abordagem na representação dos indivíduos. Diferentemente da abordagem Michigan, nessa abordagem cada indivíduo da população representa uma solução do problema. Dessa forma, a população contém vários conjuntos de regras, sendo que cada indivíduo (conjunto de regras) representa uma solução homogênea do problema. Comparativamente com a abordagem Michigan, a abordagem Pittsburgh requer um esforço computacional menor para obter a solução, embora o cálculo da aptidão dos indivíduos seja mais complexa que na outra abordagem [78].

No nosso trabalho, utilizamos a abordagem Pittsburgh, onde cada indivíduo da população é uma regra de alto nível do tipo IF-THEN e esta regra corresponde a uma solução do problema para uma determinada classe de câncer, contida na base de dados avaliada.

3.4 Aplicações de Algoritmos Genéticos na análise de Expressão Gênica

Alguns dos principais projetos desenvolvidos aplicando-se AGs na análise dos dados de expressão gênica são revisados a seguir. Vários deles utilizaram a base de dados NCI60 investigada também nessa dissertação. Essa base foi apresentada na seção 2.2 e o apêndice A apresenta um fragmento da mesma.

No trabalho de Deb e Reddy (2003) [12], o objetivo foi estabelecer pequenos conjuntos de genes preditores que tiveram suas expressões medidas a partir de amostras de câncer que possuem duas ou mais classes. Neste trabalho, foi utilizado o AG multi-objetivos conhecido por NSGA-II (*Nondominated Sorting Genetic Algorithm II*) na busca de classificadores. Foram estudadas bases de dados de classificação binária e multiclasse. Para as

bases de classificação binária, um método de classificação por *ranking* chamado *weigthed voting* (WV) [79] foi empregado. Para a classificação multiclasse, além da abordagem WV, foi utilizado o método de classificação *one-versus-all* (OVA) *binary pair-wise* [79]. Cinco bases de dados de domínio público foram analisadas neste trabalho: *leukemia* [79], *diffuse large B-cell lymphoma* [80], *Colon* [81], GCM [82] e NCI60 [20], sendo as três primeiras de classificação binária e as duas últimas multiclasses. A base de dados *leukemia* foi dividida em dois conjuntos de dados, chamados de treinamento e teste, sendo constituídos por 38 amostras e 34 amostras, respectivamente. Para esta base, foi obtido 91,4% de acerto nas 72 amostras. A base de dados *diffuse large B-cell lymphoma* é constituída de 96 amostras sendo sendo 42 amostras de *lymphoma* e 54 de outros cânceres. As amostras desta base foram divididas em quantidades iguais em treinamento e teste (50% para cada conjunto) obtendo conjuntos classificadores compostos por 8 genes que classificaram corretamente 100% das amostras em treinamento e 97,91% das amostras em teste. A base de dados GCM é composta por 198 amostras que foram divididas em 144 amostras para treinamento e 54 amostras para teste. Foi obtido um conjunto de 37 genes que obteve 86% de acerto em treinamento e 80% em teste. A última base de dados analisada foi a NCI60. Esta base é composta por 61 amostras divididas em 41 amostras de treinamento e 20 amostras de teste, sendo encontrado 92,68% de acurácia em treinamento e 90% em teste.

O trabalho de Ooi e Tan (2003) [3] foi fundamental para o desenvolvimento dessa dissertação. Os autores buscaram identificar um conjunto de genes preditivos em relação a nove classes de câncer, a partir de uma base reduzida da NCI60, contendo as expressões gênicas de 1000 genes. Foi utilizado como estratégia de classificação um classificador MLHD e um AG que otimiza a entrada do MLHD. O AG determina automaticamente os membros do grupo de genes preditivos, assim como o tamanho ótimo deste conjunto, usando para isto, um método de classificação de máxima verossimilhança (MLHD), utilizado na avaliação da afinidade destes genes selecionados. Neste trabalho foram invetigadas as bases GCM [82] e NCI60 [20]. A partir da NCI60, 4 conjuntos de genes preditores foram gerados, dois deles utilizando o método AG/MLHD investigado no trabalho e dois deles empregando técnicas de *ranking* para comparação. A seção 4.1.5 discute como esses conjuntos foram obtidos. Foram encontrados bons resultados para ambas as bases anali-

sadas. Estas bases foram divididas em dois conjuntos, treinamento e teste, tendo 2/3 e 1/3 de todas as amostras de 1000 genes, respectivamente. Para a base GCM, formada por 198 amostras divididas em 14 classes, foram obtidos conjuntos preditivos formados por 32 genes e com taxa de erro de 20,14% no método *leave-one-out cross validation* (LOOCV) e 14,81% utilizando um conjunto de teste independente. Para a base NCI60, composta de 61 amostras divididas em 9 classes, foi obtido um conjunto preditivo com 13 genes com taxa de erro de 14,63% no método LOOCV e 5% utilizando um conjunto de teste independente. É importante salientar que para se chegar nos resultados apresentados anteriormente, estas duas taxas de erro foram utilizadas na evolução do AG. Vários pesquisadores [13, 25] questionaram os resultados obtidos com essa aptidão. Neste cálculo, foi utilizada uma informação vinculada à base de teste (a taxa de erro de teste independente). Assim, o AG utiliza, de uma certa forma, a base de teste em sua evolução. Portanto, a base de teste não pode ser considerada "independente"(*blind test*). Uma segunda evolução foi realizada em [3] sem a inserção da taxa de erro de teste, encontrando um conjunto preditivo com 12 genes com taxa de erro de 9,76% no método LOOCV e 20% utilizando um conjunto de teste independente; resultado este, inferior ao encontrado com as duas taxas de erro.

Em [13], o objetivo de Liu e colaboradores (2005) foi encontrar pequenos conjuntos de genes preditivos que sejam classificadores confiáveis em bases multiclasse. Neste trabalho, foram combinados algoritmos genéticos, usados como seletores de genes, e *support vector machines* (SVM), na categorização das classes analisadas. As SVM's necessitam estar integradas a outros algoritmos para proverem classificações multiclasse, tais como *one-vs.-all* ou *all-paired* (AP). Neste trabalho foi utilizado o método AP. O AG foi utilizado para evoluir o ambiente AP-SVM na busca dos melhores classificadores para as bases NCI60 [20] e Brown [83]. Para a validação dos resultados encontrados neste trabalho, também foi utilizado o método *leave-one-out cross-validation* (LOOCV). Porém, nesse caso, os autores não dividiram a base em treinamento e teste, realizando a validação LOOCV em 100% das amostras. Bons resultados foram encontrados para ambas as bases. Para a base NCI60 foi alcançado 88,52% de acertos com um conjunto preditor composto por 40 genes. Na base Brown os resultados foram um pouco piores, alcançando 81,23% de acerto.

Mitra e Banka (2006) [14] utilizaram AGs multi-objetivos na busca de *clusters* com

altos valores de relação *intra-class* e baixos valores de relação *inter-class*. Altos valores *intra-class* significa alta afinidade entre os genes de um determinado *cluster*, enquanto que, baixos valores *inter-class* denota uma independência (ou especificidade) entre estes *clusters*. Neste trabalho foram utilizados bases de dados de leveduras e de *humam B-cell lymphoma* advindos de experimentos de *microarray*. Estas bases podem ser encontradas no endereço *http://aprep.med.harvard.edu*. Também foi utilizada neste trabalho, o AG multi-objetivos NSGA-II (*Nondominated Sorting Genetic Algorithm II*), que se mostrou efetivo na construção de *clusters* com qualidade. A Biclusterização tem sido aplicada em análises de expresão gênica envolvendo dados cancerígenos, sendo utilizada principalmente na identificação de genes correlatos, anotação de funções gênicas e classificação de amostras. A validação biológica dos genes selecionados nos *biclusters* foi realizada pelo *GO Consortium*.

Na busca de conjuntos de genes preditivos e seus respectivos coeficientes de correlação ao câncer de mama [84], Wahde e Szallassi [15], utilizaram uma pequena variação do algoritmo genético padrão na evolução de classificadores simples. Neste trabalho, há a criação de uma lista de elite dos genes (*top genes*) construída utilizando-se uma versão de *ranking* muito parecida com o método *threshold number of misclasification score* (TNoM) [85]. Após construída esta lista, o AG é utilizado na evolução de classificadores do tipo *linear, single-threshold*, que selecionam os genes dentre a elite. A base de dados utilizada era composta por 97 amostras de 5.277 genes com apenas duas classes, divididos em 78 amostras para treinamento e 19 amostras para teste. Nesta classificação binária, foram formados conjuntos de 7 genes que obtiveram bons resultados de classificação. Nas bases destinadas ao treinamento e teste, foram obtidas regras com 97,4% e 89,5% de acertos, respectivamente.

Em todos os trabalhos citados anteriormente, os AGs foram empregados com o objetivo de ajustar algum outro modelo de classificador. Por exemplo, em [3], o AG seleciona o conjunto de genes que deve ser utilizado como entrada de um classificador MLHD. Em [13], os AGs são utilizados para otimizar as SVMs, que são os classificadores de fato. Entretanto, todos esses classificadores são do tipo "caixa-preta"e não explicam o conhecimento utilizado na classificação. Poucos trabalhos foram encontrados nos quais os

AGs são empregados para encontrar regras de classificação de alto nível do tipo IF-THEN. Esses trabalhos são revisados a seguir.

Hvidsten e colaboradores (2003) [16] utilizaram uma abordagem de aprendizagem supervisionada na predição de processos biológicos advindos de experimentos de *microarray*, buscando características ou perfis de expressão que possam ser discriminantes na formação de regras de decisão. Foi utilizado o sistema Rosetta [86], ambiente este, utilizado para *data mining* e *knowledge discovery*. Este ambiente emprega um AG padrão na construção e adaptação dos modelos preditivos [87]. Cada regra IF-THEN identifica um conjunto mínimo de características discriminantes de uma determinada classe de doença. O conjunto de regras de todas as classes analisadas constituem um classificador que pode ser aplicado em novas amostras de genes. Na avaliação destes classificadores é utilizado uma curva chamada *receiver operating characteristics* (ROC), contrapondo sensibilidade e especificidade. A base utilizada neste trabalho foi extraída do *The Gene Ontology Consortium 2000*, sendo dividida em dois conjuntos, um de treinamento e um de teste, divididos em 27 classes. Os classificadores evoluídos no conjunto de treinamento foram avaliados em teste utilizando *50-fold cross validation* e obtiveram, em média, no melhor resultado, 65% de acertos.

Em 2005, Viterbo e colaboradores [17] investigaram a performance de classificadores baseados em regras *fuzzy* em cinco bases de dados distintas. O objetivo dos autores era a geração de regras pequenas e simples, conseguida através de 2 tipos de algoritmos. Um algoritmo para fazer a categorização dos valores contínuos dos níveis de expressão, e um segundo algoritmo, responsável pela descoberta das regras. Estes algoritmos combinam discretização *fuzzy*, responsável pela discretização de valores contínuos em valores tais como: baixo, médio e alto ou benigno e maligno, e operadores *fuzzy* responsáveis pela geração das regras. O ambiente é composto principalmente por quatro partes, pré-seleção dos genes, apredizado *fuzzy*, construção das regras e filtragem destas regras. Para cada gene selecionado, ocorre a discretização do seu nível de expressão em um dos três valores possíveis (baixo, médio ou alto). Após feita a discretização de todos os genes selecionados, este conjunto de dados é utilizado na construção das regras. O último passo consiste em retirar regras redundantes. Como citado no trabalho [16], na filtragem das regras foi

utilizado o ambiente Rosetta [86] por ser simples e eficiente. Assim, este conjunto final de regras pode ser utilizado para determinar a classe de qualquer novo ou desconhecido elemento. Foram utilizadas cinco bases de dados, quatro delas encontradas em [88, 89, 79, 82] e a base NCBI-NLM 2004 (*http://ncbi.nlm.nih.gov/geo*), sendo pré-selecionados 200 genes de cada base de dados, divididos em treinamento e teste. Neste trabalho foi utilizado a mesma forma de avaliação de [16], a curva ROC, alcançando 99,81% de acertos em treinamento e 96,62% em teste de média para todas as 5 bases, utilizando na validação dos resultados obtidos em teste um *5 × 2 cross-validation test* proposto por [90].

Ho e seus colaboradores (2006) [18], construiram classificadores interpretáveis baseados em regras IF-THEN *fuzzy* precisas e compactas formadas por um pequeno número de genes relevantes para dados advindos de análises de *microarray*. Neste trabalho foi construído um classificador, chamado de iGEC, que busca otimizar três objetivos: precisão máxima de classificação, número mínimo de regras e número mínimo de genes utilizados. Um dos módulos deste ambiente é uma variação do AG encontrado em [91]. Este método, chamado de IGA, é utlizado para resolver eficientemente o ajuste do AG. Este ambiente foi aplicado em oito bases de dados. Os dados extraídos de [92, 93] contêm níveis de expressão gênica de tumores cerebrais, agrupados em 5 e 4 classes, respectivamente. Os dados encontrados em [94] possuem informações sobre *diffuse large b-cell lymphomas and follicular lymphoma*, agrupados em duas classes. Em [79, 95] foram obtidos níveis de expressão de leucemia, agrupados em 3 classes, em [88] de pulmão agrupados em 5 classes, [89] de tumores de próstata, agrupados em 2 classes, e em [8] de *small, round blue cell tumors of childhood* agrupados em 4 classes. O ambiente conseguiu uma precisão média de classificação de 87,9%, com média de 3,9 regras para cada base, e cada regra formada por 5 genes em média. Para validação destes resultados, foi utilizado uma validação cruzada com 10-dobras (*10-fold cross validation*). Este ambiente se mostrou mais efetivo na classificação do que os classificadores baseados em regras *fuzzy* existentes [17] e também a outros classificadores não baseados em regras, considerando todos os três objetivos.

Em nenhum dos trabalhos citados anteriormente [3, 17, 18], que fizeram a busca de regras de classificação de alto nível (IF-THEN), foi utilizada a base NCI60, investigada na presente dissertação. Segundo Xu (2007), é muito difícil propôr regras ou critérios

na determinação de um conjunto de genes que seja discriminantes no diagnóstico de doenças, especialmente quando as bases de dados estudadas possuem um elevado número de classes, tais como a complexa NCI60 [25]. A base NCI60 é conisderada um desafio para os algoritmos de classificação por suas características peculiares: um número relativamente alto de classes (9) para um número relativamente baixo de amostras (61), resultando em número baixo de amostras por classe, variando de 4 a 9 amostras por classe.

Capítulo 4

Ambiente Evolutivo

O ambiente evolutivo implementado neste trabalho foi baseado, principalmente, no trabalho de Fidelis e colaboradores [19] e no trabalho de Ooi e Tan [3]. Em relação ao trabalho de Fidelis e colaboradores, adaptamos o modelo do AG existente neste trabalho para minerarmos dados advindos de expressão gênica, além de alterarmos os operadores genéticos de *crossover* e de reinserção. Em relação ao trabalho de Ooi e Tan, ao invés de fornecermos um conjunto de genes preditores, que funciona como um classificador caixa-preta, construimos regras de classificação do tipo IF-THEN, representando o conhecimento em alto nível.

4.1 Descrição do Ambiente Evolutivo

O modelo do AG empregado no nosso ambiente evolutivo foi adaptado a partir do modelo proposto em [19] por se tratar de um ambiente voltado à mineração de regras do tipo IF-THEN. O AG foi elaborado em [19] com o objetivo de obter regras de classificação em bases de dados clínicos de pacientes e suas principais características foram revisadas na seção 3.3. As bases de dados onde o ambiente de Fidelis e colaboradores foram aplicadas eram formadas por registros que se caracterizavam por dados do paciente, no caso, a idade e presença da doença em histórico familiar e por dados relacionados a sintomas do paciente. As características que se relacionavam aos sintomas, que eram a maioria, foram todas discretizadas em: 0 - ausente, 1 - ocorrência leve, 2 - ocorrência moderada e 3 - ocorrência

severa. O ambiente evolutivo proposto nesta dissertação foi implementado na linguagem Delphi® e precisou ser adaptado para trabalhar com bases de dados de expressão gênica, onde os registros apresentam os níveis de expressão de dezenas ou centenas de genes, que são valores contínuos e com precisão variável (números reais). Para se chegar no ambiente evolutivo utilizado neste trabalho, partimos dos parâmetros propostos em [19] e fomos, experimentalmente, ajustando-os para a nossa aplicação. Vários aspectos foram abordados, tais como: melhores métodos de seleção e reinserção, tamanho da população, número de gerações, peso, tamanho do *tour* e precisão (número de casas após a vírgula). A seguir, as principais características de nosso modelo de AG são detalhadas: codificação do indivíduo, operadores genéticos, função de avaliação e parâmetros genéticos.

Antes de prosseguirmos, estabeleceremos a seguinte convenção: tanto no domínio do problema abordado, expressão gênica, como na descrição da técnica utilizada, algoritmo genético (AG), a palavra gene é utilizada, podendo surgir dúvidas em relação ao termo. Assim, convencionaremos que *gene* (em itálico) se refere ao gene do indivíduo do AG e gene (sem itálico) se refere ao gene humano.

4.1.1 Codificação do Indivíduo

O indivíduo ou cromossomo do AG proposto é composto por n *genes*, onde n corresponde ao número de genes encontrados na base de expressão gênica avaliada. Cada i-ésima posição do indivíduo é subdividida em quatro campos: I (índice), P (peso), O (operador) e V (valor), como ilustrado na Figura 4.1. Cada *gene* corresponde a um termo da condição na parte IF da regra e o indivíduo (cromossomo) representa todo o antecedente da regra.

$Gene_1$				...	$Gene_N$			
I_1	P_1	O_1	V_1	...	I_N	P_N	O_N	V_N

Figura 4.1: Cromossomo ou Indivíduo

O campo I corresponde ao índice do gene correspondente na base de expressão gênica utilizada. O campo P é uma variável do tipo inteira e o seu valor está compreendido entre os valores 0 (zero) e 10 (dez). É importante dizer que este campo P é o responsável pela

inserção ou exclusão de um termo na condição. Caso este valor seja menor do que um valor limite para este *gene* (e o gene correspondente na base) não fará parte da regra; caso contrário, o mesmo fará. Neste trabalho, na maioria das execuções do AG, foi utilizado como limite o valor 8 (oito). Isso significa que uma condição referente ao $Gene_i$ só estará presente efetivamente na regra se o valor do campo P_i for 8, 9 ou 10. Para todos os outros valores (0 a 7), a condição não estará presente na regra, independentemente dos valores dos outros campos O_i e V_i.

$Gene_1$				$Gene_2$				...	$Gene_{15}$				...	$Gene_{20}$			
7	3	$\geq$	-0,7	11	**9**	$\geq$	0,4	...	289	**8**	$<$	-0,5	...	905	6	$<$	2,1

Figura 4.2: Exemplo de cromossomo

Por exemplo, consideremos o indivíduo dado pela Figura 4.2, onde todos os outros *genes* omitidos têm o campo $P < 7$ e o indivíduo representa uma regra que pode ter no máximo 20 condições. Ou seja, o AG é aplicado sobre uma base com níveis de expressão gênica de 20 genes. O antecedente da regra equivalente a esse indivíduo é dado por:

$$\text{SE (Gene\_11} \geq 0{,}4\text{) E (Gene\_289} < 0{,}5\text{)}$$

Ou seja, apenas o *gene* 2 e o *gene* 15, que se referem às expressões dos genes de índice 11 e 289 da base, respectivamente, estão presentes no antecedente. O conseqüente não é representado explicitamente na regra. Ao contrário, a cada execução o AG busca regras de classificação para uma classe pré-especificada. Assim, suponha que o indivíduo da Figura 4.2 represente uma regra de uma execução do AG especificada para a classe 2. Dessa forma, a regra resultante é dada por:

$$\text{SE (Gene\_11} \geq 0{,}4\text{) E (Gene\_289} < 0{,}5\text{) ENTÃO Classe} = 2$$

O campo O pode variar entre os operadores $<$ (menor) e $\geq$ (maior ou igual). O campo de V é uma variável do tipo ponto flutuante que pode variar entre o menor e o maior valor encontrados na base de expressão gênica avaliada e a precisão (número de casas decimais) utilizada nesse campo é um parâmetro de execução do AG, que se mostrou bastante importante para a convergência de nossos experimentos.

4.1.2 Função de Avaliação ou Aptidão (FA) (*Fitness Function*)

A Aptidão (ou *fitness*) refere-se ao grau de contribuição de uma determinada solução candidata para a convergência do AG, na pesquisa da melhor solução dentro do espaço de busca, avaliando a qualidade de cada regra (indivíduo). A FA aplicada foi baseada em [63]. Para o entendimento da FA aqui aplicada, alguns conceitos precisam ser elucidados. Quando aplicamos uma regra na classificação sobre os dados de uma amostra (um registro da base de expressão gênica), quatro diferentes resultados podem ser observados, dependendo da classe predita pela regra e a da verdadeira classe da amostra. São eles:

- *True Positive* (*tp*) - A regra classifica a amostra em uma determinada classe e a amostra de fato pertence a essa classe;
- *False Positive* (*fp*) - A regra classifica a amostra em uma determinada classe, mas a mesma não pertence a essa classe;
- *True Negative* (*tn*) - A regra classifica a amostra como não pertencente a uma determinada classe e a amostra é de fato de outra classe;
- *False Negative* (*fn*) - A regra classifica a amostra como não pertencente a uma determinada classe, mas a amostra pertence à classe em questão;

A FA utiliza dois indicadores comumente utilizados em domínios médicos, chamados de sensibilidade (*Se*) e especificidade (*Sp*). *Se* e *Sp* são definidos abaixo:

$$Se = \frac{tp}{(tp + fn)} \tag{4.1}$$

$$Sp = \frac{tn}{(tn + fp)} \tag{4.2}$$

A FA utilizada é definida como o produto destes dois indicadores, *Se* e *Sp*, como segue abaixo:

$$Aptidao = Se \times Sp \tag{4.3}$$

O objetivo do AG é maximizar ao mesmo tempo *Se* e *Sp* e, conseqüentemente, o valor de *Aptidão*, utilizando-se para isso, as equações 4.1, 4.2 e 4.3. Em cada execução, o AG trabalha com um problema de classificação de duas classes, isto é, quando regras de uma dada classe C estão sendo mineradas, todas as outras classes são agrupadas em uma segunda classe (*not* C).

4.1.3 Operadores Genéticos

Na seleção dos pais para o *crossover*, na maioria das execuções do AG, aplicamos o método do torneio estocástico utilizando *tour* de tamanho 3 (três). Este método foi revisado na seção 3.1.3. Sobre os pais selecionados, aplicamos *crossover* múltiplo com dois pontos de corte, gerando dois novos filhos com taxa de *crossover* de 100%. Nestes dois filhos gerados, aplicamos o operador de mutação. Os operadores de mutação utilizados neste trabalho variam com o tipo do gene avaliado e foram aplicados a uma taxa de mutação por gene no valor de 30%.

Para o campo P do gene o novo valor é dado sorteando o incremento ou o decremento de uma unidade do valor corrente. A Figura 4.3 ilustra uma mutação aplicada ao campo P onde foi sorteado o incremento de uma unidade ao valor original. Para o campo O do gene ocorre a troca do operador corrente: se o operador for $<$, troca-se por $\geq$, e vice-versa. A Figura 4.4 demonstra como é feita a mutação no campo O trocando-se o operador $\geq$ por $<$. A mutação do campo V do gene é feita sorteando-se um incremento ou um decremento de 0,1 no valor corrente. Na Figura 4.5 foi sorteado o decremento de 0,1, que foi aplicado ao valor original deste campo. Na composição dos indivíduos que irão participar da próxima geração do AG, selecionamos os melhores pais e filhos.

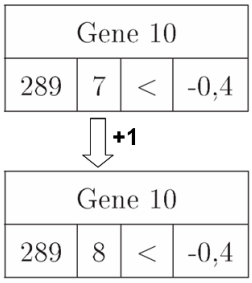

Figura 4.3: Mutação aplicada no campo P

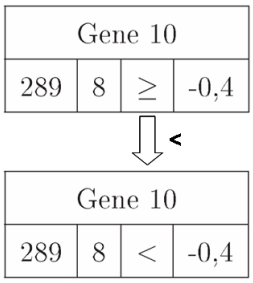

Figura 4.4: Mutação aplicada no campo *O*

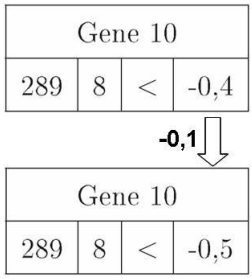

Figura 4.5: Mutação aplicada no campo *V*

4.1.4 Parâmetros Genéticos

Neste trabalho, após os ajustes que serão descritos na seção 4.2, utilizamos uma população formada por 400 indivíduos, taxa de *crossover* de 100%, taxa de mutação de 30% por gene e executamos o AG por 100 gerações. Embora essa taxa de mutação não seja usual em trabalhos que envolvem AGs, esse valor foi originalmente utilizado por Fidelis em [19]. Após avaliações experimentais, constatamos a importância de se usar essa taxa relativamente alta para uma boa convergência do AG.

4.1.5 Bases de Dados investigadas

A base de dados NCI60 descrita na seção 2.2 e apresentada no Apêndice A, foi obtida a partir de experimentos de *microarray* aplicados sobre 61 amostras de células cancerígenas resultando nos níveis de expressão de mais de 8.000 genes. Essa base foi obtida nos experimentos descritos na referência [20]. Posteriormente, Ooi e Tan [3] aplicaram alguns procedimentos simples de filtragem, excluindo os genes mais ruidosos, chegando a uma

base com a expressão de 1.000 genes. A partir dessa base, diferentes técnicas foram aplicadas para se chegar a conjuntos reduzidos de genes que fossem bons preditores das nove classes de câncer.

Nessa dissertação, utilizamos quatro conjuntos reduzidos de genes obtidos em [3]. Segundo Lin (2006) [24], a preleção gênica é necessária quando se trabalha com dados advindos de experimentos de *microarray* [24] e diversos outros trabalhos realizam algum tipo de pré-processamento antes de realizar o *data mining* propriamente dito [21, 12, 3, 13, 22, 23, 24].

O primeiro conjunto, chamado de G_1, foi minerado em [3] utilizando-se um AG e um classificador de máxima verossimilhança (MLHD) [96]. O conjunto AG/MLHD determina automaticamente quais genes farão parte do conjunto preditor. O melhor conjunto encontrado é formado por 13 genes preditivos. Estes genes são identificados pela sua posição dentro da base de 1000 genes que foi minerada. São eles: 11, 50, 97, 127, 194, 242, 289, 348, 366, 828, 839, 863 e 881.

O segundo conjunto, chamado de G_2, foi minerado utilizando um método B/W (*between-group/within-group*) empregado em [21] onde os genes são rankeados baseados na soma dos quadrados das relações entre *between-groups* e *within-groups*. Esta técnica foi proposta anteriormente em [21]. Após calcular o valor desta relação para cada gene, os mesmos foram rankeados decrescentemente e selecionados os top 20 genes. São eles: 2, 17, 18, 19, 28, 75, 97, 141, 224, 231, 235, 246, 280, 292, 302, 409, 499, 526, 637 e 843.

O terceiro conjunto, chamado de G_3, foi minerado utilizando-se uma adaptação do método descrito em [79], chamada S2N/OVA (*signal-to-noise/one-vs.-all*), podendo assim, ser aplicado em cenários multiclasse. Na formação deste conjunto, para cada classe, um conjunto de genes positivamente correlacionados (altos valores positivos para S2N) e outro, formado por genes negativamente correlacionados (pequenos valores negativos para S2N) são formados. Para cada classe foi selecionado o gene que possui o maior valor de relação S2N positivo e o gene que possui o menor valor de relação negativa para S2N, totalizando 18 genes. São eles: 2, 2, 41, 63, 97, 229, 379, 456, 475, 485, 525, 531, 637, 721, 786, 870, 890 e 929. Uma observação importante a ser colocada com relação ao conjunto B_3 refere-se à presença em duplicidade do gene 2. Em nossos experimentos retiramos todas

as duplicidades existentes; devido a este fato, o conjunto G_3 é composto por 17 e não por 18 genes. São eles: 2, 41, 63, 97, 229, 379, 456, 475, 485, 525, 531, 637, 721, 786, 870, 890 e 929.

O método empregado na construção do quarto conjunto, chamado de G_4, é uma variação do método empregado na construção do conjunto G_1. Foi empregado um AG em conjunto com um classificador MLHD [3], utilizando uma função de aptidão simplificada, ignorando uma das duas taxas de erro que compõem a função de aptidão original, utilizada na obtenção do conjunto G_1. O conjunto G_4 é composto por 12 genes: 11, 46, 177, 289, 306, 336, 380, 499, 661, 783, 865 e 950.

O objetivo da obtenção dos conjuntos G_2 e G_3 em [3], gerados a partir de técnicas de *ranking*, foi de compará-los com os genes preditivos obtidos pela técnica AG/MLHD. Uma das conclusões do trabalho é que os conjuntos reduzidos por técnicas diferentes não se sobreporam na maioria dos genes. Entretanto alguns genes aparecem em dois ou mais conjuntos. São eles:

- 2 (G_2 e G_3)
- 11 (G_1 e G_4)
- 97 (G_1, G_2 e G_3)
- 289 (G_1 e G_4)
- 499 (G_2 e G_4)
- 637 (G_2 e G_3)

A partir da composição dos quatro grupos G_1, G_2, G_3 e G_4, excluindo-se os genes duplicados, chegou-se a um total de 55 genes distintos, cujos níveis de expressão estão representados nas Tabelas 2, 3, 4, 5, 6 e 7 do apêndice B. Realizamos experimentos de mineração utilizando-se sub-conjuntos obtidos a partir desses 55 genes, que serão discutidos nas seções 5.1 e 5.2.

O objetivo dessa mineração é partir de um conjunto reduzido de genes, construídos a partir de outras técnicas de *data mining*, e chegar em regras de alto nível, do tipo IF-THEN que não só sejam associadas a cada classe individualmente, reduzindo o problema

a poucos genes por classe, mas também associando o nível de expressão gênica a cada gene que compõe a regra.

4.2 Ajuste do Ambiente Evolutivo

O ajuste do ambiente foi realizado em três etapas e partiu da configuração dos parâmetros utilizados em [19]. Na primeira etapa foram ajustados os operadores genéticos. A segunda etapa contemplou o ajuste dos parâmetros genéticos e a terceira analisou a precisão do campo O do gene.

1ª etapa: Escolha dos métodos de seleção e reinserção.

A primeira etapa consistiu em avaliar os métodos de seleção de pais para o *crossover* e reinserção. Dentre os métodos de seleção existentes, analisamos o métodos conhecidos como roleta e torneio estocástico, que foram revisados na seção 3.1.3. Os métodos de reinserção avaliados foram o elitismo e os melhores pais e filhos (*steady-state*), que também foram revisados na seção 3.1.3.

Para essa avaliação, os valores do tamanho da população e do número de gerações foram fixados em $T_p = 50$ e $N_{ger} = 50$. Avaliamos as seguintes combinações: roleta + elitismo, roleta + melhores pais e filhos, torneio estocástico + elitismo e torneio estocástico + melhores pais e filhos. Como principal conclusão dessa etapa, temos que os melhores resultados foram encontrados com a combinação torneio estocástico + melhores pais e filhos.

2ª etapa: Ajuste dos parâmetros genéticos

A segunda etapa consistiu em ajustar os parâmetros genéticos do AG. Para este ajuste, foram avaliados os valores 100, 200 e 400 para o tamanho da população (T_p); 50, 100 e 200 para o número de gerações (N_{ger}); e 2, 3 e 4 para o tamanho do *tour* do torneio estocástico.

Fixamos o método de seleção (torneio estocástico) e o método de reinserção (melhores pais e filhos), ajustados na etapa anterior, e avaliamos os resultados encontrados com a combinação de três valores para T_p (100, 200 e 400), três valores para N_{ger} (50, 100 e 200) e três valores para o *tour* do método torneio estocástico (2, 3 e 4). Os melhores resultados

foram encontrados com $T_p = 400$, $N_{ger} = 100$ e $tour = 3$. É importante salientar que mesmo ao aumentar o N_{ger} para 200 gerações, não houve uma melhoria significativa nos valores encontrados que justificasse a opção por este valor, visto que, a escolha de $N_{ger} = 200$ levaria a um aumento significativo no tempo de processamento do AG. Assim, utilizamos $N_{ger} = 100$;

3ª etapa: Precisão do campo Operador do gene

Na terceira etapa foi utilizado os valores 1, 2 e 3 para o número de casas decimais após a vírgula para o campo O do cromossomo.

Fixando o método de seleção (torneio estocástico), o método de reinserção (melhores pais e filhos), $T_p = 400$, $N_{ger} = 100$ e $tour = 3$, fizemos experimentos utilizando 1, 2 e 3 casas após a vírgula no campo O. A convergência para boas regras de classificação foi significativamente superior utilizando apenas 1 casa após a vírgula. Após esclarecimentos junto aos especialistas, que confirmaram ser essa precisão ideal para a interpretação das regras obtidas, resolvemos manter a precisão em apenas uma casa decimal.

Após todas as etapas de ajuste, chegamos a um ambiente cuja especificação foi utilizada em todos os experimentos descritos nas próximas seções.

- Método de seleção: torneio estocástico
- Método de reinserção: melhores pais e filhos
- $T_p = 400$
- $N_{ger} = 100$
- $tour = 3$
- Número de casas após a vírgula: 1

Capítulo 5

Resultados

Neste capítulo, serão apresentados os resultados dos principais experimentos conduzidos na mineração de bases de dados extraídas a partir da base NCI60 [20].

Inicialmente, o ambiente evolutivo construído para a mineração das regras foi aplicado sobre quatro bases de dados criadas a partir dos quatro conjuntos de genes obtidos em Ooi e Tan [3], citados na seção 4.1.5, chamadas nesta dissertação de B_1, B_2, B_3 e B_4. Os resultados obtidos em casa base individual foram analisados e comparados. A seção 5.1 apresenta estes resultados.

Numa segunda etapa, na tentativa de obter resultados ainda melhores que os obtidos nas bases individuais, novas bases foram criadas a partir das composição (2 a 2, 3 a 3 e completa) das bases B_1, B_2, B_3 e B_4. Os principais resultados obtidos nesta etapa são discutidos na seção 5.2.

Finalmente, a seção 5.3 faz uma análise mais detalhada dos melhores resultados obtidos nas duas etapas de experimentos. Dessa análise, dois conjuntos de regras denominados K_1 e K_2 foram extraídos dentre as melhores regras. Análises comparativas entre esses conjuntos e os principais classificadores encontrados na literatura para a base NCI60 são apresentados onde é possível constatar que os resultados obtidos nessa dissertação são bastante competitivos com os publicados por outros autores.

5.1 Experimentos com a mineração das bases reduzidas individuais

Quatro bases de dados reduzidas foram criadas a partir da base de 1.000 genes disponibilizada por Ooi e Tan [3]. Os genes utilizados nessas 4 bases correspondem aos conjuntos reduzidos também em [3], que chamamos de G_1, G_2, G_3 e G_4, resultando nas bases B_1, B_2, B_3 e B_4.

Inicialmente, o AG foi aplicado em cada uma dessas quatro bases individualmente. Conforme mencionado na seção 2.2, a base NCI60 é composta por 61 amostras categorizadas em 9 classes de câncer. Portanto, o objetivo da mineração é obter regras de classificação para essas nove classes. A avaliação da qualidade das regras mineradas foi feita inicialmente por classe.

Na avaliação por classe, cada base composta por 61 amostras foi dividida em 3 partições de tamanhos semelhantes, guardando sempre a proporcionalidade entre o número de amostras de cada classe. Duas partições foram utilizadas em treinamento e a terceira partição, chamada de teste, foi utilizada para a avaliação do nível de generalização das regras obtidas em treinamento. Isto é, as regras que foram evoluídas pelo AG, usando a junção das partições 1 e 2, posteriormente foram testadas na partição 3 (12->3). O mesmo procedimento foi realizado para as demais combinações: partições 1 e 3 em treinamento e a partição 2 em teste (13->2) e as partições 2 e 3 em treinamento e a partição 1 em teste (23->1). Cada um desses experimentos (12->3, 13->2 e 23->1) foi composto por 50 execuções para cada uma das nove classes possíveis do atributo objetivo. A avaliação de cada regra obtida é dada pela $Aptidao_{Trein}$ e $Aptidao_{Teste}$ (equações 4.1, 4.2 e 4.3). Como cada base é formada por 61 amostras, buscou-se manter a proporcionalidade entre as classes em cada partição. Assim, cada partição possui aproximadamente 20 amostras da base.

A Tabela 5.1 apresenta os resultados de $Aptidao_{Trein}$ e $Aptidao_{Teste}$ das melhores regras obtidas para a base B_1, a partir de 50 execuções do AG, para cada uma das 9 classes e para cada experimento de teste (12->3, 13->2 e 23->1).

Para estabelecermos um conjunto de regras de classificação, as melhores regras encon-

Tabela 5.1: Aptidão de treinamento e aptidão de teste das melhores regras evoluídas na base B_1

	Experimento 12->3			**Experimento 13->2**			**Experimento 23->1**	
Classes	$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	0,8	0		0,812	0		0,971	0,317
2	1	1		1	1		1	1
3	1	0,5		1	1		1	1
4	1	1		1	0,938		1	0,5
5	1	1		1	1		1	1
6	1	0		1	0,667		1	0,333
7	1	0		1	0,5		1	0,476
8	1	0,667		1	1		1	0,95
9	1	0,895		1	0		1	0,857

tradas nos experimentos da base B_1, independentemente do experimento de teste utilizado (12->3, 13->2 ou 23->1), foram agrupadas e são apresentadas na Tabela 5.2. Os valores de $Aptidao_{Trein}$ e $Aptidao_{Teste}$ dessas regras também são apresentados nesta tabela.

Tabela 5.2: Melhores regras encontradas na base de dados B_1

Classe	**Regra**	$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	if(127<0,6) and (289<0,1) and (348<-0,2) and (366≥-0,1) and (839<0,9)	0,971	0,317
2	if(11≥0,4) and (289<-0,5)	1	1
3	if(50<-2,3) and (194<-1,1) and (289≥-0,3) if(50<-2,3) and (194<-1,1) and (839≥-0,8)	1	1

4	if(11<-2,3) and (50≥-2,1) and (366<-0,1)
	if(50≥-2,1) and (127<-0,7) and (366<-0,1)
	if(50≥-2,1) and (194<-0,7) and (366<-0,1)
	if(50≥-2,1) and (348<0,2) and (366<-0,1)
	if(50≥-2,2) and (366<-0,1) and (881<-0,2)
		1	1

5	if(11≥-1,5) and (97<0,1) and (348<-1,5)	1	1
6	if(97≥-1,4) and (242<0,3) and (828<0,1) and (839≥-0,5) and (863≥-0,3)	1	0,667
7	if(97<1,4) and (194≥0,2) and (839<-0,2) if(194≥0,2) and (242≥-0,1) and (839<-0,2)	1	0,5

8	if(97≥0,7) and (127≥0,3) and (863<0,7)
	if(97≥0,7) and (348<-0,8) and (863<0,7)
	if(97≥0,7) and (863<0,8) and (881≥-0,3)
		1	1

9	if(50<-2,1) and (289<-0,3) and (839≥-1,3)	1	0,895

Para a base B_1, foram encontrados ótimos resultados (100% em treinamento e em teste) para 5 das 9 classes existentes (2, 3, 4, 5 e 8). Resultados razoáveis foram encontrados para a classe 9 (100% em treinamento e 89,5% em teste). Resultados inferiores

foram encontrados para as demais classes. O melhor resultado para a classe 1 foi 97,1% em treinamento e 31,7% em teste e para as classes 6 e 7 foram encontrados 100% em treinamento e 66,7% e 50% em teste, respectivamente. Assim, obtivemos bons resultados em 6 das 9 classes mineradas. Para algumas dessas classes (3, 4, 7 e 8), mais de uma regra foi obtida com o maior valor de aptidão.

O mesmo procedimento foi realizado utilizando-se as bases B_2, B_3 e B_4. As tabelas detalhadas por partição para estas bases são apresentados no apêndice D. As Tabelas 5.3, 5.4 e 5.5 apresentam as melhores regras obtidas (considerando-se os 3 experimentos de teste) e os valores de aptidão associados a elas. Em relação à base B_2, foram encontrados ótimos resultados (100% em treinamento e em teste) para 3 das 9 classes existentes (4, 5 e 8) e bons resultados para outras 4 classes (2, 3, 7 e 9). Para as demais classes, 1 e 6, não foram encontrados resultados satisfatórios. Assim, obtivemos bons resultados em 7 das 9 classes mineradas.

Em relação à base B_3, foram encontrados ótimos resultados (100% em treinamento e em teste) para 4 das 9 classes existentes (2, 4, 5 e 9) e bons resultados para as classes 3 e 8. Para as demais classes (1, 6 e 7) não foram encontrados resultados satisfatórios. Assim, obtivemos bons resultados em 6 das 9 classes mineradas.

Para a base B_4, foram encontrados ótimos resultados (100% em treinamento e em teste) para 3 das 9 classes existentes (2, 3 e 5) e bons resultados para as classes 4, 8 e 9. Para as demais classes (1, 6 e 7), foram encontrados resultados insatisfatórios. Assim, obtivemos bons resultados em 6 das 9 classes mineradas.

Assim, independentemente da base utilizada na mineração, foi possível encontrar regras perfeitas (100% em treinamento e em teste) ou eficazes (acima de 90% de média entre $Aptidao_{Trein}$ e $Aptidao_{Teste}$) em 6 das 9 classes analisadas: 2, 3, 4, 5, 8 e 9. Com relação à classe 7, apenas o experimento com a base B_2 foi capaz de encontrar uma regra eficaz. Para as classes 1 e 6, nenhum experimento conseguiu evoluir regras com eficácia razoável.

Uma análise conjunta desses experimentos com as bases individuais foi feita agrupando-se as melhores regras obtidas para cada classe, independentemente da base utilizada. A Tabela 5.6 apresenta essas regras assim como a $Aptidao_{Trein}$ e $Aptidao_{Teste}$, representando assim a qualidade de cada regra separadamente. Com exceção da classe 9, em todas as

Tabela 5.3: Melhores regras encontradas na base de dados B_2

Classe	**Regras**	$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	if(28<1,4) and (97<0,8) and (409≥-0,4) and (499<0,2) and (526≥-0,1)	0,944	0,472
2	if(17≥-0,4) and (97≥0,1) and (637<0,5) if(235<1) and (246≥0,8) and (302≥0,1)	1	0,952
3	if(75≥-0,7) and (246<-0,4)	1	0,875

4	if(19<-0,4) and (526<-0,9)
	if(19<-0,4) and (843<-1)
	if(224<-2,2) and (843<-1)
	if(409≥-1,8) and (843<-1)
		1	1

5	if(2<-2,4) and (18<0,2) and (28≥-0,3) and (97<0,1)
	if(2<-2,4) and (18<0,1) and (97<0,1) and (224≥-0,6)
	if(2<-1,5) and (18<0,8) and (97<0,2) and (246≥-0,7)
	if(2<-1,8) and (18≥-2,6) and (97<0,2) and(292<0,8)
	if(2<-2,5) and (28≥-0,3) and (97<0,1) and (292<0,5)
	if(2<-2,4) and (28≥-0,3) and (97<0,1) and (302<0,1)
	if(2<-2,4) and (97<0,1) and (224≥-0,6) and (292<0,5)
	if(2<-2,4) and (97<0,1) and (224≥-0,6) and (302<0,1)
	if(2<-2,5) and (97<0,1) and (246≥-0,7) and (292<0,5)
	if(2<-1,5) and (97<0,1) and (246≥-0,7) and (302<0,1)
	if(2<-2,4) and (97<0,2) and (292<0,5) and (409≥-0,9)
	if(2<-2,4) and (97<0,1) and (302<0,1) and (409≥-1)
	if(18<0,4) and (19<-1,5) and (28≥-0,7) and (97<0,1)
	if(18<0,5) and (19<-1,8) and (97<0,1) and (224≥-0,8)
	if(18<0,5) and (19<-1,8) and (97<0,1) and (246≥-0,6)
	if(18<0,5) and (19<-2,4) and (97<0,1) and (409≥-1,2)
	if(19<-2,6) and (28≥-0,4) and (97<0,1) and (292<0,6)
	if(19<-2,6) and (28≥-0,3) and (97<0,1) and (302<0,1)
	if(19<-2,4) and (28≥-0,4) and (97<0,1) and (637≥-0,3)
	if(19<-2,5) and (97<0,1) and (224≥-0,6) and 292<0,5)
	if(19<-2,5) and (97<0,1) and (224≥-0,6) and (302<0,1)
	if(19<-2,5) and (97<0,1) and (224≥-0,6) and (637≥-0,5)
	if(19<-2,6) and (97<0,1) and (246≥-0,7) and (292<0,5)
	if(19<-2,6) and (97<0,1) and (246≥-0,5) and (302<0,1)
	if(19<-2,6) and (97<0,2) and (246≥-0,7) and (637≥-0,7)
	if(19<-2,4) and (97<0,1) and (302<0,1) and (409≥-0,9)
	if19<-2,4) and (97<0,1) and (409≥-1,2) and (637≥-0,9)
		1	1

6	if(17<1,8) and (28<1,1) and (235≥-0,2) and (409<2,3) and (637≥0,4)	1	0,431
7	if(2≥-2,1) and (97<1,4) and (224≥-0,2)	0,971	1
8	if(18<1,5) and (97≥0,7) and (280≥0,1) and (409≥0,4) if(97≥0,7) and (246<1) and (280≥0,1) and (409≥0,4)	1	1

9	if(19≥-0,2) and (231<-1,4)
	if(19≥-0,3) and (499<-1,1)
	if(224≥-2,2) and (231<-1)
		1	0,952

Tabela 5.4: Melhores regras encontradas na base de dados B_3

Classe	**Regras**	$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	if(531≥0,2) and (*70≥0) an d(929<0,2)	1	0,3
2	if(229≥1,1) and (456≥-0,9)	1	1

3	if(2≥-0,4) and (379≥0,10 and (475<0,1) and (485<0,1)
	if(2≥-0,4) and (379≥0) and (475<0,1) and (929≥-1,2)
	if(63≥-0,3) and (97<0) and (379≥0,1) and (475≥-2,8)
	if(63≥-0,2) and (97<-0,2) and (379≥0,1) and (485<0)
	if(63≥-0,3) and (475<0,2) and 485<0,2) and (637<0,7)
		1	0,938

4	if(63<-0,3) and 485≥0,7)
	if(229≥-1,6) and (485≥0,7)
	if(456<1,2) and (485≥0,7)
	if(485≥0,7) and (525≥-1,1)
	if(485≥0,7) and (929≥-0,3)
		1	1

5	if(41≥-2,1) and (97<0,1) and (721≥1)
	if(97<0,1) and (379<0,1) and (721≥1)
	if(97<0,1) and (475≥-0,5) and (721≥1)
		1	1

6	if(2<-1,3) and (379≥0,2) and (456≥-1,2) and (637≥0,4)	1	0,667

7	if(63≥0,9) and (97≥-1)
	if(63≥0,9) and (379<0,7)
	if(63≥0,9) and (475≥-0,2)
	if(63≥0,9) and (890<-0,6)
		1	0,472

8	if(63<-0,4) and (97≥0,7) and (870<0,4)	1	0,875
9	if(2≥-0,5) and (485≥-1,5) and (786<-0,6)	1	1

classes para as quais foi possível encontrar regras perfeitas (100% em treinamento e teste), também foi possível encontrar mais de uma regra.

Uma outra forma de análise foi feita sobre este conjunto de regras, na qual foi elaborado um classificador composto de uma regra de cada classe, para posteriormente, verificarmos sua taxa de acertos na base completa (61 amostras). Para realizar esta análise, foi necessário selecionar apenas uma regra de cada classe, sendo que o critério adotado para a seleção destas regras foi pegar a primeira ocorrência para cada classe. De posse das 9 regras, aplicamos estas regras no conjunto de dados compreendido por 1000 genes e 61 amostras da base NCI60 [20]. O conjunto de regras do classificador avaliado é apresentado na Tabela 5.7.

Denominamos esse procedimento de análise AECD (Acerto | Erro Grave | Confusão | Desconhecimento). Este método consiste em analisar um registro da base de cada vez,

Tabela 5.5: Melhores regras encontradas na base de dados B_4

Classe	**Regras**	$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	if(46<1,8) and (289<0,5) and (306<0,3) and (783<0,1) and (865≥-1,2)	0,917	0,444
2	if(11≥0,4) and (289<-0,5)	1	1
3	if(46<-0,7) and (289≥-0,5) and (306≥-0,6) and (336<-0,3)	1	1

4	if(11<-2,7) and (289≥-0,9) and (865≥0,1)
	if(11<-2,8) and 856≥0,1) and (950≥0)
	if(289≥-0,9) and (499<-0,8) and (865≥0,1)
	if(499<-0,8) and (865≥0,1) and (950≥0)
		1	0,952

5	if(11≥-1,5) and (289≥-1,3) and (380<-0,7)
	if(177≥-1,4) and (289≥-1,3) and (380<-0,7)
	if(289≥-1,3) and (306<-1) and (380<-0,7)
	if(289≥-1,3) and (336≥-0,7) and (380<-0,7)
	if(289≥-1,5) and (380<-0,7) and (661≥-1,2)
	if(289≥-1,3) and (380<-0,7) and (865≥-0,9)
	if(289≥-1,3) and (380<-0,7) and (950≥0)
		1	1

6	if(306≥-0,9) and (380<0,2) and (661≥-0,4)	1	0,314
7	if(46≥-0,7) and (306≥-0,5) and (499<0,3)	1	0,5
8	if(46≥0,8) and (865<-0,4)	1	0,938
9	if(11≥-3,6) and (177<-2) if(177<-2,2) and (783≥-0,5)	0,974	1

correspondente a uma amostra de célula, e este registro pode ser interpretado como acerto, erro grave, confusão ou um desconhecimento, dependendo do resultado de classificação. Um acerto ocorre quando somente a regra que possui a mesma classe do registro é disparada. Por exemplo, se o registro avaliado é da classe 1 somente a regra da classe 1 dispara na classificação deste registro. Um erro grave ocorre quando a regra correspondente à sua classe não é disparada na classificação do registro e uma outra regra de classe diferente é disparada. Por exemplo, o registro é da classe 1 e na classificação a regra da classe 1 não dispara enquanto que a regra da classe 2 dispara. Uma confusão acontece quando o registro é classificado pela regra da sua classe e por uma outra regra de outra classe. Por exemplo, o registro é da classe 1 e as regras da classe 1 e da classe 2 disparam. Um desconhecimento ocorre quando nenhuma regra é disparada na classificação do registro, nem da mesma classe e nem de outras classes.

O resultado da análise AECD utilizando as regras da Tabela 5.7 como um classificador da base NCI60 retornou um percentual de acerto de de 90,16% nos 61 registros da base,

Tabela 5.6: Melhores regras encontradas para o conjunto de bases B_1, B_2, B_3 e B_4

Classes	**Regras**	$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	if(28<1,4) and (97<0,8) and (409≥-0,4) and (499<0,2) and (526≥-0,1)	0,944	0,472
2	if(11≥0,4) and (289<-0,5) if(229≥1,1) and (456≥-0,9)	1	1
3	if(50<-2,3) and (194<-1,1) and (289≥-0,3) if(50<-2,3) and (194<-1,1) and (839≥-0,8)	1	1

4	if(19<-0,4) and (526<-0,9)
	if(19<-0,4) and (843<-1)
	if(63<-0,3) and (485≥0,7)
	if(224<-2,2) and (843<-1)
	if(229≥-1,6) and (485≥0,7)
	if(409≥-1,8) and (843<-1)
	if(456<1,2) and (485≥0,7)
	if(485≥0,7) and (525≥-1,1)
	if(485≥0,7) and (929≥-0,3)
		1	1

5	if(11≥-1,5) and (97<0,1) and (348<-1,5)
	if(11≥-1,5) and (289≥-1,3) and 380<-0,7)
	if(41≥-2,1) and (97<0,1) and (721≥1)
	if(97<0,1) and (379<0,1) and (721≥1)
	if(97<0,1) and (475≥-0,5) and (721≥1)
	if(177≥-1,4) and (289≥-1,3) and (380<0,7)
	if(289≥-1,3) and (306<-1) and (380<-0,7)
	if(289≥-1,3) and (336≥-0,7) and (380<-0,7)
	if(289≥-1,5) and (380<-0,7) and (661≥-1,2)
	if(289≥-1,3) and (380<-0,7) and (865≥-0,9)
	if(289≥-1,3) and 380<-0,7) and (950≥0)
		1	1

6	if(2<-1,3) and (379≥0,2) and (456≥-1,2) and (637≥0,4)	1	0,667
7	if(2≥-2,1) and (97<1,4) and (224≥-0,2)	0,971	1

8	if(97≥0,7) and (127≥0,3) and (863<0,7)
	if(97≥0,7) and (348<-0,8) and (863<0,7)
	if(97≥0,7) and (863<0,8) and (881≥-0,3)
		1	1

9	if(2≥-0,5) and (485≥-1,5) and (786<-0,6)	1	1

sendo 55 acertos, nenhum erro grave, 4 confusões e 2 desconhecimentos.

Os resultados obtidos nas análises efetuadas nas bases individuais geraram dois artigos que foram submetidos e aprovados em dois congressos, SBAI 2007 (Simpósio Brasileiro de Automação Inteligente) e BIBE 2007 (*IEEE* 7^{th} *International Symposium on Bioinformatics and Bioengeneering*), sendo que no primeiro o artigo foi aceito completo e no segundo como resumo expandido. O artigo completo [97] é apresentado no apêndice F.

Tabela 5.7: Conjunto de regras do classificador

Classes	Regras	$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	if(28<1,4) and (97<0,8) and (409≥-0,4) and (499<0,2) and (526≥-0,1)	0,944	0,472
2	if(11≥0,4) and (289<-0,5)	1	1
3	if(50<-2,3) and (194<-1,1) and (289≥-0,3)	1	1
4	if(19<-0,4) and (526<-0,9)	1	1
5	if(11≥-1,5) and (97<0,1) and (348<-1,5)	1	1
6	if(2<-1,3) and (379≥0,2) and (456≥-1,2) and (637≥0,4)	1	0,667
7	if(2≥-2,1) and (97<1,4) and (224≥-0,2)	0,971	1
8	if(97≥0,7) and (127≥0,3) and (863<0,7)	1	1
9	if(2≥-0,5) and (485≥-1,5) and (786<-0,6)	1	1

5.2 Experimentos com a mineração das bases compostas

Conforme apresentado na seção anterior (5.1), a mineração de regras realizada pelo AG sobre as bases individuais retornou resultados bons para 7 das 9 classes envolvidas na base NCI60. Entretanto, para as classes 1 e 6 o resultado foi insatisfatório. Cabe ressaltar que essa mesma dificuldade nas classes 1 e 6 da base NCI60 foi observada por Dudoit e colaboradores em [21]. Assim, partimos para uma nova etapa de experimentos, na qual bases com um número maior de genes foram utilizadas durante a fase de treinamento realizada pelo AG. Esperávamos ser possível melhorar os resultados para essas duas classes, sem decair a eficácia das outras sete. Assim, foram realizadas diferentes composições das quatro bases B_1 (13 genes), B_2 (20 genes), B_3 (17 genes) e B_4 (12 genes), associadas 2 a 2, 3 a 3 e 4 a 4, excluindo-se os genes repetidos, gerando outras 11 bases. São elas: B_1B_2 (32 genes), B_1B_3 (29 genes), B_1B_4 (23 genes), B_2B_3 (34 genes), B_2B_4 (31 genes), B_3B_4 (29 genes), $B_1B_2B_3$ (46 genes), $B_1B_2B_4$ (41 genes), $B_1B_3B_4$ (39 genes), $B_2B_3B_4$ (45 genes) e $B_1B_2B_3B_4$ (55 genes). Nos experimentos envolvendo as 11 bases compostas, foi utilizado o mesmo procedimento empregado no caso das bases individuais: a base completa foi dividida em 3 partições contendo aproximadamente 1/3 das amostras. Depois o AG foi evoluído em três experimentos diferentes: 12->3, 13->2 e 23->1. Os resultados completos desses experimentos são apresentados por partição no apêndice D. As melhores regras obtidas em cada experimento são apresentadas no apêndice E. Na Tabela 5.8, apresentamos

os valores de aptidão de treinamento e de teste para as melhores regras evoluídas em cada experimento, independentemente do experimento de teste (partições) em que foram mineradas. Na tabela também reproduzimos os valores das melhores aptidões obtidas nas bases individuais B_1, B_2, B_3 e B_4, para facilitar a comparação com os novos experimentos.

Tabela 5.8: Resultados encontrados para as bases de dados individuais e para todas as composições

Bases	B_1			B_2			B_3			B_4	
Classes	Apt_{Trein}	Apt_{Teste}		Apt_{Trein}	Apt_{Teste}		Apt_{Trein}	Apt_{Teste}		Apt_{Trein}	Apt_{Teste}
1	0,971	0,317		0,944	0,472		1	0,3		0,917	0,444
2	1	1		1	0,952		1	1		1	1
3	1	1		1	0,875		1	0,938		1	1
4	1	1		1	1		1	1		1	0,952
5	1	1		1	1		1	1		1	1
6	1	0,667		1	0,431		1	0,667		1	0,314
7	1	0,5		0,971	1		1	0,472		1	0,5
8	1	1		1	1		1	0,875		1	0,938
9	1	0,895		1	0,952		1	1		0,974	1

Bases	B_1B_2			B_1B_3			B_1B_4			B_2B_3	
Classes	Apt_{Trein}	Apt_{Teste}		Apt_{Trein}	Apt_{Teste}		Apt_{Trein}	Apt_{Teste}		Apt_{Trein}	Apt_{Teste}
1	1	0,333		1	0,533		1	0,317		0,947	0,375
2	1	1		1	1		1	1		1	1
3	1	0,633		1	0,938		1	1		1	0,875
4	1	1		1	1		1	0,5		1	1
5	1	1		1	1		1	1		1	1
6	1	0,667		1	0,627		1	0,333		0,973	0,933
7	1	0,5		1	0,952		1	0,938		1	0,944
8	1	1		1	1		1	0,95		1	0,95
9	1	0,952		1	0,952		1	1		1	1

Bases	B_2B_4			B_3B_4			$B_1B_2B_3$			$B_1B_2B_4$	
Classes	Apt_{Trein}	Apt_{Teste}		Apt_{Trein}	Apt_{Teste}		Apt_{Trein}	Apt_{Teste}		Apt_{Trein}	Apt_{Teste}
1	1	0,283		1	0,3		1	0,3		0,972	0,389
2	1	1		1	1		1	1		1	1
3	1	0,875		1	1		1	0,938		1	0,875
4	1	1		1	1		1	1		1	1
5	1	1		1	1		1	1		1	1
6	1	0,333		1	0,633		1	0,622		1	0,533
7	1	0,944		1	0,944		1	0,952		1	0,938
8	1	0,875		1	0,938		1	1		1	0,938
9	1	1		1	0,952		1	0,952		1	0,952

Bases	$B_1B_3B_4$			$B_2B_3B_4$			$B_1B_2B_3B_4$	
Classes	Apt_{Trein}	Apt_{Teste}		Apt_{Trein}	Apt_{Teste}		Apt_{Trein}	Apt_{Teste}
1	1	0,3		0,972	0,444		0,972	0,5
2	1	1		1	1		1	1
3	1	0,938		1	0,875		1	0,938
4	1	1		1	1		1	1
5	1	1		1	1		1	1
6	1	0,333		1	0,588		1	0,549
7	1	0,952		1	0,938		1	0,952
8	1	0,938		1	0,938		1	0,667
9	1	0,952		1	1		1	1

A Tabela 5.9 apresenta os resultados obtidos para cada classe analisada, em todos os experimentos. Foi considerado um resultado satisfatório se foi encontrada uma regra perfeita (100% em treinamento e teste) ou uma regra com pelo menos 90% de treinamento e 85% de teste. O valor encontrado entre parênteses, refere-se ao número de genes presente nas bases de dados. O melhor resultado foi obtido na mineração da base composta B_2B_3, na qual foi obtido um resultado insatisfatório apenas para a classe 1. Em seguida, podemos destacar os resultados das bases B_2, B_1B_3, B_2B_4, B_3B_4, $B_1B_2B_3$, $B_1B_2B_4$, $B_1B_3B_4$ e $B_2B_3B_4$; que retornaram resultados insatisfatórios apenas para as classes 1 e 6. Quando comparamos os resultados obtidos pelas bases individuais e os resultados obtidos pelas composições de bases, percebemos que apenas a base B_2B_3 conseguiu superar os resultados encontrados para as bases individuais, que retornaram resultados insatisfatórios em duas ou três classes. Um outro ponto a ser destacado, refere-se aos resultados obtidos pela composição das quatro bases ($B_1B_2B_3B_4$) que retornou resultados inferiores aos obtidos pelas bases individuais. Exceto no experimento com essa base "completa", o nosso AG se manteve robusto, não decaindo o desempenho com o aumento de genes nas bases e até superando os resultados obtidos nas bases individuais em algumas das bases analisadas.

Tabela 5.9: Classes que obtiveram ótimos/bons e ruins resultados para todas as bases

Bases	**Classes com resultados satisfatórios**	**Classes com resultados insatisfatórios**
B_1 (13)	2, 3, 4, 5 e 9	1, 6 e 7
B_2 (20)	2, 3, 4, 5, 7, 8 e 9	1 e 6
B_3 (17)	2, 3, 4, 5, 8 e 9	1, 6 e 7
B_4 (12)	2, 3, 4, 5, 8 e 9	1, 6 e 7
B_1B_2 (32)	2, 4, 5 e 8 e 9	1, 3, 6 e 7
B_1B_3 (29)	2, 3, 4, 5, 7, 8 e 9	1 e 6
B_1B_4 (23)	2, 3, 5, 7, 8 e 9	1, 4 e 6
B_2B_3 (34)	2, 3, 4, 5, 6, 7, 8 e 9	1
B_2B_4 (31)	2, 3, 4, 5, 7, 8 e 9	1 e 6
B_3B_4 (29)	2, 3, 4, 5, 7, 8 e 9	1 e 6
$B_1B_2B_3$ (46)	2, 3, 4, 5, 7, 8 e 9	1 e 6
$B_1B_2B_4$ (41)	2, 3, 4, 5, 7, 8 e 9	1 e 6
$B_1B_3B_4$ (39)	2, 3, 4, 5, 7, 8 e 9	1 e 6
$B_2B_3B_4$ (45)	2, 3, 4, 5, 7, 8 e 9	1 e 6
$B_1B_2B_3B_4$ (55)	2, 3, 4, 5, 7 e 9	1, 6 e 8

Por outro lado, essa base contempla todos os 55 genes utilizados nos outros 14 ex-

perimentos. Portanto, potencialmente, a base $B_1B_2B_3B_4$ contém todas as informações utilizadas nos outros experimentos. Assim, o AG não foi capaz de convergir para regras eficazes. Esse fato pode sinalizar que o ajuste realizado para nosso AG começou a decair o desempenho com o aumento do número de genes manipulados. Outro fato que corrobora essa observação é que algumas bases compostas por duas individuais (B_1B_3 e B_2B_3) retornaram melhores resultados do que as bases compostas por três individuais. Assim, observamos que a convergência do AG para regras eficazes começa a decair quando analisamos conjuntos maiores que aproximadamente 40 genes.

A análise AECD também foi aplicada às regras mineradas a partir de cada composição de base. A Tabela 5.10 ilustra os resultados encontrados para todas as combinações das bases individuais B_1, B_2, B_3 e B_4. O melhor resultado foi encontrado para as regras mineradas a partir da base B_1B_2 atingindo 90,16% de acertos, ou seja, o mesmo resultado alcançado pelo conjunto de regras das melhores regras obtidas nas bases individuais, apresentado na Tabela 5.7. O segundo melhor resultado foi obtido pelas regras mineradas a partir da base B_1B_4 e a partir da base $B_1B_2B_3$, atingindo 86,89% de acertos.

Tabela 5.10: Análise AECD para todas as combinações de bases

Base	Acerto	Erro Grave	Confusão	Desconhecimento	Taxa de Acerto
B_1B_2	55	1	1	4	90,16%
B_1B_3	52	1	7	1	85,25%
B_1B_4	53	0	3	5	86,89%
B_2B_3	49	0	11	1	80,33%
B_2B_4	50	0	7	4	81,97%
B_3B_4	52	0	6	3	85,25%
$B_1B_2B_3$	53	1	5	3	86,89%
$B_1B_2B_4$	49	1	10	1	80,33%
$B_1B_3B_4$	52	1	5	3	85,25%
$B_2B_3B_4$	50	0	9	2	81,97%
$B_1B_2B_3B_4$	47	1	10	3	77,05%

5.3 Análise das melhores regras e dos melhores conjuntos

A Tabela 5.11 ilustra as melhores regras, e seus respectivos valores de aptidão, obtidas em todo o conjunto de bases, independentemente das bases utilizadas na mineração e da partição utilizada como teste. Foi possível encontrar regras perfeitas (100% em treinamento e em teste) ou eficazes (acima de 90% de média entre $Aptidao_{Trein}$ e $Aptidao_{Teste}$) em oito das nove classes analisadas: 2, 3, 4, 5, 6, 7, 8 e 9. Apenas a aptidão da melhor regra encontrada para a classe 1 foi abaixo do desejado: 76,65% em média (treinamento e teste). Quando comparamos este resultado com o encontrado para as bases individuais, constatamos uma melhoria significativa para a classe 6 e uma melhoria pouco signifcativa para a classe 1. Na classe 6, o melhor resultado encontrado, utilizando-se apenas uma base individual, foi igual a 83,35% de aptidão em média, resultado este, minerado na base B_1. Com a composição de bases, conseguimos elevar este valor para 95,3%, valor este encontrado na base B_2B_3. Para a classe 1, a melhoria foi menos significativa: o melhor resultado em base individual foi igual a 70,8% de média, minerada na base B_2, e na base composta, foi igual a 76,65% em média (B_1B_3). Assim, com a composição das bases individuais conseguimos efetivamente melhorar os valores de aptidão somente para a classe 6. Entretanto, mesmo nas outras classes onde o desempenho já havia sido satisfatório na mineração das bases individuais, foi possível encontrar um número maior de regras perfeitas.

Tabela 5.11: Melhores regras encontradas em todas as bases analisadas

Classes	**Regras**	$Aptidao_{Trein}$	$Aptidao_{Teste}$

1	if(289<0,5) and (531≥0,2) and (721≥0,1) and (870≥-0,2)
	if(289<0,5) and (839<1,9) and (531≥0,2) and (721≥0,2)
	if(289<0,5) and (863<1) and (531≥0,2) and (870≥0)
		1	0,533

2	if(11≥0,4) and (289<-0,5)
	if(11≥0,4) and (637<0,4)
	if(141<1,3) and (229≥1,1)
	if(229≥1,1) and (177<0,9)
	if(229≥1,1) and (289<-0,5)
	if(229≥1,1) and (456≥-0,9)
		1	1

	if(235<1) and (229≥1,1)
	if(289<-0,5) and (229≥1,1)
	if(637<0,4) and (11≥0,4)
	if(839≥0,5) and (637<0,4)		

3	if(50<-2,3) and (194<-1,1) and (289≥-0,3)
	if(2≥-0,4) and (46<-0,7) and (289≥-0,3)
	if(50<-2,3) and (194<-1,1) and (839≥-0,8)
	if(50<-2,3) and (242<0,6) and (289≥-0,7)
	if(50<-2,3) and (289≥-0,4) and (306≥-0,9)
		1	1

4	if(2<-0,2) and (485≥0,7)
	if(11<-2,8) and (485≥0,7)
	if(19<-0,4) and (485≥0,7)
	if(19<-0,4) and (526<-0,9)
	if(19<-0,4) and (843<-1)
	if(50≥-2) and (280<-0,7)
	if(50≥-2) and (485≥0,7)
	if(50≥-2) and (526<-0,8)
	if(50≥-2) and (843<-1)
	if(63<-0,3) and (485≥0,7)
	if(194<-0,8) and (485≥0,7)
	if(224<-2,2) and (380≥-0,2)
	if(224<-2,2) and (485≥0,7)
	if(224<-2,2) and (843<-1)
	if(224<-2,2) and (865≥0,1)
	if(224<-2,1) and (950≥0)
	if(229≥-1,6) and (485≥0,7)
	if(235≥-2,9) and (485≥0,7)
	if(235≥-3,1) and (843<-1)
	if(366≥-0,9) and (485≥0,7)
	if(366≥-0,9) and (526<-0,9)
	if(366≥-0,9) and (843<-1)
	if(409≥-1,7) and (485≥0,7)
	if(409≥-1,8) and (843<-1)
	if(456<1,2) and (485≥0,7)
	if(475≥-2,5) and (485≥0,5)
	if(485≥0,6) and (11<-2,4)
	if(485≥0,7) and (525≥-1,1)
	if(485≥0,7) and (661<-0,3)
	if(485≥0,7) and (783<0,4)
	if(485≥0,7) and (865<1,5)
	if(485≥0,7) and (929≥-0,3)
	if(526<-0,7) and (63<-0,3)
	if(526<-0,9) and (929≥-0,3)
	if(839<-0,4) and (224<-2,2)
	if(839<-0,5) and (485≥0,7)
	if(843<-1) and (63<-0,3)
		1	1

	if(843<-1) and (525≥-1,1)
	if(843<-1) and (783<0,4)
	if(843<-0,9) and (929≥-0,3)
	if(881<-0,2) and (485≥0,6)		

5	if(2<-1,9) and (289≥-1,3) and (380<-0,7)
	if(2<-1,8) and (17≥-0,8) and (229<-0,8)
	if(11≥-1,5) and (97<0,2) and (18<0,1)
	if(11≥-1,5) and (97<0,1) and (46<-0,4)
	if(11≥-1,5) and (97<0,1) and (229<-0,7)
	if(11≥-1,5) and (97<0,1) and (292<0,6)
	if(11≥-1,5) and (97<0,1) and (302<0,1)
	if(11≥-1,5) and (97<0,1) and (348<-1,5)
	if(11≥-1,6) and (97<0,1) and (379<0,1)
	if(11≥-1,7) and (97<0,1) and (380<-0,6)
	if(11≥-1,5) and (289≥-1,3) and (380<-0,7)
	if(17≥-0,8) and (19<-2,6) and (229<-0,8)
	if(17≥-0,9) and (46<-0,3) and (306<-1)
	if(17≥-0,9) and (46<-0,2) and (661<0,1)
	if(17≥-0,8) and (229<-0,8) and (306<-1)
	if(17≥-0,9) and (229<-0,8) and (380<-0,6)
	if(17≥-0,9) and (289≥-1,3) and (380<-0,7)
	if(18<0,1) and (41≥-2,2) and (721≥0,5)
	if(18<0,2) and (97<0,1) and (11≥-1,5)
	if(19<-2,9) and (41≥-2,2) and (721≥0,8)
	if(19<-2,6) and (229<-0,7) and (890≥-0,1)
	if(19<-0,9) and (289≥-1,3) and (380<-0,7)
	if(28≥-0,7) and (97<0,1) and (380<-0,7)
	if(28≥-0,8) and (97<0,2) and (721≥1)
	if(28≥-0,5) and (289≥-1,3) and (380<-0,7)
	if(41≥-2) and (46<-0,4) and (380<-0,7)
	if(41≥-2,1) and (97<0,1) and (721≥1)
	if(41≥-2) and (229<-0,8) and (380<-0,7)
	if(41≥-2,3) and (289≥-1,4) and (380<-0,7)
	if(41≥-2,3) and (456<-0,5) and (380<-0,5)
	if(41≥-2) and (721≥0,9) and (46<-0,4)
	if(41≥-2) and (721≥0,9) and (380<-0,7)
	if(41≥-2,3) and (721≥0,9) and (783<0,1)
	if(97<0,4) and (11≥-1,6) and (46<-0,2)
	if(97<0,1) and (11≥-1,7) and (380<-0,7)
	if(97<0,5) and (28≥-0,6) and (721≥1)
	if(97<0,1) and (41≥-2) and (380<-0,7)
	if(97<0,1) and (41≥-2,3) and (721≥1)
	if(97<0,1) and (721≥1) and (306<-1)
	if(97<0,1) and (194≥-1,4) and (380<-0,6)
	if(97<0,1) and (194≥-1,3) and (721≥1)
	if(97<0,1) and (224≥-0,6) and (380<-0,6)
		1	1

	if(97<0,1) and (231≥-0,3) and (306<-0,8)			
	if(97<0,1) and (242≥0,3) and (380<-0,7)			
	if(97<0,1) and (246≥-0,4) and (380<-0,7)			
	if(97<0,1) and (246≥-0,4) and (721≥1)			
	if(97≥-0,7) and (289≥-1,3) and (380<-0,7)			
	if(97<0,1) and (292<0,9) and (11≥-1,5)			
	if(97<0,1) and (302<0,1) and (11≥-1,5)			
	if(97<0,1) and (306<-1) and (380<-0,6)			
	if(97<0,1) and (379<0,1) and (380<-0,7)			
	if(97<0,1) and (379<0,1) and (721≥1)			
	if(97<0,1) and (380<-0,6) and (950≥0)			
	if(97<0,1) and (475≥-0,5) and (721≥1)			
	if(97<0,1) and (637≥0,1) and (306<-1)			
	if(97<0,2) and (721≥1) and (11≥-1,9)			
	if(97<0,1) and (721≥1) and (870≥-0,5)			
	if(97<0,1) and (870≥-0,9) and (380<-0,6)			
	if(97<0,4) and (881≥-0,9) and (721≥1)			
	if(97<0,1) and (890≥-0,8) and (380<-0,6)			
	if(141≥-1,4) and (289≥-1,6) and (380<-0,7)			
	if(177≥-1,4) and (289≥-1,3) and (380<-0,7)			
	if(194≥-1,3) and (229<-0,8) and (380<-0,5)			
	if(224≥-0,6) and (289≥-1,3) and (380<-0,7)			
	if(229<-0,8) and (890≥-0,1) and (306<-1)			
	if(229<-0,8) and (890≥-0,1) and (380<-0,6)			
	if(242≥0,3) and (41≥-2) and (46<-0,3)			
	if(242≥0,3) and (289≥-1,3) and (380<-0,7)			
	if(242≥0,1) and (302<0,1) and (41≥-2)			
	if(289≥-1,3) and (2<-1,1) and (380<-0,7)			
	if(289≥-1,3) and (17≥-1,6) and (380<-0,7)			
	if(289≥-1,3) and (19<-1) and (380<-0,7)			
	if(289≥-1,5) and (380<-0,7) and (661≥-1,2)			
	if(289≥-1,4) and (28≥-0,6) and (380<-0,6)			
	if(289≥-1,3) and (177≥-1,4) and (380<-0,7)			
	if(289≥-1,3) and (224≥-0,6) and (380<-0,7)			
	if(289≥-1,3) and (246≥-0,4) and (380<-0,7)			
	if(289≥-1,3) and (306<-1) and (380<-0,7)			
	if(289≥-1,3) and (336≥-0,5) and (380<-0,7)			
	if(289≥-1,3) and (366≥-0,6) and (380<-0,7)			
	if(289≥-1,3) and (380<-0,7) and (865≥-0,9)			
	if(289≥-1,3) and (380<-0,7) and (950≥0)			
	if(289≥-1,4) and (637≥0,2) and (380<-0,6)			
	if(289≥-1,4) and (721≥0,9) and (380<-0,6)			
	if(289≥-1,4) and (828≥-0,8) and (380<-0,6)			
	if(289≥-1,4) and (881≥-0,9) and (380<-0,6)			
	if(289≥-1,5) and (890≥-0,2) and (380<-0,4)			
	if(292<0,5) and (41≥-2) and (721≥1)			

	if(302<0,1) and (41≥-2) and (11≥-1,8)
	if(302<0,1) and (41≥-2) and (380<-0,4)
	if(302<0,1) and (41≥-2) and (721≥1)
	if(348<-1,2) and (17≥-0,8) and (229<-0,7)
	if(348<-1,5) and (41≥-2,1) and (229<0,2)
	if(348<-1,5) and (41≥-2) and (721≥0,8)
	if(475≥-0,6) and (289≥-1,3) and (380<-0,5)
	if(485≥-1,9) and (289≥-1,4) and (380<-0,7)
	if(525≥-0,9) and (289≥-1,3) and (380<-0,7)
	if(526≥-0,4) and (289≥-1,3) and (380<-0,7)
	if(531≥-0,4) and (289≥-1,4) and (380<-0,7)
	if(637≥0,2) and (289≥-1,3) and (380<-0,7)
	if(721≥1) and (289≥-1,3) and (380<-0,7)
	if(870≥-0,6) and (289≥-1,4) and (380<-0,7)
	if(881<0,9) and (41≥-2) and (721≥1)
	if(890≥-0,1) and (46<-0,3) and (306<-1)
	if(890≥-0,1) and (289≥-1,3) and (380<-0,7)		

6	if(2<1) and (17<0,4) and (637≥-0,1) and (379≥-0,1) and (456≥-1,2)	0,973	0,933
7	if (2≥-2,1) and (97<1,4) and (224≥-0,2)	0,971	1

8	if(97≥0,7) and (348<-0,8) and (863<0,7)
	if(97≥0,7) and (127≥0,3) and (863<0,7)
	if(97≥0,7) and (863<0,8) and (63<-0,4)
	if(97≥0,7) and (863<0,8) and (881≥0,1)
	if(127≥0,3) and (348<-0,7) and (863<0,7)
		1	1

9	if(18<-3,2) and (292≥-1,4)
	if(18<-3,1) and (19≥-0,2)
	if(18<-3) and (46<-0,5)
	if(18<-3,2) and (637≥-1,5)
		1	1

Todas as regras apresentadas anteriormente foram avaliadas segundo o valor de aptidão que elas retornaram no experimento em que foram evoluídas, tanto em treinamento quanto em teste. Entretanto, essa avaliação é melhor estimada pelo procedimento de validação cruzada 2:1, que é realizado através da média das melhores regras obtidas nos experimentos com as três partições de teste distintas.

A Tabela 5.12 apresenta os melhores resultados encontrados em cada uma das três partições de teste analisadas. Para cada classe, apresentamos o valor da melhor aptidão, que reproduz os valores fornecidos anteriormente para as melhores regras e a aptidão média nas três partições. Assim, embora todas as regras da Tabela 5.11 tenham sido apresentadas com suas aptidões reais calculadas nos experimentos em que as mesmas

foram evoluídas, os valores apresentados na Tabela 5.12, como aptidão média, fazem uma melhor estimativa do desempenho das mesmas. A tabela também fornece a média nas nove classes para a melhor aptidão e para a aptidão média. É importante ressaltar que, em treinamento, os valores médios encontrados são os mesmos, tanto para a aptidão média, quanto para a melhor aptidão, mostrando que em qualquer uma das três partições os valores de treinamento obtidos foram bem próximos. Quando avaliamos os resultados médios nas nove classes em teste, há uma diferença de aproximadamente de 5% entre a média entre os melhores resultados e a aptidão média (0,9406 e 0,889, respectivamente). Isso demostra que existe uma queda na eficácia das regras obtidas, em função da partição escolhida, para algumas classes. Essa queda pode ser percebida principalmente nas classes 6 e 8. Entretanto, de uma forma geral, podemos dizer que independentemente da partição escolhida, o resultado médio está muito próximo ao resultado obtido na melhor partição.

Tabela 5.12: Resultado do *cross validation*

	Exp. 12->3	**Exp. 13->2**	**Exp. 23->1**		
Classes	$Apt_{Trein/Teste}$	$Apt_{Trein/Teste}$	$Apt_{Trein/Teste}$	Melhor Aptidao	Aptidao Media
1	0,972/0,444	0,921/0,406	1/0,533	1/0,533	0,964/0,461
2	1/1	1/1	1/1	1/1	1/1
3	1/0,889	1/1	1/1	1/1	1/0,963
4	1/1	1/1	1/1	1/1	1/1
5	1/1	1/1	1/1	1/1	1/1
6	1/0,667	0,973/0,935	1/0,633	0,973/0,935	0,991/0,745
7	1/0,944	1/0,938	0,971/1	0,971/1	0,99/0,96
8	1/0,667	1/1	1/0,95	1/1	1/0,872
9	1/1	1/1	1/1	1/1	1/1
			Médias	0,9938/0,9406	0,9938/0,8990

As aptidões obtidas nas melhores regras apresentadas na Tabela 5.11, seja pela aptidão absoluta obtida no experimento em que as mesmas foram evoluídas, seja pela aptidão média nas três partições, nos fornecem avaliações da eficácia dessas regras em relação a cada classe analisada.

Entretanto, para que pudéssemos ter uma avaliação geral do conjunto de regras como um todo, na classificação de todas as amostras da base NCI60, realizamos novamente a análise AECD, onde um conjunto de 9 regras (uma para cada classe) é empregado como

um classificador caixa-preta na avaliação das 61 amostras. Essa avaliação é importante, sobretudo, para compararmos os resultados de classificação das regras com outros classificadores disponíveis na literatura, que não realizam uma avaliação por classe. Esse conjunto foi obtido selecionando a primeira regra de cada classe da Tabela 5.11, mas outros conjuntos/classificadores poderiam ser elaborados com as demais regras. Aplicando a análise AECD neste conjunto, foi obtido 86,88% de classificações corretas, sendo 53 acertos, nenhum erro grave, 7 confusões e um desconhecimento.

Resultados melhores que os 86,88%, citados anteriormente, já haviam sido obtidos utilizando-se o conjunto classificador mostrado na Tabela 5.7, elaborado a partir das regras obtidas nos experimentos com bases individuais e também pelo classificador construído a partir das regras mineradas da base B_1B_2, cujo resultado foi apresentado na Tabela 5.10. Nos dois casos, a análise AECD retornou uma taxa de 90,16% de classificações corretas.

Esse resultado, a princípio, nos pareceu inconsistente. Como seria possível obter um valor mais baixo na análise AECD com o conjunto das melhores regras, se na seleção dessas regras, todas as outras são consideradas? Após uma análise, registro a registro, dos erros de classificação, foi possível esclarecer a situação, conforme a explicação a seguir.

A métrica que utilizamos na avaliação das regras por classe, relaciona-se à sensibilidade e especificidade das regras evoluídas, e não simplesmente ao número de acertos da regra, que é a medida efetivamente utilizada na avaliação AECD. Assim, na análise simples de acertos, um erro de classificação por falso positivo ou por falso negativo não faz diferença. Por outro lado, na avaliação efetuada pelas equações 4.1 e 4.2, um falso negativo tem um peso muito maior no valor de aptidão do que um falso positivo, pois o denominador da sensibilidade, que contém o número de amostras da classe em questão, é tipicamente menor que o denominador da especificidade, que contém o número de amostras de todas as outras classes.

Dessa forma, não necessariamente o mesmo conjunto que retorna os maiores valores de aptidão, segundo a equação 4.3, retornarão o maior valor na análise de AECD. Por outro lado, nos outros trabalhos que fizeram a mineração da base NCI60, a análise é feita puramente em cima do número de acertos. Assim, realizamos novamente uma busca, considerando-se todas as melhores regras evoluídas em cada execução do AG (para todas

as bases e todas as partições de teste) e selecionamos um segundo conjunto de regras, que retornou a melhor análise AECD. Chamamos esse conjunto de K_2 e o conjunto anterior, formado pelas melhores regras segundo a aptidão, de K_1. As Tabelas 5.13 e 5.14 apresentam os dois conjuntos, com suas respectivas avaliações de aptidão, além dos erros de classificação por classe. As tabelas também apresentam os valores totais de erros para treinamento e teste.

Tabela 5.13: Conjunto K_1: regras com os maiores valores de aptidão segundo a equação 4.3

Classes	**Regras**	Apt_{Trein}	Apt_{Teste}	$Erros_{Trein}$	$Erros_{Teste}$
1	if(289< 0,5) and (531≥0,2) and (721≥0,1) and (870≥-0,2)	1	0,533	0	5
2	if(11≥0,4) and (289<-0,5)	1	1	0	0
3	if (49<-2,3) and (193<-1,1) and (289≥-0,3)	1	1	0	0
4	if (2<-0,2) and (485≥0,7)	1	1	0	0
5	if (2<-1,8) and (289≥-1,3) and (380<-0,7)	1	1	0	0
6	if (2<1) and (17<0,4) and (637≥-0,1) and (379≥-0,1) and (456≥-1,2)	0,973	0,933	1	1
7	if (97≥0,7) and (348<1,4) and (224≥-0,2)	0,971	1	1	0
8	if (97≥0,7) and (348<-0,8) and (863<0,7)	1	1	0	0
9	if (18<-3,2) and (291≥-1,4)	1	1	0	0
			Total	2	6

Tabela 5.14: Conjunto K_2: regras com o maior número de acertos na análise AECD

Classes	**Regras**	Apt_{Trein}	Apt_{Teste}	$Erros_{Trein}$	$Erros_{Teste}$
1	if(28<0,8) and (75≥0,2) and (280≥-0,3) and (498<0,1) and (843≥0)	1	0,333	0	2
2	if(11≥0,4) and (289<-0,5)	1	1	0	0
3	if (49<-2,3) and (193<-1,1) and (289≥-0,3)	1	1	0	0
4	if (2<-0,2) and (485≥0,7)	1	1	0	0
5	if (2<-1,8) and (289≥-1,3) and (380<-0,7)	1	1	0	0
6	if(17≥-1,6) and (242<0,3) and (637≥0,4) and (881<1)	1	0,667	0	1
7	if (97≥0,7) and (348<1,4) and (224≥-0,2)	0,971	1	1	0
8	if (97≥0,7) and (348<-0,8) and (863<0,7)	1	1	0	0
9	if (18<-3,2) and (291≥-1,4)	1	1	0	0
			Total	1	3

A Tabela 5.15 apresenta os resultados de sensibilidade (Se, calculado pela equação 4.1) e especificidade (Sp calculado pela equação 4.2) para os conjuntos K_1 e K_2. O que difere o

conjunto K_1 e K_2 são as regras utilizadas como classificadores para as classes 1 e 6. Para a classe 1, apesar do valor de aptidão encontrado para o conjunto K_1 (0,533) ser maior do que o encontrado para o conjunto K_2 (0,333), o número de erros encontrados para o primeiro conjunto é maior do que o encontrado para o segundo (5 e 2, respectivamente). Estes erros são apresentados na Tabela 5.13 para o conjunto K_1 e na Tabela 5.14 para o conjunto K_2. Na composição do valor de aptidão, uma classificação errada encontrada no cálculo da sensiblidade (*Se*) é mais severa do que uma classificação errada encontrada no cálculo da especificidade (*Sp*). Isso acontece devido ao tamanho do conjunto de amostras utilizado no cálculo de *Se* e *Sp*. Para o cálculo de *Se*, utilizam-se apenas as amostras de uma determinada classe. Por outro lado, no cálculo de *Sp*, utilizam-se as demais amostras da base. Consideremos, por exemplo, o cálculo de *Se* e *Sp* para a classe 1 da base NCI60. Para o cálculo do *Se*, serão avaliados apenas 7 amostras, ao passo que, ao calcular *Sp*, serão utilizadas as 54 amostras restantes da base. Um erro encontrado no cálculo de *Se*, diminuirá de 1/7 o valor de aptidão, enquanto que um erro encontrado no cálculo de *Sp*, diminuirá de 1/54 essa aptidão. Para a análise AECD, diferentemente do que acontece para o cálculo da aptidão, erros encontrados em *Se* ou *Sp* (falso negativo ou falso positivo) possuem o mesmo peso. Assim, a regra da classe 1 encontrada no conjunto K_1 possui aptidão maior do que a regra encontrada no conjunto K_2, mas possui uma quantidade de erros maior (5 ao invés de 2). Para a classe 6, o número de erros encontrados do conjunto K_1 para o conjunto K_2 decaiu de uma unidade. No conjunto K_1, foram encontrados dois erros, um na base de treinamento e outro na base de teste, ambos no cálculo da *Sp*, causando um pequeno decréscimo ao valor da aptidão. Para o conjunto K_2, foi encontrado apenas um erro na base de teste, mas este erro aconteceu no cálculo da *Se*, causando um grande decréscimo no valor da aptidão.

A partir dos resultados dos erros absolutos obtidos pelos conjuntos de regras K_1 e K_2, é possível comparar o desempenho desses classificadores com outros da literatura, que foram elaborados para a base NCI60 [21, 12, 3, 13, 22, 23, 24] e que tiveram sua taxa de acertos divulgada. Alguns desses trabalhos também fizeram duas partições, uma contendo 2/3 das amostras utilizadas no treinamento, e outra, contendo 1/3 das amostras utilizadas no teste. Estas partições são apresentadas na Tabela 5.16. Outros trabalhos divulgaram

Tabela 5.15: Sensibilidade e Especificidade das regras dos conjuntos K_1 e K_2

K_1								
	Treinamento			**Teste**			**Base Completa**	
Classes	**Se**	**Sp**		**Se**	**Sp**		**Se**	**Sp**
1	1	1		0,6666	0,8		0,8571	0,9259
2	1	1		1	1		1	1
3	1	1		1	1		1	1
4	1	1		1	1		1	1
5	1	1		1	1		1	1
6	1	0,9729		1	0,9333		1	0,9615
7	1	0,9705		1	1		1	0,9818
8	1	1		1	1		1	1
9	1	1		1	1		1	1

K_2								
	Treinamento			**Teste**			**Base Completa**	
Classes	**Se**	**Sp**		**Se**	**Sp**		**Se**	**Sp**
1	1	1		0,3333	1		0,7142	1
2	1	1		1	1		1	1
3	1	1		1	1		1	1
4	1	1		1	1		1	1
5	1	1		1	1		1	1
6	1	1		0,6666	1		0,8888	1
7	1	1		1	0,9705		1	0,9818
8	1	1		1	1		1	1
9	1	1		1	1		1	1

apenas a taxa de acertos em relação à base total e são apresentados na Tabela 5.17. O resultado obtido por Umpai [22] é uma média encontrada em 5 experimentos. O trabalho de Ooi e colaboradores [3] e o de Lin e colaboradores [24] possuem duas ocorrências, uma em cada tabela, devido ao uso das duas abordagens nestes trabalhos. É importante dizer que os resultados obtidos em Ooi_1 [3] não utilizam métodos tradicionais de teste, conforme discutido na seção 3.4. Assim, estes resultados não foram utilizados na análise de treinamento e teste, apenas na análise com base completa. O símbolo (*) encontrado na Tabela 5.17 refere-se à média do número de erros encontrados em cinco execuções do ambiente proposto por Umpai et al. [22].

Na análise comparativa considerando-se o número total de erros do conjunto de regras K_1, podemos observar que esse conjunto obteve resultados comparáveis com diversos

Tabela 5.16: Comparativo dos erros encontrados em K_1 e K_2 e de outros trabalhos, utilizando 2/3 da base em treinamento e 1/3 em teste

Base Particionada: 2/3 Treinamento e 1/3 Teste				
Referencia	**Nro de Genes**	$Erros_{Trein}$	$Erros_{Teste}$	$Erros_{Total}$
Dudoit [21]	30	-	8	$\geq$8
Deb [12]	12	3	2	5
Ooi_2 [3]	12	4	4	8
Lin_1 [24]	15	5	4	9
K_1	20	2	6	8
K_2	22	1	3	4

Tabela 5.17: Comparativo dos erros encontrados em K_1 e K_2 e de outros trabalhos, utilizando todas as amostras da base NCI60

Base Total		
Referencia	**Nro de Genes**	**Nro de Erros**
Liu [13]	40	7
Umpai [22]	30	14,5 (*)
Lin_2 [24]	15	3
Ooi_1 [3]	13	7
K_1	20	8
K_2	22	4

outros classificadores (Dudoit, Ooi_2 e Lin_1) que também fizeram a partição 2/3 de treinamento e 1/3 de teste, sendo superado significativamente apenas pelo classificador de Deb. Com relação aos classificadores que foram ajustados utilizando-se a base completa, portanto com maiores chances de encontrar um baixo número de erros, mas com um resultado de generalização questionável, também é possível dizer que o conjunto K_1 obteve resultados comparáveis aos classificadores de Liu e Ooi_1, superou o classificador de Umpai e foi superado apenas por Lin_2. Com relação ao desempenho de erros na base de teste, o conjunto K_1 superou apenas o classificador de Dudoit, sendo superado pelos demais classificadores (Deb, Ooi_2 e Lin_1). Entretanto, devemos salientar que o conjunto K_1 foi obtido considerando-se os valores de sensibilidade e especificidade das regras em suas respectivas classes, tanto na evolução do AG quanto na seleção das melhores regras, sem ser direcionado diretamente à taxa de acertos. Dessa forma, consideramos bom o desempenho do conjunto K_1 uma vez que esse classificador elaborado para a base NCI60, diferente-

mente dos demais, possui um conhecimento de alto nível e detalhado por classe, sem apresentar um decaimento significativo de taxa de acerto, em relação aos classificadores tipo caixa-preta, publicados na literatura.

Na análise comparativa considerando-se o conjunto K_2, em relação aos classificadores obtidos por meio de particionamento treinamento/teste, é possível observar que ele só é superado pelo classificador de Deb, na taxa de acertos na base de teste, por um lado, mas K_2 supera esse mesmo classificador na taxa de acertos total, assim como os classificadores de Dudoit, Ooi_2 e Lin_1. Com relação aos classificadores que usaram a base completa, o conjunto K_2 só é superado por Lin_2, lembrando que o classificador Lin_2 foi obtido usando a base completa, enquanto que cada regra de K_2 foi evoluída utilizando-se apenas 2/3 da base. Assim, embora na evolução do AG a taxa de acertos não seja utilizada diretamente, a seleção posterior das melhores regras utilizando-se a análise AECD, resultou em um conjunto/classificador competitivo com os demais do tipo caixa-preta, superando a maioria dos resultados publicados.

Concluindo, o conjunto K_1 é o que apresenta os resultados individuais por classe mais expressivos considerando-se a sensibilidade e especificidade, com uma razoável taxa de acertos em relação aos classificadores já publicados. Entretanto, o AG também foi capaz de evoluir regras eficazes em relação à taxa de acertos, sendo possível construir o conjunto K_2, que supera a maioria dos classificadores já publicados, em relação à taxa de acertos. Uma informação importante, é que na constituição dos conjuntos K_1 e K_2 nenhum esforço foi gasto na busca de conjuntos que tivessem a mesma *perfomance* e um número menor de genes, já que o método de seleção adotado foi a de buscar a primeira regra de cada classe. Assim, pode-se buscar conjuntos que utilizam um menor número de genes e que tenham o mesmo desempenho dos conjuntos K_1 e K_2 apresentados.

Capítulo 6

Conclusões e trabalhos futuros

Em nossos experimentos, foi possível observar que embora a obtenção de regras com alto índice de treinamento seja relativamente fácil de se conseguir, a qualidade dessas regras é logo diminuída em algumas classes pelo desempenho das mesmas na base de testes. Acreditamos que tal comportamento possa ser justificado pelo baixo número de amostras por classe, inerente ao problema. Para compensar essa dificuldade, procuramos efetuar um grande número de execuções do AG, para obtenção de um maior número de regras por classe, com alta taxa de desempenho na base de treinamento. Dessa forma, conseguimos obter regras eficazes em oito das nove classes.

Embora a base NCI60 tenha sido extensivamente investigada, em nenhum dos trabalhos analisados foram encontrados conjuntos de genes preditivos para cada classe, e sim, um único conjunto preditivo para todas as classes. Selecionar um conjunto de genes relacionados a uma determinada classe de câncer é relevante para o entendimento das interações moleculares (*molecular pathways*) e também para encontrar novos alvos que sejam úteis no desenvolvimento de novas drogas [23]. Outro ponto importante refere-se ao fato de nenhum dos trabalhos analisados apresentarem conhecimento de alto nível para a base NCI60. O nosso trabalho apresenta um conjunto reduzido de genes por classe (variando de 2 a 5 genes), conjunto este, apresentado na forma de regras do tipo IF-THEN, relacionando genes, intervalos de níveis de expressão e sua classe. Um outro ponto forte do projeto refere-se à avaliação de sensibilidade e especificidade de cada classe, não realizado em nenhum outro trabalho (de que tenhamos notícia) na base NCI60.

Para a validação final do ambiente foi utilizado o método de *cross validation* 2:1, que obteve em média para as 9 classes avaliadas, 99,38% de acertos em treinamento e 88,9% em teste.

Um conjunto formado por representantes das regras que apresentaram o melhor desempenho treinamento/teste, chamado K_1, além de apresentar um conhecimento de alto nível e valores aceitáveis de sensibilidade e especificidade, também apresenta um número de acertos total. Esse conjunto retornou as aptidões médias de 99,38% em treinamento e 94,06% em teste, medidos pela equação 4.3, que combina a sensibilidade e a especificidade.

Um segundo conjunto, chamado K_2, também foi elaborado a partir das melhores regras evoluídas. Embora o resultado de sensibilidade e especificidade seja inferior ao K_1, o conjunto K_2 possui uma taxa de acertos total igual a 93,44%, superando diversos métodos publicados e sendo inferior apenas ao resultado obtido por Lin e colaboradores [24] (95,08%). Entretanto, os autores usaram a base completa na evolução do AG, enquanto as regras do conjunto K_2 foram evoluídas usando 2/3 da base (para cada classe) para encontrar esse valor. Em termos do número de erros na base de teste obtido pelo K_2 (3), esse valor só é superado pelo trabalho de Deb e Reddy (2) [12].

Além dos dois conjuntos citados anteriormente, nos quais realizamos uma análise comparativa com os principais classificadores publicados na base NCI60, a Tabela 5.11 apresenta um número maior de regras por classe. Todas as regras da tabela representam o melhor desempenho obtido, com o menor número de genes possível, para cada classe correspondente. De posse dessas regras, diversos outros conjuntos/classificadores podem ser elaborados e avaliados. Além disso, essa pluralidade de regras pode fornecer mais informações aos biólogos sobre as relações entre os genes e a existência de genes homólogos (genes distintos que possuem a mesma função). Por exemplo, poderíamos construir uma regra mais complexa para a classe 4, da seguinte forma:

SE (Gene_50 $\geq$ -2 OU Gene_224 $<$ -2,2 OU Gene_235 $\geq$ -2,9)
E (Gene_485 $\geq$ 0,7 OU Gene_843 $<$ -1)
ENTÃO Classe = leucemia

Esste tipo de conhecimento pode ser utilizado pelos biólogos para investigar as relações entre os conjuntos de genes {50, 224, 235} e {485, 843} (homólogos?) e a leucemia.

Conseguimos delimitar genes relacionados a cada classe de câncer e seus respectivos níveis de expressão. Desta forma, obtemos uma associação gene/câncer e gene/gene que esperamos que possa contribuir para o diagnóstico deste tipo de câncer limitando assim o número de genes a serem analisados na busca de novos tratamentos.

Como trabalho futuro, sugerimos a construção de um AG multi-objetivo, que trabalhe com várias métricas de forma simultânea, porém isoladas. Um resumo dos AGs multi-objetivos é apresentado no apêndice C. Diferentes métricas podem ser aplicadas como objetivos, tais como: sensibilidade, especificidade, precisão, cobertura, dentre outros.

Uma outra extensão para este trabalho seria a utilização de bases com um número maior de genes. Por exemplo, na base NCI60, diversos trabalhos divulgaram conjuntos reduzidos de genes [21, 12, 3, 13, 22, 23, 24], que aplicados a algum modelo de classificador (RNA, SVM, MLHD, dentre outros), retornaram uma taxa de acertos razoável. Em nosso trabalho, partimos apenas dos genes extraídos no trabalho de Ooi e Tan [3], chegando a 55 genes na base completa ($B_1B_2B_3B_4$). Os genes extraídos em outros trabalhos poderiam ser incorporados a essa base, aumentando a disponibilidade de informações para o AG evoluir regras eficazes.

Entretanto, antes de mais nada, será necessário realizar experimentos com o objetivo de ajustar o ambiente evolutivo na manipulação de bases com um número maior de genes. Conforme ressaltamos no capítulo anterior, um resultado que nos chamou a atenção foi obtido na base completa $B_1B_2B_3B_4$ que, embora use todo o potencial de informação das expressões gênicas, retornou resultados inferiores se comparado às evoluções das bases individuais. Esse resultado mostra que o AG teve dificuldades de convergência para regras eficazes, com o aumento do número de genes. Acreditamos que tal ajuste tem forte relação com o valor limite do parâmetro peso (o valor que decide se uma condição estará presente ou não em uma regra) e o tamanho da população (quanto maior o tamanho do cromossomo, maior a necessidade de amostragem do espaço de busca). Experimentos incluindo ruído na base $B_1B_2B_3B_4$ (genes extraídos aleatoriamente dos 1000 genes da base NCI60) podem auxiliar neste ajuste.

Aplicar o ambiente em outras bases de dados públicas de expressão gênica. Estas bases de dados podem ser binárias ou multiclasse. As binárias podem ser encontradas em

[79, 95] (leucemia), [80, 94] (*diffuse large B-cell lymphoma*), [81] (Cólon), [89] (Próstata), [84] (mama). As multiclasses pode ser encontradas em [82] (GCM), [83] (Brown) e [8] (*small, round blue cell tumors of childhood*).

A partir da leitura de trabalhos publicados, foi possível observar uma diversidade de métodos aplicados pelos pesquisadores para validação dos seus resultados. Diversidade essa que prejudica inclusive a comparação do desempenho entre os diversos classificdores. Propomos como continuidade a esse trabalho, a aplicação de outras estratégias de validação, tais como: o *leave-one-out cross validation*, a técnica mais empregada e o *bootstrap*, uma técnica que vem sendo aplicada nos trabalhos mais recentes e que nos parece contornar melhor os problemas inerentes aos experimentos de *microarrays* (baixo número de amostras com um elevado número de genes).

Referências Bibliográficas

[1] D. J. Duggan, M. Bittner, Y. Chen, P. Meltzer, and J. M. Trent. Expression profiling using cdna microarrays. *Nature Genetics*, 21, 1999.

[2] W. G. C. Ticona. *Aplicação de Algoritmos Genéticos Multi-Objetivo para Alinhamento de Sequências Biológicas*. PhD thesis, Universidade de São Paulo, 2003.

[3] C. H. Ooi and P. Tan. Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. *Bioinformatics*, 19(1):37–44, 2003.

[4] A. Borém, M. Giúcide, and T. Sedyiama. *Melhoramento Genômico*. Universidade Federal de Viçosa, 2003.

[5] J. C. Setúbal and J. Meidanis. *Introduction to Computacional Molecular Biology*. PWS Publishing Company, Boston, 1997.

[6] P. Baldi and S. Brunak. *Bioinformatics: the Machine Learning approach*. MIT Press, 2 edition, 2001.

[7] Y. Xu, F. M. Selaru, J. Yin, T. T. Zou, V. Shustova, Y. Mori, F. Sato, T. C. Liu, A. Olaru, S. Wang, M. C. Kimos, K. Perry, K. Desai, B. D. Greenwald, M. J. Krasna, D. Shibata, J. M. Abraham, and S. J. Meltzer. Artificial neural networks and gene filtering distinguish between global gene expression profiles of barret's esophagus and esophageal cancer. *Cancer Research*, 2002.

[8] J. Khan, J. S. Wei, M. Ringnér, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer. Classifiction and

diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 2001.

[9] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 2000.

[10] T. A. Brown. *Genética: Um enfoque molecular*. Guanabara Koogan, Rio de Janeiro, 3 edition, 1999.

[11] I. Zwir, R. R. Zaliz, and E. H. Ruspini. Automated biological sequence description by genetic multiobjective generalized clustering. *New York Academy of Sciences*, (980):65–82, 2002.

[12] K. Deb and A. R. Reddy. Classification of two and multi-class cancer data reliably using multi-objective evolutionary algorithms. *KanGAL Report*, 2003.

[13] J. J. Liu, G. Culter, W. Li, Z. Pan, S. Peng, T. Hoey, L. Chen, and X. Ling. Multiclass cancer classification and biomarker discovery using ga-based algorithms. *Bioinformatics*, 21(11):2691–2697, 2005.

[14] S. Mitra and H. Banka. Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognition*, 2006.

[15] M. Wahde and Z. Szallasi. Improving the prediction of the clinical outcome of breast cancer using evolutionary algorithms. *Soft Comput*, 2006.

[16] T. R. Hvidsten, A. Laegreid, and J. Komorowski. Learning rule-based models of biological process from gene expression time profiles using gene ontology. *Bioinformatics*, 19(9), 2003.

[17] S. A. Vinterbo, E. Kim, and L. Ohno-Machado. Small, fuzzy and interpretable gene expression based classifiers. *Bioinformatics*, 21(9), 2005.

[18] S. Ho, C. Hsieh, H. Chenc, and H. Huangd. Interpretable gene expression classifier with an accurate and compact fuzzy rule base for microarray data analysis. *BioSystems*, 85, 2006.

[19] M. V. Fidelis, H. S. Lopes, and A. A. Freitas. Discovery comprehensible classification rules with a genetic algorithm. In *Congress on Evolutionary Computation - (CEC-2000)*, pages 805–810. La Jolla, CA, USA, 2000.

[20] D. T. Ross, U. Scherf, M. B. Eisen, C. M. Perou, C. Rees, P. Spellman, V. Iyer, S. S. Jeffrey, M. Van de Rijn, M. Waltham, A. Pergamenschikov, J. C. F. Lee, D. Lashkari, D. Shalon, T. G. Myers, J. N. Weinstein, D. Botstein, and P. O. Brown. Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics*, 2000.

[21] S. Dudoit, J. Fridlyand, and T. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457), March 2002.

[22] T. J. Umpai and S. Aitken. Feature selection and classification microarray data analysis: Evolutionary methods for identifying predictive genes. *BMC Bioinformatics*, 6(148), 2005.

[23] R. D. Uriarte and S. A. Andrés. Gene selecion and classification of microarray data using random forest. *BMC Bioinformatics*, 7(3), 2006.

[24] T. C. Lin, R. S. Liu, C. Y. Chen, Y. T. Chao, and S. Y. Chen. Pattern classification in dna microarray data of multiple tumor types. *Pattern Recognition*, 39:2426–2438, 2006.

[25] R. Xu, G. C. Anagnostopoulos, and D. C. Wunsch II. Multiclass cancer classification using semisupervised ellipsoid artmap and particle swarm optimization with gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(1), 2007.

[26] Lodish. *Biologia Celular e Molecular*. Revinter, Rio de Janeiro, 4 edition, 2002.

[27] L. C. Junqueira and J. Carneiro. *Biologia Celular e Molecular*. Guanabara Koogan, Rio de Janeiro, 6 edition, 1997.

[28] M. C. P. de Souto, A. C. Lorena, A. C. B. Delbem, and A. C. P. L. F. de Carvalho. Técnicas de aprendizado de máquina para problemas de biologia molecular. Porto Alegre, 2003. Sociedade Brasileira de Computação, Sociedade Brasileira de Computação.

[29] B. Alberts, D. Bray, and J. Lewis. *Biolgia Molecular da Célula*. Artes Médicas, 3 edition, 1997.

[30] S. Brenner, M. Johnson, J. Bridgham, G. Golda, D. H. Lloyd, D. Johnson, S. McCurdy S. Luo, M. Foy, M. Ewan, R. Roth, D. George, S. Eletr, G. Albrecht, E. Vermaas, S. R. Williams, T. B. K. Moon, R. B. M. Pallas, J. Kirchner, K. Fearon, J. Mao, and K. Corcoran. Gene expression analysis by massive parallel signature sequencing (mpss) on microbead array. *Nature Biotechnology*, 18(10):630–640, 2000.

[31] V. E. Velculescu, L. Zhang, B. Vogelstein, and K. W. Kinzler. Serial analysis of gene expression. *Science*, 270:484–487, 1995.

[32] W. M. Freeman, S. J. Walker, and K. E. Vrana. Quantitative rt-pcr: pitfalls and potentials. *Biotechniques*, 26:112–122, 1999.

[33] C. A. Harrington, C. Rosenow, and J. Retief. Monitoring gene expression using dna microarrays. *Curr. Opin. Microbiol.*, 3:285–291, 2000.

[34] N. P. Carneiro and A. A. Carneiro. *A Era Genômica - Desvendando o Código Genético*. UFLA, 2002.

[35] L. R. Amaral. Bioinformática, surge uma nova ciência. Especialização, Universidade Federal de Lavras, Lavras, 2005.

[36] T. Mitchell. *Machine Learning*. McGraw Hill, New York, 1997.

[37] Y. Su, T. Murali, V. Pavlovic, M. Schaffer, and S. Kasif. Rankgene: identification of diagnostic genes bases on expression data. *Bioinformatics*, 19(12):1578–1579, 2003.

[38] E. Fix and J. Hodges. Discriminatory analysis, noparametric discrimination: Consistency properties. Technical report, Escola de AviaçãoForça Aérea Americana, 1951.

[39] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[40] L. Breiman, J. Friedman, and R. Olshen. *C: Classification and regression trees.* Chapman & Hall, New York, 1984.

[41] B. D. Ripley. *Pattern recognition and neural networks.* Cambrigde University Press, Cambridge, 1996.

[42] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning.* Springer, New Tork, 2001.

[43] B. Efron. Estimating the error rate of a prediction rule: improvement on cross-validation. *Jounal of the American Statistical Association*, 78:316–331, 1983.

[44] M. Barnard. The secular variations of skull characters in four series of egyptian skulls. *Annals of Eugenics*, 6:352–371, 1935.

[45] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc Natl Acad Sci USA*, 99(10):6567–6572, 2002.

[46] P. Roepman, L. F. Wessels, N. Kettelarij, P. Kemmeren, A. J. Miles, P. Lijnzaad, M. G. Tilanus, R. Koole, G. J. Hordijk, P. C. van der Vliet, M. J. Reinders, P. J. Slootweg, and F. C. Holstege. An expression profile for diagnosis of lymph node metastases from primary head and neck squamous cell carcinomas. *Nature Genetics*, 37:182–186, 2005.

[47] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley, New York.

[48] R. E. Castro. *Otimização de Estrutura com Multi-Objetivos via ALgoritmos Genéticos.* PhD thesis, Universidade Federal do Rio de Janeiro, AGOSTO 2001.

[49] L. Davis. *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York, 1991.

[50] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Adison-Wesley, USA, 1989.

[51] M. Mitchell. *An Introduction to Genetic Algorithms: Complex Adapative Systems.* MIT Press, MA, 1996.

[52] J. H. Holland. *Adaptation in Natural and Artificial Systems.* MIT Press, 1975.

[53] J. R. Koza. *Genetic Programming. On the Programming of Computers by Means of Natural Selection.* MIT Press, USA, 1992.

[54] J. Tanomaru. Motivação, fundamentos e aplicações de algoritmos genéticos. In *Congresso Brasileiro de Redes Neurais*, Curitiba, 1995. III Escola de Redes Neurais.

[55] S. Austin. An introduction to genetic algorithms. *AI Expert*, 3, 1990.

[56] S. A. Oliveira. *Metaheurísticas Aplicadas ao Planejamento da Expansão da Transmissão de Energia Elétrica em Ambiente de Processamento Distribuído.* PhD thesis, UNICAMP, outubro 2004.

[57] M. A. C. Pacheco. Algoritmos genéticos: Princípios e aplicações. In *INTERCON99: V Congreso Internacional de Ingeniería Electrónica, Elétrica Y Sistemas*, pages 11–16, Lima, 1999.

[58] M. A. Potter and K. A. Jong. Cooperative coevolution: An architeture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000.

[59] D. A. Van Veldhuizen C. A. C. Coello and G.B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Kluwer Academic, New York, March 2002.

[60] D. Hand. *Construction and Assessment If Classification Rules.* John Wiley and Sons, Chichester, 1997.

[61] A. A. Freitas and S. H. Lavington. *Mining Very Large Databases with Parallel Processing*. Kluwer Academic Publishers, London, 1998.

[62] E. D. Goodman. An introduction to gallops - the genetic algorithms optimized for portability and parallelism system. Technical report, Departament od Computer Science - Michigan State University, 1996.

[63] H. S. Lopes, M. S. Coutinho, and W. C. Lima. An evolutionary approach to simulate cognitive feedback learning in medical domain. In E. Sanchez, T. Shibata, and L. A. Zadeh, editors, *Genetic Algorithms and Fuzzy Logic Systems*, pages 193–207. World Scientific, 1997.

[64] D. L. A. Araujo, H. S. Lopes, and A. A. Freitas. A parallel genetic algorithm for rule discovery in large databases. In *Systems, Man and Cybernetics*, volume 3, pages 940 – 945, Tokyo, October 1999. IEEE.

[65] D. R. Carvalho and A. A. Freitas. A hybrid decision tree/genetic algorithm for coping with the problem of small disjuncts in data mining. In *Genetic and Evolutionary Computation (GECCO-2000)*, pages 1061–1068, Las Vegas, NV, USA, Jully 2000.

[66] D. R. Carvalho and A. A. Freitas. A genetic algorithm-based solution for the problem of small disjuncts. In Springer-Verlag, editor, *Principles of Data Mining and Knowledge Discovery*, volume 1910, pages 345–352, 2000.

[67] A. A. Freitas. *Advances in Evolutionary Computation*, chapter A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery. Springer-Verlag, 2002.

[68] W. Romao, A. A. Freitas, and R. C. S. Pacheco. A genetic algorithm for discovering interesting fuzzy prediction rules: applications to science and technology data. In *Genetic and Evolutionary Computation (GECCO-2002)*, New York, July 2002.

[69] K. C. Tan, Q. Yu, C. M. Heng, and T. H. Lee. Evolutionary computing for knowledge dicovery in medical diagnosis. *Artificial Intelligence in Medicine*, (27):129–154, 2003.

[70] C. R. S. Miranda, G. M. B. Oliveira, and J. B. Santos. Algoritmos genéticos aplicados em data mining para obtenção de regras simples e precisas. In *Anais do SBAI2003*, pages 638–643, 2003.

[71] H. Ishibuchi and T. Yamamoto. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets and Systems*, (141):59–88, 2004.

[72] Daniel C. Weaver. Applying data mining techniques to library design, lead generation and lead optimization. *Science Direct*, 2004.

[73] Y. Kim and W. N. Street. An intelligent system for customer targeting: a data mining approach. *Decision Support Systems*, (37):215–228, 2004.

[74] A. Ghosh and B. Nath. Multi-objective rule mining using genetic algorithms. *Information Sciences*, 163, 2004.

[75] M. A. C. Pacheco, M. M. R. Vellasco, C. H. P. Lopes, and E. P. L. Passos. Extração de regras de associação em bases de dados por algoritmos genéticos. In *Anais do XIII Congresso Brasileiro de Automática (CBA 2000)*, Floarianópolis, Setembro 2000.

[76] M. C. S. Takiguti. Utilização de algoritmos genéticos multi-objetivos na mineração de regras precisas e interessantes. Dissertação de mestrado em engenharia elétrica, Universidade Presbiteriana Mackenzie, 2003.

[77] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. IE-Springer-Verlag, 1997.

[78] A. A. Freitas. *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. 2002.

[79] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction. *Science*, 286, October 1999.

[80] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[81] U. Alon, N. Barkai, D. D. Notterman, K. Gish, S. Ibarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of National Academy of Science, Cell Biology*, 96:6745–6750, 1999.

[82] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T. R. Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Science*, 98(26):15149–15154, 2001.

[83] K. Munagala, R. Tibshirani, and P. O. Brown. Cancer characterization and feature set extraction by discriminative margin clustering. *BMC Bioinformatics*, 5(21), 2004.

[84] L. J. van 't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. M. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536, 2002.

[85] N. Friedman I. Nachman M. Schummer A. Ben-Dor, L. Bruhn and Z. Yakhini. Tissue classification with gene expression profiles. *J. Computational Biology*, 7:559–584, 2000.

[86] J. Komorowski, A. Øhrn, and A. Skowron. *Handbook of Data Mining and Knowledge Discovery*, chapter The ROSETTA rough set software system, pages 554–559. Oxford University Press, 2002.

[87] S. Vinterbo and A. Ohrn. Minimal approximate hitting sets and rule templates. *International Journal of Approximate Reasoning*, 25(2):123–143, Outubro 2000.

[88] A. Bhattacharjee, W. G. RichardsDagger, J. Stauntondagger, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. BuenoDagger, M. Gillette, M. Loda, G. Weber, E. J. Markdagger, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. J. Sugarbaker, and M. Meyerson. Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *Proc. Natl Acad. Sci*, 98:13790–13795, 2001.

[89] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda P. W. Kantoff, T. R. Golub, and W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1:203–209, 2002.

[90] E. Alpaydin. Combined 5×2 cv f test for comparing supervised classification learning algorithms. *Neural Computation*, 11:1885–1982, 1999.

[91] S. Ho, L. Shu, and J. Chen. Intelligent evolutionary algorithms for large parameter optimization problems. *IEEE Transactions on Evolutionary Computation*, 8(6):522–541, 2004.

[92] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. H. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, R. Rifkin S. Mukherjee and, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415(6870):436–442, 2002.

[93] C. L. Nutt, D. R. Mani, R. A. Betensky, P. Tamayo, J. G. Cairncross, C. Ladd, U. Pohl, C. Hartmann, M. E. McLaughlin, T. T. Batchelor, P. M. Black, A. von Deimling, S. L. Pomeroy, T. R. Golub, and D. N. Loui. Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. *Cancer Research*, 63(7):1602–1607, 2003.

[94] M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. T. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. S. Pinkus, T. S. Ray, M. A. Koval, K. W. Last, A. Norton, T. A. Lister, J. Mesirov, D. S. Neuberg, E. S. Lander, J. C. Aster, and T. R. Golub. Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 2002.

[95] S. A. Armstrong, J. E. Staunton, L. B. Silverman, R. Pieters, M. L. den Boer, M. D. Minden, S. E. Sallan, E. S. Lander, T. R. Golub, and S. J. Korsmeyer. Mll translocations specify a distinct gene expression profile, distinguishing a unique leukemia. *Nature Genetics*, 30(1):41–47, 2002.

[96] M. James. Classification algorithms. *Wiley-Interscience*, New York.

[97] L. R. Amaral, G. Sadoyama, F. S. Espindola, and G. M. B. Oliveira. Classificação de oncogenes medidos por *microarray* utilizando algoritmos genéticos. *Anais do Simpósio Brasileiro de Automação Inteligente*, 2007.

[98] G. W. Burns and P. J. Bottino. *Genética*. Guanabara Koogan, Rio de Janeiro, 6 edition, 1991.

[99] N. Srinivas and K. Deb. Multiobjective optimization using non dominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[100] J. W. Hartmann. *Low-thurst Trajectory Optimization Using Stochastic Optimization Methods*. PhD thesis, University of Illinois-Champaign, 1999.

[101] C. M. Fonseca and P. J. Fleming. Genetics algorithms for multi-objective optimization: Formulation, discussion and generalization. In *Stephanie Forrest editor*,

San Mateo California, 1993. Proceedings of the Fifth International Conference on Genetic Algorithms.

[102] J. C. Bortot. Otimização evolutiva multi-objetivos na busca parametrizada de autômatos celulares unidimensionais. Master's thesis, Universidade Presbiteriana Mackenzie, São Paulo, 2003.

[103] I. Anciutti, A. L. Gonçalves, F. A. Siqueira, and P. S. S. Borges. Uma aplicação de data mining sobre circuitos elétricos de baixa tensão utilizando algoritmos genéticos. *1º Workshop de Ciências da Computação e Sistemas da Informação da Região Sul (WorkComp Sul)*, Maio 2004.

[104] R. Lewis. *Human Genetics - Concepts and Applications.* McGraw Hill, London, 4 edition, 2001.

[105] J. D. Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms.* PhD thesis, Vanderbilt University, 1884.

[106] P. Hajela and C. Y. Lin. Genetic search strategies in multicriterion optimal design. *Structural Optimization*, 1992.

[107] J. Horn and N. Nafpliotis. Multiobjective optimization using the niched pareto genetic algorithm. *IlliGAL Report Illinois Gentic Algorithms Laboratory*, 1993.

[108] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. *Computer Engineering and Communication Networks Lab (TIK)*, 1998.

[109] H. Ding, L. Benyoucef, and X. Xie. A simulation-based multi-objective genetic algorithm approach for networked enterprises optimization. *Engineering Applications of Artificial Intelligence*, 2005.

[110] J. D. Knowles and D. W. Corne. The pareto archived evolution strategy: A new baseline algorithm for multi-objective optimization. *IEEE Proceedings of the 1999 congress on evolutionary computation*, pages 98–105, 1999.

[111] K. Deb, S. Agarwal, A. Pratap, and T. Meyarian. A fast and elitism multiobjective genetic algorithm: Nsga ii. *IEEE Trans. Evol. Comput.*, 6:182–197, 2002.

[112] D. Corne, J. Knowles, and M. Oates. The pareto envelope-based selection algorithm for multi-objective optimization. In *The Proceedings of The Sixth International Conference on Parallel Problem Solving from Nature*, pages 839–848, 2000.

[113] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutinary algorithm. Technical Repor 103, Computer Engineering and Networks Laboratory, 2001.

[114] M. J. van de Vijver, Y. D. He, L. J. van't Veer, H. Dai, A. A. Hart, D. W. Voskuil, G. J. Schreiber, J. L. Peterse, C. Roberts, M. J. Marton, M. Parrish, D. Atsma, A. Witteveen, A. Glas, L. Delahaye, T. van der Velde, H. Bartelink, S. Rodenhuis, E. T. Rutgers, S. H. Friend, and R. Bernards. A gene-expression signature as a predictor of survival in breast cancer. *The New England Journal of Medicine*, 347(25):1999–2009, December 2002.

[115] J. Li, H. Liu, J. R. Downing, A. E. Yeoh, and L. Won. Simple rules underlying gene expression profiles of more than six subtypes of acute lymphoblastic leukemia (all) patients. *Bioinformatics*, 19(1), 2003.

[116] S. Ramaswamy, K. N. Ross, E. S. Lander, and T. R. Golub. A molecular signature of metastasis in primary solid tumors. *Nature Genetics*, 33:49–54, 2003.

[117] G. Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 43–52, San Diego, CA, USA, 1999.

[118] C. Ambroise and G. J. McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc Natl Acad Sci*, 99(10):6562–6566, 2002.

[119] B. Efron and R. J. Tibshirani. Improvements on cross-validation: the .632+ bootstrap method. *J Americam Statistical Association*, 92:548–560, 1997.

[120] Camillo Jorge Santos Oliveira. Classificação de imagens coletadas na web. Master's thesis, Universidade Federal de Minas Gerais, 2001.

[121] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

[122] Carolina Baldisserotto. Técnicas de aprendizagem de máquina para previsão de sucesso em implantes dentários. Trabalho de Conclusão de Curso de Engenharia da Computação (UFPE), 2005.

APÊNDICE A

A Tabela 1 ilustra alguns genes pertencentes à base NCI60 [20], trazendos os quatro primeiros e os quatros últimos genes presentes nesta base, além de seus níveis de expressão e sua classificação.

Tabela 1: Fragmento da base NCI60

Amostra	**Genes e suas expressões gênicas**									**Classe**
	0001	0002	0003	0004	...	0997	0998	0999	1000	
1	-0,164161	-4,8848	2,0963	-0,534775	...	0,457161	0,191355	-0,611755	0,0839329	1
2	-3,8759	-3,76918	2,01063	3,11975	...	0,186854	0,962579	-0,401814	1,78893	1
3	-4,34999	0,410967	-2,92301	-4,35091	...	-0,774644	-1,07119	1,40315	-1,83591	1
4	-5,29456	-3,09717	-3,22842	-2,18553	...	-0,248217	0,0858496	-0,989704	-0,437588	1
5	-5,19037	-4,50851	3,82538	-3,14846	...	-0,963794	1,57446	0,34804	-0,787848	1
6	-6,65517	-6,19736	4,41931	-4,2562	...	-1,13668	1,42771	-0,146111	-0,488786	1
7	-3,92652	0,267668	-2,36513	-3,61612	...	0,41846	-0,0695651	1,64288	-0,244977	1
8	0,656287	-4,92744	0,605895	0,868257	...	0,280724	1,29956	-0,201284	-0,506502	2
9	0,327138	-3,71334	2,10391	-1,51799	...	-1,30169	0,33943	-0,318589	0,583541	2
10	-2,18271	-4,89847	0,445682	0,864383	...	-0,488053	1,8196	0,288936	-0,281444	2
11	1,5362	-4,25094	0,772748	-3,02959	...	-0,112368	0,833978	-0,387631	-0,311587	2
12	1,74647	-3,93992	1,81068	-2,46432	...	0,102225	0,587927	-0,281781	1,51905	2
13	-2,04592	-4,64978	3,09726	1,34988	...	-0,00162903	0,00451977	-0,0488471	0,543234	2
14	-4,58404	2,24431	-2,08176	-4,95283	...	-0,438279	0,0359224	1,91293	0,377864	3
15	-1,74558	1,50375	-0,533707	-3,22361	...	-0,33104	-0,603297	-0,353551	0,180887	3
16	-4,20519	-0,33811	-0,354664	-4,74961	...	-1,31752	-0,904932	-0,74256	-1,00899	3
17	-3,72242	1,41686	-2,20511	-3,51661	...	0,122766	-0,93728	0,675619	-0,668483	3
18	-3,40815	1,79236	0,160562	-3,44727	...	-0,125658	-0,307894	1,23935	1,2473	3
19	-4,555	1,07009	-3,53816	-4,21284	...	-0,73219	-0,389328	-0,323507	-1,71748	3
20	-3,92233	1,26645	-0,0719507	-3,85631	...	0,0358646	-0,160007	0,148397	0,0290838	3
21	-5,32291	-4,01798	-2,14245	-5,24112	...	-0,474868	0,169888	-1,66133	-0,682513	4
22	-3,76692	-3,51577	-1,26189	-3,01223	...	-0,219042	-0,13643	-1,53748	-1,35238	4
23	-3,5108	-0,51563	-3,6797	-2,72406	...	1,6996	-0,391022	-1,14911	-0,837641	4
24	-4,11077	-4,07627	-0,855212	-4,17887	...	-0,0448836	0,600618	-1,75416	-1,02973	4
25	-3,12699	-0,250697	-1,14948	0,612616	...	-0,36596	-0,581312	-0,278901	-0,459154	4
26	-4,38922	-4,38112	-2,79697	-4,80341	...	-1,826	-0,0901879	-1,47845	0,364429	4
27	-3,90606	-2,85282	-1,9443	-0,661495	...	0,32868	0,385334	-0,468209	-0,299772	5
28	-4,04712	-3,81434	3,92145	-0,0933891	...	-0,867585	1,13333	0,192118	-1,14206	5
29	-2,93464	-3,21917	2,88026	-0,662403	...	-0,243017	1,10976	-0,425754	-0,698014	5
30	-4,58145	-4,3876	3,84132	-1,09215	...	0,456843	1,11968	-0,592796	-0,213937	5
31	-2,62531	-3,60226	2,80123	1,80131	...	-0,495891	2,00023	-0,31613	0,609825	5
32	-6,04759	-5,43341	0,735293	-2,52952	...	-0,513214	2,34576	-0,847313	-0,557261	5
33	-3,26477	-3,29238	2,05958	0,167136	...	-0,376271	1,37391	-0,649388	-0,273737	5
34	-4,2192	-4,44825	4,0366	-0,966085	...	-0,582756	1,13634	0,106253	-0,377676	5
35	0,719652	-2,22066	0,149834	-1,75318	...	0,0890649	0,617618	0,0407661	-0,863616	6
36	-2,77515	-1,38629	-0,942618	-1,97052	...	0,43488	0,216116	-0,0232552	-0,0452224	6
37	1,42846	-2,43593	0,0852813	-0,139468	...	0,376792	1,34191	0,336734	0,997303	6

38	-6,60547	-2,55737	-1,58175	0,38433	...	1,10781	0,922406	-0,371142	0,519772	6
39	1,43531	-3,07289	0,458781	-3,96217	...	-0,164157	0,423992	-0,392538	0,322683	6
40	1,10256	-2,66995	0,691393	-3,50732	...	0,789153	0,633915	0,588746	1,56927	6
41	-3,04775	0,75296	-0,676708	-3,11158	...	-0,252044	-0,862433	0,192085	-0,232298	6
42	1,45398	-3,50006	3,47098	0,86778	...	-2,21328	-0,392764	0,788429	0,867195	6
43	0,716666	-3,85022	-1,02366	-4,83664	...	-0,825387	0,770405	-0,330002	-1,15479	6
44	-3,7439	-1,04524	-2,4685	-4,1153	...	-0,380604	-0,371351	-1,32162	0,117363	7
45	0,649325	0,803413	-1,15137	-4,02577	...	0,0115508	0,0074894	-0,0617476	0,201552	7
46	-1,78431	1,09893	-1,34046	-2,98738	...	0,161457	0,78973	-0,229251	0,651669	7
47	-3,71313	-0,152521	1,1782	-2,59209	...	0,171932	-0,586802	0,187039	0,127809	7
48	1,11499	-2,0407	-0,287684	-0,0989378	...	0,292568	0,528733	1,02933	0,187235	7
49	-2,95444	0,090369	2,42615	-2,40244	...	0,131778	1,21708	0,438408	0,742201	7
50	-0,767671	-2,48372	-0,0324988	0,394634	...	-0,685856	0,258114	-0,0776372	0,583	8
51	-0,554984	-3,29686	0,730205	-3,83809	...	0,188504	-0,0833314	-0,150136	0,0805609	8
52	-3,43941	-0,805609	-0,00166909	-0,0987586	...	-1,28297	-0,1479	-0,73187	-0,449098	8
53	1,71796	-2,9929	0,941972	-0,418864	...	0,581641	1,02023	-0,516387	0,62146	8
54	-0,785637	-3,10034	0,480331	0,605166	...	-0,412405	-0,486664	-0,416042	0,743273	8
55	1,74548	-4,70277	-0,270924	-4,27518	...	-2,50858	0,8392	0,943643	0,621112	8
56	-0,164802	-0,75772	0,872185	1,04158	...	-0,608374	0,22955	-0,563887	0,238201	8
57	1,43745	-0,34705	1,05609	-0,0964912	...	-0,604294	0,575899	0,267507	1,37224	8
58	-3,12265	-0,000776301	-0,969465	-2,89194	...	2,41071	-0,2419	-1,16987	-1,1586	9
59	-3,40592	-0,495942	-2,53151	-3,70448	...	0,828309	-0,632215	-1,11793	-0,952129	9
60	-4,45113	0,24414	-1,59187	-4,00739	...	-1,22435	-1,31178	1,4053	-1,12544	9
61	-3,24625	0,538351	-2,28288	-3,61045	...	-0,0822695	-0,674677	1,40586	-0,547719	9

APÊNDICE B

As tabelas abaixo ilustram os genes, seus níveis de expressão gênica e sua classificação, dentre as 9 classes possíveis (Tabelas 2, 3, 4 5, 6 e 7).

Tabela 2: Códigos e expressão gênica dos genes da base de dados B_1

	Códigos e expressão gênica dos genes da base de dados B_1													
Amostra	0011	0050	0097	0127	0194	0242	0289	0348	0366	0828	0839	0863	0881	**Classe**
1	-0,103522	0,0749199	0,774746	-1,71629	-0,732035	1,35166	-2,07422	-2,04733	0,46645	-0,364501	-0,733991	0,834798	0,156616	1
2	-2,21418	-2,27882	-2,41233	0,462524	-2,05625	-0,541687	-2,2083	-0,564796	0,310698	-0,278658	-0,0236491	-0,00133864	0,406395	1
3	-0,82912	-0,629544	0,603895	-2,35566	-1,52958	1,03772	-3,26092	-3,00894	-0,0873521	-0,758008	-1,14556	0,939854	-0,736931	1
4	-3,51391	1,65066	0,380296	0,280169	0,0410925	-1,21763	-0,954262	-1,00747	0,583922	0,412475	-0,301249	0,652274	-0,465053	1
5	2,02585	0,882851	0,601008	-0,0862014	-0,0762004	0,61535	0,0533926	-2,19736	0,516784	1,0259	0,776571	0,580852	0,847398	1
6	0,336447	-0,169524	-0,704792	1,32164	-0,272202	0,622776	0,491263	-0,684165	0,304506	-0,724929	0,32339	0,133891	0,222303	1
7	-3,80793	-3,03094	-3,10915	-2,00082	-2,25424	-1,35933	-1,30518	-1,11802	-1,12114	-1,31643	-1,33791	-1,24776	-0,715655	1
8	1,40794	1,61487	0,179707	-0,404795	-0,554994	-0,359214	-1,62607	-2,20106	1,38448	0,412437	1,33742	1,13253	1,57853	2
9	0,586283	0,980219	0,520937	0,898916	-0,216614	1,27262	-1,25022	-1,57864	0,236351	0,141967	0,943141	1,13712	0,954878	2
10	2,00007	0,130301	0,297864	-0,448922	-0,116546	0,309864	-1,9062	-2,22374	0,071138	-0,664907	1,50146	0,671309	1,15145	2
11	0,49241	0,786922	0,367203	0,72829	0,457588	-0,304991	-0,615864	-0,729053	0,537549	0,605988	0,562647	0,732964	1,17502	2
12	0,471609	1,48882	0,847666	0,30918	-0,307932	0,925249	-2,5171	-2,64711	0,292831	-0,213347	1,04789	1,08715	0,706803	2
13	2,18319	0,896251	0,285514	0,730159	-0,292397	0,142945	-0,598794	-0,0103232	0,569385	0,67578	0,531138	0,531094	1,57358	2
14	-2,69213	-2,41228	-0,590777	-1,73579	-1,18964	0,543226	0,484663	-0,618737	0,36123	-0,614338	-0,0108392	0,597166	-0,465481	3
15	-2,29205	-2,30448	-2,80679	-1,87227	-2,18809	-0,913112	-0,217442	-1,62205	-0,57295	-1,53081	-0,696569	-2,37972	-1,54185	3
16	-4,29216	-2,80198	-1,96733	-1,94696	-1,85785	-1,79681	0,157092	-2,18478	-1,47146	-2,06176	-0,391953	1,11249	-1,31245	3
17	-2,0357	-2,88937	-1,61938	-1,81546	-1,78242	-0,269746	0,406382	-1,29552	0,00219804	-1,17216	-0,734569	-0,983874	-0,772641	3
18	-3,38821	-3,11957	-2,89618	-2,82773	-2,06026	0,385011	0,167933	-1,00965	0,108283	-0,72046	-0,723643	-1,41791	-1,28329	3
19	-2,84401	-2,79649	-2,43453	-1,91015	-1,81497	0,43657	0,608268	-1,53836	0,123147	-1,11379	0,068236	-0,280746	-0,607476	3
20	-3,68907	-4,64572	-2,42695	-0,626419	-1,78739	-0,452533	0,0974808	-1,21219	-0,419689	-0,3288	0,347022	-1,19581	-0,896939	3
21	-2,87948	-0,742572	-0,407514	-2,04875	-2,41412	-0,595085	-0,586813	-0,354366	-0,852477	-0,819172	-1,29254	0,173609	-0,692541	4
22	-6,15868	1,1322	0,283168	-3,26654	-4,01684	-2,51971	2,39585	-1,99533	-0,240053	0,244785	-1,29096	-0,0334306	-0,736428	4
23	-3,76253	-1,12904	-1,34095	-1,70678	-3,42609	0,122901	0,631448	-2,10076	-0,417069	-1,42428	-0,662854	-0,100966	-0,860344	4
24	-4,14299	1,39347	-0,527461	-3,20484	-5,16398	-2,08439	1,04189	-2,9588	-0,173309	-1,80735	-2,20392	2,07077	-1,01906	4
25	-3,28705	1,2316	0,715167	-2,81273	-3,13302	-0,695066	1,63402	-2,53046	-0,440182	-2,0684	-1,79503	1,05069	-0,794289	4
26	-3,08938	-1,96779	-3,0712	-0,720666	-0,82602	-1,91835	-0,831813	-1,6309	-0,737671	-0,565236	-0,578867	-0,900492	-0,269402	4
27	1,17806	-0,645115	0,0467692	-0,776113	-0,460068	1,15558	1,99245	-1,86931	0,372313	0,223906	-0,364876	0,69686	-0,517255	5
28	-0,410018	-2,26939	0,0851878	-1,5113	-0,444469	0,336779	-1,23846	-1,60814	-0,558132	-0,0966399	-0,866592	0,922814	-0,530827	5
29	-1,49399	-4,23973	0,0905471	-1,45846	-0,868005	0,952073	2,54113	-1,96941	-0,38758	-0,796449	-0,51404	0,305297	-0,50833	5
30	0,393862	-0,765177	-0,0405467	-1,97391	-1,10304	1,68784	0,292586	-1,82822	0,678946	-0,162251	-1,64911	0,841132	-0,150524	5
31	0,923374	-2,01437	-0,192461	-1,868	-0,527787	2,15467	1,25614	-1,83716	0,659145	0,212095	-0,312122	1,26541	0,620948	5
32	0,0855046	-2,82376	-0,664243	-1,61334	-1,0453	1,08286	0,395048	-1,52167	0,0309763	-0,482929	-0,673715	-0,125413	0,281064	5
33	-1,05559	1,10612	-0,535704	0,641826	0,862049	0,737617	0,717859	-3,2207	0,587051	-0,203198	-0,0430824	0,886055	-0,72226	5
34	-0,384449	-1,22456	-0,170168	-2,23068	-1,28553	0,586728	-0,733935	-2,54251	0,493198	0,216003	-0,638189	0,875273	0,0461134	5
35	0,31306	0,614041	0,82009	0,513128	0,698346	-0,331419	0,734916	0,726677	1,11414	-0,0145314	1,15386	0,920167	0,883913	6
36	-2,50761	-0,238344	1,07764	-0,0354999	0,541246	-1,08619	-0,464893	0,471643	0,593036	0,0484671	0,18921	0,820677	-0,0979766	6
37	-2,42524	-1,44509	1,7524	0,0539245	-0,218165	-0,636298	0,58409	1,15336	0,142132	-0,446367	-0,474743	0,119492	-0,541702	6

38	-2,41394	-0,0718925	-1,2187	-1,07679	-1,3584	-1,09522	-1,22145	0,162176	-0,080729	-0,226777	-0,0639266	-0,286434	-1,30017	6
39	-2,67337	-0,147557	0,970814	-1,07819	-1,52943	-0,783347	0,570793	0,979778	2,32308	-0,527892	-0,282687	0,916263	-0,801252	6
40	-3,24614	0,174114	-0,302184	-1,64422	-1,86895	-3,45558	-1,99476	-2,54092	0,178187	-1,06027	2,10983	-0,0528301	-1,36209	6
41	-0,199181	1,06124	-1,33686	-0,0990531	-0,335269	-1,9451	0,635635	-1,24567	1,28469	0,671287	-0,506219	-0,577277	0,942283	6
42	1,87582	1,29876	0,456242	1,52092	0,825242	0,287893	-0,440944	1,95797	0,289187	-0,119422	0,745059	1,11234	0,0901404	6
43	-2,33444	-2,28379	0,763652	-0,619977	-0,289064	-0,732774	1,4152	-1,03817	1,58669	-0,11922	1,70361	0,84011	0,607704	6
44	0,262613	3,01561	1,31018	1,40401	0,55978	1,0065	0,786736	0,718503	1,33121	0,896285	-0,393228	1,05361	0,907079	7
45	0,530779	0,477625	0,412877	1,83338	0,957326	-0,0678755	1,57048	0,765237	0,051733	0,323563	-0,510282	0,498072	1,06657	7
46	-2,10392	0,638376	-0,0791271	0,605212	1,40616	0,453661	-0,0665868	0,509037	-0,197069	-0,535656	-0,263124	-0,0384527	0,186147	7
47	-4,05749	-1,51122	0,0449173	0,106034	1,26553	1,02371	-0,990445	1,10968	0,739371	1,17027	-0,201468	-0,868864	0,957581	7
48	-3,16229	0,420312	-0,738013	-0,375574	-0,727247	0,369401	0,588803	-0,86316	1,02351	-0,138821	-0,0886083	0,256372	0,133302	7
49	-3,39343	1,08845	-0,831538	-1,25341	0,210505	-0,0554015	-0,0763907	-1,90375	0,025112	-0,942	-0,280944	-0,886571	-0,552479	7
50	1,19139	1,22537	1,73514	1,78471	0,993739	0,790705	0,881833	-0,814253	1,38997	1,99828	0,624905	0,689459	1,64018	8
51	-2,3231	-1,18463	2,59287	1,33982	0,270411	-0,592037	-0,758778	-1,45767	0,21315	0,968477	-1,66819	0,107753	0,847117	8
52	1,23518	-0,152463	0,7591	1,58115	0,425561	-4,03947	0,645642	-1,20161	0,517934	1,14885	-0,189698	-1,31459	1,37064	8
53	-0,154612	-1,30148	1,67287	0,326719	0,28034	-0,360302	0,699735	-1,81428	0,931002	-0,570926	-0,819902	0,2556	0,388869	8
54	1,44623	-0,590017	2,18464	1,483	0,405446	-0,0404916	1,16946	-1,04315	0,856753	2,16557	0,482948	0,318704	1,17886	8
55	-0,269493	0,271419	2,12176	0,734052	-0,333213	-0,22917	1,04013	-2,23434	0,178397	0,127597	-0,426538	0,344658	0,966374	8
56	0,238757	-0,899222	2,32858	1,12464	0,853164	0,828158	0,542677	-1,10833	0,45591	0,0776867	0,0119624	0,395237	0,678021	8
57	-0,440923	-0,0198836	1,74694	1,14677	-0,224056	-0,0836071	-0,789598	-1,35514	0,547858	0,593202	-0,998126	-0,11146	0,218254	8
58	-3,54653	-3,08779	-3,62545	-1,64065	-2,11245	-1,63871	-1,03034	-1,62591	-0,58583	-1,47136	-1,16989	-0,836162	-0,149909	9
59	-3,50223	-2,33792	-2,52427	-1,03333	-1,58614	-1,07298	-1,16499	-0,369161	-0,607675	-1,04261	-0,978751	-0,674263	-0,125554	9
60	-2,24822	-2,17369	-2,17727	-1,99502	-0,232019	-2,28349	-0,387918	-0,851332	-3,14675	-0,345106	-0,428909	-0,966665	-0,196995	9
61	-3,54528	-3,0864	-2,52507	-2,60718	-1,00157	-2,72377	-0,953137	-1,51944	-5,95247	-0,695914	-0,699754	-1,22513	-0,799339	9

Tabela 3: Códigos e expressão gênica dos dez primeiros genes da base de dados B_2

	Genes e suas expressões gênicas										
Amostras	0002	0017	0018	0019	0028	0075	0097	0141	0224	0231	**Classe**
1	-4,50851	-0,337059	-0,116572	-3,5135	1,26755	0,487778	0,774746	-0,601873	0,315961	0,468817	1
2	0,267668	-2,60075	-2,43295	0,343207	-0,43818	0,375522	-2,41233	-0,440962	-0,910274	-0,411599	1
3	-6,19736	-1,00459	-0,523421	-4,02536	0,790664	0,42316	0,603895	-1,10987	-0,358966	0,553991	1
4	-3,09717	-1,09976	1,47741	-2,54687	-0,961194	1,95633	0,380296	0,80071	-0,164686	1,29704	1
5	-3,76918	1,90559	1,66304	-3,59062	0,451119	0,489401	0,601008	1,75136	1,10089	0,406921	1
6	-4,8848	-0,371121	1,73563	-4,30754	0,107885	0,277914	-0,704792	1,40622	-0,111332	-0,497452	1
7	0,410967	-4,70241	-2,91712	0,676599	-2,45178	-1,82778	-3,10915	-0,853595	-0,998001	-1,87113	1
8	-4,64978	1,0533	1,31886	-3,36685	0,824513	1,36745	0,179707	-0,332799	1,13079	0,941352	2
9	-3,93992	0,18646	1,05696	-3,02071	1,23568	1,42367	0,520937	0,94428	0,343336	0,789843	2
10	-3,71334	0,0113592	2,02081	-3,01391	-0,541978	0,345248	0,297864	0,505982	0,985056	-0,838194	2
11	-4,92744	0,664066	1,63741	-2,45224	-1,50278	1,01785	0,367203	1,28713	-0,827346	0,785339	2
12	-4,25094	-0,372414	1,69849	-3,9258	1,0243	1,2503	0,847666	0,34651	-0,6642	0,408859	2
13	-4,89847	0,408468	-0,22939	-3,37565	-0,184535	0,58102	0,285514	0,454732	-0,239393	0,746571	2
14	1,26645	-0,733324	-1,47207	1,43666	-1,75672	0,0105565	-0,590777	0,297345	-1,2085	0,0769802	3
15	1,07009	-4,70419	-1,44096	1,04022	-1,32356	2,07231	-2,80679	-2,3943	-0,66684	0,585915	3
16	-0,33811	-2,62027	-0,236484	-0,245908	-0,98645	-0,255764	-1,96733	-1,71535	-2,56973	-0,09162	3
17	1,50375	-3,91724	-2,92373	1,50194	-2,48502	0,257606	-1,61938	-0,780028	-0,542577	0,204203	3
18	2,24431	-4,3415	-2,94559	2,23819	-1,32756	1,65542	-2,89618	0,878451	-0,801437	-0,149849	3
19	1,41686	-3,97608	-2,01646	1,50591	-1,96889	-0,694375	-2,43453	-1,39734	-0,705848	0,236278	3
20	1,79236	-4,61155	-3,04138	1,61087	-4,91052	1,94995	-2,42695	0,160983	-0,67818	1,15541	3
21	-0,250697	-2,71582	-0,796259	-0,419231	-2,12761	-0,82622	-0,407514	-0,0124978	-2,27384	-1,95494	4
22	-4,38112	-5,06632	-2,79815	-3,20275	-4,46021	0,925874	0,283168	-0,165052	-2,73772	0,58795	4
23	-3,51577	-3,53722	-3,26023	-3,35546	-2,40225	-2,64249	-1,34095	-1,52068	-2,76937	-2,63317	4
24	-4,01798	-4,58858	-4,31109	-3,14221	-3,93166	-3,64305	-0,527461	-3,67123	-3,37101	-3,64149	4
25	-4,07627	-4,30051	-3,31691	-3,73568	-3,51828	-3,45385	0,715167	-2,85785	-3,12645	-2,92595	4
26	-0,51563	-4,41176	-3,48574	-0,538944	-1,65469	-3,99163	-3,0712	-4,37396	-2,49494	-3,02845	4
27	-4,44825	-0,0705998	-0,590805	-3,96555	2,13124	0,681558	0,0467692	0,0200185	1,12187	0,201523	5
28	-3,29238	-0,793769	-2,54199	-3,1943	0,394608	-0,618114	0,0851878	0,179949	-0,541336	-0,028039	5
29	-5,43341	-0,121207	-1,92793	-3,52975	1,35325	-3,48629	0,0905471	1,36791	-0,298023	-0,0369958	5
30	-4,3876	-0,481043	-0,300642	-3,71414	1,45413	-0,402383	-0,0405467	-0,686775	0,0514112	-0,215646	5
31	-3,60226	0,506947	-1,12972	-3,26	1,86334	-1,63509	-0,192461	-0,010832	0,264951	0,903693	5
32	-3,21917	0,0646114	-0,169651	-3,29961	1,14384	0,704115	-0,664243	-0,440671	0,237503	0,127389	5
33	-2,85282	-0,508199	0,0909463	-3,62985	-0,22302	-0,933449	-0,535704	0,708899	0,385015	1,13853	5
34	-3,81434	-0,210048	0,0488285	-3,32461	0,780449	-0,179484	-0,170168	-1,27455	0,231756	0,35137	5
35	-2,43593	0,282223	1,52297	-1,78881	0,314759	-0,253444	0,82009	2,21143	1,13973	0,947879	6

36	-1,38629	-0,612974	-0,189354	-1,58909	-0,224662	1,01911	1,07764	0,409918	0,00474883	0,471709	6
37	-2,22066	-0,859775	0,344879	-1,91098	1,01039	0,505603	1,7524	-0,0995294	0,718602	0,233168	6
38	0,75296	-2,92049	-0,751447	0,928465	-0,459217	-2,75017	-1,2187	-0,333738	-0,91784	-0,06613	6
39	-3,50006	-1,17176	-0,846374	-4,31056	0,209739	0,702584	0,970814	0,473076	0,69259	1,35516	6
40	-3,85022	-0,419221	-2,895	-3,38718	-2,55519	-3,39945	-0,302184	-0,0553295	-4,3584	0,929327	6
41	-2,55737	-0,544937	1,71558	-2,22152	0,258625	1,86874	-1,33686	1,21499	0,918059	0,508529	6
42	-3,07289	-0,922308	0,103946	-3,26766	-0,357875	0,264378	0,456242	0,840762	-0,381537	0,915253	6
43	-2,66995	-0,593702	0,528374	-2,24688	0,244994	1,02921	0,763652	0,776011	1,35472	1,11443	6
44	0,090369	-1,54622	1,82909	0,299648	2,15323	2,47786	1,31018	0,940882	-0,000424902	1,61928	7
45	-2,0407	-0,0256595	0,287134	-1,66494	-1,03034	-0,22504	0,412877	1,55535	-0,136178	1,85885	7
46	0,803413	-0,397069	-0,796434	0,817594	0,000907711	0,722143	-0,0791271	-0,275441	0,0873685	-0,402553	7
47	1,09893	2,18631	-1,23411	1,33306	-0,161957	-0,815032	0,0449173	0,20689	0,729631	0,741929	7
48	-0,152521	-2,68387	-0,12263	-0,237039	0,895815	0,438666	-0,738013	-0,110954	1,23254	0,808001	7
49	-1,04524	-1,13258	-1,42209	-0,958473	1,03653	-0,689701	-0,831538	-0,642301	-0,186687	0,324727	7
50	-0,34705	0,920139	1,20869	-0,195479	2,04715	0,250675	1,73514	1,53571	1,81311	1,34029	8
51	-0,75772	-1,91798	-0,265394	-0,71091	1,06212	0,397646	2,59287	-0,229233	-0,0967748	0,726098	8
52	-4,70277	-2,84048	1,45396	-3,0756	-0,0160491	0,130352	0,7591	2,11516	-0,450487	1,37938	8
53	-2,48372	-0,918659	0,789364	-2,489	1,77544	-0,0965556	1,67287	0,946485	-0,371652	-0,0849264	8
54	-2,9929	-0,321648	1,35242	-2,64201	1,85912	0,772677	2,18464	1,23691	1,15106	0,758469	8
55	-3,10034	0,158647	0,973178	-2,62379	1,56061	0,570008	2,12176	0,670583	0,451686	0,43378	8
56	-3,29686	0,12611	1,34305	-3,43822	1,26926	0,616999	2,32858	0,795452	-0,0757759	0,583863	8
57	-0,805609	-1,69713	1,05458	-0,645084	1,1909	0,765307	1,74694	0,292051	0,429605	0,826174	8
58	0,24414	-4,50053	-5,35013	0,372298	-2,30905	-2,47339	-3,62545	-1,24874	-0,621853	-1,57274	9
59	0,538351	-3,67387	-3,29306	0,453264	-1,75937	-1,57628	-2,52427	-0,625486	-0,467699	-1,43954	9
60	-0,000776301	-4,038	-3,31019	0,160958	-3,27065	-2,34785	-2,17727	-2,879	-1,51753	-2,17496	9
61	-0,495942	-4,64065	-5,03725	-0,143366	-4,83851	-2,87258	-2,52507	-3,70694	-2,14193	-3,35003	9

Tabela 4: Códigos e expressão gênica dos dez últimos genes da base de dados B_2

	Genes e suas expressões gênicas										
Amostra	0235	0246	0280	0292	0302	0409	0499	0526	0637	0843	**Classe**
1	-0,515733	0,921301	-0,0401955	-0,183671	0,0996163	1,26979	0,179124	0,77547	0,864763	0,647987	1
2	-0,514945	-0,122969	0,358636	-2,18912	-1,50235	-0,957949	-0,74193	-0,35398	1,39079	1,11175	1
3	-1,13805	1,10688	-0,222135	-0,532276	-0,538659	1,6218	-0,385338	0,694388	1,11697	0,270405	1
4	0,613584	0,502628	0,597919	1,17556	1,46077	-0,27492	0,000975853	1,53539	-0,024048	0,0324528	1
5	1,57028	2,06523	0,670213	1,31194	1,5436	0,191686	-0,0765836	0,537336	0,56749	1,07641	1
6	1,17388	1,36603	0,365911	1,42796	1,43696	0,40482	-0,748932	0,81097	-1,15699	0,615342	1
7	-1,01837	-1,48972	-1,6195	-1,56851	-1,3536	-0,360527	-2,62518	-0,0954974	0,581828	-0,192222	1
8	-0,206956	1,51256	0,522783	1,2079	0,910651	1,00432	0,0354808	0,577905	-0,951725	1,4127	2
9	0,735171	1,05532	0,455076	0,931361	0,865049	0,535273	0,975584	0,512237	0,194887	0,982211	2
10	0,349158	0,849092	-0,282803	1,49248	1,75127	-0,505967	-0,138112	0,280999	0,384641	-0,0948019	2
11	0,901404	1,14948	0,93913	1,42787	1,69294	-0,159616	0,651218	0,994903	0,382547	0,463685	2
12	0,29231	1,04147	0,259673	1,51549	1,60014	0,146323	0,570392	0,752681	-0,0739046	1,13354	2
13	0,444304	1,18875	0,709836	0,433777	0,434897	-0,184408	0,391723	1,10974	0,251034	0,488672	2
14	0,174037	-0,433162	0,27555	-1,04093	-0,425406	-0,721133	-1,03218	-0,358862	0,557731	-0,526954	3
15	-1,18534	-2,33145	-1,19883	-1,62237	-1,25117	-1,39229	-0,348718	-0,0964686	-1,00557	-0,95192	3
16	-1,69009	-0,536451	-1,08472	-0,380437	-0,328425	-0,84824	-0,915682	-1,32633	0,0317292	-0,570794	3
17	-0,558247	-2,46283	-1,3241	-1,81861	-1,63576	-0,705241	-0,326291	-0,829957	0,482984	-0,157082	3
18	0,940425	-1,20444	-0,202131	-2,42948	-1,94396	-0,788282	-0,13458	1,03244	0,194685	-0,15746	3
19	-0,577635	-2,26164	-0,0468545	-1,43587	-1,34597	-1,30886	-0,791337	0,0457742	0,396511	0,357897	3
20	-0,479448	-1,83272	0,263095	-1,17136	-1,2496	-0,283526	0,658042	0,615285	0,613426	0,113116	3
21	0,0429842	-0,672263	-1,05513	-0,844333	-0,576423	-0,313843	-1,47812	-0,926148	-1,37604	-1,09016	4
22	-0,150066	0,831934	-4,37371	-1,85119	-1,68556	-1,31549	-0,81591	-1,30677	0,892716	-1,75618	4
23	-1,21856	-2,40949	-1,47857	-1,88168	-1,94408	-1,53647	-2,75069	-2,30437	-1,58867	-1,83726	4
24	-2,86355	-1,49415	-2,68145	-1,85501	-1,86911	-1,64162	-2,43304	-2,60234	-2,34061	-1,48298	4
25	-2,73031	-2,78264	-1,97788	-1,65072	-0,265489	-0,840606	-3,16452	-2,28318	-1,92305	-1,39911	4
26	-2,64758	-1,39417	-2,22538	-2,11434	-1,54839	-1,49512	-1,59626	-2,2853	-2,8577	-1,8224	4
27	-0,13707	1,25712	-0,00129788	-0,550194	-0,539786	1,73343	-0,173191	-0,0127347	1,29602	0,478424	5
28	0,220008	-0,292395	-0,401752	-1,77723	-1,82431	0,870253	-0,572574	-0,202331	1,45603	-0,414298	5
29	1,24733	0,526358	-1,53867	-1,79738	-1,41599	-0,809556	-1,81007	-0,228377	1,29602	0,589857	5
30	-1,09845	0,721852	-0,0390497	0,071329	-0,0749881	0,33696	-0,441043	1,00939	0,357202	0,343044	5
31	-0,0881586	1,68473	0,433627	-0,303109	-0,852309	-0,173336	0,223922	0,184095	1,15226	0,933484	5
32	-0,521074	0,900311	0,301705	-0,324231	-0,224173	0,533193	-0,272054	-0,241965	0,24597	1,07669	5
33	0,678049	0,488721	0,37144	0,0124113	-0,0987403	2,40047	0,234297	0,729877	0,625644	-0,584566	5
34	-1,28754	0,857237	-0,274175	0,434505	0,00742046	1,15413	-0,219753	0,164703	0,898542	1,00901	5
35	2,20137	1,58049	1,67931	1,44773	1,39261	1,10752	0,607226	1,93184	1,11067	0,724547	6

36	0,443835	0,202692	0,314256	-0,0948575	-0,0672807	0,0919746	0,286794	0,192691	1,18658	0,642552	6
37	0,0799567	0,0993102	-0,361401	0,263231	0,391199	0,436611	0,859523	-0,0926942	0,799958	-0,0178569	6
38	-0,20669	-2,08462	-0,483935	-0,78114	-0,651463	-1,21258	-0,396336	-0,422318	-0,0991803	-0,109373	6
39	0,0139885	-0,0305028	0,784135	-0,86835	-0,568267	-1,29473	0,333106	0,0761062	0,777935	0,0879266	6
40	0,00309237	-2,50973	0,0174379	-1,57034	-1,89659	-1,44153	-0,720087	0,10558	0,858501	-0,379475	6
41	1,25358	0,646247	0,412215	1,40206	1,63481	0,684364	0,941268	1,37442	0,552731	0,613798	6
42	1,06917	0,780539	1,15764	0,0606785	0,0524001	-0,66813	1,37872	0,56802	0,980727	1,2155	6
43	0,699917	-0,119836	0,500983	0,602303	0,873755	0,20443	0,5692	0,470464	1,58602	0,977005	6
44	1,15069	1,11158	1,60627	1,88111	1,72065	0,371542	0,263784	1,62342	0,240454	0,702232	7
45	1,22695	1,0792	0,016269	0,687002	0,387864	0,938265	0,894085	1,60876	1,30805	0,945032	7
46	0,00673733	-1,55831	-0,313856	-0,757684	-0,464091	-0,407693	-0,176819	-0,726978	-0,191607	0,572707	7
47	0,448488	0,533118	-0,0354362	-1,02527	-0,723791	-0,433021	0,0344003	0,373211	-0,274207	0,286855	7
48	-0,218801	-0,480561	0,863039	-0,090101	-0,0935466	0,00310246	-0,0573801	-0,297245	0,86006	-0,343652	7
49	-0,63859	-0,217089	-0,220408	-1,36038	-1,19598	0,447424	-0,503265	0,433422	0,245934	0,0608847	7
50	1,44175	0,687947	1,6785	1,07952	1,10036	1,17539	1,26264	-0,154571	1,42497	0,795398	8
51	-0,0144558	0,745998	1,04923	-0,154444	-0,346267	0,772124	-0,0376131	0,605829	0,385668	-0,386962	8
52	2,01045	0,525438	0,324409	1,19664	1,34072	0,429513	1,01812	0,385935	0,689491	1,38688	8
53	0,961054	0,92634	1,09664	0,651677	0,706478	1,34541	-0,152585	0,0657119	0,0664101	-0,678707	8
54	1,01441	0,566964	1,4402	1,07258	1,10136	0,90853	0,733709	0,704871	0,640853	0,512456	8
55	0,260542	0,500643	0,417555	0,726732	0,885937	1,46527	-0,143604	0,251878	-0,0345037	-0,354784	8
56	0,494335	0,411906	0,18915	0,971616	0,902151	1,93853	-1,41486	-0,211052	0,735241	0,0612963	8
57	0,164695	0,0453306	0,890685	0,956029	0,946043	0,619726	-0,184181	-0,0244084	0,935621	-0,5282	8
58	-1,43964	-1,75754	-1,83603	-1,11122	-1,99439	-1,30637	-2,46838	-0,164288	0,368978	-0,245405	9
59	0,000376812	-0,789906	-0,991365	-0,461935	-1,35075	-0,667448	-1,86002	0,333864	0,72217	0,330521	9
60	-3,74875	-0,419895	-2,51329	-0,98359	-1,8793	-3,09597	-1,14517	-2,14324	-0,671	-1,3924	9
61	-3,97687	-0,776504	-3,17569	-1,38751	-1,97772	-2,151	-1,65968	-2,02353	-1,48907	-2,05269	9

Tabela 5: Códigos e expressão gênica dos nove primeiros genes da base de dados B_3

	Genes e suas expressões gênicas									
Amostra	0002	0041	0063	0097	0229	0379	0456	0475	0485	**Classe**
1	-4,50851	-2,52832	-1,10453	0,774746	-0,68317	-2,30162	-1,13307	0,253977	-1,3027	1
2	0,267668	-3,08724	1,2974	-2,41233	-0,693557	-0,883685	2,53544	-0,788185	-0,854117	1
3	-6,19736	-2,6686	-1,50648	0,603895	-1,40403	-1,9673	-1,73029	-0,482072	-2,06446	1
4	-3,09717	-2,15246	-1,82798	0,380296	-1,90856	0,868439	-1,35365	-0,148024	-2,20109	1
5	-3,76918	3,13381	-1,83426	0,601008	1,50088	0,382379	-1,22149	0,241089	-0,317325	1
6	-4,8848	-0,342498	-1,33807	-0,704792	2,13738	1,5046	-1,03519	0,0206586	-0,865357	1
7	0,410967	-3,47007	-2,27014	-3,10915	-2,16559	-0,420848	0,6851	-1,52805	-1,96287	1
8	-4,64978	1,29692	-1,72917	0,179707	1,18536	-0,165679	-0,02842	0,256157	-1,5374	2
9	-3,93992	-1,72083	-0,500528	0,520937	1,54418	-0,763413	1,21861	0,702087	-0,638692	2
10	-3,71334	-1,34378	-4,57514	0,297864	1,2302	-0,815058	0,0719127	-0,561465	-1,80847	2
11	-4,92744	-0,0608777	-1,99941	0,367203	1,59903	1,55635	-0,772605	0,300259	-0,014546	2
12	-4,25094	-3,2904	-1,36056	0,847666	2,01684	-1,4006	0,875864	0,00560726	-1,22024	2
13	-4,89847	0,3557	-1,03606	0,285514	1,35518	0,0634192	-0,868488	0,407251	-0,229996	2
14	1,26645	-2,87056	0,056438	-0,590777	-1,13908	1,42038	0,891777	0,00968717	-0,793581	3
15	1,07009	-3,5447	-0,18016	-2,80679	-2,07075	0,334916	-1,4809	-1,46642	-1,29189	3
16	-0,33811	-3,94117	-0,198896	-1,96733	-2,70728	0,100886	-1,47118	-1,73425	-1,93637	3
17	1,50375	-3,27332	0,164072	-1,61938	-1,44721	0,896244	-0,585318	-0,209419	-1,03479	3
18	2,24431	-3,52506	0,345721	-2,89618	-1,73499	1,45138	1,69596	-1,6507	-0,714454	3
19	1,41686	-2,81741	-0,0247991	-2,43453	-0,935801	0,955075	-0,310487	-2,7076	-0,893618	3
20	1,79236	-3,89177	1,97558	-2,42695	-2,99905	1,33129	-0,399999	-0,277586	-0,0638095	3
21	-0,250697	0,0719375	-2,76306	-0,407514	-1,05565	-0,750952	1,16672	-0,792581	0,809274	4
22	-4,38112	-3,83613	-1,41645	0,283168	0,377106	-1,20533	-0,59488	1,21572	2,10007	4
23	-3,51577	-3,48903	-0,338194	-1,34095	-1,45356	0,0817432	0,644299	-0,552416	2,31652	4
24	-4,01798	-2,23379	-1,21051	-0,527461	-0,683611	1,66608	-1,28351	0,595915	2,58904	4
25	-4,07627	-3,35838	-1,8356	0,715167	-1,04757	1,41301	-1,01671	-0,0699641	2,1719	4
26	-0,51563	-4,14401	-0,37554	-3,0712	-1,32695	1,13343	0,555626	-2,24337	0,784443	4
27	-4,44825	-0,648502	-1,76319	0,0467692	-1,69723	-1,40571	-0,891902	0,9426	-0,684821	5
28	-3,29238	0,242232	-2,29874	0,0851878	-1,55661	-1,30119	-0,98036	0,324644	-1,01967	5
29	-5,43341	-1,93687	-0,809961	0,0905471	-1,09358	0,0596664	-1,22576	-0,0840419	-1,82033	5
30	-4,3876	-1,07374	-1,98235	-0,0405467	-1,42295	-1,04607	-1,40023	-0,0291383	-0,637698	5
31	-3,60226	1,57608	-0,336561	-0,192461	-1,44113	-0,107695	-1,01486	0,167182	-0,468927	5
32	-3,21917	-0,598032	-1,48473	-0,664243	-1,7241	-0,859252	-0,994975	-0,00675686	-1,04267	5
33	-2,85282	-0,758641	-4,46472	-0,535704	-0,967703	-0,470282	-1,35513	1,27071	-0,2401	5
34	-3,81434	-1,67727	-1,47696	-0,170168	-0,883976	-1,23516	-1,22346	-0,436748	-1,66858	5
35	-2,43593	-0,436431	-0,515188	0,82009	0,985526	1,66317	-1,17491	0,804229	-0,347734	6

36	-1,38629	-1,76737	-1,33305	1,07764	-0,211299	0,791063	-0,598743	0,320321	-1,15394	6
37	-2,22066	-2,05428	-0,144939	1,7524	-0,406675	1,14683	-0,968743	-0,0907589	-0,758136	6
38	0,75296	-2,76227	-2,65854	-1,2187	-0,706613	0,395961	-0,344097	0,292664	-0,921065	6
39	-3,50006	0,924406	4,51673	0,970814	-3,07615	2,49773	1,84698	-0,0571699	-0,553393	6
40	-3,85022	-3,77511	-1,46191	-0,302184	0,173081	1,51156	-0,971103	-0,600641	-1,48489	6
41	-2,55737	0,307192	-3,67632	-1,33686	-0,480241	0,268793	-0,0713054	-0,262824	-1,06391	6
42	-3,07289	-2,93876	-1,83723	0,456242	-0,144715	0,458751	-1,03852	1,39474	-0,921014	6
43	-2,66995	-2,72691	0,824269	0,763652	0,899415	0,999218	-0,440553	0,924992	-0,187128	6
44	0,090369	-2,16374	2,67404	1,31018	-0,670896	0,671503	-0,399219	1,00723	-0,0778896	7
45	-2,0407	-0,998101	0,32451	0,412877	0,222206	0,793046	-1,96062	0,943491	-0,944933	7
46	0,803413	-3,39259	0,903812	-0,0791271	-1,2897	-0,446136	-0,145526	-0,0253096	-1,16651	7
47	1,09893	-2,79507	1,54208	0,0449173	-0,761665	-0,595433	-0,727845	0,373393	-0,766875	7
48	-0,152521	-2,57756	1,70697	-0,738013	-1,11329	0,300915	-1,21619	-0,128055	-1,28333	7
49	-1,04524	-3,32476	1,53542	-0,831538	-1,34376	0,00607664	-1,32557	0,415411	-1,1763	7
50	-0,34705	0,126747	-0,587027	1,73514	-1,11392	0,668416	-0,411544	0,412843	-0,647733	8
51	-0,75772	1,11612	-0,510696	2,59287	-1,2955	-1,85866	-2,55255	0,524935	-1,3433	8
52	-4,70277	-2,43803	-1,22622	0,7591	1,43544	2,0147	-1,35985	0,0666861	-1,08054	8
53	-2,48372	0,168968	-2,71116	1,67287	0,944128	-1,40027	-1,8406	-0,498949	-1,43398	8
54	-2,9929	0,00132952	-1,55708	2,18464	0,746293	0,0805751	0,214438	0,238967	-0,422169	8
55	-3,10034	0,926016	-1,73179	2,12176	0,380485	-0,99761	-1,28998	0,49072	-0,989493	8
56	-3,29686	-3,66508	-0,453838	2,32858	-0,520676	-1,4597	-1,24256	-0,114044	-1,07316	8
57	-0,805609	-0,509364	-1,53939	1,74694	-1,97221	0,296118	-1,70272	0,220399	-1,22222	8
58	0,24414	-4,19064	-1,85419	-3,62545	-2,03389	-0,85803	0,982003	-1,25292	-1,40653	9
59	0,538351	-3,18997	-3,02091	-2,52427	-1,74269	-0,153515	0,684054	-0,413168	-0,652202	9
60	-0,000776301	-2,9095	0,70997	-2,17727	-1,75721	1,30706	1,25025	-3,23233	1,22267	9
61	-0,495942	-3,67004	-0,0341226	-2,52507	-1,05313	1,0676	1,09423	-4,1886	0,50919	9

Tabela 6: Códigos e expressão gênica dos oito últimos genes da base de dados B_3

	Genes e suas expressões gênicas								
Amostra	0525	0531	0637	0721	0786	0870	0890	0929	**Classe**
1	-0,34274	0,226085	0,864763	1,50271	0,71234	1,15217	0,404458	-0,486982	1
2	-0,0751498	0,248734	1,39079	0,266416	-0,0500898	0,0028393	-0,306922	0,143486	1
3	-0,742543	0,419914	1,11697	1,68135	0,516872	1,12689	-0,650673	-0,56773	1
4	-0,849267	0,874263	-0,024048	0,325731	0,298347	0,830463	1,159	0,08838	1
5	0,910028	1,35653	0,56749	0,768198	1,50777	0,488041	0,678679	0,0909462	1
6	0,772914	0,295314	-1,15699	0,309573	1,54846	1,12688	1,29196	0,612791	1
7	-1,16793	-1,26033	0,581828	-0,874761	-1,05424	-0,708219	0,227464	-0,752572	1
8	0,728028	2,97911	-0,951725	-0,350192	0,242187	2,07317	0,787704	0,499288	2
9	0,804104	0,813123	0,194887	-0,403697	-0,876943	1,99649	0,93556	0,218734	2
10	-0,627531	-1,34602	0,384641	-0,589293	0,968493	0,24069	-0,368826	0,393078	2
11	0,478564	2,00909	0,382547	0,023364	-0,651698	0,72531	-0,0260698	0,482588	2
12	0,619733	0,279664	-0,0739046	-0,384601	-1,74438	1,64153	1,32416	0,31967	2
13	0,865112	1,19911	0,251034	0,329994	-0,430974	1,31909	0,0208059	0,345522	2
14	-0,222525	0,524755	0,557731	-0,338098	0,0668613	-0,461642	0,943419	0,811608	3
15	-0,740761	-0,420995	-1,00557	-0,680318	1,29371	-1,13567	0,0822061	-1,01882	3
16	-1,47044	-0,991887	0,0317292	-0,471803	-0,794357	-1,6322	-1,19527	-0,638073	3
17	-0,476278	-0,159554	0,482984	-0,189103	0,469378	-0,190227	-0,489558	-0,269691	3
18	-0,363043	0,251814	0,194685	1,48121	0,0127595	-0,665707	1,3913	0,200639	3
19	0,0909494	0,502212	0,396511	-0,611977	-0,325317	-0,754419	-1,05149	-0,148307	3
20	0,585554	1,25937	0,613426	0,171295	1,21783	0,0911495	-0,488017	0,33498	3
21	-0,994972	-0,0475314	-1,37604	-1,40338	0,0498236	-0,48263	-0,681265	-0,173116	4
22	-0,0403817	0,193005	0,892716	-0,839313	0,975476	-0,314582	-1,97632	0,686159	4
23	-1,07353	-0,271704	-1,58867	-0,079441	0,203455	0,00336786	-1,12196	0,279146	4
24	-0,637169	-0,0595157	-2,34061	0,762345	-1,23619	-0,389625	-0,733183	-0,214183	4
25	0,027684	0,286907	-1,92305	0,272856	-1,49418	-0,817086	-0,541836	0,0673154	4
26	-1,08354	0,140835	-2,8577	0,2231	-0,915117	-0,851546	-1,01261	0,0174157	4
27	0,0826743	0,146953	1,29602	1,73229	1,29134	0,706108	0,740924	0,615231	5
28	-0,395351	-0,122192	1,45603	1,86567	0,10062	0,180805	0,647504	-0,319451	5
29	-0,278504	0,366459	1,29602	2,55948	0,14013	0,97713	0,291277	0,658379	5
30	-0,222466	-0,00892935	0,357202	1,75981	0,371587	2,06302	0,0355741	-1,41093	5
31	0,0269611	0,423546	1,15226	1,8803	0,125448	1,80607	-0,0644408	-0,822778	5
32	-0,215317	0,259296	0,24597	1,62243	-0,429031	0,524412	0,100622	-0,168547	5
33	-0,596783	0,317561	0,625644	1,09604	-0,20713	-0,403128	-0,0922052	0,427898	5
34	-0,746354	0,151035	0,898542	1,91113	0,699588	1,01546	0,442522	-1,20445	5
35	0,244216	2,37495	1,11067	0,298101	0,99684	1,17266	1,2681	1,35781	6

36	-0,621673	0,499473	1,18658	0,196106	0,98067	-0,314285	-1,48846	0,254491	6
37	-0,95228	0,146302	0,799958	0,132907	0,136013	-0,441987	-0,939279	1,26553	6
38	-0,550086	0,300122	-0,0991803	-0,970357	0,900778	-0,500487	-1,12269	-0,597517	6
39	-5,5012	-2,07778	0,777935	0,483952	0,0476712	0,0168069	-0,713809	0,320174	6
40	1,53309	0,484279	0,858501	1,59887	-1,94447	-1,41523	-0,828002	1,02142	6
41	-0,208807	1,3317	0,552731	0,947681	0,926666	0,975643	-0,70923	2,23521	6
42	-0,302731	0,715826	0,980727	0,263438	-0,00848574	0,196704	-1,39634	0,744581	6
43	1,20451	1,46404	1,58602	1,04283	2,18708	0,533722	-0,875466	1,12093	6
44	0,642603	1,3977	0,240454	1,15181	0,849571	-0,225222	-0,76924	2,60042	7
45	-1,12697	0,575814	1,30805	1,48436	-0,174304	-0,10913	-0,614419	0,658237	7
46	0,774201	-0,0145072	-0,191607	-0,10246	0,215837	-0,11828	-0,691352	-0,487457	7
47	1,43953	0,106395	-0,274207	-0,817699	0,208731	0,21093	-0,728166	-0,0340787	7
48	-1,2258	0,561699	0,86006	-0,428225	1,5531	-0,445368	-0,929281	-0,145585	7
49	-0,279961	-0,338307	0,245934	-0,432619	-0,965539	-0,581884	-1,37741	-0,289138	7
50	0,575231	4,96096	1,42497	1,05185	0,951399	0,325106	0,281058	1,01376	8
51	-1,4782	-0,234988	0,385668	0,151716	-0,104245	-0,13042	-1,65096	-1,13767	8
52	-0,319569	1,25405	0,689491	1,47288	-0,465439	0,295017	0,0223647	1,04232	8
53	-0,711824	-0,139605	0,0664101	-0,191405	-0,0523665	0,222589	-1,06272	-0,352899	8
54	0,237857	0,71673	0,640853	1,45947	1,28062	0,195471	-0,651795	1,00524	8
55	-0,490034	0,198918	-0,0345037	0,454319	0,196879	0,297011	-0,564416	0,251085	8
56	0,00971388	-0,139301	0,735241	0,408146	0,134313	-0,00299571	-0,99056	-0,355118	8
57	-1,09205	0,139997	0,935621	0,972988	-0,383086	-1,10921	-1,13112	0,344182	8
58	-0,924537	-0,212501	0,368978	-1,08786	-1,14611	-0,340339	0,5933	-1,51451	9
59	-0,809208	-0,0295062	0,72217	-0,286489	-0,677789	-0,397545	0,899428	0,376444	9
60	-1,56496	-0,147492	-0,671	0,430535	-0,712466	-0,456266	-0,725124	-2,13024	9
61	-1,59469	-0,565252	-1,48907	-0,101767	-1,24792	-0,735205	-1,38796	-0,578589	9

Tabela 7: Códigos e expressão gênica dos genes da base de dados B_4

	Genes e suas expressões gênicas												
Amostra	0011	0046	0177	0289	0306	0336	0380	0499	0661	0783	0865	0950	**Classe**
1	-0,103522	-2,48838	0,176967	-2,07422	-0,924886	0,959412	-1,91527	0,179124	0,150177	-0,479867	0,329944	0,316965	1
2	-2,21418	-1,30646	-0,88801	-2,2083	0,555071	-1,02417	0,252388	-0,74193	0,901572	0,250217	0,666481	0,881566	1
3	-0,82912	-2,64821	-0,31992	-3,26092	-1,38575	0,513009	-3,31315	-0,385338	-0,438888	-0,928142	-0,0524833	0,155949	1
4	-3,51391	-2,36697	1,18908	-0,954262	-0,735781	0,334405	0,117847	0,000975853	-1,46559	0,0432044	-0,734477	-0,0401793	1
5	2,02585	1,73176	1,9747	0,0533926	-1,30594	0,580403	-0,280089	-0,0765836	0,372751	-0,682724	0,632387	0,549099	1
6	0,336447	-0,0140956	1,22492	0,491263	-2,67064	-0,429315	0,165418	-0,748932	0,314684	-0,0834582	-0,122622	0,490055	1
7	-3,80793	-2,01708	-2,57359	-1,30518	-0,139116	-1,51598	0,356728	-2,62518	-1,48154	-0,795396	-1,08543	-0,109984	1
8	1,40794	2,70436	0,408334	-1,62607	-3,10208	0,587364	-2,17883	0,0354808	-0,174667	0,117736	0,088305	0,135017	2
9	0,586283	1,06841	-1,12526	-1,25022	-0,289796	0,685389	0,174856	0,975584	-1,17261	0,746857	0,3353	0,491329	2
10	2,00007	2,00387	-0,381004	-1,9062	-2,98732	0,334627	-0,725257	-0,138112	-1,13135	-1,18742	-0,232572	-0,27841	2
11	0,49241	2,27148	0,745534	-0,615864	-1,68801	0,400118	0,268942	0,651218	-0,230067	0,619998	0,732934	0,920534	2
12	0,471609	0,962593	-1,55739	-2,5171	-1,09525	0,76577	-0,939498	0,570392	-0,431971	0,489305	-0,495295	0,465243	2
13	2,18319	2,60307	0,880008	-0,598794	-1,71649	0,0974071	0,603713	0,391723	-0,55477	0,373359	1,16762	5,38997	2
14	-2,69213	-0,793645	-1,10374	0,484663	1,47899	-0,457014	1,13982	-1,03218	0,124446	0,283804	0,0205586	0,521259	3
15	-2,29205	-1,34381	-0,882882	-0,217442	1,12258	-2,53959	0,446929	-0,348718	-1,61294	-0,199066	-1,23563	-0,526895	3
16	-4,29216	-2,25507	-2,49246	0,157092	-0,50488	-1,65877	-0,872941	-0,915682	-1,2557	-1,09791	-2,01237	-0,969143	3
17	-2,0357	-1,08097	-0,786041	0,406382	0,259164	-1,31715	0,333941	-0,326291	0,848475	0,53774	-0,862668	0,227918	3
18	-3,38821	-1,12264	-0,219609	0,167933	1,22098	-1,66679	0,531316	-0,13458	0,795706	-0,236675	0,368894	0,881061	3
19	-2,84401	-1,15293	-1,03054	0,608268	0,482787	-2,13564	0,0198364	-0,791337	-0,587042	0,424831	-1,34799	0,360368	3
20	-3,68907	-4,19063	0,227054	0,0974808	2,10709	-0,88108	0,496236	0,658042	-1,25611	1,52144	-0,91944	0,540566	3
21	-2,87948	-0,621942	-1,62467	-0,586813	-1,0763	-0,357629	-0,101685	-1,47812	-0,408847	-0,849783	0,294804	0,0560722	4
22	-6,15868	-1,79896	-0,558465	2,39585	-1,29114	0,216313	0,818326	-0,81591	-0,984011	-0,463078	0,462982	0,686806	4
23	-3,76253	-0,938614	-3,25891	0,631448	-2,3601	-0,944374	0,39111	-2,75069	-1,4048	-1,02396	0,195634	0,106811	4
24	-4,14299	-0,538856	-1,77329	1,04189	-0,70371	-0,363397	0,743707	-2,43304	-1,34193	0,35585	0,64798	0,512498	4
25	-3,28705	-1,3707	-1,95025	1,63402	0,229056	0,692486	0,876797	-3,16452	-1,72512	-0,415365	0,194832	0,714595	4
26	-3,08938	-1,42215	-2,92028	-0,831813	-1,44895	-1,37263	0,00283497	-1,59626	-0,376525	0,389742	1,4665	0,228567	4
27	1,17806	-1,13268	-0,212427	1,99245	-2,9224	-0,119626	-2,56629	-0,173191	-0,791282	-1,63645	0,746923	0,720839	5
28	-0,410018	-0,411902	-1,10428	-1,23846	-2,52693	0,270167	-2,97911	-0,572574	-0,317532	-0,103838	-0,0254608	0,564717	5
29	-1,49399	-2,39773	-0,650742	2,54113	-2,57663	0,703971	-3,28649	-1,81007	-0,226765	-0,915591	-0,676565	0,215229	5
30	0,393862	-1,36544	-0,341117	0,292586	-1,18758	-0,0471184	-1,39694	-0,441043	0,055679	-0,125704	-0,456401	0,393523	5
31	0,923374	-0,996664	0,138396	1,25614	-1,42864	-0,166603	-1,48479	0,223922	-0,0398563	-0,0205932	0,368428	1,05965	5
32	0,0855046	-0,739171	-0,269847	0,395048	-1,64137	-0,401288	-2,6505	-0,272054	-1,09416	-1,15519	-0,896413	0,179421	5
33	-1,05559	-0,863154	0,673953	0,717859	-1,31697	-0,346321	-0,756546	0,234297	-0,6459	-0,00258346	0,0837397	0,0836734	5
34	-0,384449	-0,416057	-1,21153	-0,733935	-1,02014	0,456665	-2,25105	-0,219753	-0,799929	-0,408618	-0,3182	0,035387	5
35	0,31306	0,697408	1,72724	0,734916	0,627948	0,74885	-0,344213	0,607226	-0,318526	-0,2367	-0,23254	1,4554	6
36	-2,50761	-0,120982	-0,486662	-0,464893	-0,783659	0,953878	-0,387236	0,286794	-0,312372	1,27622	-0,667667	0,580942	6
37	-2,42524	-0,164986	-0,601221	0,58409	-0,893132	1,59602	-0,404837	0,859523	-0,393439	1,16599	-1,03535	0,592844	6

38	-2,41394	-1,14152	-1,17827	-1,22145	-0,510174	-1,09905	-0,286842	-0,396336	-1,52703	-0,460737	-1,46465	-0,753949	6
39	-2,67337	-4,85106	-0,434891	0,570793	-0,0914322	1,29077	0,573653	0,333106	-1,49628	-0,843176	-1,21034	0,14602	6
40	-3,24614	-1,07968	-0,831468	-1,99476	0,531332	-0,249754	-0,799438	-0,720087	0,230159	-0,235537	-0,851123	-0,0977335	6
41	-0,199181	2,27561	1,02125	0,635635	-0,255267	-1,84484	-0,195853	0,941268	-0,238748	0,222077	-0,206601	-0,153513	6
42	1,87582	-1,31443	0,359139	-0,440944	-0,717543	0,399384	-0,435952	1,37872	-0,174989	-1,28754	-1,5885	-0,0615455	6
43	-2,33444	0,716273	0,744879	1,4152	-0,424991	0,857202	0,107447	0,5692	0,65671	1,23678	0,00929223	1,11083	6
44	0,262613	1,95457	0,959328	0,786736	0,619025	1,06336	0,287557	0,263784	1,52805	0,690394	0,152091	1,5154	7
45	0,530779	2,10995	2,22554	1,57048	0,0682614	0,360342	-0,312103	0,894085	1,01768	0,664355	-0,397782	0,667135	7
46	-2,10392	-0,344371	0,483867	-0,0665868	0,621992	-0,515484	0,709341	-0,176819	0,643673	0,0696317	-0,776344	-0,330831	7
47	-4,05749	0,57678	-0,264026	-0,990445	0,360804	0,172254	1,17078	0,0344003	1,99257	0,46818	-0,604145	0,966317	7
48	-3,16229	0,70144	0,664467	0,588803	0,00510869	-0,683513	-0,113256	-0,0573801	-0,837996	-2,3793	-1,57285	-0,0856573	7
49	-3,39343	-0,656646	-0,137477	-0,0763907	-0,411465	-0,744654	0,292118	-0,503265	-0,594271	0,260469	-1,47753	-0,0416137	7
50	1,19139	2,48906	1,78815	0,881833	-0,10148	1,57285	0,945307	1,26264	0,248347	2,04276	-1,09925	0,761662	8
51	-2,3231	1,2179	0,326187	-0,758778	-1,10194	2,4363	0,0798933	-0,0376131	1,22505	0,866978	-1,37084	-0,357101	8
52	1,23518	3,27244	1,34967	0,645642	-3,2129	0,759319	-0,023826	1,01812	-0,118074	-0,347821	-0,916156	0,173905	8
53	-0,154612	0,809561	0,639092	0,699735	-2,68745	1,28848	-0,494016	-0,152585	-0,713812	1,85276	-1,18189	-0,284966	8
54	1,44623	1,26268	1,45025	1,16946	-0,199099	1,7401	0,342975	0,733709	1,27224	1,71161	-0,49677	0,126953	8
55	-0,269493	1,69096	1,23719	1,04013	-1,80672	1,66425	0,0972347	-0,143604	0,749387	0,766306	-0,866869	0,380299	8
56	0,238757	1,09746	0,271356	0,542677	-0,804326	2,06999	0,230407	-1,41486	-1,09825	1,28228	-1,08495	-0,263232	8
57	-0,440923	1,4572	0,295587	-0,789598	-1,04364	1,35565	0,402032	-0,184181	0,391813	1,28835	-2,27481	-0,345308	8
58	-3,54653	-1,78467	-2,61657	-1,03034	0,110498	-2,43066	1,24889	-2,46838	-3,48737	-0,495359	-1,201	-0,384081	9
59	-3,50223	-1,25066	-2,26539	-1,16499	0,378708	-1,81102	1,06597	-1,86002	-0,858616	-0,278594	-1,05076	0,112206	9
60	-2,24822	-1,10386	-2,57193	-0,387918	-0,634427	-1,1645	0,580011	-1,14517	0,119495	0,795191	1,79798	0,306238	9
61	-3,54528	-1,23443	-3,07856	-0,953137	-1,52316	-1,38622	-0,0208435	-1,65968	-0,663206	0,530379	0,955755	-0,395708	9

APÊNDICE C

Algorimos Genéticos Multi-Objetivos

Devido a sua forma de trabalhar com uma gama de soluções a cada geração, os AGs são capazes de encontrar várias soluções não dominadas ao longo do processo de otimização. Essa propriedade aliada a sua adaptabilidade a diferentes tipos de problemas, tornam os AGs importantes ferramentas de otimização multiobjetivo.

Muitos problemas do mundo real envolvem uma otimização simultânea de múltiplos objetivos [48], isto é, existem vários critérios que devem ser balanceados. Na otimização de um único objetivo, tenta-se obter o melhor resultado, ou a melhor decisão, o que usualmente é o mínimo ou o máximo global. No caso de múltiplos objetivos, pode não haver uma melhor solução (ótimo global) com respeito a todos os objetivos. Em um problema de otimização multi-objetivos, existe um conjunto de soluções que são superiores às demais dentro de um espaço de busca onde todas as possíveis soluções são consideradas [99]. Esse conjunto de soluções é conhecido como o Ótimo de Pareto ou soluções não dominadas [102].

Ótimo de Pareto

O Ótimo de Pareto foi formulado pelo sociólogo e economista Vilfredo Pareto (1848 - 1923) e tornou-se o princípio de otimização quando há a competição de múltiplos objetivos.

A solução ótima de Pareto não é única, mas sim, um conjunto de pontos os quais são considerados igualmente bons em função do vetor objetivo. Esse espaço pode ser visto como um espaço de busca de soluções, no qual cada objetivo poderia ser aperfeiçoado, mas seria melhorado às custas de pelo menos outro objetivo [100].

Não dominância versus dominância

A busca pelo ótimo de Pareto tem sido conhecida como otimização simultânea de múltiplos objetivos. Uma conceituação alternativa seria pensar que uma solução é a ótima de Pareto se, para um dado conjunto de objetivos, não exista nenhuma outra solução que seja superior a ela, considerando-se todos os objetivos [102]. Para elucidar este conceito será utilizado como exemplo a compra de um automóvel, onde várias decisões precisam

ser tomadas, priorizando custo ou conforto, fatores estes conflitantes. A figura 1 ilustra várias opções de escolha [2].

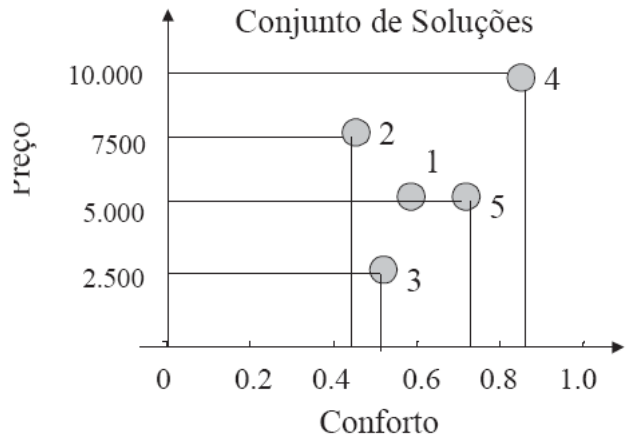

Figura 1: Exemplo que ilustra várias opções de compra de automóvel (1-5), considerando o seu custo e conforto [2]

Neste exemplo, o objetivo é minimizar o custo e maximizar o conforto. Neste caso, existem cinco opções de compra. Intuitivamente, descarta-se a solução 1, já que a solução 5 oferece mais conforto pelo mesmo custo. A solução 2 também é descartada pela mesma razão. Restam então três boas alternativas de compras: 3, 4 e 5. Em termos quantitativos nenhuma solução é melhor que a outra, pois o acréscimo no nível de conforto do automóvel traz consigo um aumento no custo do mesmo. Em raciocínio análogo, ao diminuir o custo do automóvel, diminui-se também o nível de conforto do mesmo [2].

Em outras palavras, a solução ótima de Pareto seria aquela para a qual não exista outra no espaço de busca que a domine. Similarmente, uma solução não é considerada uma solução ótima de Pareto se ela for categoricamente dominada por, pelo menos, uma solução do conjunto de candidatas. Quando consideramos soluções dominadas, devemos pensar que diferentes níveis de dominância são possíveis. Uma solução dominada será sempre categoricamente inferior às soluções não dominadas do conjunto de soluções do ótimo de Pareto. Entretanto, uma solução dominada pode também dominar outra solução. Por exemplo, a solução 1 é dominada pela solução 5 e domina a solução 2. Esses níveis de dominância permitem caracterizar totalmente o grupo de soluções, separando-as dentro de fronteiras de não dominância, que inclui a fronteira correspondente aos ótimos de Pareto.

Esse aspecto do paradigma é muito importante no projeto de um Algoritmos Genético para Problemas Multi-objetivos (AGMO). Usando o conceito de não doninância e dominância, a análise de Pareto pode ser simplificada como a busca da não dominância. A busca consiste em classificar as soluções candidatas em grupos de soluções não dominadas, pois são elas as favoritas. Dessa forma, a otimização de Pareto pode ser vista como uma otimização clássica, onde a dominância global é o atributo desejado [100]. Se os pontos não dominados estão em um espaço contínuo, pode-se desenhar uma curva. Todos os pontos contidos na curva formam a **Frente de Pareto** ou **Fronteira de Pareto** [2].

O AG requer uma informação de avaliação escalar para poder trabalhar. Isso significa dizer que, para a solução de problemas envolvendo múltiplos critérios, necessitamos escalonar um vetor de objetivos. Um dos problemas, é que nem sempre é possível derivar um critério global baseado na formulação do problema. Na ausência de informação, os objetivos tendem a ter uma importância equivalente. Por outro lado, quando temos uma certa compreensão do problema, podemos combiná-los de acordo com a informação existente, provavelmente atribuindo maior importância a alguns objetivos. Otimizar uma combinação de objetivos tem a vantagem de produzir apenas uma solução simples, não exigindo uma iteração posterior para a tomada de decisão [101].

A utilização de AGs como método de otimização permite que uma abordagem efetivamente multi-objetivos, levando-se em consideração os conceitos de dominância e ótimo de Pareto, seja utilizada sem a necessidade de se combinar os objetivos através de pesos de importância relativa. Nos últimos anos muitos pesquisadores têm modificado as idéias iniciais propostas por Goldberg em seu livro [50] para tratamento de problemas multi-objetivos. Podemos citar alguns desses principais métodos:

- VEGA (*Vector Evaluated Genetic Algorithms*) [105]
- Agregação dos objetivos por pesos variáveis [106]
- MOGA (*Multi-objective Optimization Genetic Algorithm*) [101]
- NPGA (*Niched Pareto Genetic Algorithm*) [107]
- NSGA (*Nondominated Sorting Genetic Algorithm*) [99]

- SPEA (*Strength Pareto Evolutionary Algorithm*) [108]
- PAES (*Pareto Archieved Evolution Strategy*) [110]
- NSGA-II (*Nondominated Sorting Genetic Algorithm II*) [111]
- PESA (*Pareto Enveloped-based Selection Algorithm*) [112]
- SPEA2 (*Strength Pareto Evolutionary Algorithm 2*) [113]
- PMOGA (*Pareto Multiobjective Genetic Algorithm*) [48]

APÊNDICE D

Tabela 8: Aptidão de treinamento e aptidão de teste das melhores regras evoluídas na base B_2

	Experimento 12->3			**Experimento 13->2**			**Experimento 23->1**	
Classes	$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	0,944	0,472		0,921	0,406		1	0
2	1	0		1	0,5		1	0,952
3	1	0,5		1	0,875		1	0,633
4	1	0		1	1		1	1
5	1	0,941		1	1		1	0,283
6	1	0,431		0,892	0,489		0,969	0,283
7	1	0,5		1	0,938		0,971	1
8	1	0,667		1	1		1	0,85
9	1	0,947		1	0,941		1	0,952

Tabela 9: Aptidão de treinamento e aptidão de teste das melhores regras evoluídas na base B_3

	Experimento 12->3			**Experimento 13->2**			**Experimento 23->1**	
Classes	$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	1	0		0,947	0		1	0,3
2	1	1		1	1		1	1
3	1	0,472		1	0,938		1	0,633
4	1	0		1	1		1	0,952
5	1	1		1	1		1	0,333
6	1	0,667		0,973	0,622		1	0,317
7	1	0,472		1	0		1	0,452
8	1	0,667		1	0,875		1	0,6
9	1	0		1	1		1	0,905

Tabela 10: Aptidão de treinamento e aptidão de teste das melhores regras evoluídas na base B_4

	Experimento 12->3			**Experimento 13->1**			**Experimento 23->1**	
Classes	$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	0,917	0,444		0,812	0		0,971	0,3
2	1	1		1	1		1	1
3	1	0,778		1	1		1	0,333
4	1	0,444		1	0,438		1	0,952
5	1	1		1	1		1	0,95
6	1	0,314		0,973	0,267		0,937	0
7	1	0,5		1	0,469		1	0,476
8	1	0,667		1	0,938		1	0,667
9	0,974	1		1	0		1	0,952

Tabela 11: Aptidão de treinamento e aptidão de teste das melhores regras evoluídas na base B_1B_2

	Experimento 12->3			**Experimento 13->2**			**Experimento 23->1**	
Classes	$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	0,861	0,417		1	0		1	0,333
2	1	1		1	1		1	1
3	1	0,5		1	0,438		1	0,633
4	1	0		1	1		1	0,5
5	1	1		1	1		1	0,283
6	1	0,667		0,973	0,667		1	0,333
7	1	0,5		1	0,5		1	0,5
8	1	0,667		1	1		1	0,95
9	1	0,947		1	0		1	0,952

Tabela 12: Aptidão de treinamento e aptidão de teste das melhores regras evoluídas na base B_1B_3

	Experimento 12->3			**Experimento 13->2**			**Experimento 23->1**	
Classes	$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	1	0		0,921	0		1	0,533
2	1	1		1	1		1	1
3	1	0,889		1	0,938		1	0,633
4	1	0		1	1		1	0,952
5	1	0,549		1	1		1	0,333
6	1	0,627		1	0,622		1	0,317
7	1	0,5		1	0,5		1	0,952
8	1	0,667		1	1		1	0,567
9	1	0		1	0		1	0,952

Tabela 13: Aptidão de treinamento e aptidão de teste das melhores regras evoluídas na base B_1B_4

	Experimento 12->3			**Experimento 13->2**			**Experimento 23->1**	
Classes	$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	0,944	0		0,921	0,375		1	0,317
2	1	1		1	1		1	1
3	1	0,5		1	1		1	1
4	1	0,444		1	0,438		1	0,5
5	1	1		1	1		1	0,95
6	1	0,275		1	0,222		1	0,333
7	1	0,5		1	0,938		1	0,476
8	1	0,667		1	0,875		1	0,95
9	1	1		1	0		1	0,905

Tabela 14: Aptidão de treinamento e aptidão de teste das melhores regras evoluídas na base B_2B_3

	Experimento 12->3			**Experimento 13->1**			**Experimento 23->1**	
Classes	$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	1	0		0,947	0,375		1	0,317
2	1	1		1	1		1	1
3	1	0,5		1	0,875		1	0,633
4	1	0		1	1		1	0,952
5	1	0,549		1	1		1	0,333
6	1	0,275		0,973	0,933		1	0,333
7	1	0,944		1	0		1	0,452
8	1	0,667		1	0,875		1	0,95
9	1	0		1	1		1	1

Tabela 15: Aptidão de treinamento e aptidão de teste das melhores regras evoluídas na base B_2B_4

	Experimento 12->3			**Experimento 13->2**			**Experimento 23->1**	
Classes	$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	0,972	0		0,974	0		1	0,283
2	1	1		1	1		1	1
3	1	0,5		1	0,875		1	0,633
4	1	0		1	1		1	1
5	1	1		1	1		1	0,95
6	1	0,275		1	0,289		1	0,333
7	1	0,944		1	0,938		1	0,476
8	1	0,667		1	0,875		1	0,85
9	1	1		1	1		1	0,952

Tabela 16: Aptidão de treinamento e aptidão de teste das melhores regras evoluídas na base B_3B_4

	Experimento 12->3			**Experimento 13->2**			**Experimento 23->1**	
Classes	$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	1	0		0,947	0,344		1	0,3
2	1	1		1	1		1	1
3	1	0,5		1	1		1	0,633
4	1	0		1	1		1	0,952
5	1	0,882		1	1		1	0,333
6	1	0,588		1	0,578		1	0,633
7	1	0,944		1	0,938		1	0,476
8	1	0,667		1	0,938		1	0,667
9	1	0		1	0		1	0,952

Tabela 17: Aptidão de treinamento e aptidão de teste das melhores regras evoluídas na base $B_1B_2B_3$

	Experimento 12->3			**Experimento 13->2**			**Experimento 23->1**	
Classes	$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	0,972	0		0,974	0		1	0,3
2	1	1		1	1		1	1
3	1	0,5		1	0,938		1	0,633
4	1	0		1	1		1	0,952
5	1	0,549		1	1		1	0,333
6	1	0,314		1	0,622		1	0,333
7	1	0,944		1	0,5		1	0,952
8	1	0,667		1	1		1	0,85
9	1	0		1	0		1	0,952

Tabela 18: Aptidão de treinamento e aptidão de teste das melhores regras evoluídas na base $B_1B_2B_4$

	Experimento 12->3			**Experimento 13->1**			**Experimento 23->1**	
Classes	$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	0,972	0,389		0,947	0,344		1	0,317
2	1	1		1	1		1	1
3	1	0,5		1	0,875		1	0,633
4	1	0		1	1		1	0,5
5	1	1		1	1		1	0,95
6	1	0,51		1	0,533		1	0,333
7	1	0,833		1	0,938		1	0,476
8	1	0,667		1	0,938		1	0,85
9	1	0		1	0		1	0,952

Tabela 19: Aptidão de treinamento e aptidão de teste das melhores regras evoluídas na base $B_1B_3B_4$

	Experimento 12->3			**Experimento 13->1**			**Experimento 23->1**	
Classes	$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	1	0		0,947	0		1	0,3
2	1	1		1	1		1	1
3	1	0,889		1	0,938		1	0,633
4	1	0		1	1		1	0,952
5	1	0,882		1	1		1	0,333
6	1	0,275		1	0,289		1	0,333
7	1	0,5		1	0,938		1	0,952
8	1	0,667		1	0,938		1	0,333
9	1	0		1	0		1	0,952

Tabela 20: Aptidão de treinamento e aptidão de teste das melhores regras evoluídas na base $B_2B_3B_4$

	Experimento 12->3			**Experimento 13->2**			**Experimento 23->1**	
Classes	$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	0,972	0,444		1	0		1	0,267
2	1	1		1	1		1	1
3	1	0,5		1	0,875		1	0,633
4	1	0		1	1		1	0,952
5	1	0,882		1	1		1	0,333
6	1	0,588		1	0,578		1	0,3
7	1	0,5		1	0,938		1	0,476
8	1	0,667		1	0,938		1	0,85
9	1	0		1	1		1	0,952

Tabela 21: Aptidão de treinamento e aptidão de teste das melhores regras evoluídas na base $B_1B_2B_3B_4$

	Experimento 12->3			**Experimento 13->2**			**Experimento 23->1**	
Classes	$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$		$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	1	0		0,974	0		1	0,5
2	1	1		1	1		1	1
3	1	0,472		1	0,938		1	0,633
4	1	0		1	1		1	0,952
5	1	0,667		1	1		1	0,333
6	1	0,549		1	0,289		1	0,333
7	1	0,389		1	0,938		1	0,952
8	1	0,667		1	0,5		1	0,667
9	1	1		1	0		1	1

APÊNDICE E

Tabela 22: Melhores regras encontradas na base de dados B_1B_2

Classe	Regras	Apt_{Trein}	Apt_{Teste}
1	if(28<0,8) and (75≥0,2) and (280≥-0,3) and (499<0,1) and (843≥0)	1	0,333

2	if(11≥0,4) and (289<-0,5)
	if(11≥0,4) and (637<0,4)
	if(839≥0,5) and (637<0,4)
		1	1

3	if(2≥1) if(19≥1)	1	0,633

4	if(19<-0,4) and (526<-0,8)
	if(19<-0,4) and (843<-1)
	if(50≥-2) and (843<-0,9)
	if(224<-2,2) and (843<-0,8)
	if(366≥-0,9) and (526<-0,9)
	if(366≥-0,9) and (843<-1)
	if(409≥-1,8) and (843<-1)
	if(839<-0,4) and (224<-2,2)
		1	1

5	if(11≥-1,5) and (97<0,2) and (18<0,2)
	if(11≥-1,6) and (97<0,1) and (292<0,6)
	if(11≥-1,5) and (97<0,3) and (302<0,1)
		1	1

6	if(242<0,3) and (881<1) and (17≥-1,6) and (637≥0,4)	1	0,667

7	if(2≥-1,4) and (28≥-0,2) and (409<0,6)
	if(2≥-1,3) and (224≥-0,4) and (409<0,6)
	if(19≥-1,3) and (224≥-0,2) and (409<0,5)
	if(50≥0,4) and (2≥-2,4) and (17<0,5)
	if(50≥0,2) and (2≥-2,2) and (224<1,7)
	if(50≥0,1) and (2≥-2,4) and (409<1)
	if(50≥0,1) and (2≥-2,4) and (499<1,2)
	if(50≥0,2) and (2≥-2,1) and (637<1,4)
	if(50≥0,4) and (17<0,1) and (19≥-1,7)
	if(50≥0,1) and (19≥-1,9) and (235<1,4)
	if(50≥0,2) and (97<1,4) and (19≥-1,7)
	if(50≥0,2) and (828<0,9) and (2≥-2,2)
	if(50≥0,2) and (828<1) and (19≥-1,7)
	if(50≥0,4) and (839<0,1) and (2≥-2,1)
	if(50≥0,4) and (839<0,2) and (19≥-1,7)
	if(50≥0,4) and (881<1,5) and (19≥-1,7)
	if(97<1,5) and (19≥-1,1) and (224≥-0,3)
	if(97<1,6) and (194≥0,2) and (839<-0,2)
	if(194≥0,2) and (242≥-0,1) and (839<-0,2)
	if(194≥0,2) and (839<-0,2) and (843≥-0,1)
		1	0,5

8	if(97≥0,5) and (127≥0,3) and (863<0,9)	1	1

	if(97≥0,7) and (348<0,1) and (863<0,8) if(97≥0,7) and (863<0,8) and (881≥0,1)		

9	if(18<-2,1) and (292≥-1,5)
	if(19≥-0,3) and (231<-1,4)
	if(289<-0,3) and (18<-3)
	if(366<-0,5) and (18<-3)
		1	0,952

Tabela 25: Melhores regras encontradas na base de dados B_2B_3

Classes	**Regras**	Apt_{Trein}	Apt_{Teste}
1	if(17<-0,3) and (224<0,4) and (843≥-0,2) and (531<0,9) and (890≥-1)	0,947	0,375

2	if(141<1,3) and (229≥1,1)
	if(229≥1,1) and (456≥-0,9)
	if(235<1) and (229≥1,1)
		1	1

3	if(75≥-0,7) and (246<-0,3)	1	0,875

4	if(2<-0,2) and (485≥0,7)
	if(19<-0,4) and (485≥0,7)
	if(19<-0,4) and (526<-0,7)
	if(19<-0,4) and (843<-1)
	if(63<-0,3) and (485≥0,7)
	if(224<-2,2) and (485≥0,7)
	if(229≥-1,7) and (485≥0,6)
	if(235≥-2,9) and (485≥0,6)
	if(409≥-1,7) and (485≥0,7)
	if(485≥0,5) and (525≥-1,3)
	if(485≥0,7) and (929≥-0,4)
	if(526<-0,9) and (929≥-0,5)
	if(843<-1) and (525≥-1,4)
	if(843<-0,8) and (929≥-0,3)
		1	1

5	if(2<-1,8) and (17≥-0,9) and (229<-0,8)
	if(17≥-0,9) and (19<-2,6) and (229<-0,8)
	if(19<-2,6) and (229<-0,7) and (890≥-0,1)
	if(28≥-0,8) and (97<0,2) and (721≥1)
	if(97<0,2) and (246≥-0,4) and (721≥1)
	if(97<0,2) and (379<0,1) and (721≥1)
	if(97<0,2) and (475≥-0,6) and (721≥1)
	if(97<0,2) and (721≥1) and (870≥-0,6)
	if(292<0,5) and (41≥-2) and (721≥1)
		1	1

6	if(2<1) and (17<0,4) and (637≥-0,1) and (379≥-0,1) and (456≥-1,2)	0,973	0,933
7	if(63≥0,1) and (379<0,8)	1	0,944
8	if(75≥-0,1) and (409≥0,4) and (870<0,5) if(97≥0,6) and (280≥0,1) and (409≥0,4)	1	0,95

	if(97≥0,6) and (409≥0,4) and (870<0,4)		
	if(280≥0,1) and (409≥0,4) and (870<0,4)		
9	if(17<-3,6) and (75<-1,2) and (292≥-1,5)	1	1
	if(17<-3,6) and (231<-0,5) and (292≥-1,5)		
	if(17<-3,4) and (292≥-1,4) and (499<-1,1)		
	if(17<-3,6) and (292≥-1,4) and (525<-0,7)		
	if(18<-2,5) and (231<-0,3) and (292≥-1,4)		
	if(18<-3,2) and (292≥-1,4) and (499<-1,1)		
	if(19≥-0,2) and (231<-1,3) and (409<-0,5)		
	if(19≥-0,2) and (231<-0,9) and (485≥-1,6)		
	if(19≥-0,3) and (231<-0,9) and (499≥-2,6)		
	if(19≥-0,2) and (231<-1,4) and (531≥-0,6)		
	if(19≥-0,2) and (292≥-1,4) and (499<-1,1)		
	if(19≥-0,2) and (499<-1,1) and (531≥-1,1)		
	if(19≥-0,4) and (531≥-0,6) and (786<-0,6)		
	if(75<-1,2) and (141<-0,5) and (292≥-1,5)		
	if(75<-1,2) and (231<-1) and (292≥-1,5)		
	if(75<-1) and (292≥-1,5) and (456≥0,6)		
	if(75<-1,2) and (292≥-1,5) and (475<-0,1)		
	if(75<-1,3) and (292≥-1,4) and (499<-1,1)		
	if(75<-1,2) and (292≥-1,5) and (525<-0,7)		
	if(75<-1,2) and (292≥-1,5) and (786<-0,6)		
	if(75<-1,1) and (499≥-2,6) and (456≥0,6)		
	if(231<-0,5) and (292≥-1,5) and (302<-0,8)		
	if(231<-0,2) and (292≥-1,5) and (786<-0,6)		
	if(292≥-1,5) and (302<-0,8) and (525<-0,7)		
	if(409<-0,6) and (456≥0,6) and (786<-0,6)		

Tabela 27: Melhores regras encontradas na base de dados $B_1B_2B_3$

Classes	**Regras**	Apt_{Trein}	Apt_{Teste}
1	if(246≥-0,4) and (531≥0,2) and (929<0,2)	1	0,3
2	if(11≥0,4) and (289<-0,5)	1	1
	if(141<1,4) and (229≥1,1)		
	if(289<-0,5) and (229≥1,1)		
	if(839≥0,5) and (637<0,4)		
3	if(50<-2,3) and (63≥-0,4)	1	0,938
4	if(2<-0,1) and (485≥0,7)	1	1
	if(19<-0,3) and (485≥0,7)		
	if(19<-0,4) and (526<-0,9)		
	if(19<-0,2) and (843<-1)		
	if(63<-0,3) and (485≥0,7)		
	if(224<-2,2) and (485≥0,1)		

	if(235≥-3) and (485≥0,3)
	if(235≥-3,1) and (843<-1)
	if(366≥-0,9) and (485≥0,7)
	if(366≥-0,9) and (526<-0,9)
	if(409≥-1,8) and (485≥0,3)
	if(475≥-3,1) and (485≥0,1)
	if(485≥0,4) and (525≥-1,1)
	if(485≥0,6) and (929≥-0,3)
	if(526<-0,7) and (63<-0,3)
	if(843<-1) and (63<-0,2)
	if(843<-1) and (525≥-1,1)
	if(843<-0,6) and (929≥-0,3)		

5	if(2<-1) and (17≥-0,8) and (229<-0,7)
	if(11≥-1,8) and (97<0,2) and (18<0,1)
	if(11≥-1,5) and (97<0,2) and (292<0,7)
	if(11≥-1,7) and (97<0,1) and (302<0,2)
	if(17≥-0,8) and (19<-0,8) and (229<-0,7)
	if(97<0,5) and (28≥-0,6) and (721≥1)
	if(97<0,1) and (41≥-2,5) and (721≥1)
	if(97<0,1) and (194≥-1,3) and (721≥1)
	if(97<0,1) and (246≥-0,4) and (721≥1)
	if(97<0,1) and (379<0,1) and (721≥1)
	if(97<0,1) and (721≥1) and (870≥-0,5)
	if(242≥-0,1) and (302<0,1) and (41≥-2)
	if(302<0,1) and (41≥-2) and (721≥1)
	if(348<-1,2) and (17≥-0,8) and (229<-0,7)
	if(348<-1,5) and (41≥-2) and (721≥0,8)
	if(881<0,9) and (41≥-2) and (721≥1)
		1	1

6	if(828<0,3) and (2<0,9) and (637≥-0,4) and (379≥0,2) if(828<0,3) and (19<1) and (637≥-0,4) and (379≥0,2)	1	0,622
7	if(242≥-0,1) and (63≥0,3)	1	0,952
8	if(97≥0,7) and (348<-0,2) and (863<0,7)	1	1

9	if(18<-3,2) and (19≥-2,2)
	if(18<-3,2) and (485<1,7)
	if(19≥-0,2) and (231<-1,3)
	if(19≥-0,2) and (525<-0,8)
	if(19≥-0,2) and (786<-0,5)
	if(50<-1,7) and (18<-3,2)
	if(289<-0,2) and (18<-3,2)
		1	0,952

Tabela 30: Melhores regras encontradas na base de dados $B_2B_3B_4$

Classes	**Regras**	Apt_{Trein}	Apt_{Teste}
1	if(409≥-0,4) and (929≥-0,8) and (289<0,5) and (783<0,1)	0,972	0,444

2	if(11≥0,3) and (289<-0,5)
	if(141<1,4) and (229≥1,1)
	if(229≥1,1) and (177<1)
	if(229≥1,1) and (289<-0,5)
	if(229≥1,1) and (456≥-0,9)
	if(235<1) and (229≥1)
		1	1

3	if(75≥-0,8) and (246<-0,4)	1	0,875

4	if(2<-0,2) and (485≥0,7)
	if(19<-0,4) and (485≥0,7)
	if(19<-0,4) and (843<-1)
	if(63<0,2) and (485≥0,7)
	if(224<-2,2) and (380≥-0,3)
	if(224<-2,2) and (485≥0,7)
	if(224<-2,2) and (865≥0,1)
	if(235≥-3,7) and (485≥0,7)
	if(409≥-1,7) and (485≥0,6)
	if(475≥-2,7) and (485≥0,4)
	if(485≥0,5) and (525≥-1,1)
	if(485≥0,7) and (661<-0,1)
	if(485≥0,7) and (783<0,4)
	if(485≥0,7) and (929≥-0,3)
	if(526<-0,6) and (63<-0,3)
	if(843<-0,7) and (63<-0,3)
	if(843<-0,9) and (929≥-0,3)
		1	1

5	if(2<-0,8) and (17≥-0,8) and (229<-0,8)
	if(2<-1,3) and (289≥-1,4) and (380<-0,7)
	if(11≥-2) and (289≥-1,4) and (380<-0,7)
	if(17≥-0,8) and (19<-2,5) and (229<-0,8)
	if(17≥-0,9) and (229<-0,7) and (380<-0,6)
	if(17≥-0,9) and (289≥-1,4) and (380<-0,7)
	if(18<0,1) and (41≥-2,2) and (721≥0,5)
	if(18<0,2) and (97<0,1) and (11≥-1,6)
	if(19<-2,9) and (41≥-2,2) and (721≥0,8)
	if(19<-0,9) and (289≥-1,4) and (380<-0,7)
	if(28≥-1) and (97<0,6) and (721≥1)
	if(41≥-2) and (46<-0,3) and (380<-0,4)
	if(41≥-2,2) and (721≥0,5) and (46<-0,1)
	if(97<0,1) and (41≥-2) and (380<-0,7)
	if(97<0,1) and (246≥-0,4) and (380<-0,7)
	if(97<0,1) and (292<1) and (11≥-1,5)
	if(97<0,1) and (379<0,1) and (380<-0,7)
	if(97<0,2) and (379<0,1) and (721≥1)
	if(97<0,1) and (637≥-0,3) and (306<-0,9)
		1	1

	if(97<0,2) and (721≥1) and (306<-1)
	if(224≥-0,6) and (289≥-1,4) and (380<-0,7)
	if(229<-0,8) and (890≥-0,3) and (306<-1)
	if(229<-0,8) and (890≥-0,2) and (380<-0,6)
	if(289≥-1,4) and (306<-0,8) and (380<-0,7)
	if(289≥-1,6) and (336≥-1,5) and (380<-0,7)
	if(289≥-1,4) and (380<-0,7) and (865≥-1,4)
	if(289≥-1,4) and (380<-0,7) and (950≥-0,6)
	if(302<0,1) and (41≥-2) and (11≥-1,8)
	if(302<0,1) and (41≥-2) and (380<-0,4)
	if(485≥-1,9) and (289≥-1,4) and (380<-0,7)
	if(525≥-0,9) and (289≥-1,4) and (380<-0,7)
	if(531≥-0,4) and (289≥-1,4) and (380<-0,7)
	if(637≥0,1) and (289≥-1,6) and (380<-0,7)
	if(721≥0,8) and (289≥-1,6) and (380<-0,7)
	if(870≥-1) and (289≥-1,6) and (380<-0,7)
	if(890≥-0,3) and (46<-0,3) and (306<-1)
	if(890≥-0,2) and (289≥-1,3) and (380<-0,7)		

6	if(2<-0,2) and (637≥0,1) and (379≥0,1) and (306≥-1)	1	0,588
7	if(63≥0,3) and (46≥-0,7)	1	0,938
8	if(46≥0,8) and (865<-0,4)	1	0,938
9	if(18<-3,1) and (19≥-0,3)	1	1

Tabela 31: Melhores regras encontradas na base de dados $B_1B_2B_3B_4$

Classe	**Regras**	Apt_{Trein}	Apt_{Teste}
1	if(289<0,5) and (75≥0,1) and (721≥0,2)	1	0,5

2	if(11≥0,4) and (289<-0,5)
	if(289<-0,5) and (229≥1,1)
	if(839≥0,5) and (637<0,5)
		1	1

3	if(50<-2,3) and (63≥-0,5)	1	0,938

4	if(19<-0,3) and (485≥0,6)
	if(19<-0,3) and (526<-0,9)
	if(50≥-2) and (280<-0,7)
	if(50≥-2) and (485≥0,2)
	if(63<-0,3) and (485≥0,6)
	if(194<-0,8) and (485≥0,7)
	if(224<-1,7) and (950≥0)
	if(366≥-0,9) and (485≥0,7)
	if(409≥-1,8) and (485≥0,3)
	if(485≥0,5) and (525≥-1,1)
	if(485≥0,3) and (783<0,4)
	if(485≥0,7) and (929≥-0,3)
		1	1

	if(526<-0,7) and (63<-0,3)
	if(526<-0,9) and (929≥-0,3)
	if(843<-0,9) and (63<-0,3)
	if(843<-1) and (783<0,4)
	if(843<-0,8) and (929≥-0,4)		

5	if(17≥-0,8) and (19<-2,1) and (229<-0,7)
	if(17≥-0,8) and (229<-0,8) and (306<-1)
	if(17≥-0,9) and (229<-0,8) and (380<-0,3)
	if(41≥-2) and (721≥0,6) and (46<0,8)
	if(41≥-2,3) and (721≥0,9) and (783<0,1)
	if(97<0,1) and (306<-0,8) and (380<-0,3)
	if(97<0,1) and (379<0,1) and (380<-0,4)
	if(97<0,1) and (379<0,2) and (721≥0,3)
	if(97<0,5) and (721≥0,7) and (306<-0,8)
	if(229<-0,7) and (890≥-0,1) and (380<-0,5)
	if(242≥0,1) and (41≥-2,4) and (46<-0,3)
	if(242≥0,1) and (302<0,2) and (41≥-2,4)
	if(289≥-1,4) and (2<-1,1) and (380<-0,6)
	if(289≥-1,4) and (224≥-1,4) and (380<-0,6)
	if(289≥-1,4) and (637≥0,2) and (380<-0,6)
	if(289≥-1,4) and (721≥0,9) and (380<-0,6)
	if(890≥-0,1) and (46<-0,3) and (306<-1)
		1	1

6	if(242<0,3) and (637≥0,5) and (661<0,3)	1	0,549
7	if(242≥-0,1) and (63≥0,3)	1	0,952

8	if(46≥1) and (865<-0,4)
	if(97≥1,5) and (177≥0,2)
	if(97≥1,6) and (280≥-0,3)
	if(97≥1,6) and (348<-0,1)
	if(97≥1,2) and (409≥0,6)
	if(97≥1,6) and (881≥0,2)
		1	0,667

9	if(18<-3) and (46<-0,5)
	if(18<-3,1) and (292≥-1,6)
	if(18<-3,2) and (637≥-1,5)
		1	1

Tabela 23: Melhores regras encontradas na base de dados B_1B_3

Classes	**Regras**	Apt_{Trein}	Apt_{Teste}
1	if(289<0,5) and (531≥0,2) and (721≥0,1) and (870≥-0,2)	1	0,533
	if(289<0,5) and (839<1,9) and (531≥0,2) and (721≥0,2)		
	if(289<0,5) and (863<1) and (531≥0,2) and (870≥0)		
2	if(11≥0,1) and (289<-0,5)	1	1
	if(11≥0,4) and (637<0,5)		
	if(229≥1,1) and (456≥-0,9)		
	if(289<-0,5) and (229≥1,1)		
	if(839≥0,5) and (637<0,4)		
3	if(50<-2,3) and (63≥-0,2)	1	0,938
4	if(2<-0,1) and (485≥0,6)	1	1
	if(11<-2,7) and (485≥0,6)		
	if(50≥-2,1) and (485≥0,7)		
	if(63<-0,3) and (485≥0,7)		
	if(194<-0,8) and (485≥0,7)		
	if(366≥-0,9) and (485≥0,7)		
	if(475≥-2,8) and (485≥0,5)		
	if(485≥0,7) and (525≥-1,1)		
	if(485≥0,7) and (929≥-0,4)		
	if(839<-0,5) and (485≥0,6)		
	if(881<-0,2) and (485≥0,6)		
5	if(11≥-1,6) and (97<0,1) and (229<-0,7)	1	1
	if(11≥-1,6) and (97<0,1) and (379<0,1)		
	if(97<0,4) and (41≥-2,3) and (721≥1)		
	if(97<0,3) and (194≥-1,6) and (721≥1)		
	if(97<0,1) and (379<0,1) and (721≥1)		
	if(97<0,4) and (881≥-0,9) and (721≥1)		
	if(348<-1,5) and (41≥-2,1) and (229<0,2)		
6	if(242<0,3) and (881<1) and (2<-1,2) and (637≥0,5)	1	0,627
	if(828<0,7) and (2<-1,1) and (379≥0,2) and (637≥0,4)		
7	if(242≥-0,1) and (63≥0,3)	1	0,952
8	if(97≥0,7) and (127≥0,1) and (863<0,7)	1	1
	if(97≥0,7) and (348<-0,8) and (863<0,7)		
	if(97≥0,7) and (863<0,8) and (63<-0,4)		
	if(97≥0,7) and (863<0,8) and (881≥0,1)		
	if(127≥0,3) and (348<-0,7) and (863<0,7)		
9	if(242<-0,9) and (456≥0,6)	1	0,952

Tabela 24: Melhores regras encontradas na base de dados B_1B_4

Classes	**Regras**	Apt_{Trein}	Apt_{Teste}
1	if(194<0,2) and (289<0,1) and (839<0,8) and (177≥-1,1) and (865≥-0,9)	1	0,317
2	if(11≥0,4) and (289<-0,5)	1	1

3	if(50<-2,3) and (194<-1,1) and (289≥-0,3)
	if(50<-2,3) and (242<0,6) and (289≥-0,7)
	if(50<-2,3) and (289≥-0,4) and (306≥-0,9)
		1	1

4	if(194<-2,4)	1	0,5

5	if(11≥-1,5) and (97<0,1) and (46<-0,3)
	if(11≥-1,7) and (97<0,1) and (348<-1,4)
	if(11≥-1,7) and (97<0,1) and (380<-0,6)
	if(11≥-1,5) and (289≥-1,3) and (380<-0,7)
	if(97<0,1) and (194≥-1,4) and (380<-0,6)
	if(97<0,1) and (242≥0,3) and (380<-0,7)
	if(97≥-1,2) and (289≥-1,6) and (380<-0,6)
	if(97<0,1) and (306<-0,9) and (380<-0,6)
	if(242≥0,3) and (289≥-1,3) and (380<-0,7)
	if(289≥-1,5) and (177≥-1,8) and (380<-0,6)
	if(289≥-1,3) and (306<-1) and (380<-0,6)
	if(289≥-1,3) and (336≥-0,6) and (380<-0,7)
	if(289≥-1,4) and (366≥-0,6) and (380<-0,5)
	if(289≥-1,6) and (380<-0,6) and (661≥-1,2)
	if(289≥-1,3) and (380<-0,7) and (865≥-0,9)
	if(289≥-1,3) and (380<-0,7) and (950≥-0,5)
	if(289≥-1,4) and (828≥-0,8) and (380<-0,6)
	if(289≥-1,6) and (881≥-1) and (380<-0,5)
		1	1

6	if(127<0,1) and (242<-0,6) and (366≥-0,1) if(242<-0,6) and (366≥-0,1) and (177<1,1)	1	0,333
7	if(839<0,4) and (46≥-0,7) and (306≥-0,6)	1	0,938
8	if(97≥0,7) and (863<0,8)	1	0,95
9	if(50<-2,1) and (177<-1,9) and (783≥-0,7)	1	1

Tabela 26: Melhores regras encontradas na base de dados B_2B_4

Classes	**Regras**	Apt_{Trein}	Apt_{Teste}
1	if(97<0,8) and (246≥-0,2) and (177≥-1) and (289<0,2) and (306≥-1,7)	1	0,283
2	if(11≥0,4) and (289<-0,5)	1	1
	if(637<0,4) and (11≥0,4)		
3	if(75≥-0,7) and (246<-0,3)	1	0,875
4	if(19<-0,4) and (526<-0,9)	1	1
	if(19<-0,4) and (843<-1)		
	if(224<-2,2) and (380≥-0,2)		
	if(224<-2,2) and (843<-0,9)		
	if(224<-2,2) and (865≥0,1)		
	if(224<-1,8) and (950≥0)		
	if(409≥-1,9) and (843<-1)		
	if(843<-1) and (783<0,4)		
5	if(2<-1,9) and (289≥-1,3) and (380<-0,7)	1	1
	if(17≥-0,9) and (46<-0,2) and (306<-1)		
	if(17≥-0,9) and (289≥-1,3) and (380<-0,7)		
	if(18<0,7) and (97<0,2) and (11≥-1,5)		
	if(19<-0,7) and (289≥-1,3) and (380<-0,7)		
	if(28≥-0,7) and (97<0,1) and (380<-0,7)		
	if(28≥-0,5) and (289≥-1,3) and (380<-0,7)		
	if(97<0,4) and (11≥-1,6) and (46<-0,2)		
	if(97<0,1) and (11≥-1,7) and (380<-0,7)		
	if(97<0,1) and (224≥-0,6) and (380<-0,6)		
	if(97<0,1) and (246≥-0,4) and (380<-0,6)		
	if(97≥-0,7) and (289≥-1,3) and (380<-0,7)		
	if(97<0,2) and (292<0,9) and (11≥-1,5)		
	if(97<0,1) and (302<0,1) and (11≥-1,5)		
	if(97<0,1) and (306<-1) and (380<-0,5)		
	if(97<0,1) and (380<-0,6) and (950≥0)		
	if(97<0,1) and (637≥0,1) and (306<-0,5)		
	if(141≥-1,4) and (289≥-1,6) and (380<-0,7)		
	if(224≥-0,7) and (289≥-1,3) and (380<-0,7)		
	if(289≥-1,6) and (336≥-0,5) and (380<-0,5)		
	if(289≥-1,3) and (380<-0,7) and (865≥-1)		
	if(289≥-1,3) and (380<-0,7) and (950≥-0,8)		
	if(526≥-0,4) and (289≥-1,3) and (380<-0,7)		
6	if(2<0,8) and (28<0,6) and (141≥-0,4) and (499≥-1) and (177<1,1) and (306≥-1)	1	0,333
7	if(19≥-1,7) and (224≥-0,2) and (306≥-0,5)	1	0,944
8	if(46≥0,7) and (865<-0,4)	1	0,875
9	if(18<-3,1) and (19≥-0,2)	1	1
	if(18<-3,1) and (637≥-1,5)		

Tabela 28: Melhores regras encontradas na base de dados $B_1B_2B_4$

Classes	**Regras**	$Aptidao_{Trein}$	$Aptidao_{Teste}$
1	if(242<1,5) and (348<1,8) and (366<0,9) and (19<1,3) and (75<2,2) and (235≥-1,2) and (843≥-0,3) and (783<0,2) and (950≥-0,2)	0,972	0,389
2	if(11≥0,4) and (289<-0,5) if(839≥0,5) and (637<0,4)	1	1
3	if(75≥-0,7) and (246<-0,3)	1	0,875

4	if(19<-0,4) and (526<-0,8)
	if(19<-0,4) and (843<-1)
	if(50≥-2) and (280<-0,7)
	if(50≥-2) and (526<-0,8)
	if(50≥-2) and (843<-1)
	if(224<-2,1) and (950≥0)
	if(366≥-0,9) and (526<-0,9)
	if(366≥-0,9) and (843<-1)
		1	1

5	if(11≥-2,2) and (289≥-1,3) and (380<-0,5)
	if(17≥-1) and (46<-0,3) and (306<-1)
	if(17≥-0,9) and (46<-0,2) and (661<0,1)
	if(97<0,1) and (224≥-0,6) and (380<-0,2)
	if(97<0,1) and (231≥-0,3) and (306<-0,8)
	if(97<0,1) and (242≥0,1) and (380<-0,2)
	if(97≥-1,6) and (289≥-1,3) and (380<-0,5)
	if(242≥0,3) and (289≥-1,3) and (380<-0,7)
	if(289≥-1,3) and (2<-1) and (380<-0,7)
	if(289≥-1,3) and (17≥-1,6) and (380<-0,7)
	if(289≥-1,3) and (19<-1) and (380<-0,7)
	if(289≥-1,4) and (28≥-0,6) and (380<-0,6)
	if(289≥-1,3) and (224≥-0,6) and (380<-0,7)
	if(289≥-1,3) and (246≥-0,4) and (380<-0,7)
	if(289≥-1,3) and (366≥-1,2) and (380<-0,7)
	if(289≥-1,3) and (380<-0,7) and (865≥-1,2)
	if(289≥-1,3) and (380<-0,7) and (950≥-0,1)
	if(289≥-1,4) and (881≥-0,9) and (380<-0,6)
		1	1

6	if(828<0,1) and (19<1,1) and (141≥-0,4) and (306≥-1)	1	0,533
7	if(2≥-2,3) and (224≥-0,4) and (336<1,1) if(194≥-1) and (839<0,1) and (306≥-0,5)	1	0,938
8	if(46≥0,8) and (865<-0,4)	1	0,938
9	if(18<-3,1) and (19≥-0,6) if(19≥-0,2) and (231<-1,2)	1	0,952

Tabela 29: Melhores regras encontradas na base de dados $B_1B_3B_4$

Classes	**Regras**	Apt_{Trein}	Apt_{Teste}
1	if(531≥0,2) and (870≥0) and (929<0,2)	1	0,3

2	if(11≥0,2) and (289<-0,5)
	if(11≥0,4) and (637<0,4)
	if(229≥1) and (177<1)
	if(229≥1,1) and (456≥-0,9)
	if(289<-0,5) and (229≥1,1)
	if(839≥0,5) and (637<0,4)
		1	1

3	if(50<-2,3) and (63≥-0,5)	1	0,938

4	if(2<-0,2) and (485≥0,6)
	if(11<-2,8) and (485≥0,7)
	if(63<-0,3) and (485≥0,7)
	if(194<-0,8) and (485≥0,7)
	if(366≥-0,9) and (485≥0,6)
	if(475≥-2,5) and (485≥0,1)
	if(485≥0,7) and (661<-0,3)
	if(485≥0,6) and (783<0,4)
	if(485≥0,7) and (865<1,5)
	if(485≥0,7) and (929≥-0,3)
	if(839<-0,5) and (485≥0,7)
	if(881<-0,2) and (485≥0,6)
		1	1

5	if(11≥-1,5) and (97<0,1) and (46<-0,4)
	if(11≥-1,5) and (97<0,1) and (229<-0,5)
	if(41≥-2,2) and (46<0,8) and (380<-0,6)
	if(41≥-2,1) and (229<0,1) and (380<-0,6)
	if(41≥-2,1) and (721≥0,6) and (380<-0,6)
	if(97<0,1) and (41≥-3) and (380<-0,6)
	if(97<0,1) and (306<-0,9) and (380<-0,6)
	if(97<0,1) and (379<0,1) and (380<-0,6)
	if(97<0,3) and (379<0,2) and (721≥1)
	if(97<0,1) and (637≥0,1) and (306<-0,8)
	if(97<0,1) and (721≥1) and (306<-1)
	if(97<0,1) and (870≥-0,9) and (380<-0,6)
	if(97<0,1) and (890≥-0,8) and (380<-0,6)
	if(194≥-1,3) and (229<-0,8) and (380<-0,5)
	if(229<-0,7) and (890≥-0,4) and (306<-1)
	if(242≥0,3) and (41≥-2) and (46<-0,1)
	if(289≥-1,3) and (177≥-1,4) and (380<-0,7)
	if(289≥-1,5) and (721≥0,9) and (380<-0,4)
	if(289≥-1,5) and (890≥-0,2) and (380<-0,4)
		1	1

6	if(242<-0,6) and (366≥-0,1) and (890<-0,7) if(242<-0,6) and (890<-0,1) and (499≥-0,8)	1	0,333
7	if(242≥-0,1) and (63≥0,3)	1	0,952
8	if(46≥0,8) and (865<-0,4)	1	0,938
9	if(242<-1) and (456≥0,6)	1	0,952

APÊNDICE E

Classificação de Oncogenes medidos por *Microarray* utilizando Algoritmos Genéticos

Laurence Rodrigues do Amaral
Universidade Federal de Uberlândia
Laboratório de Inteligência Artificial
Av. João Naves de Ávila, 2160 Uberlândia, Brasil
lramaral@pos.facom.ufu.br
Centro Universitário do Cerrado-Patrocínio

Gina Maira B. Oliveira
Universidade Federal de Uberlândia
Laboratório de Inteligência Artificial
Av. João Naves de Ávila, 2160 Bloco B Uberlândia, Brasil
gina@facom.ufu.br

Foued Salmen Espindola
Universidade Federal de Uberlândia
Instituto de Genética e Bioquímica
Laboratório de Bioquímica e Biologia Molecular
Av. Pará, 1720 Bloco 2E39A Uberlândia, Brasil
foued@ufu.br

Geraldo Sadoyama Leal
Centro Universitário do Cerrado-Patrocínio
Laboratório de Imunologia, Genética e Microbiologia
Av. Arthur Botelho, S/N Patrocínio, Brasil
geraldosadoyama@unicerp.edu.br

Abstract

Técnicas de Inteligência Artificial (IA) têm se tornado cada vez mais importantes na solução de problemas biológicos. Neste artigo, utilizamos um Algoritmo Genético (AG) na busca de regras de alto nível do tipo IF-THEN. Este AG foi aplicado na classificação de uma base de dados de expressão gênica de células cancerígenas advindas de experimentos de microarray, buscando assim, relações entre os níveis de expressões gênicas e os nove tipos de classes de câncer analisados.

1. Introdução

Uma das áreas em que a aplicação de técnicas computacionais inteligentes tem se mostrado mais promissora é a Biologia Molecular [35].

Devido à grande quantidade e complexidade da informação, as ferramentas baseadas na computação convencional têm se mostrado limitadas na abordagem de problemas biológicos complexos. Uma das explicações para essa dificuldade é a ineficiência das ferramentas convencionais em lidar com grandes volumes de dados. Técnicas advindas da Inteligência Artificial (IA), tais como, os algoritmos genéticos e as redes neurais artificiais, são cada vez mais empregadas para tratar problemas em Biologia Molecular. A aplicabilidade dessas técnicas advém de sua capacidade de aprender automaticamente a partir de grandes

volumes de dados e produzir hipóteses úteis [4].

Um fragmento de DNA pode conter diversos genes. A propriedade mais importante dos genes está no fato de que eles contêm o código genético para a expressão do mRNA (RNA mensageiro) que será traduzido em proteínas, componentes estes, essenciais a todo ser vivo [9]. As proteínas são polipeptídeos compostas por conjuntos de aminoácidos. Estes aminoácidos são representados por trincas (códons) de nucleotídeos (Adenina - A, Uracila - U, Citosina - C e Guanina - G) no DNA. O processo pelo qual as seqüências de nucleotídeos dos genes são interpretados na produção de proteínas é denominado expressão gênica [9]. Mensurar e analisar informações de expressão gênica é de grande interesse para as Ciências Biológicas. Esse tipo de análise pode fornecer informações importantes sobre as funções de uma célula, uma vez que as mudanças na fisiologia de um organismo são geralmente acompanhadas por mudanças nos padrões de expressão dos genes [1]. Uma das técnicas mais difundidas para esta medição são os *Microarrays* de DNA [5] [38] [15] [23].

Diferentes técnicas de IA foram aplicadas na análise de dados de expressão gênica, tais como: redes neurais artificiais [41] [25], *Support Vector Machines* [18] [6] e algoritmos genéticos [42] [31] [10] [28] [30] [39]. Em todos os projetos citados anteriormente, o objetivo é encontrar conjuntos de genes (clusters) que possam ser utilizados como classificadores confiáveis, com uma elevada taxa de classificação e um bom desempenho de generalização. Dessa forma, os conjuntos minerados podem auxiliar na classificação de novos casos, facilitando o diagnóstico e o tratamento de doenças. Entretanto, em nenhum desses trabalhos, encontramos classificadores baseados em regras de alto nível, como por exemplo, regras do tipo IF-THEN. Ao contrário, os classificadores obtidos são do tipo caixa-preta, onde a entrada são os dados de expressão de uma determinada amostra de células e a saída é a classe à qual essa amostra provavelmente pertence. Por exemplo, essa saída associada pode ser uma classe de doença. Assim, a partir de um conjunto de dados de milhares de genes chega-se a um pequeno conjunto de poucas dezenas de genes que sejam discriminantes para o problema.

Neste trabalho o enfoque será a busca (mineração) de regras de alto nível, que não só sejam associadas a cada classe individualmente, reduzindo o problema a poucos genes por classe, mas também associando o nível de expressão gênica a cada gene que compõe a regra. Acreditamos que esse tipo de informação possa ser de grande utilidade aos especialistas que buscam entender o mecanismo por detrás de alterações nos padrões de expressão gênica associadas ao aparecimento de determinadas doenças. Para tal, elaborou-se um Algoritmo Genético para a obtenção de regras do tipo IF-THEN a partir de bases de dados de expressões gênicas. Este ambiente evolutivo foi aplicado na classificação de uma base de dados de expressões gênicas de células cancerígenas, advindas de experimentos de *microarray* [34]. O principal objetivo é a busca das relações entre os níveis de expressões gênicas e nove classes de câncer: mama, sistema nervoso central, colom, leucemia, melanoma, pulmão, ovário, renal e reprodutivas. Como ponto de partida, utilizamos conjuntos reduzidos de genes que foram minerados a partir de trabalhos anteriores nessa mesma base de dados [31] [13] e [20].

2. Algoritmos Genéticos (AGs)

AGs são métodos computacionais de busca baseados nos mecanismos da evolução natural e na genética, simulando a teoria da seleção natural de Darwin [19]. Os AGs fazem parte da Computação Evolutiva, área da Inteligência Artificial baseada nas Ciências Biológicas e que se baseia na teoria da evolução das espécies de Charles Darwin.

O AG é um algoritmo que manipula, em paralelo, um conjunto de indivíduos (população), tipicamente cadeias de símbolos de tamanho fixo, que representam cromossomos. A cada indivíduo está associada uma avaliação. O AG transforma a população corrente em uma nova população usando operações de reprodução e sobrevivência, segundo critérios baseados na função de avaliação [27].

Em AGs, uma população de possíveis soluções para o problema em questão evolui de acordo com operadores probabilísticos concebidos a partir de metáforas biológicas, de modo que há uma tendência de que, na média, os indivíduos representem soluções cada vez melhores à medida que o processo evolutivo continua [37].

2.1. Aplicações de Algoritmos Genéticos em *Data Mining* e em Expressão Gênica

Data Mining é um conjunto de técnicas e ferramentas aplicado para a descoberta do conhecimento em bases de dados. O conhecimento minerado é utilizado em nível estratégico, para a tomada de decisão. As aplicações de *data mining* encontram em outras áreas de estudo a construção de abordagens mistas, isto é, soluções multidisciplinares que obtenham melhores resultados, acrescentando desempenho, confiabilidade e permitindo a otimização do processo de mineração de dados [2].

A tarefa de classificação é uma das várias estudadas em *data mining*. Em essência, o problema consiste em atribuir valores para os registros pertencentes a um pequeno conjunto de classes, e assim, descobrir algum relacionamento entre estes atributos. Cada registro é composto de um conjunto de atributos preditos e um atributo objetivo [22] [17].

O conhecimento descoberto é usualmente representado na forma de regras de predição do tipo IF-THEN. Este

tipo de regra se destaca devido ao seu alto nível de entendimento e pela representação do conhecimento simbólico, contribuindo para compreensibilidade das informações descobertas. As regras descobertas podem ser construídas de acordo com vários critérios, tais como: grau de confiança da predição, taxa de acerto da classificação para amostras de classes desconhecidas, compreensibilidade, dentre outros [14].

Dentre os vários trabalhos que foram desenvolvidos utilizando AGs na solução de tarefas de *data mining* podemos citar [14] [40] [26] [24] [11] [12] [36] [16] [3] [8] [7] [33] [32].

Uma outra área onde os AGs estão contribuindo para a descoberta de conhecimento é a área de expressão gênica. Na maioria destes projetos, buscamos clusterizar conjuntos de genes na busca de relações entre estes genes, objetivando assim, encontrar conjuntos de genes que são classificadores confiáveis, que auxiliam na classificação de novos casos, facilitando o diagnóstico e o tratamento de tumores cânceríngenos. Podemos citar [42] [30] [10] [31], [28]. [39],

3 Ambiente Evolutivo

O modelo do AG empregado em nosso ambiente evolutivo foi adaptado a partir do modelo de AG proposto em [14]. O AG em [14] foi desenvolvido na ferramenta GALOPPS [21] e foi elaborado com o objetivo de obter regras de classificação do tipo IF-THEN em bases de dados clínicos de pacientes. Dessa forma, as bases de dados onde o ambiente de Fidelis e colaboradores ([21]) foram aplicadas eram formadas por registros que se caracterizavam por dados do paciente, no caso, a idade e presença da doença em histórico familiar e por dados relacionados a sintomas da paciente, tal como, presença abundante de manchas brancas na face. As características que se relacionavam aos sintomas, que eram a maioria, foram todas discretizadas em: 0-ausente, 1-ocorrência leve, 2- ocorrência moderada e 4-ocorrência severa. Nosso ambiente evolutivo, implementado na linguagem Delphi®, precisou ser adaptado para trabalhar com bases de dados de expressão gênica, onde os registros apresentam os níveis de expressão de dezenas (centenas ou milhares) de genes, que são valores contínuos e com precisão variável (números reais). A seguir as principais características de nosso modelo de AG são detalhadas: codificação do indivíduo, operadores genéticos e função de avaliação.

3.1 Cromossomo ou Indivíduo

O indivíduo ou cromossomo do nosso AG é composto por *N* genes, onde cada gene do indivíduo está relacionado a uma condição envolvendo um atributo (um gene do *dataset*), onde *N* é o número de genes encontrados na base de expressão gênica. A primeira posição do indivíduo corresponde ao primeiro gene encontrado na base de dados e assim sucessivamente até que todos os genes de cada *dataset* estejam representados. O indíviduo é ilustrado na figura 1

$Gene_1$			...	...	$Gene_N$		
P_1	O_1	V_1	...	...	P_N	O_1	V_1

Figura 1. Cromossomo ou Individuo

Cada i-ésima posição do indivíduo é subdividida em três campos: Peso, Operador e Valor, como ilustrado acima. Cada gene corresponde a uma condição na parte SE da regra e o individuo (cromossomo) a toda a parte SE da regra. O campo Peso é uma variável do tipo inteira e o seu valor está compreendido entre os valores 0 (zero) e 10 (dez). É importante dizer que este campo Peso é o responsável pela inserção ou exclusão do gene na regra. Caso este valor seja menor do que um valor limite este gene não fará parte da regra, caso contrário o mesmo fará. Neste trabalho foi utilizado como limite o valor 8 (oito). O campo Operador pode variar entre as operações $<$ (menor) e $\geq$ (maior ou igual). O campo de Valor é uma variável do tipo ponto flutuante que pode variar entre o menor e o maior valor encontrados na base de expressão gênica avaliada.

3.2 Operadores Genéticos

Na seleção dos pais para *crossover* aplicamos o método do Torneio Estocástico utilizando *tour* de tamanho 3 (três). Nestes pais selecionados, aplicamos *crossover* múltiplo com dois pontos de corte, gerando dois novos filhos com taxa de *crossover* de 100%. Nestes dois filhos gerados, aplicamos o operador de mutação. Os operadores de mutação utilizados neste trabalho variam com o tipo do gene avaliado e possui taxa de mutação por gene no valor de 30%. Para o gene Peso o novo valor é dado sorteando o incremento ou o decremento de um (1) ao valor original. Para o gene Operador ocorre o sorteio de um novo operador dentre os possíveis excluindo o encontrado originalmente. Neste trabalho foi utilizado apenas dois operadores ($\geq$ e $<$), levando à troca de um pelo outro quando aplica-se o operador de mutação ao gene Operador. Na composição dos individuos que irão participar da próxima geração do AG, selecionamos os melhores pais e filhos.

3.3 Função de Avaliação ou Aptidão (FA) (*Fitness Function*)

A Aptidão (ou *fitness*) refere-se ao grau de contribuição de uma determinada solução candidata para a convergência

do AG na busca da melhor solução dentro do espaço de busca.

Neste trabalho a FA avalia a qualidade de cada regra (indivíduo). A FA aqui aplicada pode ser encontrada em [29]. Para o perfeito entendimento da FA aqui aplicada, alguns conceitos precisam ser reforçados. Quando utilizamos uma determinada regra na classificação de um exemplo, quatro diferentes tipos de resultados podem ser observados, dependendo da classe predita pela regra e a verdadeira regra do exemplo. São eles:

- *True Positive* (tp) - A regra prediz que o exemplo pertence a uma determinada classe e o mesmo pertence;
- *False Positive* (fp) - A regra prediz que o exemplo pertence a uma determinada classe mas o mesmo não pertence;
- *True Negative* (tn) - A regra prediz que o exemplo não pertence a uma determinada classe e o mesmo não pertence;
- *False Negative* (fn) - A regra prediz que o exemplo não pertence a uma determinada classe mas o mesmo pertence;

A FA utiliza dois indicadores comumente utilizados em domínios médicos, chamados de sensibilidade (*Se*) e especificidade (*Sp*). Se e Sp são definidos abaixo:

$$Se = \frac{tp}{(tp + fn)} \quad (1)$$

$$Sp = \frac{tn}{(tn + fp)} \quad (2)$$

Finalmente, a FA utilizada é definida como o produto destes dois indicadores, *Se* e *Sp*, como segue abaixo:

$$Aptidao = Se * Sp \quad (3)$$

O objetivo do trabalho é maximizar ao mesmo tempo *Se* e *Sp* e consequentemente *Aptidao*, utilizando para isso, as equações 1, 2 e 3. Em cada execução, o nosso AG trabalha com um problema de classificação de duas classes, isto é, quando o AG está procurando por regras de uma dada classe, todas as outras classes são agrupadas em uma única classe.

3.4 Bases de dados

As bases utilizadas no nosso trabalho, foram extraídas dos trabalhos [31], [13] e [20]. Cada um destes trabalhos partiram de conjuntos de genes extraídos da base NCI60 [34] composta por dados de expressão gênica, advindos de experimentos de *microarray*, contendo informações sobre células cancerígenas de 9 (nove) classes. São elas: mama, sistema nervoso central, cólom, leucemia, melanoma, pulmão, ovário, renal e reprodutivas. Cada um destes trabalhos chegaram a um conjunto de genes preditores para todas as classes de câncer citadas acima. No trabalho [31] foi obtido um conjunto preditor, chamado no trabalho de B1, constituído de 13 genes respectivamente.

No trabalho [13] o conjunto preditor, chamado de B2, é constituído por 20 genes e no trabalho [20] por 17 genes (B3).

4 Resultados

Na obtenção destes resultados utilizamos, como parâmetros do AG, população inicial de 400 indivíduos e o executamos por 100 gerações.

Como é possível observar na Tabela 1, embora o resultado de treinamento seja quase sempre 100% nas três bases avaliadas, o resultado de generalização dessas regras não é tão bom, pois ao aplicarmos as mesmas sobre a terceira partição dos registros que ficaram de fora da evolução do AG, o resultado de classificação das regras cai bastante. Acreditamos que esse desempenho se deva ao baixo número de amostras por classe que, em alguns casos chega a apenas 3 (três) registros por classe. Entretanto, essa é uma característica peculiar aos experimentos de microarray, devido ao seu alto custo e dificuldade de execução. Assim, realizamos várias execuções do AG na esperança de que ao obtermos uma variedade de regras com 100% de treinamento para cada classe, pelo menos uma delas tivesse uma boa capacidade de generalização (alto valor de teste).

Tabela 1. Média geral

	Média Geral	
Base	Treinamento	Teste
B1	0,996481	0,433
B2	1	0,386852
B3	1	0,304148

A Tabela 2 traz os melhores resultados obtidos nessa busca, apresentando as melhores regras descobertas pelo nosso AG. Para cada cada classe, nosso ambiente evolutivo foi executado 50 (cinqüenta) vezes, variando a semente randômica utilizada na geração da população inicial. A melhor regra encontrada nas 50 execuções, levando em consideração seu valor de treinamento em dois terços dos registros (e usando o menor número de genes como critério de desempate) foi selecionada como a regra preditora da classe. Cada uma destas regras foi aplicada separadamente em uma nova amostra de teste (1/3 dos registros), para avaliar o do nível de generalização de cada regra obtida em treinamento.

Tabela 2. Melhrores Resultados

C	Regra	Trein	Teste	Base
1	if(Gene_28<0,7) and (Gene_409≥0,4) and (Gene_499<0,2)	1	0.5	B2
	if(Gene_63<-1) and (Gene_379≥-0,5) and (Gene_890≥0,1)	1	0.5	B3
2	if(Gene_289<-0,5) and (Gene_839≥-0,2)	1	1	B1
3	if(Gene_97<-1,4) and (Gene_231≥-0,4)	1	1	B2
4	if(Gene_485≥0,7)	1	0.5	B3
5	if(Gene_97<0,6) and (Gene_242≥0,5) and (Gene_348<-1,1)	1	1	B1
6	if(Gene_2≥-3,1) and (Gene_229≥-0,7)	1	0.933	B3
7	if(Gene_63≥0) and (Gene_379<0,5)	1	1	B3
8	if(Gene_97≥0,9) and (Gene_348<-0,2)	1	1	B1
	if(Gene_97≥1) and (Gene_292≥0,5)	1	1	B2
9	if(Gene_366<-0,6)	1	0	B1
	if(Gene_409<-1,7)	1	0	B2
	if(Gene_525<-1,3)	1	0	B3

Para cada regra encontrada na tabela 2 mostramos informações do seu desempenho em um conjunto de treinamento e teste, obtidos através da equação 3, além de qual base de dados a regra provêm.

Das nove classes avaliadas, em cinco delas (classes 2, 3, 5, 7 e 8) foi possível atingir 100% de classificação, tanto em treinamento quanto em teste. Na classe 6 o resultado também foi bom, pois encontramos uma regra que obteve 100% de acertos em treinamento e 93,3% em teste. Infelizmente, nas três classes restantes, embora a regra tenha atingido 100% em treinamento, o desempenho em teste não foi bom: 50% para as classes 1 e 4 e 0% para a classe 9. Assim, consideramos que o desempenho foi muito bom em seis das nove classes, mas bem abaixo do satisfatório nas outras três.

5 Conclusão e Trabalhos Futuros

Com nossos experimentos de *crossvalidation*, foi possível observar que embora a obtenção de regras com alto índice de treinamento seja relativamente fácil, a qualidade dessas regras é logo descartada pelo desempenho das mesmas na base de testes. Tal comportamento, acreditamos que possa ser justificado pelo baixo número de amostras por classe, inerente ao problema. Para compensar essa dificuldade, procuramos executar um grande número de execuções do AG, para obtenção de um maior número de regras por classe, com alta taxa de desempenho na base de treinamento. Dessa forma, conseguimos obter excelentes regras em seis das nove classes. Entretanto, em três classes não foi possível obter regras satisfatórias. Animados com os resultados promissores desse trabalho, pretendemos dar continuidade ao mesmo com os seguintes passos: (i) análise de uma quarta base (B4) também extraída de [31] que também provocou uma redução da base de dados de expressões gênicas em [34], obtendo um conjunto de 11 genes; (ii) aplicar a metodologia adotada nesse trabalho em novas bases criadas a partir da composição das quatro bases já existentes na literatura [31], [13] e [20], obtendo-se 11 novas bases (B1+B2, B1+B3, B1+B4, B2+B3, B2+B4, B3+B4, B1+B2+B3, B1+B2+B4, B2+B3+B4, B1+B3+B4, B1+B2+B3+B4). Esses experimentos já se encontram em andamento e até o momento conseguimos regras com pelo menos 75% de desempenho na base de testes. Com as regras de alto nível obtidas, e com as que ainda serão obtidas em novos experimentos, conseguimos delimitar possíveis genes relacionados a cada classe de câncer e seus respectivos níveis de expressão, conseguindo assim, uma associação gene/câncer e gene/gene que esperamos que possa contribuir para o diagnóstico deste tipo de câncer limitando assim o número de genes a serem analisados na busca de novos tratamentos.

Referências

[1] B. Alberts, D. Bray, and J. Lewis. *Biolgia Molecular da Célula*. Artes Médicas, 3 edition, 1997.

[2] I. Anciutti, A. L. Gonçalves, F. A. Siqueira, and P. S. S. Borges. Uma aplicação de data mining sobre circuitos elétricos de baixa tensão utilizando algoritmos genéticos. *1º Workshop de Ciências da Computação e Sistemas da Informação da Região Sul (WorkComp Sul)*, Maio 2004.

[3] D. Araujo, H. Lopes, and A. Freitas. A parallel genetic algorithm for rule discovery in large databases. In *Systems, Man and Cybernetics*, volume 3, pages 940 – 945, Tokyo, October 1999. IEEE.

[4] P. Baldi and S. Brunak. *Bioinformatics: the Machine Learning approach*. MIT Press, 2 edition, 2001.

[5] S. Brenner, M. Johnson, J. Bridgham, G. Golda, D. H. Lloyd, D. Johnson, S. M. S. Luo, M. Foy, M. Ewan, R. Roth, D. George, S. Eletr, G. Albrecht, E. Vermaas, S. R. Williams, T. B. K. Moon, R. B. M. Pallas, J. Kirchner, K. Fearon, J. Mao, and K. Corcoran. Gene expression analysis by massive parallel signature sequencing (mpss) on microbead array. *Nature Biotechnology*, 18(10):630–640, 2000.

[6] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, and D. Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Stanford University of Medicine*, 1999.

[7] D. Carvalho and A. Freitas. A genetic algorithm-based solution for the problem of small disjuncts. In Springer-Verlag, editor, *Principles of Data Mining and Knowledge Discovery*, volume 1910, pages 345–352, Lyon, France, 2000. 4th European, Lecture Notes in Artificial Intelligence.

[8] D. Carvalho and A. Freitas. A hybrid decision tree/genetic algorithm for coping with the problem of small disjuncts in data mining. In *Genetic and Evolutionary Computation (GECCO-2000)*, pages 1061–1068, Las Vegas, NV, USA, Jully 2000.

[9] M. C. P. de Souto, A. C. Lorena, A. C. B. Delbem, and A. C. P. L. F. de Carvalho. Técnicas de aprendizado de máquina para problemas de biologia molecular. Sociedade Brasileira de Computação, 2003.

[10] K. Deb and A. R. Reddy. Classification of two and multi-class cancer data reliably using multi-objective evolutionary algorithms. *KanGAL Report*, 2003.

[11] H. Ding, L. Benyoucef, and X. Xie. A simulation-based multi-objective genetic algorithm approach for networked enterprises optimization. *Engineering Applications of Artificial Intelligence*, 2005.

[12] C. R. dos Santos Miranda, G. M. B. de Oliveira, and J. B. dos Santos. Algoritmos genéticos aplicados em data mining para obtenção de regras simples e precisas. In *Anais do SBAI2003*, pages 638–643, 2003.

[13] S. Dudoit, J. Fridlyand, and T. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. in press 576, Berkeley Stat. Dept. Technical Report, JASA, 2000.

[14] M. V. Fidelis, H. S. Lopes, and A. A. Freitas. Discovery comprehensible classification rules with a genetic algorithm. In *Congress on Evolutionary Computation - (CEC-2000)*, pages 805–810. La Jolla, CA, USA, 2000.

[15] W. M. Freeman, S. J. Walker, and K. E. Vrana. Quantitative rt-pcr: pitfalls and potentials. *Biotechniques*, 26:112–122, 1999.

[16] A. A. Freitas. *Advances in Evolutionary Computation*, chapter A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery. Springer-Verlag, 2002.

[17] A. A. Freitas and S. H. Lavington. *Mining Very Large Databases with Parallel Processing*. Kluwer Academic Publishers, London, 1998.

[18] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Oxford University Press*, 2000.

[19] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Adison-Wesley, USA, 1989.

[20] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction. *Science*, 286, October 1999.

[21] E. D. Goodman. An introduction to gallops - the genetic algorithms optimized for portability and parallelism system. Technical report, Departament od Computer Science - Michigan State University, 1996.

[22] D. Hand. *Construction and Assessment If Classification Rules*. John Wiley and Sons, Chichester, 1997.

[23] C. A. Harrington, C. Rosenow, and J. Retief. Monitoring gene expression using dna microarrays. *Curr. Opin. Microbiol.*, 3:285–291, 2000.

[24] H. Ishibuchi and T. Yamamoto. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. *Fuzzy Sets and Systems*, (141):59–88, 2004.

[25] J. Khan, J. S. Wei, M. Ringnér, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer. Classifiction and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 2001.

[26] Y. Kim and W. N. Street. An intelligent system for customer targeting: a data mining approach. *Decision Support Systems*, (37):215–228, 2004.

[27] J. R. Koza. *Genetic Programming. On the Programming of Computers by Means of Natural Selection*. MIT Press, USA, 1992.

[28] J. J. Liu, G. Culter, W. Li, Z. Pan, S. Peng, T. Hoey, L. Chen, and X. Ling. Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. *Oxford University Press*, 21(11 2005):2691–2697, 2005.

[29] H. S. Lopes, M. S. Coutinho, and W. C. Lima. An evolutionary approach to simulate cognitive feedback learning in medical domain. In E. Sanchez, T. Shibata, and L. A. Zadeh, editors, *Genetic Algorithms and Fuzzy Logic Systems*, pages 193–207. World Scientific, 1997.

[30] S. Mitra and H. Banka. Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognition*, 2006.

[31] C. H. Ooi and P. Tan. Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. *Bioinformatic*, 19(1):37–44, 2003.

[32] M. A. C. Pacheco, M. M. R. Vellasco, C. H. P. Lopes, and E. P. L. Passos. Extração de regras de associação em bases de dados por algoritmos genéticos. In *Anais do XIII Congresso Brasileiro de Automática (CBA 2000)*, Floarianópolis, Setembro 2000.

[33] W. Romao, A. A. Freitas, and R. C. S. Pacheco. A genetic algorithm for discovering interesting fuzzy prediction rules: applications to science and technology data. In *Genetic and Evolutionary Computation (GECCO-2002)*, New York, July 2002.

[34] D. T. Ross, U. Scherf, M. B. Eisen, C. M. Perou, C. Rees, P. Spellman, V. Iyer, S. S. Jeffrey, M. V. de Rijn, M. Waltham, A. Pergamenschikov, J. C. F. Lee, D. Lashkari, D. Shalon, T. G. Myers, J. N. Weinstein, D. Botstein, and P. O. Brown. Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics*, 2000.

[35] J. C. Setúbal and J. Meidanis. *Introduction to Computacional Molecular Biology*. PWS Publishing Company, Boston, 1997.

[36] K. C. Tan, Q. Yu, C. M. Heng, and T. H. Lee. Evolutionary computing for knowledge dicovery in medical diagnosis. *Artificial Intelligence in Medicine*, (27):129–154, 2003.

[37] J. Tanomaru. Motivação, fundamentos e aplicações de algoritmos genéticos. In *Congresso Brasileiro de Redes Neurais*, Curitiba, 1995. III Escola de Redes Neurais.

[38] V. E. Velculescu, L. Zhang, B. Vogelstein, and K. W. Kinzler. Serial analysis of gene expression. *Science*, 270:484–487, 1995.

[39] M. Wahde and Z. Szallasi. Improving the prediction of the clinical outcome of breast cancer using evolutionary algorithms. *Soft Comput*, 2006.

[40] D. C. Weaver. Applying data mining techniques to library design, lead generation and lead optimization. *Science Direct*, 2004.

[41] Y. Xu, F. M. Selaru, J. Yin, T. T. Zou, V. Shustova, Y. Mori, F. Sato, T. C. Liu, A. Olaru, S. Wang, M. C. Kimos, K. Perry, K. Desai, B. D. Greenwald, M. J. Krasna, D. Shibata, J. M. Abraham, and S. J. Meltzer. Artificial neural networks and gene filtering distinguish between global gene expression profiles of barret's esophagus and esophageal cancer. *Cancer Research*, 2002.

[42] I. Zwir, R. R. Zaliz, and E. H. Ruspini. Automated biological sequence description by genetic multiobjective generalized clustering. *New York Academy of Sciences*, (980):65–82, 2002.