UNIVERSIDADE DE CAXIAS DO SUL
Curso de Bacharelado em Ciência da Computação

Raquel de Lima Machado

DESENVOLVIMENTO DE UM ALGORITMO IMUNOLÓGICO PARA AGRUPAMENTO DE DADOS

Caxias do Sul

2011

Raquel de Lima Machado

DESENVOLVIMENTO DE UM ALGORITMO IMUNOLÓGICO PARA AGRUPAMENTRO DE DADOS

Trabalho de Conclusão de Curso para obtenção do Grau de Bacharel em Ciência da Computação da Universidade de Caxias do Sul.

Carine G. Weber
Orientadora

Caxias do Sul

2011

“A mente que se abre a uma nova idéia,
jamais voltará ao seu tamanho original.”
Albert Einstein.

AGRADECIMENTOS

Agradeço primeiramente e especialmente aos meus pais, sem os quais eu não conseguiria ter chegado até aqui, pelo apoio e auxílio durante todo o curso tanto financeiramente como emocionalmente, por terem me ajudado a realizar esse sonho e objetivo de vida. Ao meu namorado Marcelo, pela força nos momentos mais difíceis onde tudo parecia obscuro e que nunca ia acabar pela compreensão dos momentos que tivemos que abrir mão e das atividades que tivemos que parar. A minha prima Simone por sempre me dar uma palavra de apoio e por acreditar que eu conseguiria. A minha cunhada Priscila por sempre estar disposta a ajudar quando precisei. A minha orientadora Carine pelas dicas e auxílio durante toda a execução deste trabalho. Ao colega Luis por todas as dicas e sugestões durante a execução do trabalho. E as minhas colegas e amigas, Daline e Priscila, por todos os finais de semana em que passamos juntas para realização de trabalhos e estudo para provas, mas principalmente pela amizade construída ao longo desses anos e que espero que durem muitos mais.

RESUMO

A tarefa de agrupar dados é de grande valia no mundo atual, visto que a quantidade de dados armazenados e manipulados tem crescido muito. O principal objetivo de um processo de agrupamento é formar grupos de instâncias, de forma a aumentar a homogeneidade dentro do grupo e a heterogeneidade entre grupos. Um processo de agrupamento é composto por cinco etapas principais: (a) preparação dos dados, (b) escolha de uma medida de similaridade, (c) seleção de uma técnica de agrupamento, (d) validação dos grupos formados e por fim, (e) a interpretação dos resultados. Este trabalho apresenta uma revisão de cada uma dessas etapas separadamente. Sabe-se que cada problema de agrupamento pode ser melhor resolvido por um método adequado , porém nem todos os métodos conseguem resolver todos os problemas de uma maneira satisfatória. O objetivo deste trabalho é propor um algoritmo para agrupamento de dados com base no comportamento do sistema imunológico. Os sistemas imunológicos artificiais se baseiam nos processos do sistema imune dos vertebrados, em especial na teoria da seleção clonal, no princípio da seleção negativa e na teoria das redes imunológicas. Todas elas auxiliam na construção de novos algoritmos, como o sistema imunológico de reconhecimento de padrão e alguns algoritmos de agrupamento descritos neste trabalho. Partindo-se do estudo de agrupamento de dados e das teorias do sistema imunológico artificial é proposto um algoritmo que tem como entrada um dataset, e como saída um conjunto de grupos dos dados. O algoritmo foi desenvolvido utilizando a linguagem Java e sua arquitetura foi feita em três camadas. Foram realizados testes em três datasets públicos: Iris Plants Database, Wisconsin Breast Cancer Database e Diabetes Data Set. Os resultados obtidos de cada um dos datasets foram comparados com os resultados do algoritmo k-means. O algoritmo imunológico se mostrou tão ou mais eficiente que o algoritmo k-means em todos os datasets testados.

Palavras-chave: agrupamento de dados, sistema imunológico artificial, teoria da seleção clonal, teoria da seleção negativa.

ABSTRACT

The task of grouping data is of great value in today's world, since the amount of data stored and manipulated has grown tremendously. The main goal of a clustering process is to form groups of instances, to increase the homogeneity within the group and heterogeneity between groups. A clustering process consists of five main steps: (a) data preparation, (b) choice of a similarity measure, (c) selection of a clustering technique, (d) validation of the groups formed and finally, (e) the interpretation of results. This paper presents a review of each of these steps separately. It is known that every problem of grouping can be better solved by a suitable method, but not all methods can solve all the problems in a satisfactory manner. The objective of this study is to propose an algorithm for grouping data based on the behavior of the immune system. The artificial immune systems are based on the processes of the immune system of vertebrates, especially in the clonal selection theory, the principle of negative selection and the theory of immune networks. All of them help in the construction of new algorithms, such as the immune system of pattern recognition and some clustering algorithms described in this paper. Based on the study of grouping of data and theories of artificial immune system is proposed an algorithm that takes as input a dataset, and as output a set of groups of data. The algorithm was developed using the Java language and its architecture is made of three layers. Tests were conducted on three public datasets: Iris Plants Database, Wisconsin Breast Cancer Database and Diabetes Data Set Results from each of the datasets were compared with the results of k-means algorithm. The immune algorithm proved equally or more efficient than the k-means algorithm in all datasets tested.

Keywords: clustering, artificial immune system, clone selection theory, negative selection theory.

LISTA DE FIGURAS

Figura 1 - Diferente formas de agrupamento para um conjunto de animais (BARANAUSKA, 2003) 20
Figura 2 - Diversas formas no agrupamento de cartas de baralho (KASNAR e GONÇALVES, 2007) 21
Figura 3 - Formas de agrupamento para um conjunto de objetos no plano cartesiano (JAIN et al, 1999) 22
Figura 4 - Agrupamento de documentos textuais (WIVES, 1999) 23
Figura 5 - Etapas de um processo de agrupamento 24
Figura 6 - Atividades realizadas na etapa de preparação de padrões 24
Figura 7 - Representação de um objeto x no plano cartesiano com duas características x_1 e x_2 (ROCHA e LACHI, 2005) 25
Figura 8 – Representação de dois objetos que formam o conjunto de objetos X (ROCHA e LACHI, 2005) 26
Figura 9 - Representação de uma pessoa com atributos quantitativos (ROCHA e LACHI, 2005) 26
Figura 10 - Resultado da aplicação da fórmula de distância euclidiana (OLIVEIRA, 2005) 30
Figura 11 - Resultado da aplicação da fórmula de distância euclidiana quadrática (OLIVEIRA, 2005) 31
Figura 12 - Resultado da aplicação da fórmula de distância manhattan (OLIVEIRA, 2005) 32
Figura 13 - Métodos de agrupamento e seus respectivos algoritmos 35
Figura 14 - Fluxograma do algoritmo k-médias 38
Figura 15 - Demonstração do algoritmo k-médias (LENZ, 2009) 39
Figura 16 - Exemplo de grade em um conjunto de dados bidimensional (ALVES, 2007) 42
Figura 17 - Execução do método de agrupamento DBSCAN (HAN e KAMBER, 2001) 44
Figura 18 - Representação de grupos (LINDEN, 2009) 45
Figura 19 - Estrutura multicamada do sistema imunológico (CASTRO, 1999) 51
Figura 20 - Princípio da seleção clonal (CASTRO, 1999) 53
Figura 21 - Etapas do algoritmo CLONALG 55
Figura 22 - Fase de censoriamento do algoritmo de seleção negativa (Forrest et AL, 1994) 58
Figura 23 - Fase de monitoramento do algoritmo de seleção negativa (FORREST et AL, 1994) 59
Figura 24 - União entre receptores e epítopos (CASTRO, 1999) 61
Figura 25 - Algoritmo AIRS 63
Figura 26 - Algoritmo imunológico de agrupamento para controle de tráfego (JIA et al, 2006) 67
Figura 27 - Questões de formalização de um algoritmo (PEREIRA et al, 2009) 69
Figura 28 - Formas de representação de um algoritmo (PREUSS, 2002) 70
Figura 29 - Principais estruturas de controle (PREUSS, 2002) 71
Figura 30 - Questão de formalização do algoritmo imunológico 73
Figura 31 - Espaço de formas de Perelson e Oster (1979) 74
Figura 32 - Fluxograma algoritmo imunológico para agrupamento de dados 76
Figura 33 - Esquema do estudo de caso de avaliação do algoritmo imunológico utilizado pelo programador 82

Figura 34 - Esquema do estudo de caso do algoritmo imunológico utilizado pelos usuários ..83
Figura 35 - Plataforma de desenvolvimento ..84
Figura 36 - Diagrama de pacotes ..85
Figura 37 - Interfaces da biblioteca substance ..87
Figura 38 - Tela inicial do algoritmo imunológico ..89
Figura 39 - Tela de informações do algoritmo imunológico ..90
Figura 40 - Tela de resultados do algoritmo imunológico ..91
Figura 41 - Índice de homogeneidade e separação para distância euclidiana na base da Íris ..94
Figura 42 - Índice de homogeneidade e separação para distância euclidiana quadrática na base da Íris ..94
Figura 43- Índice de homogeneidade e separação para distância euclidiana na base do Câncer ..100
Figura 44 - Índice de homogeneidade e separação para distância euclidiana quadrática na base do Câncer ..100
Figura 45 - Índice de homogeneidade e separação para distância euclidiana na base da Diabete ..103
Figura 46 - Índice de homogeneidade e separação para distância euclidiana quadrática na base da Diabete ..103

LISTA DE TABELAS

Tabela 1 - Dados quantitativos de pessoas a serem agrupadas 27
Tabela 2 - Exemplo de cálculo de distância para dados numéricos 32
Tabela 3 - Conjunto de objetos com características de valores categorizados 33
Tabela 4 - Métricas de similaridade para valores categorizados 33
Tabela 5 - Objetos primários no formalismo do sistema adaptativo 72
Tabela 6 - Objetos secundários no formalismo do sistema adaptativo 72
Tabela 7 - Objetos primários no formalismo do algoritmo imunológico 74
Tabela 8 - Objetos secundários no formalismo do algoritmo imunológico 75
Tabela 9 - Equações para cálculo de afinidade em SIA 81
Tabela 10 - Distribuição de dados da Iris por classe 92
Tabela 11 - Comparação entre o algoritmo k-means e imunológico através do agrupamento 93
Tabela 12 - Comparação entre o algoritmo k-means e imunológico através do tempo 95
Tabela 13 - Comparação para o algoritmo imunológico através da normalização 95
Tabela 14 - Comparação do fator de mutação no algoritmo imunológico 96
Tabela 15 - Comparação do número de clones no algoritmo imunológico 97
Tabela 16 - Distribuição de dados da base de Câncer por classe 98
Tabela 17 - Comparação entre o algoritmo k-means e imunológico através do agrupamento 99
Tabela 18 - Comparação entre o algoritmo k-means e imunológico através do tempo 101
Tabela 19 - Distribuição de dados da base de Diabete por classe 102
Tabela 20 - Comparação entre o algoritmo k-means e imunológico através do agrupamento 102
Tabela 21 - Comparação entre o algoritmo k-means e imunológico através do tempo 104

LISTA DE FÓRMULAS

Equação 1 - Distância euclidiana 29
Equação 2 - Distância euclidiana quadrática 30
Equação 3 - Distância de manhattan 31
Equação 4 - Distância euclidiana para AIRS 64
Equação 5 - Cálculo do vetor de distância entre antígeno e anticorpo 66
Equação 6 - Cálculo da afinidade entre antígeno e anticorpo 66
Equação 7- Fórmula para recalcular linfócitos 79
Equação 8 - Cálculo do percentual do atributo 79
Equação 9 - Cálculo para mutação do atributo do linfócito 80
Equação 10 - Cálculo para normalização 80
Equação 11 - Distância euclidiana para SIA 81
Equação 12 - Distância euclidiana quadrática para SAI 81
Equação 14 - Fórmula da homogeneidade 81
Equação 15 - Fórmula da separação 82

LISTA DE ALGORITMOS

Algoritmo 1- K-médias....40
Algoritmo 2 - CLONALG....56
Algoritmo 3 - Seleção negativa fase de censoriamento....60
Algoritmo 4 - Seleção negativa fase de monitoramento....60
Algoritmo 5 - AIRS....65
Algoritmo 6 - Algoritmo imunológico para agrupamento de dados....78

LISTA DE ABREVIATURAS E SIGLAS

Sigla	Significado em Português	Significado em Inglês
AIRS	Sistema de reconhecimento imunológico artificial	*Artificial immune recognition system*
AINET	Rede imunológica artificial	*Artificial immune networks*
ARB	Bola de reconhecimento artificial	*Artificial recognition ball*
BIRCH	Redução e agrupamento equilibrado iterativo usando hierarquias	*Balanced iterative reducing and clustering using hierarchies*
CLONALG	Algoritmo de seleção clonal	*Clonal selection algorithm*
CURE	Agrupamento usando representação	*Clustering using representatives*
DBSCAN	Agrupamento espacial baseado em densidade de aplicação com ruído	*Density-based spatial clustering of applications with noise*
DENCLUE	Agrupamento baseado em densidade	*Density-based clustering*
EM	Maximização da expectativa	*Expectation maximization*
ISCA	Algoritmo imunológico de agrupamento supervisionado	*Immune supervised clustering algorithm*
ITS	Sistema de transporte inteligente	*Intelligent transport system*
MST	Árvore geradora mínima	*Minimum spanning tree*
SIA	Sistema imunológico artificial	*Artificial immune system*
STING	Método baseado em grade de informação estatística	*Statistical information grid-based method*
SOM	Mapa de auto-organização	*Self organizing map*
NSA	Algoritmo de seleção negativa	*Negative selection algorithm*
TOD	Controle do dia	*Time of day*
ACUM	Acumulador	*Accumulator*
CONT	Contador	*Counter*
ATRIB	Atributo	*Attribute*
PERCINF	Percentual Informado	*Percentage Reported*
PERCATRIB	Percentual do Atributo	*Percentage of the Attribute*
MINATRIB	Mínimo do Atributo	*Minimum Attribute*
MAXATRIB	Máximo Atributo	*Maximum Attribute*
VI	Virginica	*Virginica*
SE	Setosa	*Setosa*
VE	Versicolour	*Versicolour*
MA	Maligno	*Malignant*
BE	Benigno	*Benign*
POS	Positivo	*Positive*
NEG	Negativo	*Negative*

SUMÁRIO

1 Introdução 15
1.1 Objetivos 16
1.2 Estrutura do texto 17
2 Agrupamento de dados 18
2.1 Agrupamento de dados 18
2.2 Etapas do processo de agrupamento 23
2.3 Preparação de padrões 24
2.4 Medidas de similaridade 28
2.4.1 Medidas de similaridade para dados numéricos 28
2.4.1.1 Distância euclidiana 29
2.4.1.2 Distância euclidiana quadrática 30
2.4.1.3 Distância de manhattan 31
2.4.2 Medidas de similaridade para dados categorizados 33
2.5 Agrupamento 34
2.5.1 Métodos hierárquicos 36
2.5.2 Métodos particionais 37
2.5.3 Métodos baseados em modelo 40
2.5.4 Métodos baseados em grades 41
2.5.5 Métodos baseados em densidade 43
2.6 Representação de grupos 44
2.7 Validação de grupos 45
2.8 Interpretação dos resultados 47
2.9 Aplicações 47
2.10 Considerações finais 49
3 Sistemas imunológicos artificiais 50
3.1 Contextualização 50
3.2 Seleção clonal 52
3.2.1 Inspiração biológica 52
3.2.2 Algoritmo da seleção clonal 54
3.3 Seleção negativa 57
3.3.1 Inspiração biológica 57
3.3.2. Algoritmos de seleção negativa 57
3.4 Sistema imunológico artificial de reconhecimento 60
3.4.1 Inspiração biológica 60
3.4.2 Algoritmo do sistema imunológico artificial de reconhecimento 61
3.5 Aplicações 65
3.6 Considerações finais 68
4 Proposta de um algoritmo imunológico para agrupamento de dados 69
4.1 Método 69
4.2 Formalismo 71
4.3 Algoritmo 73
4.3.1 Geração de linfócitos 79
4.3.2 Normalização 80
4.3.3 Cálculos de afinidade entre elementos 80
4.3.4 Índices de validação 81

4.4 Execução do algoritmo 82
4.5 Considerações finais 83
5 Algoritmo Imunológico 84
5.1 Plataforma de desenvolvimento 84
5.2 Pacote de classes do sistema 85
5.3 Interface 87
5.4 Cenário de uso 91
5.4.1 Preparação dos dados 92
5.4.2 Base de dados íris 92
5.4.3 Base de dados do câncer 98
5.4.4 Base de dados da diabete 101
5.5 Considerações finais 104
6 Conclusões 105
6.1 Síntese do tcc 105
6.2 Contribuições do trabalho 106
6.2 Trabalhos futuros 106
7 Refêrencias 108
8 Anexos 113
8.1 Anexo A – Arquivo de log 113
8.2 Anexo B – Arquivo de índices 117
8.3 Anexo C – Arquivo dos clusters 120
8.4 Anexo D – Diagrama de classes 121

1 INTRODUÇÃO

Segundo Backer (1995), agrupar objetos em categorias é uma atividade que vem sendo muito utilizada, devido ao crescente número de informações que precisam ser manipuladas pelas pessoas atualmente. Entretanto a tarefa de agrupar objetos (*em inglês clustering*) não é um conceito recente.

As pessoas já realizam este processo manualmente (ou mentalmente) há muito tempo, agrupando e categorizando os objetos conforme algum critério, como por exemplo: grupos de pessoas conhecidas e desconhecidas, grupo de pessoas do sexo masculino e feminino, entre outras. Tornando assim mais fácil a localização das informações desejadas (ALVES, 2007).

Na área da Ciência a categorização em grupos sempre desempenhou um papel essencial. No século XVIII, Linnaeus e Sauvages classificaram de forma extensiva animais, plantas, minerais e doenças (ALVES, 2007).

Na área da informática poucos anos depois que os computadores passaram a ser utilizados por órgãos governamentais americanos, os primeiros algoritmos de agrupamento de objetos já surgiram e foram implementados (WIVES, 1999).

Atualmente a técnica de agrupamento de dados está sendo cada vez mais utilizada, pois com o uso massivo dos computadores e a evolução da internet, a quantidade de informação disponível e acessível em todo o mundo cresceu muito. Essas informações muitas vezes precisam ser classificadas, ordenadas, analisadas e eventualmente agrupadas. Considerando essa questão pode-se dizer que a tarefa de agrupamento de dados, apesar de ser uma tarefa antiga, vem sendo cada vez mais difundida e utilizada, o que nos leva a uma grande motivação em mais pesquisas dentro dessa área (PIRES, 2008).

A tarefa de agrupamento visa construir grupos conforme a similaridade dos elementos. O processo de agrupamento possui cinco etapas principais: preparação de padrões, a escolha de uma medida de similaridade, o agrupamento dos dados, a validação dos grupos formados e a interpretação dos mesmos. Atualmente existem diversos algoritmos para agrupamento de dados cada um seguindo uma técnica diferente, as quais serão melhor abordadas no decorrer deste trabalho.

Apesar dos avanços nessa área ainda existem algumas questões a se responder

como, por exemplo, como formar grupos de uma maneira mais eficaz de forma que os elementos do mesmo conjunto sejam muito similares entre si. Visando essa e outras questões, este trabalho faz uma proposta de um algoritmo para agrupamento de dados com base nas teorias do Sistema Imunológico Artificial (SIA).

A área dos SIA, apesar de recente, vem sendo bastante estudada, inclusive dentro da UCS temos trabalhos que englobam essa área desde 2005. Segundo Castro (2001), os SIA estão sendo amplamente utilizados para resolução de diversos problemas computacionais e tem auxiliado na resolução de problemas complexos. Os SIA se baseiam no sistema humano dos vertebrados, originando algumas teorias, dentre elas a teoria da seleção clonal e o princípio da seleção negativa, que tem sido muito relevante na construção de novos algoritmos.

1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver um algoritmo para resolução de problemas de agrupamento de dados baseado no modelo computacional do Sistema Imunológico. Os objetivos específicos do trabalho são os seguintes:

- Realizar um estudo dos principais algoritmos dentro da área de Sistema Imunológico Artificial e das aplicações já existente para agrupamento de dados que utilizam esses algoritmos.
- Selecionar datasets públicos que serão utilizados para os testes e um algoritmo bem conhecido já implementado, que resolva a questão de agrupamento de dados o qual será comparado com o algoritmo imunológico.
- Elaborar um algoritmo imunológico com base no SIA, o qual terá como principal objetivo o agrupamento de dados dos datasets escolhidos previamente.
- Implementar o algoritmo imunológico escolhendo uma arquitetura e uma linguagem de programação apropriada.
- Realizar testes com os datasets escolhidos previamente, tanto no algoritmo imunológico como no algoritmo já conhecido pré-definido.
- Fazer uma análise dos resultados dos dois algoritmos, comparando cada um deles através de uma análise qualitativa.

1.2 ESTRUTURA DO TEXTO

O presente trabalho está estruturado em seis capítulos. O capítulo 2 apresenta o tema agrupamento de dados, definindo os principais conceitos acerca do assunto bem como os principais algoritmos dessa área. O capítulo 3 aborda os Sistemas Imunológicos Artificiais, revisando os conceitos do sistema imunológico e os principais algoritmos já definidos dentro da área de SIA. O capítulo 4 apresenta uma proposta de algoritmo imunológico para agrupar dados definindo seu formalismo e metodologia. O capítulo 5 apresenta o algoritmo imunológico sua modelagem, interface e uma análise com alguns datasets. O capítulo 6 apresenta as conclusões referentes a este trabalho, contribuições e trabalhos futuros que podem ser realizados.

2 AGRUPAMENTO DE DADOS

Agrupar dados é uma tarefa que tem sido muito utilizada, visto que a quantidade de informações que precisam ser analisadas tem aumentado nos últimos tempos. A análise de dados a partir de técnicas de agrupamento permite que se descubram comportamentos que se repetem constituindo um padrão pelo qual são formados os grupos. O objetivo principal do agrupamento é formar grupos que contenham objetos muito similares entre si dentro do mesmo grupo, e que sejam muito diferentes dos elementos de outros grupos.

O capítulo a seguir apresenta os principais conceitos da área de agrupamento de dados, desvendando as principais etapas do processo de agrupamento e alguns dos algoritmos já existentes dentro dessa área. Serão apresentadas algumas das aplicações atuais que utilizam a área de agrupamento, como a descoberta de padrões de consumo utilizada na construção de novas campanhas de marketing, no reconhecimento de usuários que utilizam as redes sociais, na categorização de genes e doenças, entre outras.

2.1 AGRUPAMENTO DE DADOS

Agrupamento de dados (*em inglês clustering*) é uma técnica de aprendizado não supervisionado utilizada para formar grupos com base em suas similaridades (JAIN e DUBES, 1988; HAN, 1996). A aprendizagem não supervisionada busca extrair informações relevantes de dados não rotulados (dados que não possuem classe pré-definida), formando assim grupos que possuam algum significado e propriedades em comum. O processo de aprendizagem deve definir o número de grupos e as características que revelam semelhanças entre cada um desses grupos (ZUBEN e MOSCATO, 2002).

Um conceito mais atual pode ser encontrado em Santos (2009), que define agrupamento de dados como sendo o nome dado ao conjunto de técnicas que tem por objetivo separar objetos em grupos. São utilizadas medidas de similaridades para definição desses grupos, assim os objetos de um grupo são muito similares entre si e muito diferentes de objetos pertencentes a outros grupos (KASNAR e GONÇALVES,

2007; PIRES, 2008).

Segundo Hruschka e Ebecken (2003), temos a seguinte definição formal para agrupamento de dados:

Considerando o agrupamento de um conjunto de *n* objetos $X = \{X_1, X_2, ..., X_n\}$, onde cada $X_i \in \Re^p$ é um vetor de *p* medidas reais que dimensionam as características do objeto, estes devem ser agrupados em grupos disjuntos $C = \{ C_1, C_2, ..., C_k\}$ onde *k* é o número de grupos. Conforme Rodrigues (2009), o valor de *k* pode ser conhecido ou não. Caso o valor de *k* seja fornecido como parâmetro para a solução, o problema é conhecido na literatura como "problema de k-agrupamento". Caso contrário, se o *k* é desconhecido, o problema é conhecido como "problema de agrupamento automático". O conjunto C é considerado um agrupamento com *k* grupos caso as seguintes condições sejam satisfeitas:

1. $C_1 \cup C_2 \cup ... \cup C_k = X$

   As características do conjunto serão definidas pela união dos objetos deste conjunto, visto que, eles foram agrupados por suas similaridades. O resultado é o grupo formado com os objetos similares.

2. $C_i \neq \emptyset, \forall\, i, 1 \leq i \leq k$

   Cada grupo tem que possuir ao menos um objeto.

3. $C_i \cap C_j = \emptyset, \forall\, i \neq j, 1 \leq i \leq k$ e $1 \leq j \leq k$

   Um objeto não pode pertencer a mais de um grupo.

O principal objetivo do agrupamento é formar conjuntos de tal modo que, os elementos do grupo tenham alguma propriedade em comum, maximizando a homogeneidade dentro de cada grupo e ao mesmo tempo maximizando a heterogeneidade entre grupos. Quanto menor o número de grupos formados mais fácil será de gerenciar os mesmos, porém deve-se tomar cuidado, pois a diminuição do número de grupos acarreta numa diminuição na homogeneidade dentro dos mesmos. Por isso, é necessário que exista um balanceamento entre o número de grupos e a similaridade entre eles. O resultado do agrupamento será um conjunto de dados "rotulados", sendo que cada rótulo será definido pelos próprios padrões que compõem cada grupo (COLE, 1998; PUNTAR, 2003; PRASS e OGLARI, 2007; NALDI, 2010).

A seguir são descritos e ilustrados quatro exemplos de agrupamento de dados: o primeiro exemplo apresenta um grupo de animais e os diversos grupos que podem ser formados pelo mesmo conforme determinadas características. No segundo exemplo tem-se um conjunto de 16 cartas de um baralho e as diferentes formas de agrupar essas

cartas conforme os padrões escolhidos. O terceiro exemplo traz um problema comum de agrupamento de dados em um plano cartesiano e a forma resultante da divisão de grupos entre os mesmos. E o último exemplo traz o agrupamento de documentos textuais.

O primeiro exemplo é ilustrado pela figura 1. Nela temos um exemplo comum de agrupamento de objetos. Muitas vezes o conjunto de objetos que devemos agrupar representa algum mecanismo já existente do mundo real. Na figura 1 pode-se verificar as diversas formas de agrupamento de um conjunto de animais. Na parte (a) temos um conjunto de objetos originais que se quer agrupar. Na parte (b) temos a primeira forma de agrupamento, conforme as características físicas desses animais. Na parte (c) temos uma segunda forma de agrupamento pelas questões climáticas. Na parte (d) temos uma terceira forma de agrupamento por classes de animais vertebrados.

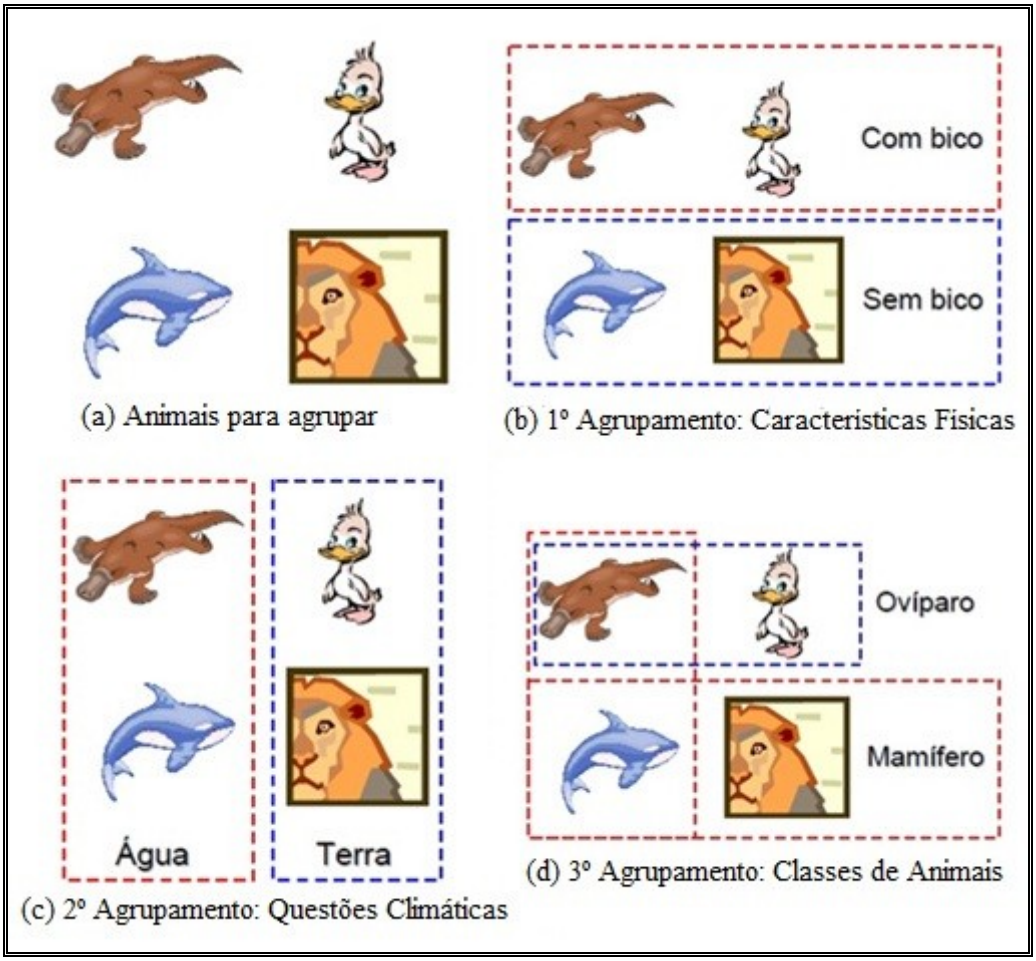


Figura 1 - Diferente formas de agrupamento para um conjunto de animais (BARANAUSKA, 2003)

Conforme exemplo demonstrado na figura 1 percebe-se que é possível dividir os objetos em grupos diversos. Tudo vai depender dos critérios de semelhança que estamos levando em consideração para cada situação e também que um objeto poderá pertencer

a mais de um grupo quando o algoritmo que gera os grupos permitir (métodos probabilísticos), como no caso da figura 1 parte (d), onde o animal ornitorrinco pode pertencer tanto ao grupo de animais ovíparos como ao grupo de animais mamíferos.

No segundo exemplo é demonstrada a natureza da dificuldade na definição de grupos naturais. Considere a ordenação de 16 cartas figuradas de um baralho convencional em grupos similares. Existe uma maneira de formar um único grupo com 16 cartas figuradas. Existem 32.767 maneiras de dividir as cartas figuradas em dois grupos (de tamanhos variados). Existem 7.141.686 maneiras de ordenar as cartas figuradas em três grupos (de tamanhos variados), e assim por diante. Na figura 2 podemos ver algumas maneiras de se agrupar essas cartas conforme algumas características pré-definidas. Na parte (a) temos as cartas individuais que devem ser agrupadas. Na parte (b) temos um agrupamento por naipe, na parte (c) temos um agrupamento pela cor do naipe, na parte (d) temos um agrupamento por cartas maiores e menores e na parte (e) temos um agrupamento por face da carta (KASNAR e GONÇALVES, 2007).

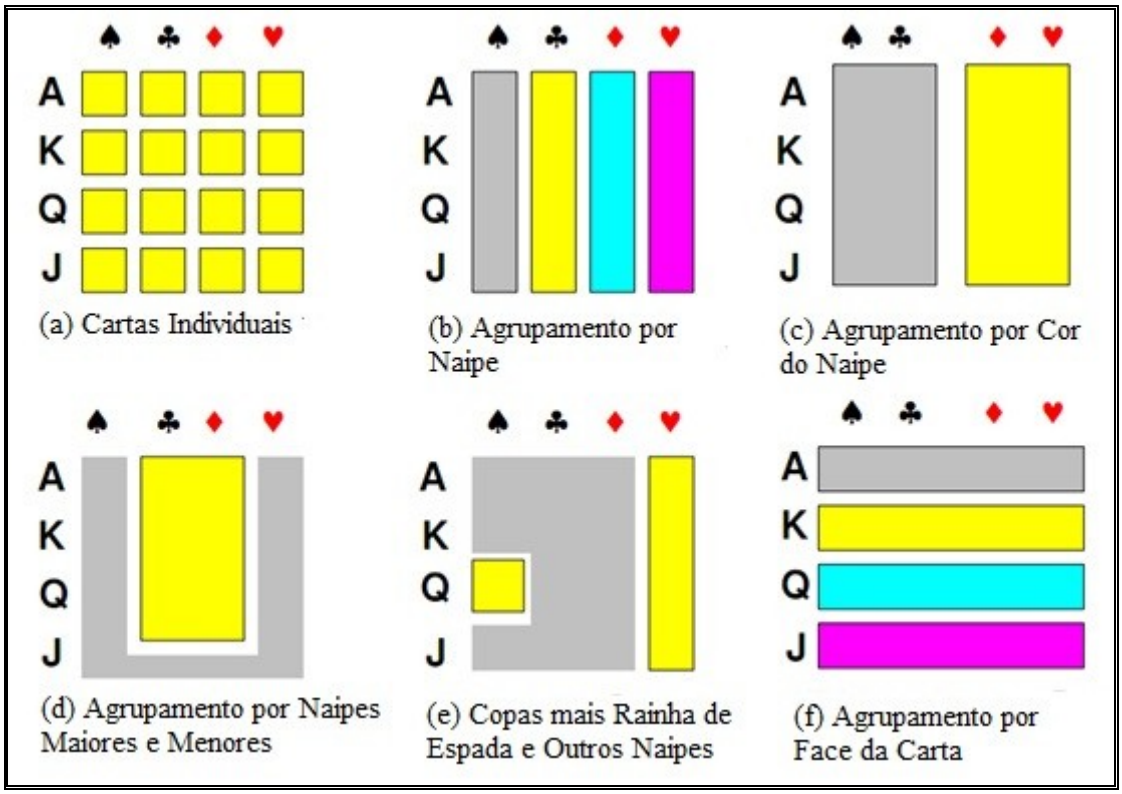


Figura 2 - Diversas formas no agrupamento de cartas de baralho (KASNAR e GONÇALVES, 2007)

Assim percebe-se que o agrupamento será realizado conforme os critérios que serão levados em consideração e que podem existir diversos grupos distintos para um

mesmo conjunto de objetos.

Na figura 3 é apresentado um exemplo ilustrativo de um agrupamento. Na parte (a) é apresentado um conjunto de dados de entrada, onde cada elemento é representado pelo símbolo *x*. Na parte (b) é apresentado o resultado do agrupamento realizado sobre esse conjunto de dados de entrada, onde cada elemento *x* foi rotulado com um número identificador.

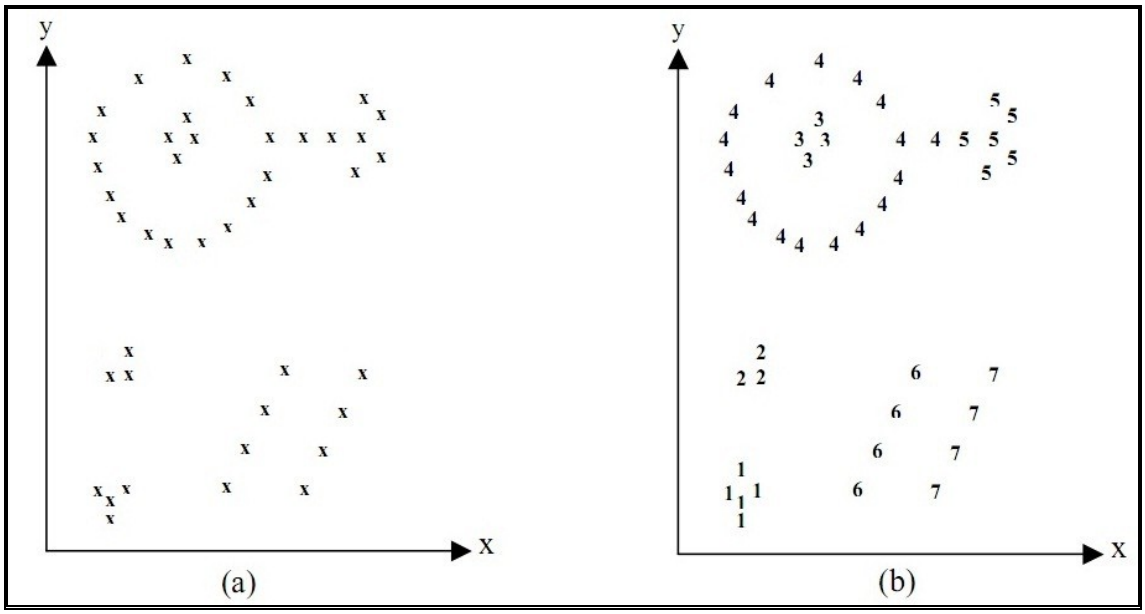


Figura 3 - Formas de agrupamento para um conjunto de objetos no plano cartesiano (JAIN et al, 1999)

Percebe-se na figura 3, que os dados que pertencem ao mesmo grupo apresentam o mesmo número, e que para formar esses grupos foi levada em consideração a distância de cada um dos objetos do plano cartesiano, ou seja, elementos mais próximos pertencerão ao mesmo grupo.

No último exemplo é apresentado um agrupamento de documentos textuais. O objetivo do agrupamento de informações textuais é separar uma série de documentos dispostos de forma organizada em um conjunto de grupos que contenham documentos de assuntos similares. Na figura 4 é ilustrada a forma de separação de um conjunto de documentos por assunto.

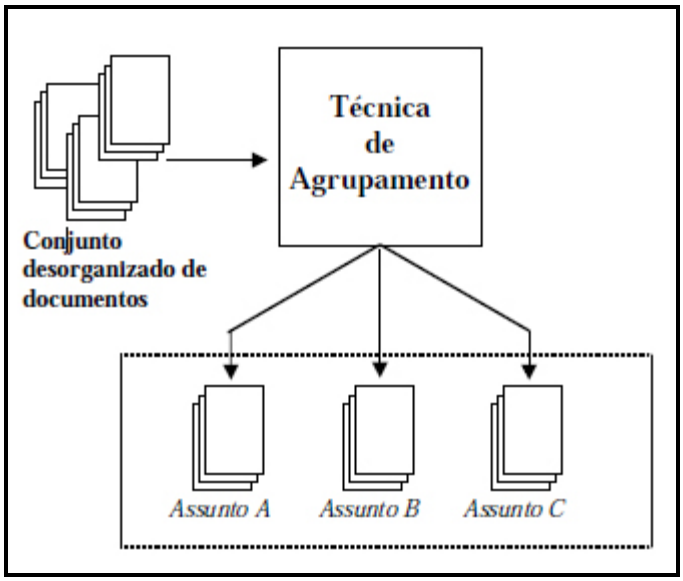


Figura 4 - Agrupamento de documentos textuais (WIVES, 1999)

O agrupamento é feito a partir do assunto tratado em cada objeto. Todos os objetos que possuírem o assunto A, ficarão no primeiro grupo, os que pertencem ao assunto B ficarão no segundo grupo e, se pertencerem ao assunto C, ficarão no terceiro grupo.

2.2 ETAPAS DO PROCESSO DE AGRUPAMENTO

Algumas propriedades básicas em um processo de agrupamento de dados são as seguintes (HAN e KAMBER, 2001, AZEVEDO, 2010):

- Escalabilidade.
- Habilidade em manipular diferentes tipos de dados.
- Requisitos mínimos no conhecimento do domínio do problema para definir os parâmetros de entrada do algoritmo.
- Insensibilidade à ordem dos registros de entrada.
- Permitir a incorporação de restrições específicas do usuário.
- Usabilidade e interpretabilidade.
- Descoberta de grupos com formas arbitrárias.
- Capacidade de tratar dados com ruídos.
- Alta dimensionalidade.

Segundo Bordigon (2010), o processo de agrupamento de dados é definido como

as etapas executadas para encontrar grupos conforme as informações pré-definidas. Existem cinco etapas no processo de agrupamento de dados: preparação dos padrões, seleção de medida de similaridade, execução de um algoritmo de agrupamento, validação dos resultados do agrupamento e interpretação dos grupos identificados.

A figura 5 foi desenvolvida a partir de leituras em (JAIN e DUBES,1988; JAIN et al, 1999; ZUBEN e MOSCATO, 2002; FACELI, 2007; MELLO, 2008; OLIVEIRA, 2008; NALDI, 2010); Nela é ilustrada a sequência dessas etapas que serão descritas detalhadamente nas próximas seções.

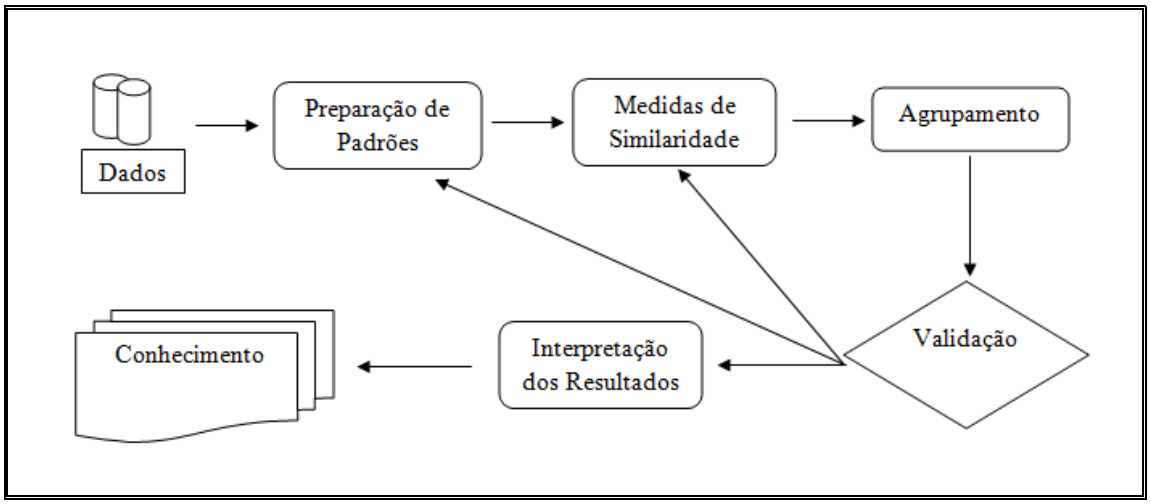


Figura 5 - Etapas de um processo de agrupamento

2.3 PREPARAÇÃO DE PADRÕES

A etapa de preparação de padrões compreende as seguintes atividades: representação de objetos, normalização e extração/seleção de características conforme ilustradas na figura 6 elaborada a partir de Rocha e Lachi (2005) e Naldi (2010).

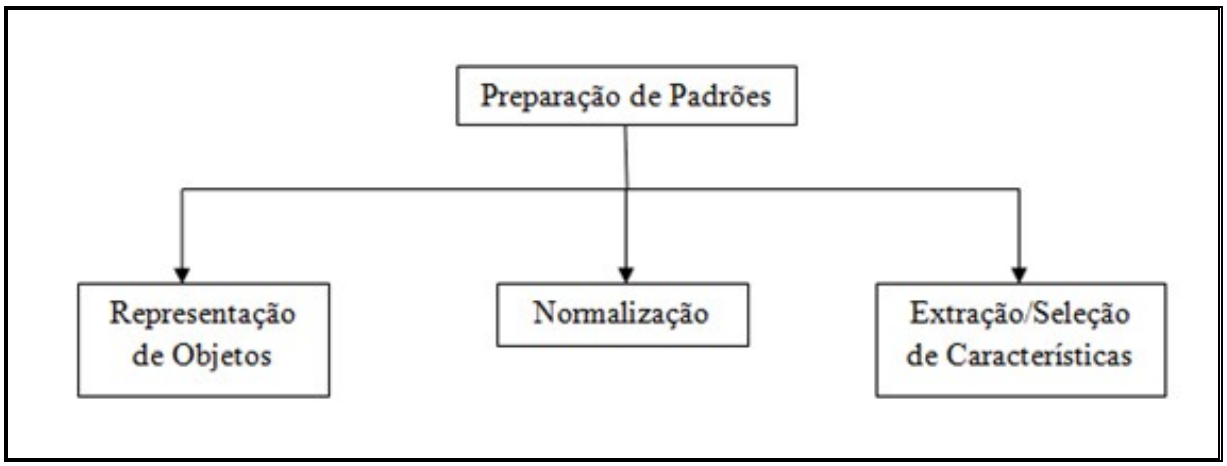


Figura 6 - Atividades realizadas na etapa de preparação de padrões

Um conjunto de dados é formado por objetos também conhecidos como padrões ou instâncias. A atividade de representação dos objetos determina como os objetos serão representados no conjunto de dados durante o processo de agrupamento (NALDI, 2010).

Um objeto *x* é um dado usado pelos algoritmos de agrupamento. Ele geralmente é representado por um vetor de *d* características sendo: $x = \{x_1, x_2, ..., x_d\}$. Cada componente escalar x_1, presente no vetor de características do objeto *x* é denominado característica do atributo *i* do objeto *x*. Além disso, o valor *d* presente na definição de *x* é denominado a dimensão do objeto, ele representa o número de características que o objeto *x* possui. Na figura 7 é apresentado um esquema de um objeto *x* representado por um vetor com duas características x_1 e x_2 (ROCHA e LACHI, 2005).

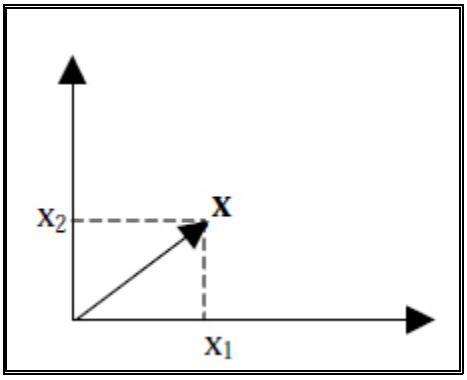


Figura 7 - Representação de um objeto x no plano cartesiano com duas características x_1 e x_2 (ROCHA e LACHI, 2005)

Se existir um conjunto de objetos, este será denotado por *X,* logo, tem-se que $X = \{x_1, ..., x_n\}$ onde cada objeto x_i pertence a este conjunto, representa, por sua vez um vetor de características $x_i = (x_1, x_2, ..., x_d)$ de dimensão *d*. Consequentemente, o conjunto de objetos *X* pode ser visualizado como uma matriz *n x d.* Na figura 8 é ilustrado um exemplo dessa situação onde são representados dois objetos x_i e x_j. O objeto x_i possui características x_2' e x_2'', e o objeto x_j possui as características x_1' e x_1''. A união dos dois objetos x_i e x_j forma o conjunto de objetos X (ROCHA e LACHI, 2005).

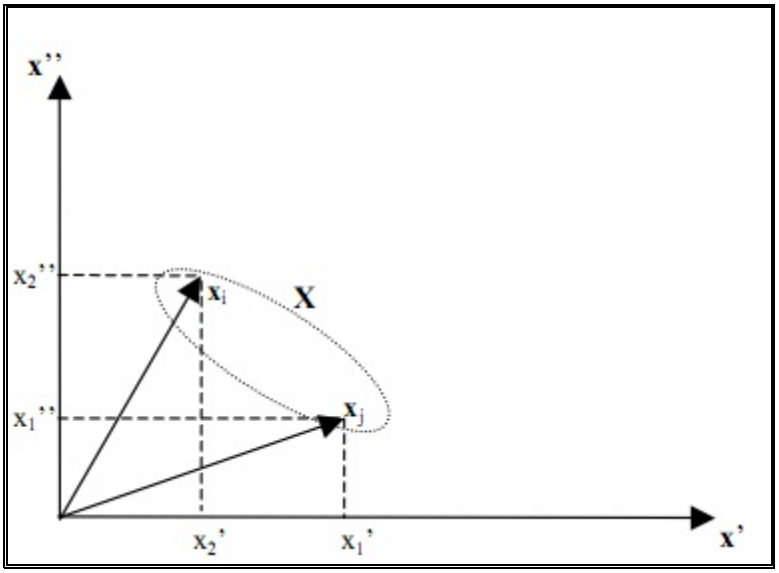


Figura 8 – Representação de dois objetos que formam o conjunto de objetos X (ROCHA e LACHI, 2005)

Os objetos a serem apresentados são geralmente descritos como vetores de características, que podem ser físicas, como cor e peso, ou abstratas, como tipo de sorriso. E pode-se dividi-los inicialmente em dois tipos básicos: qualitativos ou quantitativos (ROCHA e LACHI, 2005).

Atributos qualitativos são utilizados para expressar qualidade dos objetos, enquanto atributos quantitativos expressam quantidade, relações ou medidas. Um exemplo é ilustrado na figura 9 com um vetor de duas características quantitativas (altura e idade) representando uma pessoa com 1.75m de altura e 18 anos (ROCHA e LACHI, 2005; NALDI, 2010).

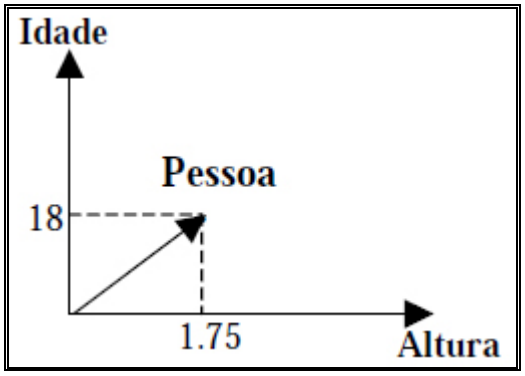


Figura 9 - Representação de uma pessoa com atributos quantitativos (ROCHA e LACHI, 2005)

Outro exemplo é visto a seguir. Para agrupar um conjunto de pessoas de acordo com os dados da Tabela 1, cada objeto é definido pelas características de uma pessoa.

Portanto, para as pessoas Mariana, José e Carlos teremos respectivamente os seguintes vetores: X = (Mariana, Jose, Carlos) e cada um desses terá um vetor com suas três características. Mariana = (1.73, 65, 23), Jose = (1.78, 85, 47) e Carlos = (1.82, 95, 24) (MELLO, 2008).

Tabela 1 - Dados quantitativos de pessoas a serem agrupadas

Objetos	Idade	Peso	Altura
Mariana	23	65	1.73
José	47	85	1.78
Carlos	24	95	1.82

Pode-se utilizar um número maior de características para representação de cada objeto. No entanto, é recomendado que o projetista procure usar o mínimo necessário de características para o seu problema, pois isso permitirá um agrupamento mais eficiente e mais simples de ser inspecionado por seres humanos (ROCHA e LACHI, 2005).

O conhecimento do tipo da característica a ser representada é muito importante na escolha da medida de similaridade e dos algoritmos a serem empregados para definir um bom agrupamento. Objetos bem representados podem permitir facilmente a interpretação dos grupos formados, enquanto uma representação complexa das características pode tornar essa interpretação complexa e difícil de ser compreendida. Por este motivo, às vezes, é necessário efetuar a transformação das características, a qual é feita através de um processo chamado de normalização, que tem como objetivo realizar transformações nos dados para que estes estejam em escala adequada para os métodos de agrupamento (OLIVEIRA, 2008; NALDI, 2010).

A normalização é importante, pois algumas vezes, os objetos apresentam características de escalas diferentes ou a representação dos dados não é adequada para o algoritmo de agrupamento escolhido. Para que as medidas de similaridade funcionem corretamente é necessário que todas as variáveis estejam dentro do mesmo intervalo. Isso evita que as unidades de medida influenciem o agrupamento dos objetos. Portanto, é necessário que a padronização ou normalização das variáveis seja realizada antes do processo de agrupamento (JAIN e DUBES 1988; HAN e KAMBER, 2001; NALDI, 2010).

Nem sempre todas as características dos objetos são de interesse para o agrupamento e podem apresentar problemas para os algoritmos, como a falta de valores ou erros na obtenção destes valores. Na tentativa de amenizar estes problemas pode-se utilizar a seleção e/ou extração de características desses dados (OLIVEIRA, 2008).

Na seleção de características identifica-se o subconjunto das características originais que seja mais apropriado para descrever cada uma delas. Já a extração de características define-se o uso de uma ou mais transformações nas características originais para produzir novas características. A utilização desses métodos possibilita encontrar um conjunto de características que melhor represente as similaridades entre os objetos (OLIVEIRA, 2008; NALDI, 2010).

Como resultado dessa etapa, temos a representação dos dados, que envolve a definição do número, tipo e modo de apresentação dos atributos que descrevem cada objeto (OLIVEIRA, 2008; NALDI, 2010).

2.4 MEDIDAS DE SIMILARIDADE

A segunda etapa do processo de agrupamento é a escolha de uma medida de similaridade. Essa medida é calculada, em geral, por meio de uma função de similaridade definida entre pares de objetos (OLIVEIRA, 2008).

Similaridade é um conceito fundamental para o processo de agrupamento, pois se dois objetos são similares de acordo com algum critério utilizado pela técnica de agrupamento, então serão agrupados em um mesmo grupo, caso contrário, serão agrupados em grupos diferentes (CARLNATONIO, 2001; ROCHA e LACHI, 2005).

A medida de similaridade tem por objetivo mensurar o quanto dois objetos são semelhantes. Para escolha dessa medida é importante levar em consideração alguns fatores como: a natureza das variáveis (discreta, contínua, binária), as escalas de medida (nominal, ordinal, intervalo) e o conhecimento específico do assunto (KASZNAR e GONÇALVES, 2007; MELLO, 2008).

Segundo Silva (2006) as medidas podem ser divididas em dois tipos, de acordo com a natureza dos valores dos atributos dos objetos: medidas de similaridade para dados numéricos e medidas de similaridade para dados categorizados.

2.4.1 MEDIDAS DE SIMILARIDADE PARA DADOS NUMÉRICOS

As medidas de similaridade consideram cada objeto x como um vetor no espaço n-dimensional, onde n corresponde à quantidade de características de cada objeto x. De acordo com essa interpretação, medidas geométricas de distância podem ser utilizadas

para determinar a similaridade entre os objetos a serem agrupados (SILVA, 2006).

A seguir são apresentadas as três principais métricas para cálculo de distância entre objetos: distância euclidiana, distância euclidiana quadrática e distância de manhattan.

2.4.1.1 DISTÂNCIA EUCLIDIANA

A distância euclidiana, cujo o cálculo é representado na fórmula 1 é a mais popular e amplamente utilizada. Foi definido pelo matemático grego Euclides e representa a menor distância existente entre dois objetos no plano multidimensional. Sua principal aplicação é no algoritmo de agrupamento k-médias (ROCHA e LACHI, 2005).

$$D_{xy} = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

Equação 1 - Distância euclidiana

Onde:

- D_{ij} é a distância Euclidiana do objeto i para o objeto j.
- *l* é o índice do atributo do vetor de d atributos dos objetos.
- x_{il} é o l-ésimo atributo do objeto i.
- x_{jl} é o l-ésimo atributo do objeto j.

Um exemplo desta fórmula é ilustrado na figura 10. Considere dois objetos x e y com os seguintes valores respectivamente (6,2) e (3,1). Qual é a distância entre os objetos x e y?

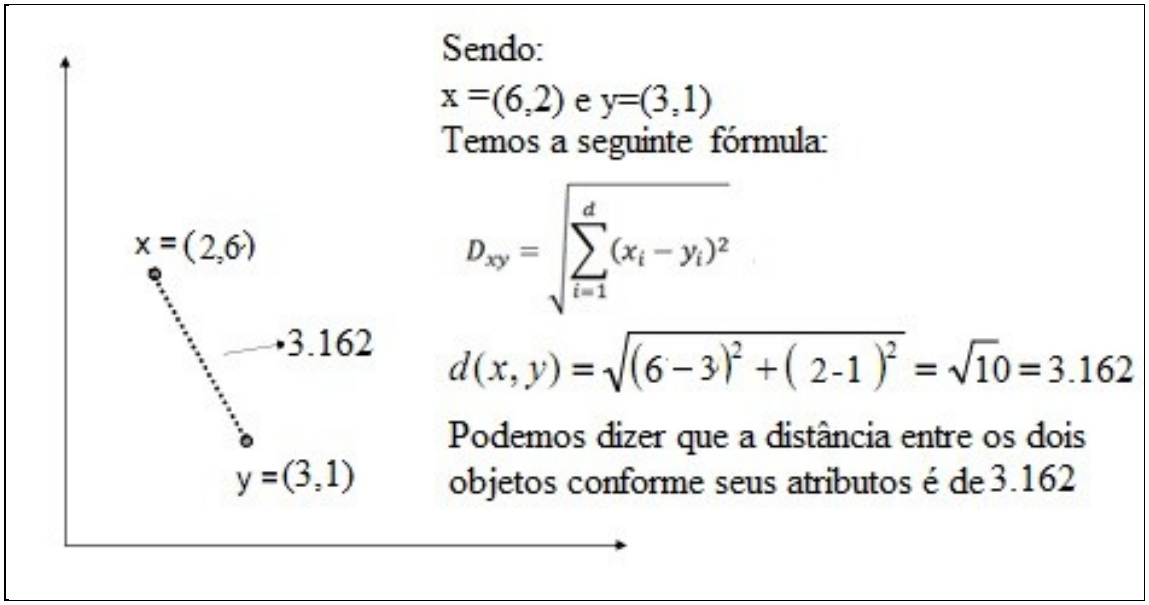


Figura 10 - Resultado da aplicação da fórmula de distância euclidiana (OLIVEIRA, 2005)

Pela figura 10 vemos que a distância entre os objetos x e y é de 3.162, conforme a distância euclidiana. Essa forma de calcular a similaridade entre padrões é bastante intuitiva. No entanto, essa métrica tem tendência de fazer com que as características que tenham os maiores valores dominem o resultado. Esse problema pode ser facilmente resolvido utilizando-se esquemas de normalização sobre os valores das características, ou então, aplicando pesos diferentes para cada uma das características (RODRIGUES, 2009).

2.4.1.2 DISTÂNCIA EUCLIDIANA QUADRÁTICA

A distância euclidiana quadrática serve como um artifício para aumentar o peso dos objetos mais distantes, ressaltando a diferença entre os grupos. Ela é representada por $X = (x_1, x_2, x_3,..., x_n)$ e $Y = (y_1, y_2, y_3,..., y_n)$ e definida na fórmula 2:

$$d_{xy} = (x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_p - y_p)^2 = \sum_{i=1}^{p}(x_i - y_i)^2$$

Equação 2 - Distância euclidiana quadrática

Na figura 11 temos um exemplo da aplicação da distância euclidiana quadrática para dois objetos x e y com os seguintes valores respectivamente (6,2) e (3,1). Qual é a

distância entre os objetos x e y?

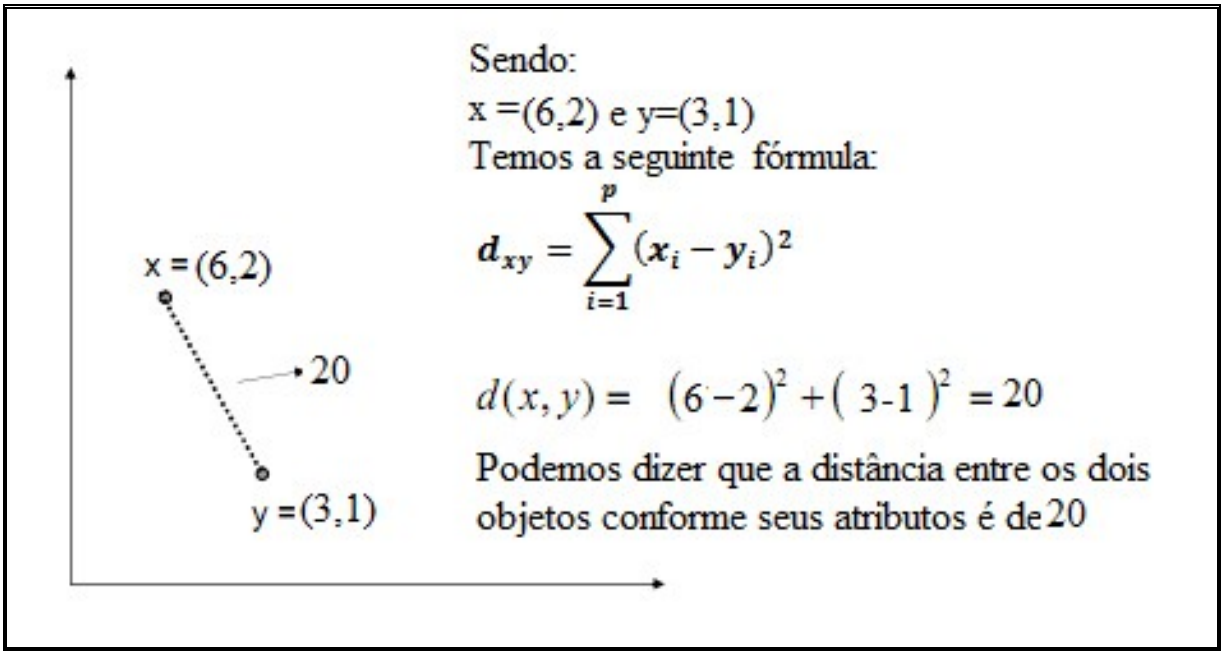


Figura 11 - Resultado da aplicação da fórmula de distância euclidiana quadrática (OLIVEIRA, 2005)

Podemos verificar que neste caso para os mesmos valores aplicados na figura 10 o resultado aumentou muito. Esta fórmula deve ser utilizada para objetos mais distantes.

2.4.1.3 DISTÂNCIA DE MANHATTAN

Segundo Almeida (2004), a Distância Manhattan (*em inglês City Block*) definida por Hermann Minkowski é também uma métrica bem conhecida para calcular a distância entre cada par de objetos i e j. Para utilizar esta métrica aplica-se a Fórmula 3.

$$D_{xy} = \sum_{l=1}^{n} |x_{il} - y_{jl}|$$

Equação 3 - Distância de manhattan

Onde:

- D_{xy} é a distância de Manhattan do objeto x para o objeto y.
- *l* é o índice do atributo do vetor de d atributos dos objetos.
- x_{il} é o l-ésimo atributo do objeto x.
- y_{jl} é o l-ésimo atributo do objeto y.

Na figura 12 temos um exemplo da aplicação da distância de manhattan para dois objetos x e y, com os seguintes valores respectivamente, (6,2) e (3,1). Qual é a distância entre os objetos x e y?

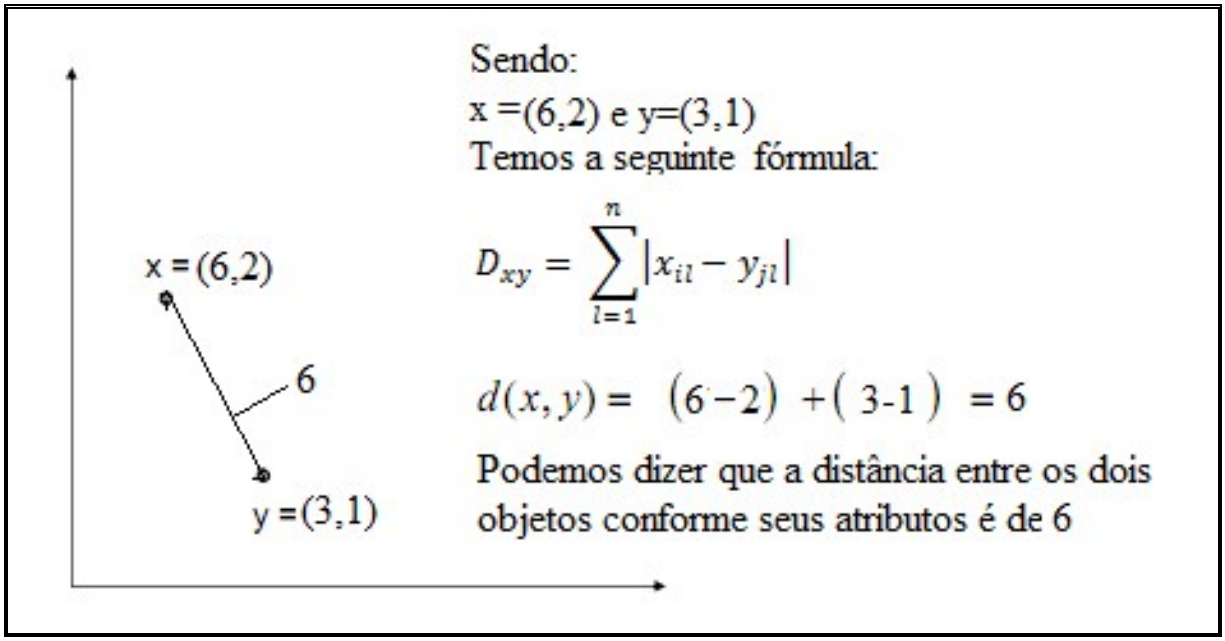


Figura 12 - Resultado da aplicação da fórmula de distância manhattan (OLIVEIRA, 2005)

A tabela 2 apresenta um resumo das medidas apresentadas, bem como um exemplo para cada uma delas. A tabela a seguir considera dois objetos x e y sendo x=(6,2) e y=(3,1).

Tabela 2 - Exemplo de cálculo de distância para dados numéricos

Métrica	Fórmula	Exemplo
Distância Euclidiana	$D_{ij} = \left(\sum_{l=1}^{d} \lvert x_{il} - x_{jl} \rvert^2 \right)^{1/2}$	$D_{ij} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$ $D_{ij} = \sqrt{(6-2)^2 + (3-1)^2}$ $D_{ij} = \sqrt{9^2 + 1^2}$ $D_{ij} = 3.162$
Distância Euclidiana Quadrática	$D_{xy} = \sum_{i=1}^{p} (x_i - y_i)^2$	$D_{xy} = (x_1 - y_1)^2 + (x_2 - y_2)^2$ $D_{xy} = (6-2)^2 + (3-1)^2$ $D_{xy} = 4^2 + 2^2$ $D_{xy} = 20$
Distância de Manhattan	$D_{xy} = \sum_{l=1}^{n} \lvert x_{il} - y_{jl} \rvert$	$D_{xy} = (x_1 - y_1) + (x_2 - y_2)$ $D_{xy} = (6-2) + (3-1)$ $D_{xy} = 4 + 2$ $D_{xy} = 6$

A tabela 2 descreve os três cálculos apresentados. Conforme o resultado dos exemplos, percebe-se que a distância euclidiana quadrática define grupos mais distantes

uns dos outros enquanto a distância euclidiana define grupos mais próximos. A distância de Manhattan, possui um cálculo mais simples e seria um valor intermediário entre as duas outras medidas.

2.4.2 MEDIDAS DE SIMILARIDADE PARA DADOS CATEGORIZADOS

Segundo Rodrigues (2009) existem problemas de agrupamento em que a distância não pode ser utilizada como medida de similaridade, tendo em vista que os valores dos atributos não são numéricos. Como exemplo, ao tratar de um problema de agrupamento que envolve atributos como sexo e endereço, são necessárias outras medidas que demonstrem o grau de similaridade entre os objetos. Esses dados são ditos dados categorizados e podem ser de dois tipos: nominais ou ordinais.

Uma característica nominal é aquela em que seus valores não estão associados a uma ordem. Por exemplo, uma variável representando cor pode ter valores como verde, azul, marrom, preto e branco. Já em características ordinais, uma ordenação dos valores é necessária. Um exemplo é dado no atributo de hierarquia de cargos: diretor, coordenador, operador (BOSCARIOLI, 2008; RODRIGUES, 2009).

Um exemplo de atributos categorizados pode ser visto na tabela 3, que apresenta um conjunto de objetos com características de valores categorizados. Tendo cor dos olhos e religião como categoria nominal e estatura, sexo e grau de instrução como categoria ordinal.

Tabela 3 - Conjunto de objetos com características de valores categorizados

Pessoa	Cor dos Olhos	Religião	Sexo	Estatura	Grau de Instrução
João	Verde	Católica	M	Alta	Nível-fundamental
Maria	Azul	Budista	F	Alta	Nível médio
José	Verde	Católica	M	Baixa	Nível médio
Pedro	Castanho	Muçulmana	M	Alta	Nível médio
Ana	Azul	Católica	F	Mediana	Graduação
Paulo	Castanho	Ateu	M	Baixa	Mestrado

A tabela 4 resume as principais medidas para tipos de dados categorizados: são elas Jaccard, Overlap, Dice e Co-seno. Nas expressões correspondentes a cada uma dessas métricas, V1 e V2 são conjuntos de valores de características de dois objetos, e Card(V) corresponde a cardinalidade do conjunto V, ou seja, a quantidade de elementos que pertencem ao conjunto.

Tabela 4 - Métricas de similaridade para valores categorizados

Jaccard	$Card\ (V1 \cap V2\ ) / Card\ (V1 \cup V2)$
Sobreposição (Overlap)	$Card(V1\ V2) / \min(Card\ (V1), Card(V2))$
Co-seno	$Card(V1 \cap V2) / \sqrt{Card(V1) x\ Card\ (V1)}$
Dice	$(2\ x\ Card\ (V1 \cap V2) / (Card\ (V1) + Card\ (V2))$

Para um exemplo da aplicação de Jaccard consideram-se dois objetos da tabela 3: João e Maria. Temos um X com as seguintes características X = {cor dos olhos, religião, sexo, estatura e nível de ensino}. Sendo João o V1 e tendo as seguintes características (Verde, Católica, M, Alta, Nível Fundamental) e Maria sendo V2 tendo as características (Azul, Budista, F, Alta, Nível Médio). Aplicando a fórmula tem-se:

J(V1,V2) = Card (Joao ∩ Maria) / Card (Joao U Maria)
J(V1,V 2) = Card (cor dos olhos, religião,sexo, nivel de ensino) / (estatura)
J(V1,V2) = 1/9
J(V1,V2) = 0,1111

Assim a distância entre João e Maria é de 4 ou seja se o β definido for menor que 4 os objetos, João e Maria pertencem ao mesmo grupo, senão eles pertencerão a grupos distintos.

Segundo Metz (2006) não existe uma regra geral que define qual métrica de distância deve ser utilizada em determinada situação. A escolha por uma determinada métrica se dá, geralmente, após a realização de vários testes com diferentes métricas. O resultado dessa etapa será a escolha de uma medida de similaridade que mais se adapte aos dados que irão ser agrupados.

2.5 AGRUPAMENTO

Após a escolha da medida de similaridade é realizado o agrupamento, que consiste na técnica de formar grupos através de um algoritmo (NALDI, 2010).

Segundo Rodrigues (2009) o problema de agrupamento apresenta uma complexidade de ordem exponencial. Um exemplo desse crescimento é dada pela seguinte equação. Para combinar 10 elementos em 2 grupos, 100 elementos em 2 grupos e 1000 elementos em 2 grupos, tem-se os seguintes resultados:

- N(10,2) = 511 formas para agrupar.
- N(100,2) = 6, 332825 x 10^{29} formas para agrupar.
- N(1000,2) = 5.3575x10^{300} formas para agrupar.

Isto quer dizer que algoritmos de força bruta, como enumerar todos os possíveis grupos e escolher a melhor configuração, não são viáveis, pois demorariam uma

eternidade para serem resolvidos. Para resolver este problema temos então os métodos de agrupamento (RODRIGUES, 2009).

Segundo Faceli (2007), métodos de agrupamento são instrumentos valiosos na análise exploratória dos dados, pois permitem a construção de importantes ferramentas para as quais existe pouco ou nenhum conhecimento prévio. A escolha errônea de um tipo de algoritmo pode levar a resultados incorretos ou sem sentido. Cabe salientar que os algoritmos existentes não dão conta, simultaneamente, de todos esses requisitos de forma efetiva. A escolha por aplicar um determinado algoritmo depende do tipo de dados que se pretende explorar, da aplicação e de seus objetivos particulares (PRASS e OGLARI, 2004; BOSCARIOLI, 2008).

Segundo Naldi (2010) os métodos de agrupamento são divididos em cinco categorias principais: métodos particionais, métodos hierárquicos, métodos por densidade, métodos por modelo e métodos por grade. Cada um desses métodos possui algoritmos próprios para resolver a questão de agrupamento. Os métodos e alguns dos seus algoritmos são ilustrados na figura 13 elaborada a partir de Zuben e Moscato (2002), Rocha e Lachi (2005), Pereira (2007), Rodrigues (2009), Metz (2006) e são tratados na próxima seção.

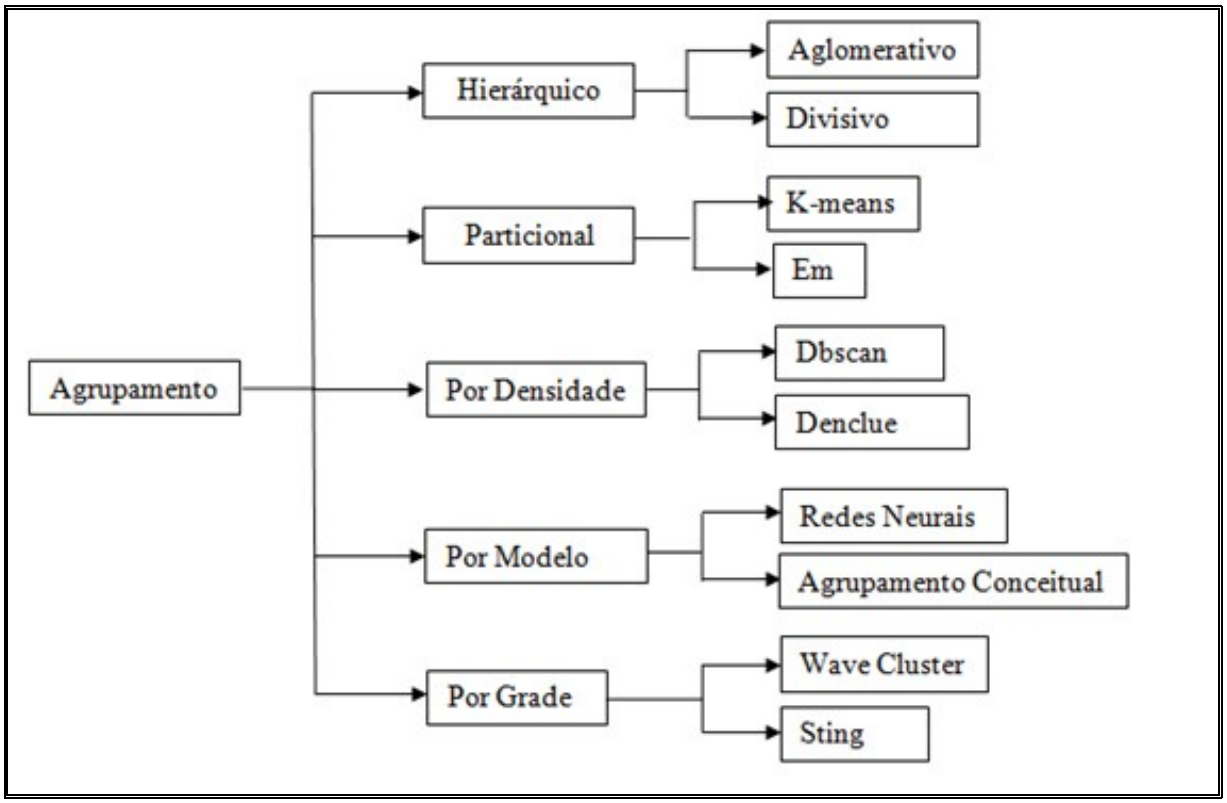


Figura 13 - Métodos de agrupamento e seus respectivos algoritmos

2.5.1 MÉTODOS HIÉRÁRQUICOS

Os métodos hierárquicos realizam sucessivas uniões ou divisões, criando uma hierarquia de classes dos objetos. O agrupamento hierárquico busca unir grupos menores formando assim, grupos maiores, ou dividir grupos grandes em outros de maior similaridade interna. O resultado do algoritmo é um gráfico tipo árvore chamado de "dendrograma", que mostra como os grupos são relacionados. O método de agrupamento hierárquico é, na maioria das vezes, utilizado em análises exploratórias dos dados com o intuito de identificar possíveis agrupamentos (ALVES, 2007; DAMASO, 2008; SAMPAIO, 2008; MELLO, 2008).

Os métodos de agrupamento hierárquicos são divididos conforme a estrutura de classes criadas e podem ser de dois tipos: aglomerativos ou divisivos (HAN e KAMBER, 2001).

Os métodos hierárquicos aglomerativos (*bottom-up)* iniciam com cada objeto do conjunto de dados representando um grupo. A cada passo o algoritmo une dois grupos que são mais similares. O algoritmo prossegue até que todos os grupos tenham sido unidos em um grupo final, ou até que um número total de grupos desejado tenha sido alcançado ou a distância entre os dois grupos mais próximos esteja acima de um limite estipulado. O critério para unir dois grupos é baseado em alguma medida de similaridade, que geralmente é calculada através da distância entre os centros dos grupos (HAN e KAMBER, 2001; PEREIRA 2007; SAMPAIO, 2008).

Existem vários métodos de agrupamentos hierárquicos aglomerativos. Os principais (mais comuns e disponíveis na maioria dos softwares) são: método de ligação simples (*Single Link*), método de ligação completa (*Complete Link*), método da medida das distâncias (*Average Link*), método do centróide (*Centroid Method*) e método de ward (SAMPAIO, 2008).

Os métodos hierárquicos divisivos (*top-down*) iniciam com um único grupo com todos os objetos do conjunto de dados. Este grupo é subdividido em dois subgrupos de tal forma que exista o máximo de semelhança entre os objetos do mesmo grupo e o máximo de diferença entre objetos de grupos diferentes. Estes subgrupos são posteriormente subdivididos em outros subgrupos distintos. Este processo será repetido até que haja tantos subgrupos quantos objetos a serem agrupados ou até que algum outro critério pré-definido seja alcançado (ALVES, 2007; PEREIRA, 2007). Como algoritmos desse tipo temos BIRCH proposto por ZHANG et al(1996) e o CURE

proposto por GUHA, RASTOGI e SHIM (1998).

Algumas das vantagens dos métodos hierárquicos são a simplicidade, o uso de diferentes medidas de similaridades e a rapidez. E suas desvantagens são redução dos impactos dos pontos fora de padrão (*outliers*), e não são bons para grandes conjuntos de dados (CASTRO et al, 2010).

2.5.2 MÉTODOS PARTICIONAIS

Dado um conjunto de dados de tamanho n e o número de grupos k, um algoritmo de particionamento constrói k partições de dados, onde cada uma destas partições representa um grupo de objetos, sendo $k \leq n$. Nessas k partições, cada grupo deve conter pelo menos um objeto e cada objeto deve pertencer apenas a um grupo. O número de partições k é um parâmetro de entrada para os métodos de particionamento. Com esse valor, são construídas k partições iniciais e através de uma técnica de realocação iterativa o algoritmo tenta melhorar o esquema de partições, movimentando os objetos de um grupo para o outro. De maneira geral, o critério adotado para orientar o algoritmo é alocar objetos similares no mesmo grupo e, objetos distintos em grupos diferentes (RODRIGUES, 2009; OLIVEIRA, 2008; MELLO, 2008; PIRES, 2008).

É necessário avaliar todas as possíveis combinações de partições para encontrar um valor ótimo global para a qualidade do agrupamento. No entanto, percorrer todas as partições é inviável mesmo para bancos de dados de tamanho razoável, pois é um procedimento muito demorado. Para evitar isso, muitos métodos trabalham com heurísticas para a escolha das partições (HAN e KAMBER, 2001; MELLO, 2008).

Duas das principais vantagens dos métodos particionais são, que um item pode mudar de agrupamento com a evolução do algoritmo e, há possibilidade de se operar um maior conjunto de dados, pois os métodos particionais são rápidos de serem processados, visto que eles não guardam na memória a matriz de similaridade durante seu processamento (MAXIMILIANO e CORDEIRO, 2008).

A principal desvantagem destes métodos é que eles não trabalham bem em grupos de formas complexas e tamanhos diferentes (PEREIRA, 2007).

Segundo Pereira (2007) os dois algoritmos mais populares são o k-médias e o EM.

O algoritmo K-médias *(K-means),* foi o primeiro e mais famoso algoritmo de

agrupamento desenvolvido por Hartingan (1975). Atualmente é um dos mais conhecidos e utilizados, por isso é uma referência contra o qual os outros algoritmos são comparados. Seu objetivo é encontrar a melhor divisão de *P* dados em *K* grupos, de maneira que a distância total entre os dados de um grupo e o seu respectivo centro, somada por todos os grupos, seja minimizada. Para isso, ele agrupa dados em um número de grupos pré-definido, e os grupos são representados por um vetor centróide, que representa o ponto central daquele grupo (PEREIRA, 2007; DAMASO, 2008, LENZ, 2009, NALDI, 2010). Conforme Rodrigues (2009), o algoritmo é descrito pelos seguintes passos e ilustrado na figura 14.

1. Dividir o conjunto de dados em k grupos e definir os centróides de cada um desses grupos.
2. Cada elemento do conjunto de dados é comparado com cada centróide definido, através de uma medida de distância, por exemplo, a distância euclidiana. O elemento é alocado ao grupo cuja distância é a menor.
3. Recalcula-se o centróide de cada grupo, redefinindo cada um, em função dos atributos de todos os objetos pertencentes ao grupo.
4. Repetir os passos 2 e 3 até que os dados se estabilizem, ou seja, os centróides não mudem ou até o número máximo de iterações pré-definido .

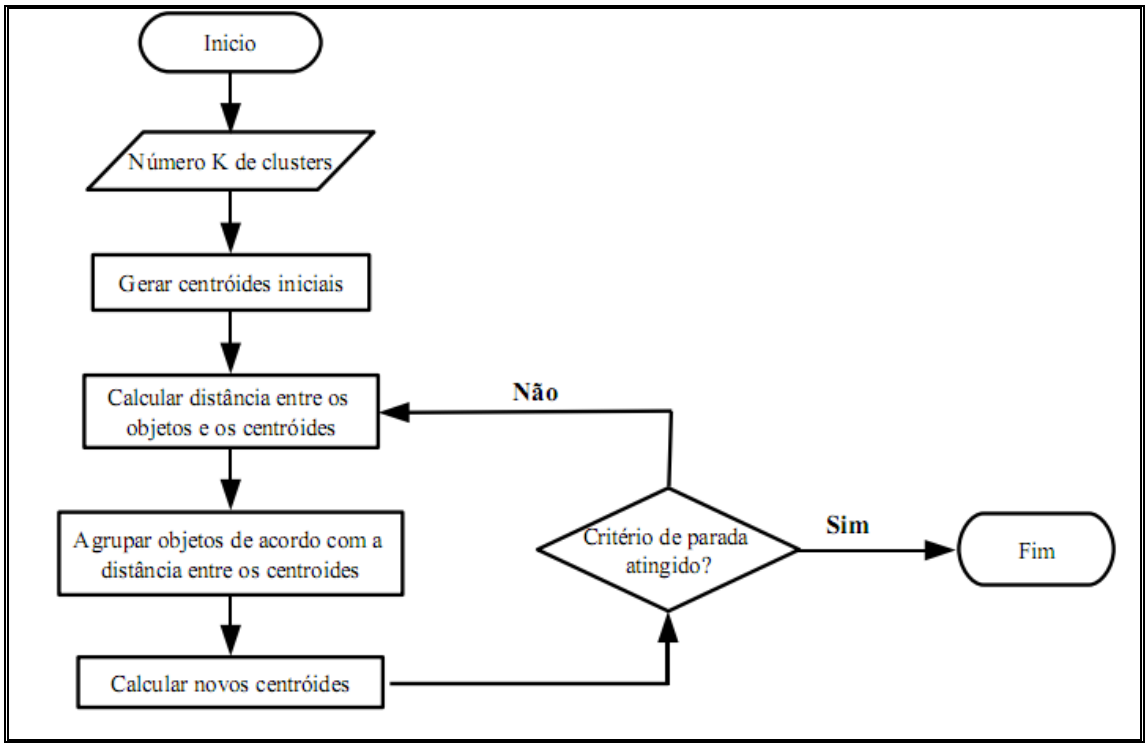


Figura 14 - Fluxograma do algoritmo k-médias

A figura 15 traz uma ilustração desse algoritmo. Na parte (a) o conjunto de

objetos de entrada é particionado em conjuntos iniciais. Esta partição pode se valer de alguma heurística ou simplesmente ser escolhido aleatoriamente. É calculado o centróide de cada partição. Na parte (b) os objetos são associados ao centróide mais próximo. Na parte (c) os centróides são atualizados para o centro de cada grupo. Na parte (d) vemos a repetição da parte (b) e (c) até o algoritmo atingir a convergência, ou seja, quando os centróides não são mais alterados.

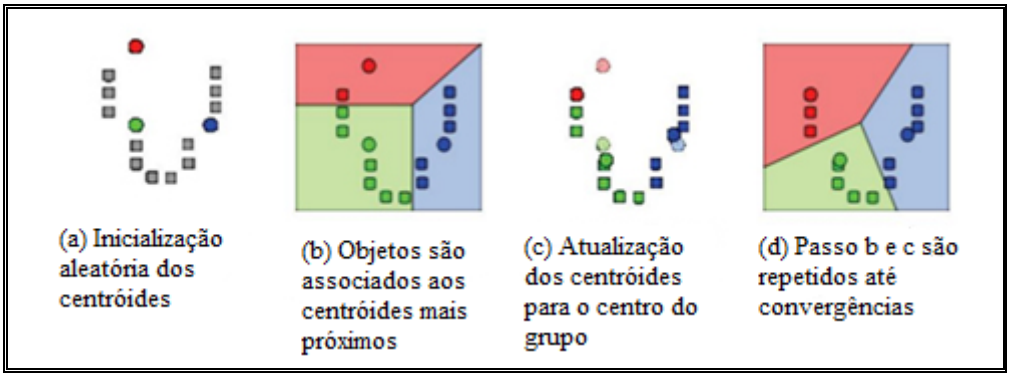


Figura 15 - Demonstração do algoritmo k-médias (LENZ, 2009)

O algoritmo k-médias é simples de implementar e de entender, possui uma rápida convergência do algoritmo de minimização e tem escalabilidade em relação aos dados de entrada (PEREIRA, 2007; DAMASO, 2008; LENZ, 2009).

Sua desvantagem é a necessidade de estimar o parâmetro k, visto que conforme são realizadas as partições iniciais é possível obter resultados diferentes. Outros pontos negativos são que o k-médias não possui imunidade a ruídos, o que acaba influenciando na média, favorecendo a formação de grupos esféricos e, principalmente, o algoritmo de minimização pode estar sujeito a mínimos locais (JAIN et. al., 1999; HAN e KAMBER, 2001; LINDEN 2009; LENZ, 2009).

Conforme Naldi (2010) segue o pseudo-código do algoritmo k-médias:

1. *Seja k o número de grupos e t o número máximo de iterações:*
2. *Se k >= 2 e t >=1*
3. *Inicializa os centróides dos grupos c_1, c_2...., c_k*
4. *Repita*
5. *Para todos $x_j \in X$ faça*
6. *Para todos c_i faça*
7. *Calcula a dissimilaridade $d(x_j, c_i)$*
8. *Fim para*

9. Adiciona o objeto x_j ao grupo C_i para o qual $i = arg\ min\ d(x_j, c_i)$
10. Fim para
11. Para todos c_i faça
12. Recalcula ci como centróide do grupo C_i
13. Fim para
Até convergência ser alcançada ou o número de iterações exceder um dado limite t.

Algoritmo 1- K-médias

O algoritmo EM (*Expectation Maximization*) foi proposto por Demspter et al(1977). Cada grupo é representado matematicamente por uma distribuição de probabilidade de forma que estas se adéquem à distribuição dos dados. O algoritmo faz uma estimativa de máxima verossimilhança dos parâmetros, ou seja, estima parâmetros que sejam os mais consistentes com os dados da amostra no sentido de maximizar a função de verossimilhança (LUNA, 2004; LENZ, 2009).

Esse algoritmo utiliza um processo de iteração para que possam ser aproximadas soluções de equações não lineares, a partir de um valor inicial pré-definido, até que haja convergência a um valor final. A premissa básica do algoritmo é que deve existir um ponto ótimo a partir do qual não é preciso mais qualquer tentativa de um novo modelo, como novos parâmetros e neste momento as iterações terminam. Segundo Lenz (2009) o algoritmo é definido por dois passos:

1. Passo E: Encontram-se os valores esperados das estatísticas suficientes para os dados completos Y, conforme os dados incompletos Z e as estimativas atuais dos parâmetros.
2. Passo M: Utilizam-se estas estatísticas para fazer uma estimativa de máxima verossimilhança.

A principal vantagem do algoritmo é a utilização de uma validação cruzada para determinar o número de grupos. Sua desvantagem é que pode-se obter um máximo local, dependendo dos parâmetros iniciais, o que não é desejado (LENZ, 2009).

2.5.3 MÉTODOS BASEADOS EM MODELO

Os métodos de agrupamento baseados em modelo têm por objetivo utilizar um modelo hipotético para cada grupo, e agrupar os objetos de acordo com esses modelos.

Os grupos são definidos através de funções de densidade que refletem a distribuição espacial dos objetos, pois supõem que os dados são gerados por uma mistura de distribuições de probabilidades (PEREIRA, 2007; MELLO, 2008).

Uma das vantagens destes métodos é que os mesmos levam em conta um modo de determinar automaticamente o número de grupos, levando o ruído em conta, rendendo assim métodos de agrupamentos robustos (MELLO, 2008).

Para Han e Kamber (2001) existem duas abordagens principais para estes tipos de métodos de agrupamento: agrupamento conceitual e agrupamento por redes neurais.

O agrupamento conceitual é uma forma de agrupamento em aprendizagem de máquina que, dado um conjunto de objetos não nomeados, produz um esquema de classificação sobre eles. Esta abordagem se difere das demais, pois além de identificar grupos de objetos semelhantes, também gera conceitos/descrições para cada grupo encontrado. Um algoritmo que utiliza essa abordagem é o COBWEB desenvolvido por Fisher (1987). Esse algoritmo realiza uma análise de probabilidade e a partir da definição de conceitos, os utiliza como modelos para os grupos (LENZ, 2009; PEREIRA, 2007; BOSCARIOLI, 2008).

As redes neurais representam os grupos como exemplares, que servem de protótipo para que novos objetos possam ser atribuídos ao grupo cujo protótipo lhe seja mais próximo, baseado isso em alguma medida de distância. Um método desta categoria é o SOM (*Self Organizing Map*) definido por Kohonen e Haykin na década de 90, tem o objetivo de encontrar um conjunto de neurônios para referência e associar cada exemplo do conjunto de dados ao neurônio de referência mais próximo. O resultado será um conjunto de neurônios que definem implicitamente os grupos (METZ, 2006; PEREIRA, 2007; BOSCARIOLI, 2008; MELLO, 2008).

2.5.4 MÉTODOS BASEADOS EM GRADES

Os métodos baseados em grades utilizam a abordagem de dividir o espaço de variáveis onde os objetos estão contidos em células que formam uma estrutura de dados em grade. Todas as operações de agrupamento serão realizadas sobre a estrutura em grade, ou seja, os algoritmos irão quantificar o espaço dentro de um número finito de células e realizar as operações neste espaço. Dessa forma todos os pontos pertencentes a uma mesma célula podem ser agregados e tratados como um único objeto. Há uma

probabilidade elevada de que todos os pontos pertencentes a um mesmo grupo estejam em uma mesma célula da grade. Na figura 16 é ilustrado um exemplo deste método (BOSCARIOLI, 2008; MELLO, 2008; PEREIRA, 2007).

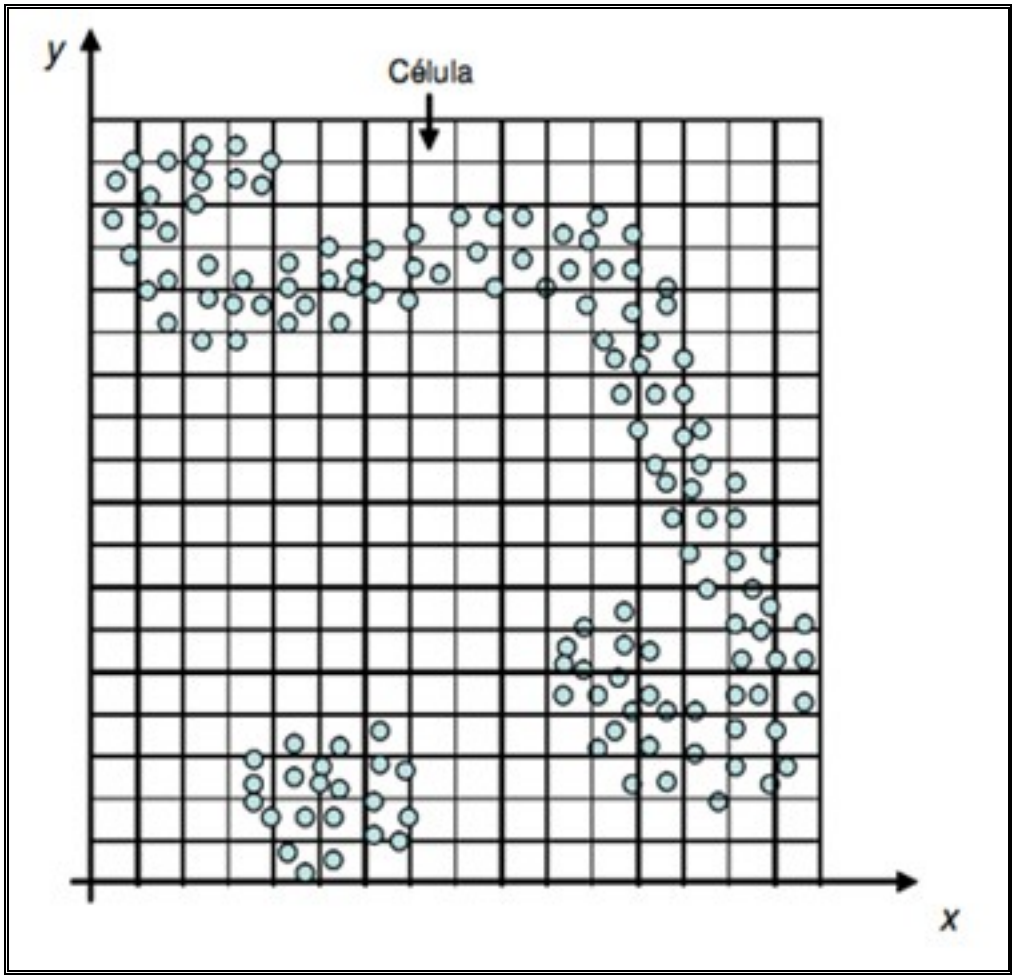


Figura 16 - Exemplo de grade em um conjunto de dados bidimensional (ALVES, 2007)

A principal vantagem dessa abordagem é seu rápido tempo de processamento, que é independente do número de objetos de dados, porém é dependente do número de células em cada dimensão do espaço dividido. Além disso, este método é capaz de encontrar grupos de formatos arbitrários, não apresenta alta sensibilidade aos pontos fora do padrão (*outliers*) e é muito eficiente para grande conjunto de dados (PEREIRA, 2007; MELLO,2008; BOSCARIOLI, 2008; METZ, 2006).

Dois métodos que podem ser citados neste caso são: STING e WaveCluster .

O método STING (*STatistical INformation Grid-based method*) é um método criado por Wang et al. (1997) com o objetivo de solucionar problemas relacionados a agrupamentos e consultas orientadas a regiões. A ideia desse algoritmo é capturar informações estatísticas associadas com as células de maneira que classes inteiras de

consultas e problemas de agrupamento possam ser respondidos sem recorrer a objetos individuais (PEREIRA,2007; MELLO, 2008).

O método WaveCluster definido por Sheikholeslami et al (2000) é uma abordagem de agrupamento multiresolução que aplica a transformação de wavelet no espaço de características. Uma transformação de wavelet é uma técnica de processamento de sinais que decompõe o sinal em diferentes sub-bandas de frequência (CARVALHO, 2002).

2.5.5 MÉTODOS BASEADOS EM DENSIDADE

Os métodos de agrupamento por densidade utilizam critério de agrupamento local, por considerarem a densidade de ligações entre os dados. Sua ideia geral é definir um grupo baseado na densidade de sua vizinhança no espaço multidimensional. Cada exemplo do grupo deve manter uma vizinhança com um número mínimo de vizinhos dentro de uma esfera com raio R. Os exemplos que possuem uma vizinhança com densidade mínima e estão numa distância menor que R pertencem ao mesmo grupo (METZ, 2006; OLIVEIRA, 2008; MELLO, 2008).

A possibilidade de encontrar grupos de formas arbitrárias (grupos de tamanhos variados e com formatos diferentes) e o fato de não precisar da definição do número de grupos como parâmetro inicial são as principais vantagens dos métodos baseados em densidade, visto que a maioria dos métodos de agrupamento se baseia na distância entre os objetos, esses métodos conseguem encontrar apenas grupos de forma esférica e têm dificuldades em encontrar grupos de forma arbitrária. Alguns algoritmos que seguem essa abordagem são: DBSCAN e DENCLUE (OLIVEIRA 2008; PEREIRA, 2007).

O método DBSCAN foi desenvolvido por ESTER et al. (1996) e procura por grupos através do uso de esferas no espaço multidimensional. De acordo com o número de pontos contidos na esfera, novos objetos são adicionados aos grupos. Os centros das esferas são definidos pelos objetos contidos em um grupo. A figura 17 ilustra esse método (MELLO, 2008, HAN e KAMBER, 2001).

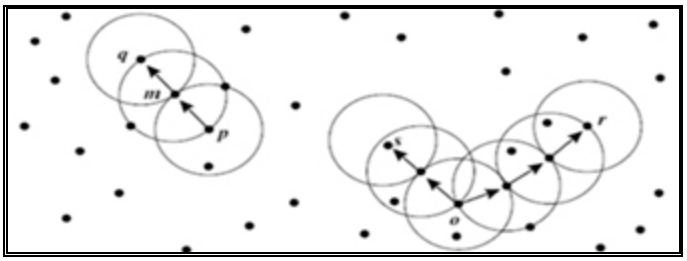


Figura 17 - Execução do método de agrupamento DBSCAN (HAN e KAMBER, 2001)

DENCLUE (*DENsity-based CLUstEring)* definido por Hinneburg e Keim (1998) permite uma descrição matemática compacta de grupos de forma arbitrária para dados multidimensionais. Utiliza células em grade, porém guarda informações apenas sobre aquelas que realmente contém pontos e manipula essa células em uma estrutura de acesso tipo árvore. Os grupos podem ser determinados matematicamente pela identificação de atratores de densidade. Atratores de densidade são máximos locais da função densidade global. Esse algoritmo é mais rápido do que os algoritmos existentes, porém precisa de uma enorme quantidade de parâmetros.

Segundo Metz (2006) deve ser levado em consideração que nenhum método é melhor ou pior que o outro, eles simplesmente atendem a diferentes objetivos. Por isso antes de escolher um deles para o agrupamento deve-se ter consciência do objetivo que se pretende alcançar e verificar a adequabilidade de cada abordagem a essa aplicação.

O resultado da etapa de agrupamento é a divisão do conjunto de dados inicial em grupos. Este resultado pode ser exclusivo, em que um objeto pertence ou não-pertence a um dado grupo; não exclusivos, em que os objetos podem pertencer a múltiplos grupos ou possuir um grau de pertinência em cada grupo; ou ainda que o objeto não pertença a nenhum grupo, isto é, os dados podem ser não agrupáveis (NALDI, 2010).

2.6 REPRESENTAÇÃO DE GRUPOS

Existem várias formas distintas de representar os grupos criados pelos algoritmos de agrupamento. As maneiras mais usuais são ilustradas na figura 18. Alguns formatos de representação estão associados a métodos específicos, como por exemplo, os dendrogramas, que são associados aos métodos hierárquicos e são representados na figura 18 parte(d). Outros são mais genéricos, como a divisão de barras e os diagramas de Venn representados pela figura 18 parte (a) e figura 18 parte(b), respectivamente. As

tabelas representadas pelas figuras 18 parte (c) são mais úteis no caso de métodos que usam lógica nebulosa (LINDEN, 2009).

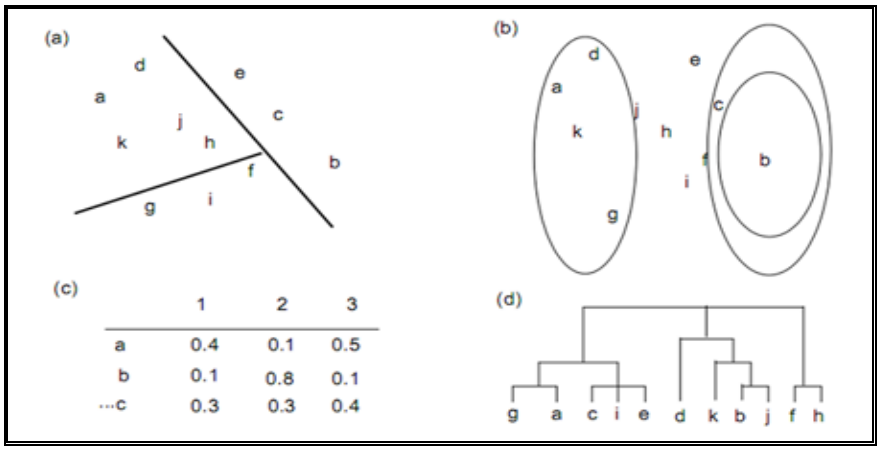


Figura 18 - Representação de grupos (LINDEN, 2009)

2.7 VALIDAÇÃO DE GRUPOS

Após a formação dos grupos, deve ser realizada a validação desse agrupamento. Vários fatores como o agrupamento escolhido, os parâmetros de entrada e a ordem de apresentação dos dados, podem levar a agrupamentos distintos, para um mesmo conjunto de dados. Portanto, critérios e padrões efetivos de avaliação são importantes para prover aos usuários um resultado com certo grau de confiança. Para isso, são utilizadas medidas de validação, que tentam avaliar de maneira objetiva os grupos encontrados (OLIVEIRA, 2008; NALDI, 2010).

A validação dos agrupamentos é realizada com base em índices estatísticos que julgam de uma maneira quantitativa, os grupos formados. A maneira pela qual um índice é aplicado para validar um agrupamento e dada pelo critério de validação. Sendo assim, um critério de validação expressa a estratégia utilizada para validar uma estrutura de agrupamento, enquanto que um índice é um valor estatístico pelo qual a validade é testada. Existem três tipos de critérios que podem ser utilizados: critérios externos, critérios internos e critérios relativos (METZ, 2006; NALDI, 2010).

Os critérios externos avaliam o agrupamento de acordo com uma estrutura pré-definida dos dados, que pode ser uma partição existente nos dados, ou um agrupamento constituído por um especialista da área. A qualidade do agrupamento é medida pelo grau em que o algoritmo conseguiu reconstituir a distribuição de categorias previamente existente. Para averiguar os grupos é montada uma matriz de similaridade para

comparar os resultados do agrupamento com os rótulos externos (categorias), atribuídos a cada objeto. Em uma situação ideal, cada grupo criado pelo algoritmo conteria somente objetos da mesma categoria. A análise de Monte Carlo oferece uma maneira de calcular uma estimativa de tal distribuição, embora sua aplicação seja difícil e custosa. Outro índice utilizado para este critério é o Índice Rand Ajustado (JAIN e DUBES, 1988; PIRES, 2008; FACELI, 2007; SILVA, 2006; OLIVEIRA, 2008).

Os critérios internos não dependem de conhecimento prévio. Eles utilizam os dados originais para medir a qualidade do agrupamento realizado através da matriz de similaridade. Geralmente são divididos em medidas de coesão, que determinam o quão relacionado os objetos do conjunto estão e em medidas de separação que dizem o quão separado um grupo está do outro. Alguns exemplos de índices que utilizam esses critérios são: Índice PBM e o Índice Davies-Bouldin (JAIN e DUBES, 1988; BOSCARIOLI, 2008; PIRES, 2008; OLIVEIRA, 2008).

Os critérios relativos comparam diferentes grupos, dando uma referência de qual destes grupos melhor satisfaz as características reais de seus objetos. Baseiam-se em avaliar a consistência ou estabilidade dos algoritmos de agrupamento, comparando os grupos obtidos pelo mesmo algoritmo sob diferentes condições. Estabelece o melhor valor para um parâmetro usado pelo algoritmo ou para comparar resultados de dois algoritmos diferentes (BOSCARIOLI, 2008; FACELI, 2007; PIRES, 2008; OLIVEIRA, 2008).

Por exemplo, dado um conjunto de partições com diferentes números de grupos como o K-médias, determina qual o número de grupos mais apropriado. O número de grupos K é um exemplo de parâmetro que um critério relativo pode auxiliar na escolha. Para isso o algoritmo de agrupamento é executado para todos os possíveis valores de K em um intervalo K min e K max. Ao apresentar em um gráfico os valores dos índices em funcão de K, o número ideal de grupos é dado pelo valor mínimo, máximo ou inflexão na curva do gráfico construído com a sequência (BOSCARIOLI, 2008; OLIVEIRA, 2008).

A validação dos grupos pode levar a redefinição dos atributos escolhidos e/ou a escolha da medida de similaridade, definidos nas etapas anteriores. A estrutura resultante de um agrupamento é valida se apresenta valores elevados de qualidade, segundo algum nível de referência e esses grupos então passarão para a etapa de interpretação (OLIVEIRA, 2008; NALDI, 2010).

2.8 INTERPRETAÇÃO DOS RESULTADOS

Nessa etapa, analisam-se os grupos encontrados pelo algoritmo para definir um rótulo para cada um desses grupos. Esse rótulo é definido conforme o significado de cada grupo sendo o mesmo relacionado ao domínio da aplicação. Para isso é necessária a participação de um especialista no domínio do problema, para avaliar os grupos encontrados e descobrir a existência de algum significado, definindo assim o rótulo (MELLO, 2008; METZ, 2006).

Muitas vezes é complicado interpretar o resultado do agrupamento, pois ele não apresenta conceitos concretos, mas sim apenas um conjunto de dados agrupados descritos por determinados valores estatísticos e índices de similaridade. A partir deste problema vê-se a necessidade da utilização de novas técnicas e ferramentas para tentar auxiliar o especialista no entendimento do conceito dos grupos. Assim o conhecimento pode ser extraído de uma maneira útil à aplicação (METZ, 2006).

2.9 APLICAÇÕES

O agrupamento de dados está sendo aplicado em diversas áreas atualmente, tais como: Psiquiatria, Psicologia, Sociologia, Biologia, Medicina, Educação, Economia, Engenharia, Marketing, Arqueologia, Física, Visão Computacional e Sensoriamento Remoto, dentre outras (FACELI, 2007; SAMPAIO, 2008; COSTA, 2009).

Em marketing há um grande interesse na utilização de agrupamento para descoberta de grupos de clientes diferentes, com objetivo de obter o conhecimento sobre padrões de consumo dos mesmos. Essa caracterização irá auxiliar no desenvolvimento de campanhas de marketing dirigidas, ou seja, permitir que produtos específicos sejam direcionados a clientes específicos (HAN e KAMBER, 2001). Um exemplo pode ser citado no trabalho de Cardoso et al (2011) que fez uma pesquisa acerca dos hábitos alimentares de adolescentes com objetivo de identificar perfis de consumo e descrever as prevalências desses perfis, com base nas distâncias dos indivíduos aos perfis obtidos. Foram gerados quatro perfis e os resultados apontam para a necessidade de promoção da alimentação saudável nesta população.

Na área agrária o agrupamento é utilizado para verificar a possibilidade de locação de uso da terra para fins agrários e/ou urbanos (HAN e KAMBER, 2001). Um

estudo nessa área pode ser visto em Meireles et al (2011) no qual foram avaliadas famílias da parte baixa da bacia hidrográfica do riacho Faé, a partir do modelo agrícola rural existente. A análise de agrupamento se mostrou eficiente para a distinção dos grupos de famílias homogêneas na área estudada, independentemente de sua localização geográfica.

Em técnicas de geoprocessamento utiliza-se o agrupamento para agrupar regiões cujas características climáticas ou socioeconômicas são semelhantes conforme alguma característica (HAN e KAMBER, 2001). Um estudo pode ser visto em Palacio et al (2011) que identificou a similaridade dos reservatórios superficiais do Ceará, com relação à salinidade das águas para identificar os fatores determinantes da qualidade das águas. O agrupamento formou quatro grupos distintos, sendo determinantes os tipos e concentrações dos sais analisados. A análise desses grupos promoveu a redução de sete características das águas superficiais do Estado Ceará, para dois componentes.

Uma área que utiliza o agrupamento também é a engenharia na criação de especificações e projetos associados à construção de obras em geral (COLE, 1998). Oliveira e Correia (2010) compararam os principais aeroportos brasileiros em relação ao movimento de cargas utilizando técnicas quantitativas de análise multivariada como ferramentas de auxílio à gestão. A comparação dos dez aeroportos baseou-se em fatores de desempenho cuidadosamente escolhidos. A análise por agrupamento hierárquico promoveu a formação de seis grupos, sendo quatro deles unitários. A análise por componentes principais reduziu o espaço amostral dos treze fatores para um espaço bidimensional, o que facilitou a análise qualitativa dos fatores dos aeroportos.

A internet necessita dessa área para agrupamento de documentos de acordo com similaridades semânticas, de forma a melhorar os resultados oferecidos por sites de busca, ou ainda para descobrir grupos de padrões de acessos similares que podem corresponder a diferentes padrões de usuários (ANKERST et al,1999). Neste último caso temos um trabalho de Junior e Mantovani (2010) que fizeram um estudo acerca de seis comunidades do Orkut com objetivo de se verificar a percepção do usuário a respeito da interação nas discussões dessas comunidades. O agrupamento encontrou quatro grupos de usuários distintos: o primeiro grupo que possui baixa interação, o segundo grupo que possui alta interação, o terceiro grupo que possui alta interação e maior percepção da relevância da comunidade para o contexto profissional e o quarto grupo que possui uma interação mediana.

O agrupamento de doenças por sintoma ou curas pode levar a taxonomias muito úteis (COLE, 1998). No trabalho de Garcia et al (2010) foi realizada uma análise sobre o estilo de vida saudável e sua relação com o abandono de serviços de cuidados clínicos preventivos pelos espanhóis adultos diabéticos. Foi visto que a adesão de serviços de prevenção está em níveis desejados, pois os indivíduos que mais necessitam destes serviços são os menos propensos a recebê-los.

Uma área que vem abrindo fronteira é a área biológica que categoriza genes de funções semelhantes, taxonomias de animais e plantas, e ganha entendimento em estruturas inerentes em populações, e atualmente na Genética (projeto *GENOMA Humano*) (Wives, 1999, HAN e KAMBER 2006). Um estudo recente é encontrado em Jangareli et al (2011) onde foram utilizados diferentes níveis de significância genômica na seleção assistida por marcadores para estimar o valor fenotípico. Aplicou-se o método de agrupamento por ligação composta adotando a distância euclidiana como medida de dissimilaridade entre as significâncias genômicas.

2.10 CONSIDERAÇÕES FINAIS

O agrupamento de dados permite formar grupos de objetos similares para que estes possam ser classificados posteriormente. Existem cinco etapas fundamentais para o agrupamento: preparação de padrões, medidas de similaridade, agrupamento, validação e interpretação dos resultados.

Existem vários métodos para agrupamento, cada um com um propósito específico, o que torna complexa a escolha de um ou outro, e mesmo assim nem todos compreendem todos os problemas aos quais são expostos.

Visando definir um novo algoritmo de agrupamento, na seção seguinte será feita uma revisão na área de SIA, a qual está sendo amplamente utilizada na resolução de problemas complexos.

3 SISTEMAS IMUNOLÓGICOS ARTIFICIAIS

Os Sistemas Imunológicos Artificiais (SIA) surgiram como tentativa de se reproduzir o funcionamento do sistema imunológico dos vertebrados através de algoritmos, para resolver determinados problemas. Técnicas oriundas desta abordagem têm obtido sucesso e sido alvo de estudos em diversas áreas, incluindo a área de agrupamento de dados.

A construção dos SIA se baseia principalmente no sistema adaptativo o qual se desenvolve ao longo de nossa vida. Vários algoritmos foram propostos a partir destes sistemas, entre esses a teoria da seleção clonal e o princípio da seleção negativa. Este capítulo propõe uma revisão sobre os principais algoritmos já desenvolvidos dentro da área de SIA bem como algumas das aplicações já propostas para o agrupamento de dados.

3.1 CONTEXTUALIZAÇÃO

A área de SIA surgiu no início dos anos 90 e tem crescido muito nos últimos anos, trazendo consigo diversos casos de sucesso. Dentre eles, destacam-se os algoritmos inspirados em imunologia teórica e experimental com o objetivo de resolver problemas complexos (CASTRO, 2006).

Os SIA se baseiam no sistema imunológico dos vertebrados, por sua capacidade de adaptação ao ambiente em que é exposto e um mecanismo de defesa contra organismos estranhos. Estas são duas características altamente desejáveis em qualquer sistema de computação, visto que flexibilidade (capacidade de gerar diversidade) e segurança (proteção ao sistema) são duas palavras chaves na atualidade (AMARAL, 2006; FORREST et al, 1998).

Segundo Castro (1999) para poder defender o organismo o sistema imunológico dos vertebrados provê uma arquitetura em camadas, conforme vista na figura 19.

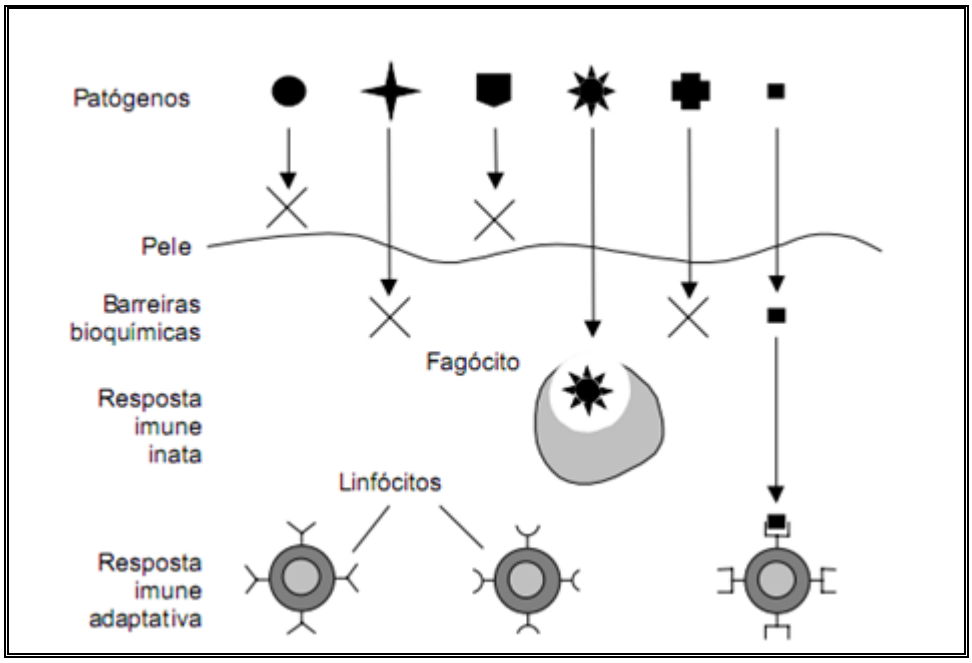


Figura 19 - Estrutura multicamada do sistema imunológico (CASTRO, 1999)

A pele é a primeira barreira contra infecções. Uma segunda barreira é formada por elementos bioquímicos, como o ph e a temperatura corporal, que auxiliam na defesa contra organismos estranhos. Caso essas duas barreiras não resolvam e os agentes patogênicos entrem no organismo, estes serão combatidos pelo sistema imune inato e pelo sistema imune adaptativo (CASTRO, 1999; AMARAL, 2006).

O sistema imune inato nasce junto com o indivíduo e permanece durante toda sua vida. Ele é composto por dois tipos de células, os macrófagos e granulócitos. Estas células possuem a capacidade de ingerir os organismos estranhos e apresentam antígenos, respectivamente. Como na figura 19 podemos ver uma das células do sistema chamada fagócito, que irá combater o organismo estranho representando pelo símbolo de estrela. No entanto, às vezes o sistema inato não consegue reconhecer e eliminar os agentes patogênicos e neste caso entra em ação o sistema imune adaptativo (CASTRO, 1999; SILVA, 2001).

O sistema imune adaptativo está presente somente nos animais vertebrados e é composto por linfócitos (glóbulos brancos), que são responsáveis por reconhecer e eliminar os agentes patogênicos, desenvolvendo para isso os anticorpos. Estes linfócitos podem ser divididos em dois tipos: as células B e as células T. Essas células são derivadas das células tronco da medula óssea e passam por um processo de maturação na própria medula e no timo, respectivamente (SILVA, 2001).

O sistema imune adaptativo possui um processo de aprendizagem que se divide

em dois tipos de resposta: primária e secundária. A resposta primária é um processo lento e acontece quando um novo patógeno entra no organismo, podendo levar semanas para eliminar a infecção. A resposta secundária ocorre quando o sistema reencontra o mesmo patógeno, combatendo-o mais rapidamente por causa do que foi aprendido na resposta primária. Neste processo o sistema desenvolve uma memória imunológica, adquirida na resposta primária, e que dura por um longo período da vida do hospedeiro (BROWNLEE, 2011).

Esse conjunto de camadas forma um sistema eficaz na defesa contra organismos estranhos, garantindo nossa resistência contra as enfermidades, apesar de estarmos sempre cercados por germes.

Os SIA são inspirados em três tipos de algoritmos: algoritmo da seleção clonal, algoritmo da seleção negativa e teoria da rede imunológica. A partir destes surgiram vários outros com intuito de resolver os mais diversificados problemas. Nas seções seguintes são apresentados os principais algoritmos relevantes para este trabalho.

3.2 SELEÇÃO CLONAL

3.2.1 INSPIRAÇÃO BIOLÓGICA

O algoritmo de seleção clonal é inspirado na teoria da seleção clonal que foi proposta por Burnet (1959). Esta teoria se refere à resposta imune adaptativa. Os linfócitos que se ligam a antígenos irão se proliferar formando “clones”. Estes poderão ser divididos em células de plasma, que possuem vários anticorpos e tem um tempo de vida curto, ou em células de memória, que auxiliam no reconhecimento de um antígeno que já passou pelo organismo. O princípio da seleção clonal é ilustrado na figura 20 (CASTRO, 1999; BROWNLEE, 2011).

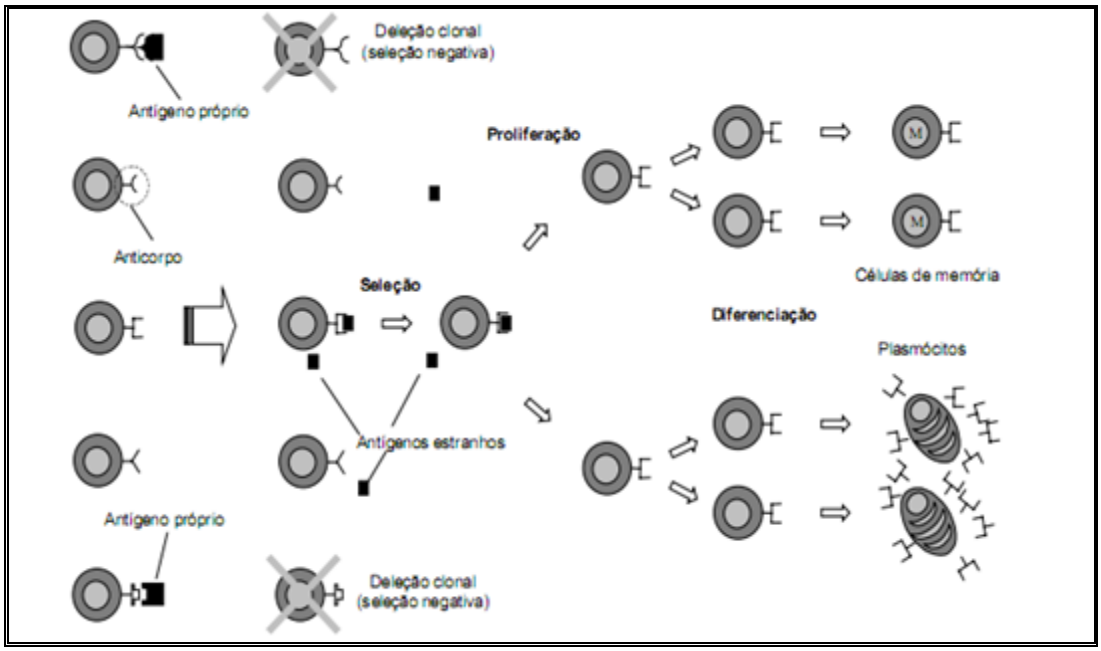


Figura 20 - Princípio da seleção clonal (CASTRO, 1999)

A característica mais importante desta teoria é que pode haver uma pequena quantidade de clones com erro na proliferação da célula, este processo é chamado de hipermutação somática. Em consequência de mudar a forma dos receptores, pode haver problemas na capacidade de reconhecimento de antígeno dos anticorpos nos linfócitos, e dos anticorpos que as células de plasma produzem (BROWNLEE, 2011).

Segundo essa teoria, o sistema é capaz de modificar a si mesmo, conforme um repertório inicial de células imunes e experiências que o mesmo teve com o ambiente. Ele ainda é capaz de prover um mecanismo de defesa antes mesmo que o patógeno tenha tido algum contato com o hospedeiro. Isso é realizado através de um processo de aprendizagem, que envolve uma população de adaptação de unidades de informação (cada uma representando uma solução do problema). Haverá uma concorrência entre essas populações que irão ser selecionadas, em seguida duplicadas e sofrerão uma mutação, resultando em uma população melhor para a defesa do organismo (BROWNLEE, 2011).

3.2.2 ALGORITMO DA SELEÇÃO CLONAL

Segundo Castro e Zuben (2000), os principais aspectos considerados para o desenvolvimento do algoritmo CLONALG (*Clonal Selection Algorithm*) foram: manutenção das células de memória funcionalmente independentes do repertório, seleção e reprodução (clonagem) das células mais estimuladas, morte das células menos estimuladas, maturação de afinidade e reseleção dos clones com maiores afinidades antigênicas, geração e manutenção de diversidade, hipermutação proporcional à afinidade celular.

As principais etapas do algoritmo são descritos a seguir e podem ser visualizadas na figura 21.

1. Gerar um conjunto inicial de anticorpos (linfócitos).
2. Determinar os melhores indivíduos da população gerada com base em uma medida de afinidade.
3. Gerar uma população temporária de clones da população gerada no passo 2.
4. Aplicar o processo de hipermutação somática na população de clones gerada no passo 3 (quanto maior a afinidade, menor a taxa de mutação).
5. Selecionar os melhores indivíduos desta população para memória imunológica.
6. Substituir o conjunto da população geral pelos novos anticorpos.

O processo é repetido até que sejam verificados todos os antígenos

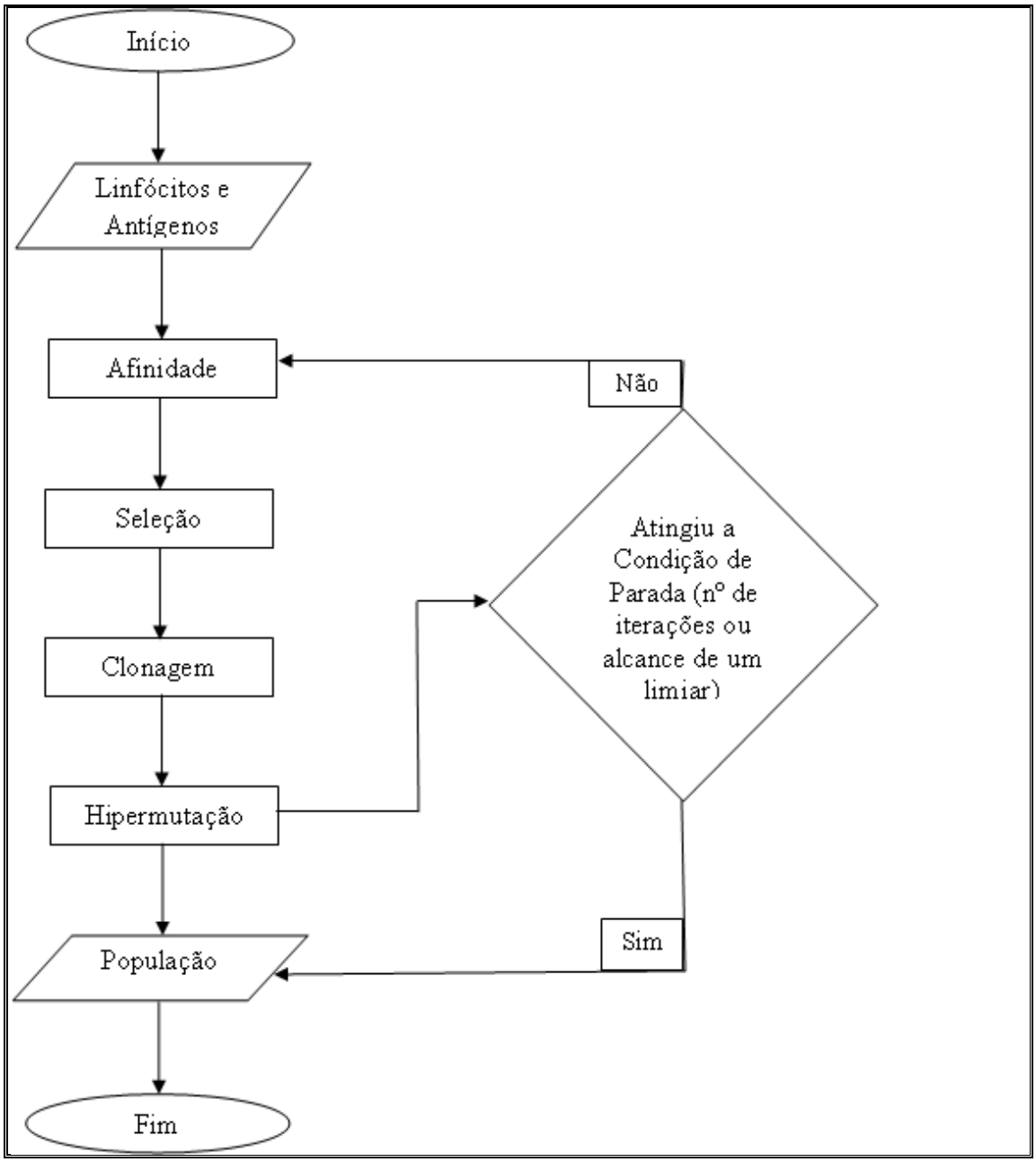


Figura 21 - Etapas do algoritmo CLONALG

Segundo Brownlee (2011) o algoritmo 2 apresenta um pseudo-código do algoritmo CLONALG. O modelo geral do CLONALG envolve a seleção de anticorpos (soluções candidatas) baseada em suas afinidades: pela combinação do linfócito com o antígeno padrão ou através de uma avaliação pelo padrão de uma função custo. Anticorpos selecionados são sujeitos a clonagem proporcional à afinidade, e a hipermutação dos clones é inversamente proporcional a afinidade do clone, quanto mais afinidade com o antígeno menor será a mutação. O resultado da clonagem compete com a população de anticorpos existentes por um lugar na próxima geração. Além disso, membros da população com baixa afinidade são substituídos por anticorpos gerados

aleatoriamente. A variação do padrão de reconhecimento do algoritmo inclui a manutenção da memória de solução que na sua totalidade representa uma solução para o problema.

Entrada: Tamanho da População, Tamanho da Seleção, Tamanho do Problema, Número de Células Aleatórias, Taxa de Clones, Taxa de Mutação
Saída: População

1. *População ← Criar Células Aleatórias (Tamanho da População, Tamanho do Problema);*
2. *Enquanto Condição de Parada() faça*
3. *Para cada $p_i \in$ População faça*
4. *Afinidade (p_i)*
5. *Fim para*
6. *Seleção da População ← Selecionar (População, Tamanho da Seleção)*
7. *Clones da População ← ϕ*
8. *Para cada $p_i \in$ População faça*
9. *Clones da População ← Clonar (p_i, Taxa de Clones)*
10. *Fim para*
11. *Para cada $p_i \in$ Clones da População faça*
12. *Hipermutação (p_i, Taxa de Mutação)*
13. *Afinidade (p_i)*
14. *Fim para*
15. *População ← Selecionar (População, Clones da População, Tamanho da População)*
16. *População Aleatória ← Criar Células Aleatórias (Número de Células Randômicas);*
17. *Substituir (População, População Aleatória)*
18. *Fim enquanto*
19. *Retorna População*

Algoritmo 2 - CLONALG

3.3 SELEÇÃO NEGATIVA

3.3.1 INSPIRAÇÃO BIOLÓGICA

A teoria da seleção negativa foi proposta por Forrest et al (1994), seu principal objetivo é fornecer tolerância às células próprias evitando assim que as células do corpo se destruam (auto-imunidade). Esse problema é conhecido como "discriminação do próprio/não-próprio", que distingue as células próprias das não próprias do corpo. O processo de distinção geralmente é feito no timo, onde as células T passam por um processo de monitoração. As células que reagirem às proteínas do próprio corpo são destruídas, somente aquelas que não se conectarem ao próprio organismo vão deixar o timo e irão realizar a proteção do organismo.

O paradigma imunológico do próprio/não próprio não é intuitivo, pois ele propõe uma modelagem de um domínio desconhecido a partir do que é conhecido. Isso é realizado a partir de modelos de mudanças, anomalias de dados desconhecidos que são construídos através da geração de padrões, que se diferem das células existentes. Este modelo é usado para monitorar a existência de dados normais ou fluxos de dados, buscando novos padrões para os dados desconhecidos (BROWNLEE, 2011).

3.3.2. ALGORITMOS DE SELEÇÃO NEGATIVA

O algoritmo de seleção negativa (NSA) tem como ideia principal gerar um conjunto de detectores aleatórios e então descartar aqueles que reconhecem os dados de entrada, que são os dados relacionados ao próprio. Estes detectores podem ser usados posteriormente para detecção de anomalias. Este algoritmo é capaz de executar tarefas como reconhecimento de padrões, armazenando informações sobre o conjunto complementar (não-próprio) ao conjunto dos padrões que se deseja proteger (FORREST et al, 1994). O algoritmo NSA é executado em duas fases: fase de censoriamento e fase de monitoramento, descritas a seguir e ilustradas nas figuras 22 e 23, respectivamente.

Na fase de censoriamento temos os seguintes passos:

1. Definir um conjunto de strings próprias (S) que se deseja verificar.
2. Gerar strings aleatoriamente e avaliar as strings próprias com as strings geradas verificando se as duas combinam conforme uma medida de afinidade.

3. Caso a afinidade seja superior a um determinado limiar pré-definido, rejeita-se essa string, caso contrário essa string é armazenada em um conjunto de detectores (R).

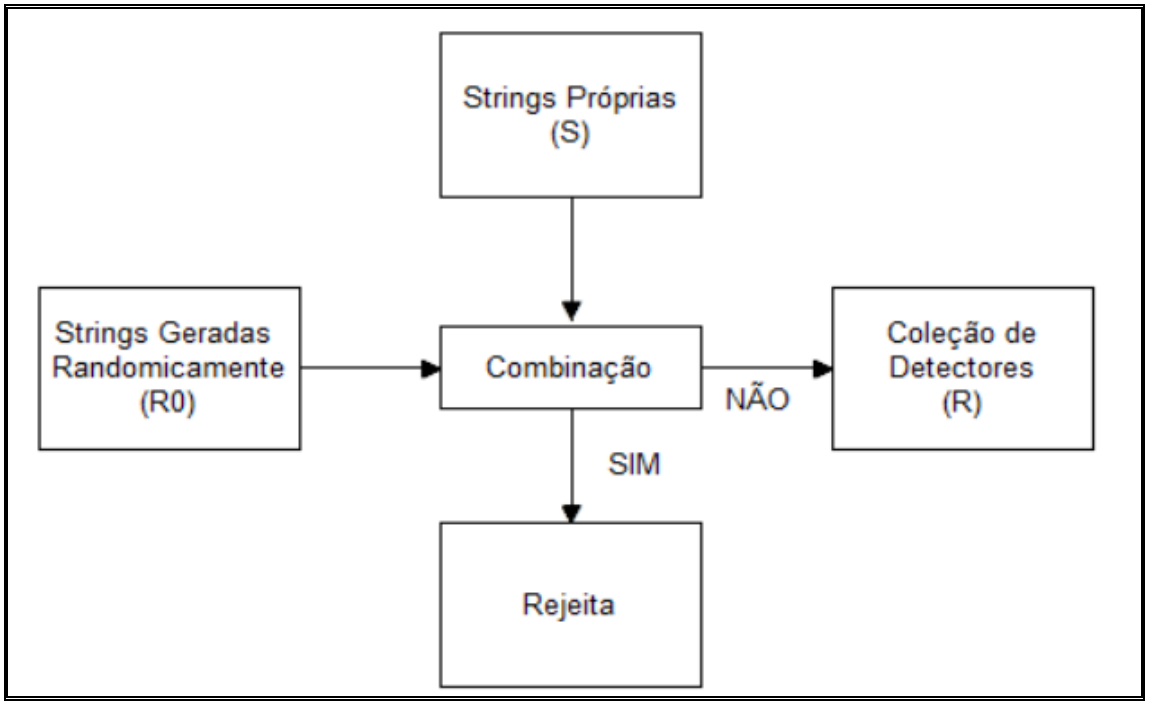


Figura 22 - Fase de censoriamento do algoritmo de seleção negativa (Forrest et AL, 1994)

Na fase de monitoramento temos o seguinte passo*:*

1. Verificar a afinidade entre as strings protegidas (S) e a coleção de detectores (R0). Se as duas combinarem conforme um limiar pré-definido, então um elemento não-próprio foi identificado.

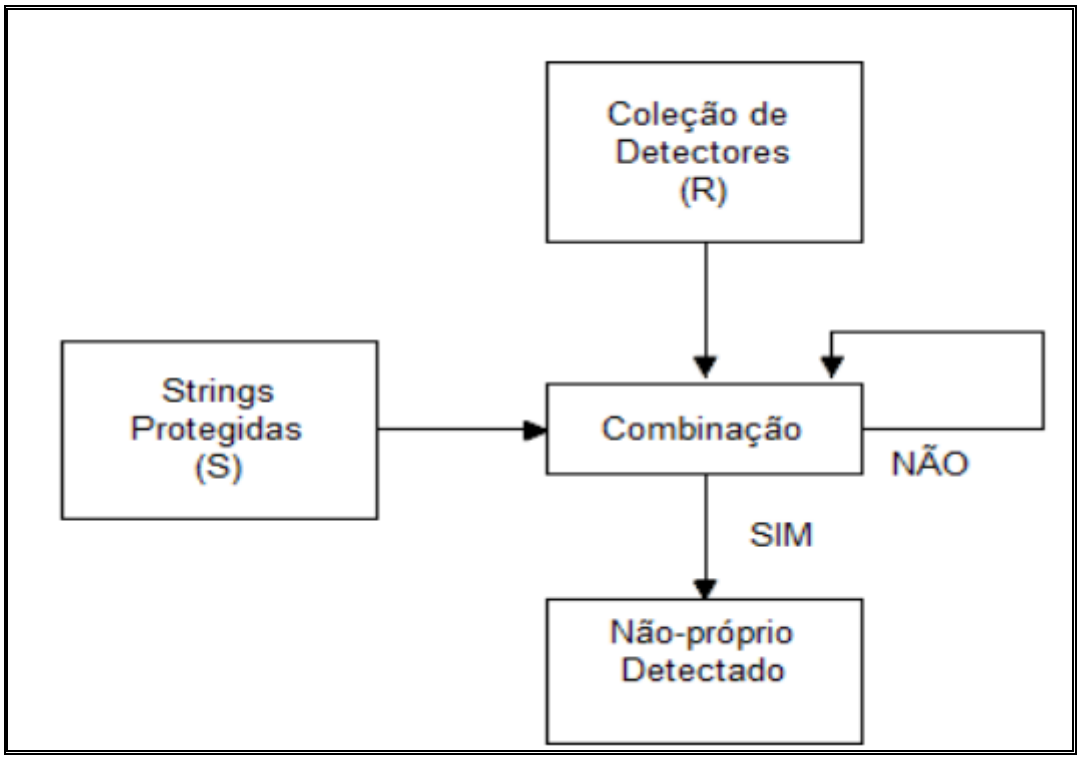


Figura 23 - Fase de monitoramento do algoritmo de seleção negativa (FORREST et AL, 1994)

O componente principal do algoritmo de seleção negativa é a escolha da regra de detecção, que determina a similaridade entre dois padrões de modo a classificar as amostras em próprio/não-próprio. Se o limiar pré-definido for muito alto, os detectores gerados irão se tornar sensíveis a qualquer anomalia nos dados e, portanto, um maior número de detectores será necessário para atingir o grau de confiabilidade desejado. Por outro lado, se o limiar for muito baixo, pode ser impossível gerar um conjunto de detectores de tamanho razoável a partir do conjunto próprio disponível. Isto sugere que o valor do limiar pode ser usado para ajustar a sensibilidade da detecção e diminuir o risco de falsos positivos (AMARAL, 2006).

Brownlee (2011) apresenta um pseudo-código para o procedimento de seleção negativa na fase de censoriamento e monitoramento, ilustrada nos algoritmos 3 e 4, respectivamente.

Entrada: Dado Próprio
Saída: Repertório

1. *Repertório* $\leftarrow \phi$
2. *Enquanto Condição de Parada () faça*
3. *Detectores* $\leftarrow$ *Gerar Detectores Randomicamente()*
4. *Para cada* $Detector_i \in$ *Repertório faça*
5. *Se Combinar*($Detector_i$, *Dado Próprio) então*
6. *Repertório* $\leftarrow Detector_i$

```
7.        Fim se
8.     Fim para
9.  Fim enquanto
10. Retorna Repertório
```

Algoritmo 3 - Seleção negativa fase de censoriamento

```
Entrada: Amostra de Entrada, Repertório
1.  Para cada Entrada_i ∈ Amostra de Entrada faça
2.        Classe de Entrada ← "não próprio"
3.        Para cada Detector_i ∈ Repertório faça
4.              Se Combinar(Entrada_i , Detector_i) então
5.                 Classe de Entrada_i ← "próprio"
6.                 Break;
7.              Fim se
8.        Fim para
9.  Fim para
```

Algoritmo 4 - Seleção negativa fase de monitoramento

3.4 SISTEMA IMUNOLÓGICO ARTIFICIAL DE RECONHECIMENTO

3.4.1 INSPIRAÇÃO BIOLÓGICA

O sistema imunológico artificial de reconhecimento (AIRS) foi proposto por Watkins e Bogges (2004), e é inspirado em algoritmos de aprendizagem supervisionada, como o algoritmo da célula dendritica, o algoritmo da seleção clonal (seção 3.2) e o algoritmo da seleção negativa (seção 3.3).

A principal característica do AIRS é o reconhecimento de antígenos pelo corpo, o qual é feito pelos linfócitos (as células B e as células T). Cada um desses linfócitos possui um receptor molecular único chamado epítopo, o qual irá se unir ao antígeno indicando se há uma ligação forte, conforme podem ser vistos na figura 24 (CASTRO, 1999; AMARAL, 2006).

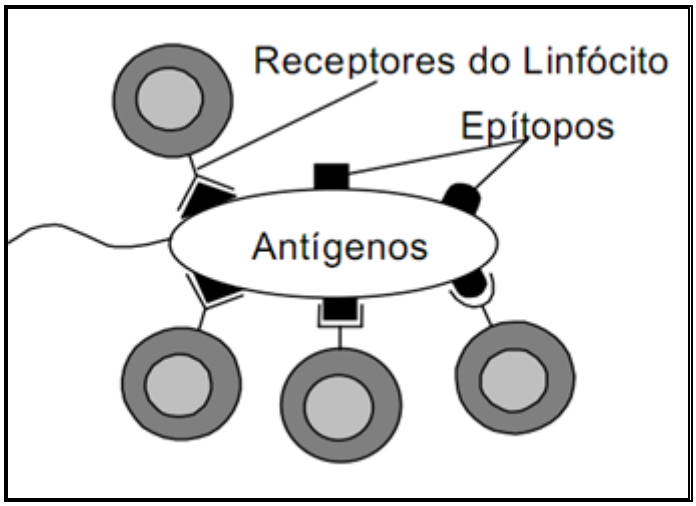


Figura 24 - União entre receptores e epítopos (CASTRO, 1999)

O número de receptores que se unem aos agentes patogênicos irá determinar a afinidade do linfócito ao antígeno. Para que o linfócito possa iniciar a defesa, a afinidade entre ele e o antígeno deve ultrapassar um certo limiar. À medida que este limiar aumenta, o número de diferentes epítopos capazes de ativar o linfócito diminui, isto é, o subconjunto de similaridade torna-se menor (AMARAL, 2006).

3.4.2 ALGORITMO DO SISTEMA IMUNOLÓGICO ARTIFICIAL DE RECONHECIMENTO

O objetivo desse algoritmo é desenvolver um conjunto de células de memória que poderão ser utilizadas para classificação de dados. O sistema mantém um reservatório de células de memória que, através de uma iteração simples dos dados de treinamento (antígenos), procuram encontrar soluções em potencial (células B). As células de memórias que forem mais estimuladas serão clonadas, e esses clones vão competir uns com os outros para entrar no reservatório de memória, baseados na quantidade de recursos. No algoritmo utilizamos o conceito de ARB (Bola de Reconhecimento Artificial), que representa um grupo de células similares.

Segundo Watkins e Bogges (2004) o algoritmo AIRS, possui quatro estágios principais que podem ser visualizados na figura 25 e são citados a seguir:

1. Normalização dos dados de entrada e inicialização do reservatório de memória.

2. Identificação das células de memória e geração das ARB's.
3. Competição de recursos e desenvolvimento de uma célula de memória candidata.
4. As células de memória bem sucedidas são adicionadas à lista final.

Este processo continua até que todos os antígenos tenham sido apresentados para o sistema.

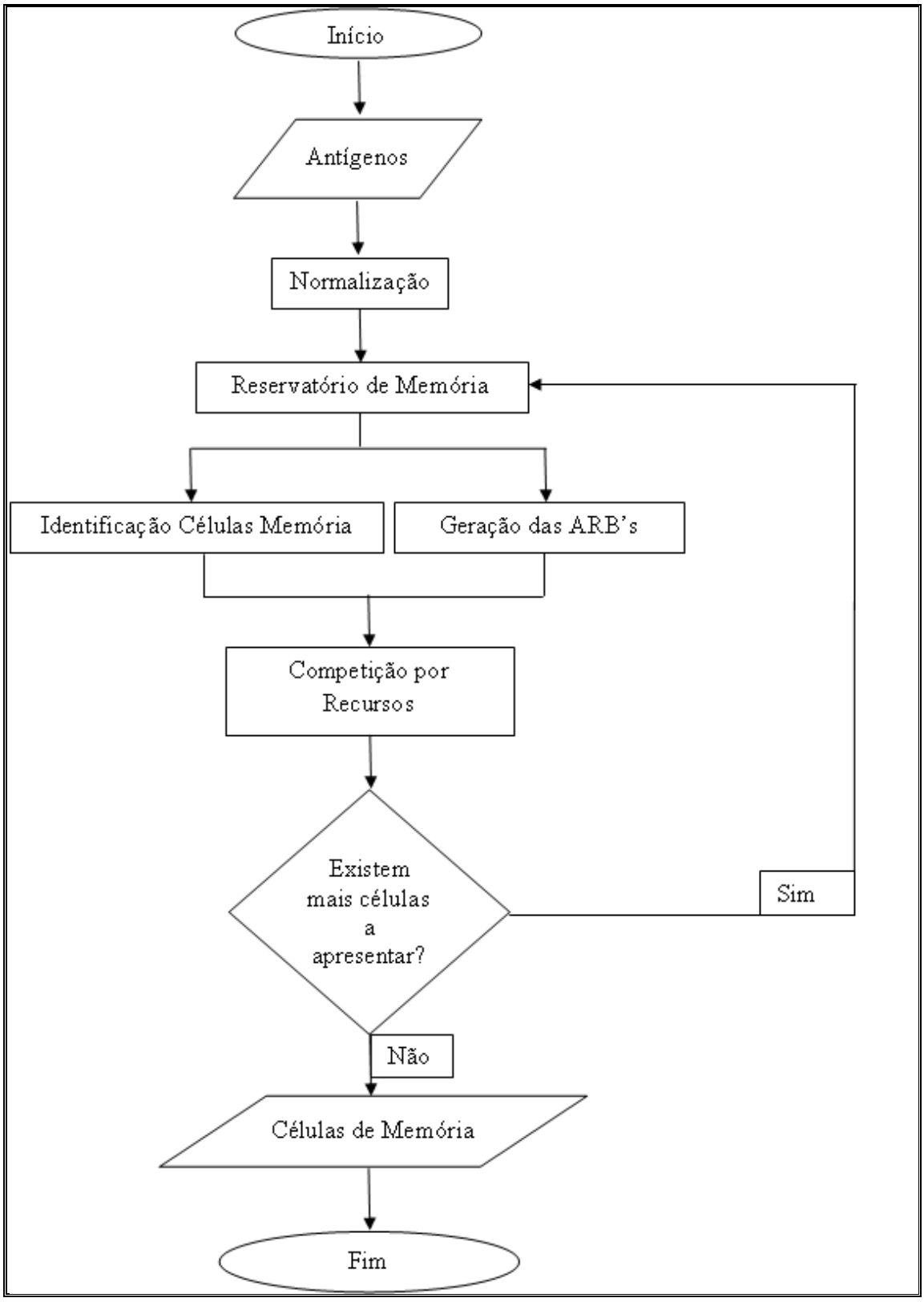


Figura 25 - Algoritmo AIRS

O algoritmo 5 apresenta um pseudo-código para preparar os vetores da célula de memória utilizado no AIRS. Uma medida de afinidade (distância) entre os padrões dados deve ser definida. Para vetores de valores reais, geralmente é usada a distância

euclidiana.

$$dist\ (x, c) = \sum_{i=1}^{n} (x_i - c_i)^2$$

Equação 4 - Distância euclidiana para AIRS

Onde:

- *n* é o número de atributos.
- *x* é o vetor de entrada.
- *c* é a célula de vetor dada.

A variação das células durante a clonagem ocorre inversamente proporcional ao estímulo de uma célula dada a um padrão de entrada.

Entrada: Padrão de Entrada, Taxa de Clones, Taxa de Mutação, Limiar de estimulação, Recurso Máximo, Limiar de afinidade
Saída: Células de Memória

1. *Células de Memória ← Inicializar Reservatório Memória (Padrão de Entrada)*
2. *Para cada Padrão de Entrada$_i$ ∈ Padrão de Entrada faca*
3. *Melhor Célula ← Buscar Célula Mais Estimulada (Padrão de Entrada$_i$, Células de Memória)*
4. *Se Melhor Classe Célula != Classe Padrão de Entrada$_i$ então*
5. *Células de Memória ← Criar Nova Célula Memória (Padrão de Entrada$_i$)*
6. *Senão*
7. *Número de Clones ← Melhor Célula Estimulada X Taxa de Clones X Taxa de Mutação*
8. *Células Clone ← Melhor Célula*
9. *Para i até Número de Células faça*
10. *Células Clone ← Clonar e Mutar (Melhor Célula)*
11. *Fim para*
12. *Enquanto Média da Simulação(Células Clone) <= Limiar de estimulação faça*
13. *Para cada Célula(i) ∈ Células Clone faça*
14. *Células Clone ← Clonar e Mutar(Célula$_i$)*
15. *Fim para*
16. *Estimular (Células Clone, Padrão de Entrada)*
17. *Reduzir Reservatório Para Recurso Máximo (Células Clone, Recurso Máximo)*
18. *Fim enquanto*
19. *Célula$_c$ ← Buscar Célula Mais Estimulada (Padrão de Entrada$_i$, Células*

Clone)
20. Se Célula Estimulada$_c$ > Melhor Célula Estimulada então
21. Célula de Memória ← Célula$_c$
22. Se Afinidade (Célula $_c$, Melhor Célula) <= Afinidade Estimulada então
23. Deletar Célula(Melhor Célula, Célula de Memória)
24. Fim se
25. Fim se
26. Fim se
27. Fim se
28. Retorna Células de Memória

Algoritmo 5 - AIRS

3.5 APLICAÇÕES

Segundo Castro (2001), os sistemas imunológicos artificiais estão sendo amplamente utilizados para resolução de diversos problemas computacionais como: problemas relacionados à segurança computacional, proteção contra vírus, detecção de anomalias, monitoramento de processos, reconhecimento de padrões, controle de robôs, sistemas classificadores, softwares tolerantes a falhas, criação de componentes mais velozes, softwares mais dinâmicos com grande tolerância a mudança de requisitos, novas topologias de redes de classificação, aprendizagem de máquinas, agrupamento de dados, entre outros.

A seguir, serão apresentados alguns trabalhos relacionados à área de agrupamento de dados, utilizando a tecnologia dos SIA.

Jia et al (2006), propuseram um algoritmo imunológico artificial de agrupamento para o controle de tráfego urbano. Um modelo inicial foi proposto por Hauser onde era utilizado um agrupamento hierárquico. Este modelo foi melhorado por Park que introduziu o conceito de algoritmos genéticos ao mesmo. Depois de aplicado, perceberam-se algumas falhas neste modelo, como o surgimento de grupos isolados e sujos, o que faz com que os engenheiros tenham que atribuir grupos adjacentes ou técnicas de algoritmos especiais para refinar esses grupos.

O algoritmo proposto por Jia et al (2006) foi aplicado ao reconhecimento de padrões de tráfego, para o qual pode-se programar os intervalos de controle do dia (TOD) automaticamente, facilitando também a separação dos intervalos do dia, que era

feita manualmente. Uma das principais características do algoritmo proposto é possuir parâmetros que podem ser ajustados. Isto é uma vantagem, pois o torna viável para diferentes aplicações, visto que os intervalos de parâmetros podem ser definidos. Por outro lado, deve-se tomar cuidado na definição dos mesmos, pois ela influencia diretamente nos efeitos do programa em execução e nos grupos formados.

O algoritmo de agrupamento imunológico baseia-se na teoria da rede imunológica e na teoria da seleção clonal (seção 3.2). Na figura 26 é ilustrado o algoritmo de agrupamento artificial que consiste nos seguintes passos:

1. Escolher um número aleatório de anticorpos para inicializar e definir uma variável de controle de parada.
2. Inicializar os dados a serem agrupados como antígenos.
    2.1 Calcular o vetor de distância e o vetor de afinidade entre o antígeno e anticorpo que podem ser vistas na equação 5 e 6, respectivamente.

$$d_{ji} = \left\| Ab_i - Ag_j \right\|_2 = \left(\sum_{k=1}^{p} \left(Ab_{ik} - Ag_{jk} \right)^2 \right)^{1/2}$$

Equação 5 - Cálculo do vetor de distância entre antígeno e anticorpo

$$af_{ji} = \left(1 + d_{ji}\right)^{-1}$$

Equação 6 - Cálculo da afinidade entre antígeno e anticorpo

    2.2 Escolher os anticorpos com maior afinidade com o antígeno e ordenar as células de maior para menor afinidade.
    2.3 Clonar os anticorpos.
    2.4 Realizar a maturação de afinidade nas células clonadas.
    2.5 Calcular a afinidade entre o antígeno e as células clone.
    2.6 Selecionar os clones que possuem maior afinidade com o antígeno e adicionar os mesmos nas células de memória.
3. Calcular o vetor de similaridade entre as células de memória e eliminar os anticorpos cuja similaridade seja menor que um limiar pré-definido.
4. Produzir anticorpos aleatórios para substituir a baixa afinidade destes com os anticorpos iniciais.
5. Ajustar o valor das variáveis de controle e voltar ao passo 2 para continuar a próxima iteração, até atingir um número de iterações definido ou satisfazer os requerimentos das configurações da análise de agrupamento.

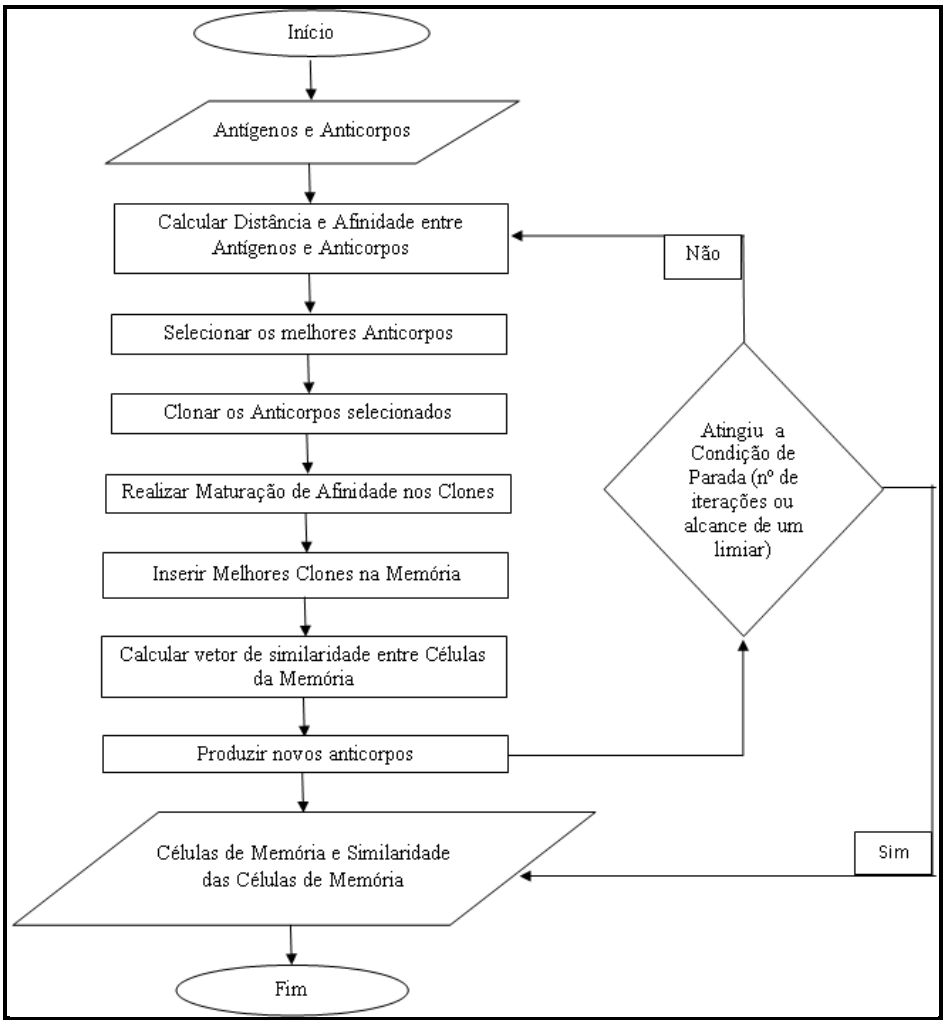


Figura 26 - Algoritmo imunológico de agrupamento para controle de tráfego (JIA et al, 2006)

O resultado do algoritmo será um grupo de dados de memória e a similaridade das células de memória. De acordo com as distâncias entre os dados iniciais e as células de memória, ou a afinidade entre antígenos e células de memória, pode-se então realizar o agrupamento dos antígenos.

O algoritmo se mostrou eficiente em reduzir as informações redundantes dos agrupamentos de dados. Ele pode ser utilizado em dados em massa, onde métodos de agrupamento tradicional não conseguem ser bem executados. Para a aplicação de controle de tráfego o algoritmo obteve sucesso e programou os esquemas TOD automaticamente no Sistema de Transporte Inteligente (ITS).

Castro e Zuben (2001) propuseram o Ainet (*Artificial Immune Networks*) que é um modelo simplificado da rede imunológica artificial, composto por um conjunto de nós, chamados anticorpos, e conjuntos de pares de nós chamados arestas, com peso atribuído associado a cada extremidade conectada. Esse algoritmo foi inspirado nas

redes imunológicas e na teoria da seleção clonal, e consiste em gerar anticorpos (protótipos) que identificam grupos de antígenos (dados apresentados). Os objetivos do Ainet são: a automação da descoberta de conhecimento, mineração de dados redundantes e a partição de agrupamento automático.

Outra proposta para algoritmo de agrupamento pode ser vista em Xu et al (2006) que propôs um algoritmo imunológico de agrupamento supervisionado denominado ISCA (*immune supervised clustering algorithm*). O objetivo do algoritmo é identificar classes uniformes que tenham uma alta densidade probabilística e que os grupos sejam realizados pelo algoritmo imunológico. A abordagem baseia-se em distâncias métricas e agrupamentos baseados em representação. A performance do algoritmo foi comparada com três outros algoritmos TDS, SRIDHCR, que são algoritmos de agrupamento supervisionado, e um algoritmo de agrupamento tradicional denominado PAM. Os resultados experimentais mostraram que o agrupamento supervisionado aumenta a "pureza" das classes de 9 para 25% sobre o algoritmo tradicional PAM. Além disso, foi verificado que apesar do ISCA ter um tempo extenso de execução, este encontrou a maioria dos grupos do experimento.

3.6 CONSIDERAÇÕES FINAIS

Os SIA têm sido utilizados como base para desenvolver soluções para problemas complexos, em diversas áreas da tecnologia. A principal teoria utilizada dentro da área computacional é a forma como o sistema imune adaptativo funciona. Existem vários algoritmos que foram definidos conforme os SIA, como o algoritmo da seleção clonal, o algoritmo da seleção negativa e o AIRS. Na área de agrupamento de dados, já foram propostos algoritmos com base nas teorias da rede imunológica e na teoria da seleção clonal os quais obtiveram sucesso formando grupos consistentes.

No próximo capítulo será proposto um novo algoritmo imunológico para agrupamento de dados, baseado principalmente na teoria da seleção negativa e na teoria da seleção clonal.

4 PROPOSTA DE UM ALGORITMO IMUNOLÓGICO PARA AGRUPAMENTO DE DADOS

O trabalho proposto visa apresentar um novo algoritmo imunológico para agrupamento de dados, utilizando como base a teoria da seleção clonal. Para resolver problemas computacionais a técnica mais utilizada é a construção de um algoritmo, que é uma sequência de passos finitos utilizados para chegar a uma solução. Neste capítulo serão revistos os princípios básicos na construção de um novo algoritmo e alguns formalismos propostos para construção de novos algoritmos na área de SIA. Serão apresentados os formalismos do algoritmo imunológico, bem como a proposta de um fluxograma e um pseudo-código para o mesmo.

4.1 MÉTODO

Na resolução de problemas computacionais deve-se encontrar uma maneira de descrevê-lo de uma forma clara e precisa. Para auxiliar nessa tarefa é descrito um algoritmo que é um conjunto finito de regras, bem definidas, para a solução de um problema em um tempo finito e com um número finito de passos. Uma breve ilustração de uma formalização de um algoritmo pode ser vista na figura 27 (PREUSS, 2002; PEREIRA et al, 2009).

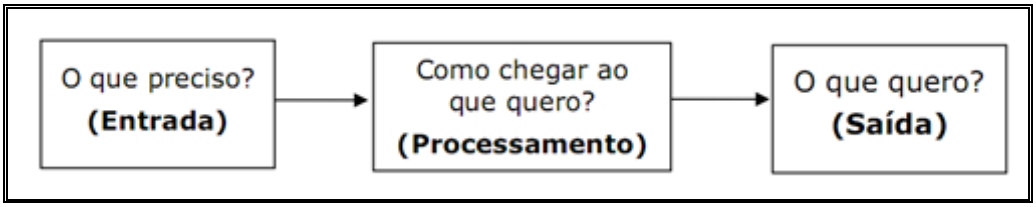


Figura 27 - Questões de formalização de um algoritmo (PEREIRA et al, 2009)

Segundo Pereira et al (2009), um algoritmo deve possuir algumas características fundamentais. Deve sempre terminar após um número finito de passos, cada passo deve ser preciso, sem possíveis ambigüidades, deve-se informar zero ou mais entradas e possuir uma ou mais saídas. Uma condição de fim deve sempre ser atingida para quaisquer entradas e num tempo finito.

Para construir um algoritmo podem-se seguir os seguintes passos (Silva, 2006):

1. Compreender o problema a fim de identificar o objetivo a ser atingido.

2. Identificar os dados de entrada (a serem fornecidos).
3. Identificar os dados de saída a serem gerados como resultado da solução.
4. Determinar o que é preciso para transformar dados de entrada em dados de saída (processamento).
5. Construir o algoritmo, fazendo uso de uma das formas de representação de algoritmos.
6. Testar a solução para validar sua execução ou detectar possíveis erros.

Existem três tipos de formas de representação de algoritmos mais conhecidas: descrição narrativa, fluxograma e pseudo-código.

Na descrição narrativa os algoritmos são descritos diretamente em uma linguagem natural. Esta forma não é muito utilizada, visto que o uso de uma linguagem natural muitas vezes dá oportunidade a ambigüidades (PREUSS, 2002).

Uma maneira mais simples e direta na representação de algoritmos é o fluxograma, que trabalha com gráficos para representar a origem dos dados, o destino das informações e os tipos de armazenamento dos processos do sistema. As caixas representam as instruções a serem executadas e são ligadas por setas que indicam o fluxo das ações. A principal vantagem dessa representação é a facilidade na compreensão do funcionamento do algoritmo. Porém, a construção de algoritmos mais complexos e longos, pode se tornar extremamente trabalhosa. A figura 28 mostra as principais formas geométricas usadas em fluxogramas (SILVA, 2006; PEREIRA et al, 2009; MEDINA e FERTIG, 2005).

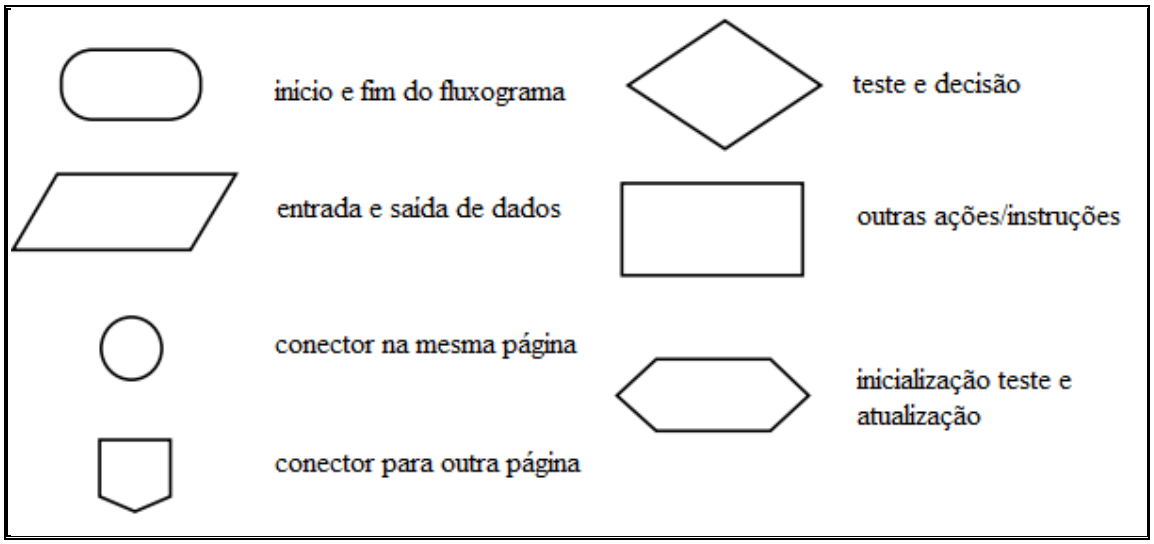


Figura 28 - Formas de representação de um algoritmo (PREUSS, 2002)

Na construção de algoritmos utiliza-se muitas vezes estruturas de controle. Na

figura 29 são ilustradas as três principais estruturas através das quais pode se construir um algoritmo.

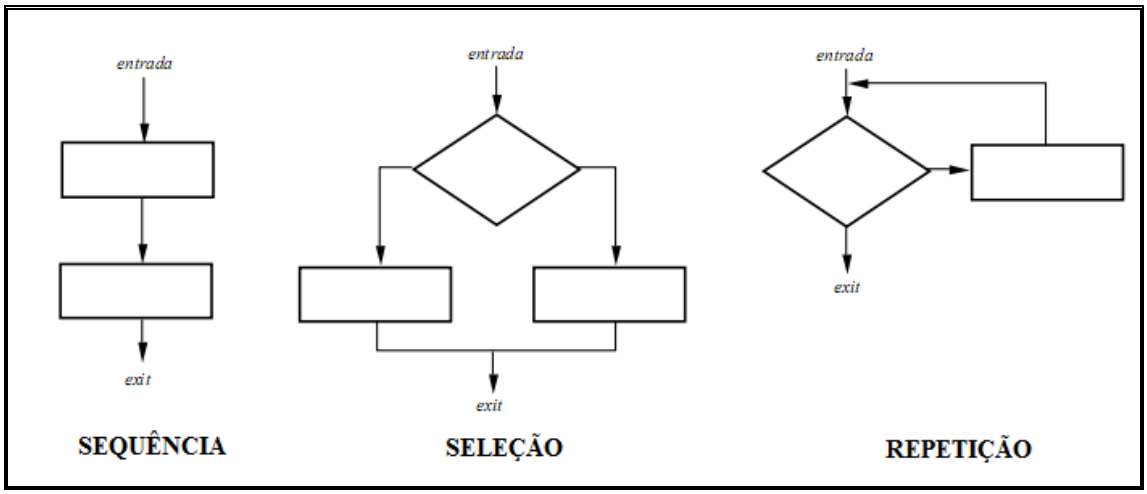


Figura 29 - Principais estruturas de controle (PREUSS, 2002)

Na estrutura sequencial os comandos de um algoritmo são executados numa sequência pré-estabelecida. Cada comando é executado somente após o término do comando anterior. Na seleção ou decisão, o fluxo de processamento pode seguir por uma das duas vias dependendo do valor lógico (verdadeiro ou falso) da expressão avaliada no início da estrutura. As estruturas de repetição são feitas de acordo com o conhecimento prévio do número de vezes que o conjunto de comandos será executado (PREUSS, 2002; PEREIRA et al, 2009).

Uma última maneira bastante utilizada na representação dos algoritmos são os pseudo-códigos, também conhecida como português estruturado ou portugol, pois é bastante rica em detalhes. Essa representação torna fácil a tradução do algoritmo para uma linguagem de programação específica (PREUSS, 2002).

4.2 FORMALISMO

Holland (1975) propôs um formalismo sobre sistemas adaptativos que fornecem uma maneira geral de representação para construção de novos algoritmos. Este formalismo é dividido em duas seções.

Na primeira seção são estudados os objetos primários que podem ser vistos na tabela 5. Estes objetos são convencionais de um sistema adaptativo: o ambiente *E,* a estratégia ou plano adaptativo que cria soluções no ambiente *S,* e a utilidade atribuída para criar soluções *U.*

Tabela 5 - Objetos primários no formalismo do sistema adaptativo

Termo	Objeto	Descrição
E	Ambiente	O ambiente do sistema submetido a adaptações.
S	Estratégia	O plano adaptativo que determina modificações estruturais sucessivas em resposta ao ambiente.
U	Utilidade	A medida da performance de diferentes estruturas no ambiente. Mapeia uma solução dada (A) para a avaliação de um número real.

Na segunda seção são estudados os objetos secundários que podem ser vistos na tabela 6. Estes fornecem mais detalhes do formalismo, pois sugerem um contexto mais amplo que aquele da instância específica dos objetos primários, permitindo assim a avaliação e comparação dos conjuntos de objetos com os planos (*S*), ambiente (*E*), espaços de pesquisa (*A*) e operadores (*O*).

Tabela 6 - Objetos secundários no formalismo do sistema adaptativo

Termo	Objeto	Descrição
A	Espaço de pesquisa	O conjunto de possíveis estruturas, soluções, e o domínio da ação para um plano adaptativo
E	Ambientes	O alcance de diferentes ambientes, onde *E* é uma instância. Ele pode representar as estratégias desconhecidas sobre o ambiente.
O	Operadores	Conjunto de operadores aplicados a uma instância de A no tempo $t(A_t)$ para transformar em A_{t+1}.
S	Estratégias	Conjunto de planos aplicáveis a um ambiente dado (onde *S* é uma instância), que usa operadores do conjunto *O*.
X	Critérios	Usados para comparar estratégias (no conjunto *S*), sob o conjunto de ambientes *E*. Leva em conta a eficiência de um plano em diferentes ambientes.
I	Respostas (Feedback)	Conjunto de entradas e sinais possíveis do ambiente fornecendo informação dinâmica ao sistema sobre a performance de uma solução particular *A*, em um ambiente particular *E*.
M	Memória	A memória ou partes retidas da história de entrada *I* para uma solução *A*.

Um plano adaptativo age em tempo distinto *t*, no qual é uma simplificação útil para a análise e simulação por computador. Um framework para um sistema adaptativo dado requer a definição de um conjunto de estratégias *S*, um conjunto de ambientes *E*, e critérios para um ranking de estratégias *X*. Dados os seguintes conjuntos de objetos: espaço de pesquisa *A*, conjunto de operadores *O*, e a resposta do ambiente *I*.

4.3 ALGORITMO

O objetivo do algoritmo imunológico para agrupamento de dados é formar grupos bem definidos constituídos por instâncias similares, utilizando os conceitos da área de SIA. Para isso o algoritmo foi inspirado na teoria da seleção clonal (apresentado na seção 3.2) e no princípio da seleção negativa (apresentado na seção 3.3).

A formalização proposta (na seção 4.1) foi utilizada para definir os elementos básicos do algoritmo imunológico (ilustrada na figura 30). Os dados de entrada são representados por cada uma das linhas do dataset informado. Mais especificamente cada uma dessas linhas representa um antígeno definidos por Ag_i, sendo i as linhas do dataset. O conjunto de antígenos pode ser descrito como $Ag_i = \{Ag_1, Ag_2,..., Ag_i\}$ e representa os elementos que devem ser agrupados. A partir dos dados de entrada é realizado o processamento através do algoritmo imunológico, definido posteriormente neste trabalho. A saída do algoritmo é representada pelos grupos formados, os anticorpos. Cada anticorpo é definido por Ab_j, sendo j o número do grupo. O conjunto de anticorpos pode ser definido por $Ab_j = \{Ab_1, Ab_2,..., Ab_j\}$, sendo que cada grupo deve conter um ou mais antígenos.

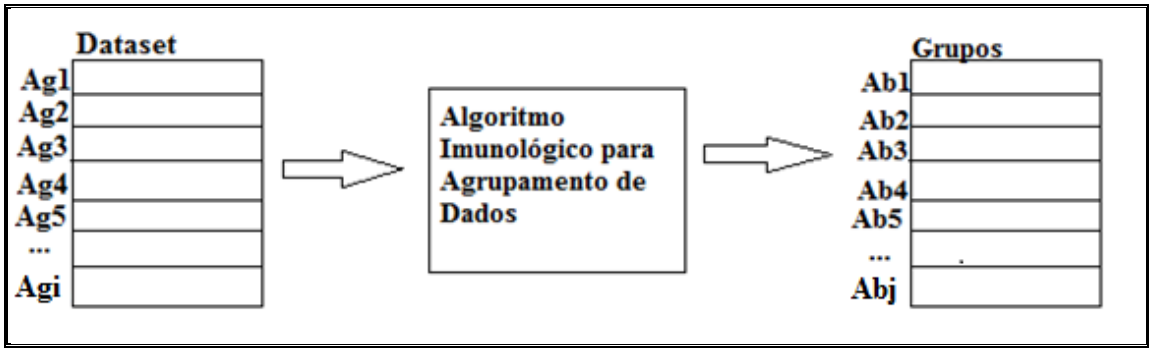


Figura 30 - Questão de formalização do algoritmo imunológico

Para construir o algoritmo imunológico foi levado em consideração também o formalismo proposto por Holland(1975) na seção 4.2. Em um primeiro momento serão estudados os objetos primários utilizados no algoritmo imunológico.

O ambiente E é definido pelo espaço de formas proposto por Perelson e Oster (1979). O espaço de formas é um espaço n-dimensional, onde as dimensões são tomadas como as propriedades generalizadas envolvidas na interação antígeno(Ag) e anticorpo(Ab). Esses elementos são considerados pontos dentro do espaço. Anticorpos possuem uma hiper-região de reconhecimento dentro deste espaço, no qual os antígenos podem desencadear uma resposta imunológica. A natureza complementar das interações

Ag-Ab pode ser ignorada, de modo que, mede a distância de similaridade e pode modelar a afinidade entre duas moléculas dadas dentro do espaço. Uma interação Ab-Ag só ocorre se o Ag está acima de um limiar de afinidade pré-definido. A figura 31 ilustra o espaço de forma S, existe uma região V na qual a forma do paratopo (.) e do complemento do epítopo (x) estão localizados. Um anticorpo é capaz de reconhecer qualquer epítopo (ou idiotopo) cujo elemento esteja situado em uma região V(e) em torno do paratopo (Browlee, 2007).

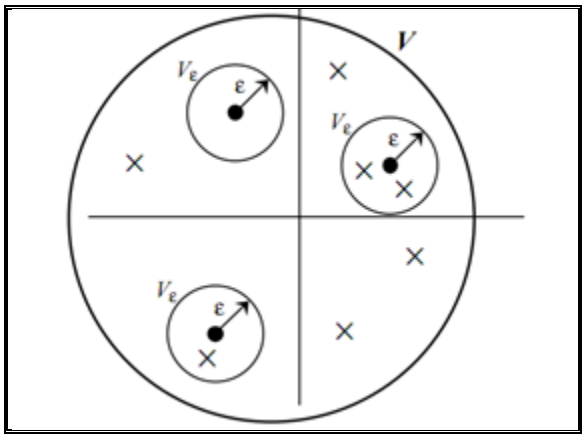


Figura 31 - Espaço de formas de Perelson e Oster (1979)

A estratégia *S* será inspirada na teoria da seleção clonal e no princípio da seleção negativa, além de utilizar a técnica dos métodos hierárquicos (seção 2.5.1). A utilidade *U* poderá ser medida através dos critérios de validação de grupos (seção 2.7). Os objetos primários do algoritmo imunológico são representados na tabela 7.

Tabela 7 - Objetos primários no formalismo do algoritmo imunológico

Termo	Objeto	Descrição
E	Ambiente	Espaço de forma definido por Perelson e Oster (1979) que descreve as interações entre as células e moléculas do sistema imunológico e antígenos.
S	Estratégia	É baseado na teoria da seleção clonal e no princípio da seleção negativa e nas técnicas de agrupamentos hierárquicos.
U	Utilidade	Poderão ser utilizados os índices relativos para validação dos grupos formados.

Em um segundo momento são estudados os objetos secundários que fornecem alguns detalhes a mais sobre o formalismo e podem ser vistos na tabela 8. O espaço de pesquisa *A* define as estruturas possíveis e o domínio de ação para o plano adaptativo. Ele representa então as estruturas que serão utilizadas para chegarmos a solução do

problema. No caso do algoritmo imunológico, poderá ser representado pelo vetor de antígenos e linfócitos. O ambiente *E* neste caso será representado pelo dataset informado, o qual contém os elementos a serem agrupados. Os operadores *O* utilizados na questão de agrupamento poderão ser definidos pelo cálculo da afinidade entre os elementos, os quais servirão para informar se um elemento pertence ou não a um dado grupo. As estratégias *S* são basicamente a verificação sobre a qual grupo o antígeno pertence conforme a afinidade entre o mesmo e o linfócito. Os critérios *X* poderão representar o limiar de afinidade entre um antígeno e um linfócito. A resposta *I* poderá ser medida pelos grupos formados pelo algoritmo imunológico. A memória *M* por sua vez, poderá ser representada pela memória imunológica, que será um mapa que conterá os principais padrões pelos quais os antígenos estão sendo agrupados.

Tabela 8 - Objetos secundários no formalismo do algoritmo imunológico

Termo	Objeto	Descrição
A	Espaço de pesquisa	Antígenos, Linfócitos e Memória Imunológica
E	Ambientes	Dataset pré-definido.
O	Operadores	Cálculo de afinidade entre antígenos e linfócitos.
S	Estratégias	Verificação a qual grupo o antígeno pertence, seja pela memória imunológica ou pela combinação com o linfócito.
X	Critérios	Limiar de afinidade entre antígeno e linfócito.
I	Respostas (Feedback)	Grupos formados chamados anticorpos.
M	Memória	Memória imunológica definida por um mapa de padrões, que conterá os padrões de cada grupos e seu respectivo grupo.

Para apresentar o algoritmo imunológico para agrupamento de dados foram escolhidas duas formas de representação, conforme definidas na seção 4.1. A primeira representação será ilustrada por um fluxograma, visto sua facilidade de entendimento, e a segunda representação será através de um pseudo-código no qual o algoritmo é definido em mais detalhes. O fluxograma é ilustrado na figura 32, e representa as principais etapas do algoritmo imunológico.

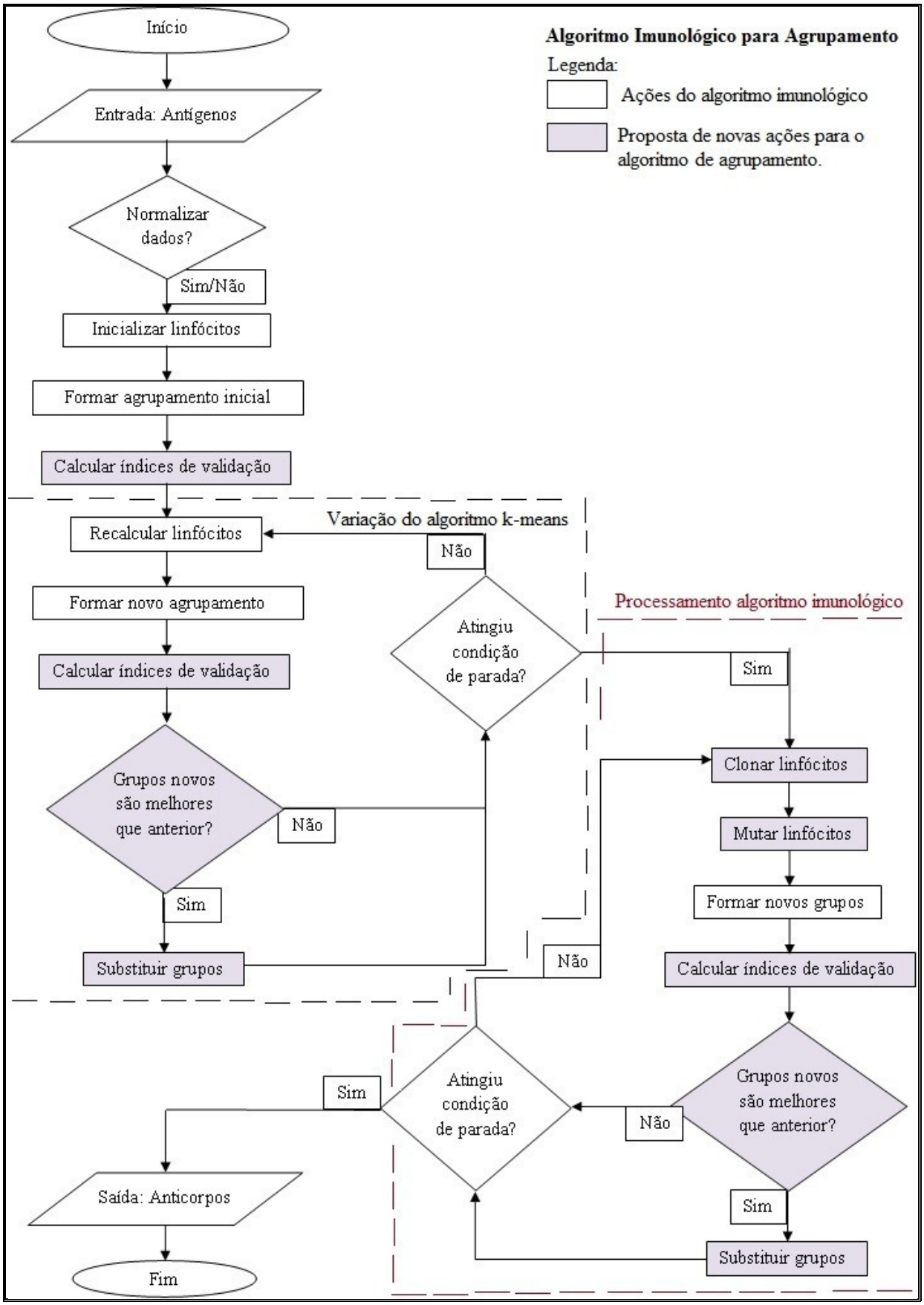


Figura 32 - Fluxograma algoritmo imunológico para agrupamento de dados

O algoritmo 6 apresenta o pseudo-código do algoritmo imunológico para agrupamento de dados. O algoritmo inicia com *n* antígenos correspondentes as *n* linhas do arquivo a serem agrupadas, sendo que cada linha será representada por um elemento. Como visto no capítulo 2, uma das principais etapas a ser realizada no agrupamento é a normalização dos dados, em seguida é criado um conjunto de linfócitos que serão confrontados com os antígenos para a criação dos grupos. Para criar os linfócitos serão selecionados alguns antígenos aleatoriamente, estes servirão como um "padrão" inicial para os grupos. Para formar os grupos é verificada a afinidade entre os linfócitos e o antígeno em questão através de cálculos de similaridade.

Após as iterações iniciais os linfócitos serão clonados e mutados e será recalculada a afinidade entre estes e os antígenos, assim poderá ser verificado se os grupos melhoraram ou não conforme alguns índices de validação. O algoritmo irá ser realizado até que atinja determinada condição de parada definida pelo usuário. O resultado do algoritmo será um grupo de anticorpos que são os elementos agrupados.

Entrada: Antígenos

Saída: Grupo de Anticorpos

1. Ler arquivo de entrada (antígenos)
2. Normalizar dados (quando solicitado)
3. Selecionar linfócitos iniciais através do dataset original
4. Formar agrupamento inicial
5. Calcular homogeneidade de cada grupo
6. Calcular homogeneidade global
7. Calcular separação
8. Repetir
    - 8.1 Mover linfócitos para centro do grupo
    - 8.2 Formar novo agrupamento
    - 8.3 Calcular homogeneidade de cada grupo
    - 8.4 Calcular homogeneidade global
    - 8.5 Calcular separação
    - 8.6 Selecionar o melhor agrupamento através dos índices calculados (homogeneidade e separação)

   Até atingir critério de parada
9. Repetir
    - 9.1 Clonar linfócitos
    - 9.2 Gerar mutação dos novos linfócitos
    - 9.3 Formar novos grupos
    - 9.4 Calcular homogeneidade de cada grupo
    - 9.5 Calcular homogeneidade global
    - 9.6 Calcular separação
    - 9.7 Selecionar o melhor agrupamento através dos índices calculados (homogeneidade e separação)

Até atingir critério de parada

10. Retorna melhor agrupamento (anticorpos)

Algoritmo 6 - Algoritmo imunológico para agrupamento de dados

Nas seções seguintes serão detalhadas algumas das principais funcionalidades do algoritmo imunológico.

4.3.1 GERAÇÃO DE LINFÓCITOS

Serão criados linfócitos para a verificação da afinidade entre os antígenos, e será através desta afinidade que pode-se dizer se o antígeno pertence ou não a determinado grupo. Os linfócitos representarão o "centro" do grupo e a criação destes será realizada de forma distinta e pode ser dividida em três etapas.

A primeira etapa irá ocorrer na inicialização do algoritmo, onde serão selecionados alguns antígenos aleatoriamente do arquivo original. A quantidade de linfócitos a serem inicializados será informada pelo usuário.

A segunda etapa ocorre na iteração do algoritmo k-means. Nesta etapa os linfócitos irão ser recalculados conforme os grupos formados. A equação 7 traz o cálculo para o novo linfócito denotado por linf de C_i. Considere C_i o enésimo grupo para o qual estamos tentando recalcular o linfócito, Acum é um vetor onde cada posição contém a soma dos atributos de todos os antígenos de C_i e Cont é um vetor onde cada posição contém a soma da quantidade de todos os antígenos de C_i.

$$linf[C_i] = Acum[C_i]/Cont[C_i]$$

Equação 7- Fórmula para recalcular linfócitos

A terceira e última etapa diz respeito ao processamento do algoritmo imunológico, os linfócitos irão ser clonados e será aplicada uma fórmula para a mutação dos mesmos. O usuário irá informar um percentual que ele deseja que seja aplicado nos atributos do linfócito para mutação. Primeiro temos que achar o percentual do atributo que será aplicado ao seu valor original. A fórmula para encontrar este percentual pode ser vista na equação 8, onde $Atrib_i$ é o valor do atributo em questão, *percInf* é o percentual informado pelo usuário e *percAtrib* é o percentual do atributo.

$$percAtrib = Atrib_i * (percInf/100)$$

Equação 8 - Cálculo do percentual do atributo

Com o percentual do atributo, podemos então somar ou diminuir o mesmo do valor do atributo original. Esta soma ou subtração será feita aleatoriamente, ou seja, na primeira iteração pode-se somar já na segunda, poderá ser subtraído o valor. O resultado deste cálculo será o novo valor do atributo mutado, a fórmula para esta mutação pode ser vista na equação 9, onde *percAtrib* é o percentual do atributo calculado

anteriormente, *valorAtributo* é o valor original e *valorMutado* é o resultado da mutação.

$$valorMutado = percAtrib +/- valorAtributo$$

Equação 9 - Cálculo para mutação do atributo do linfócito

A mutação irá garantir uma melhor forma de explorar o espaço, pois estaremos modificando sempre os pontos do linfócito em questão. Levando em consideração um percentual iremos garantir que a mutação seja proporcional. Caso dois atributos sejam muito distintos, por exemplo, se um primeiro atributo do linfócito for 1 e o segundo atributo for 1000, a mutação neste linfócito será proporcional. Outra coisa a se levar em consideração é que a soma ou subtração deste valor será feita aleatoriamente, assim estaremos movendo esses pontos dentro do espaço de uma forma aleatória, podendo portanto cobrir melhor o espaço.

4.3.2 NORMALIZAÇÃO

Conforme visto (na seção 2.3) uma das principais etapas do agrupamento é a normalização dos dados, porém nem sempre se deseja normalizar os dados a serem agrupados. Conforme algumas pesquisas realizadas, verificou-se que apesar da normalização ser uma boa prática, muitas pessoas não utilizam. Assim, para este algoritmo foi decidido deixar a etapa de normalização opcional, sendo realizada apenas se o usuário assim desejar. A fórmula para normalização pode ser vista na equação 10. Considere $atrib_i$ o valor a ser normalizado, $minAtrib_i$ é o menor valor do atributo para todos os antígenos do arquivo e $maxAtrib_i$ é o maior valor do atributo para todos os antígenos do arquivo.

$$dadoNormalizado = atrib_i - (minAtrib_i / \mathrm{maxAtrib_i})$$

Equação 10 - Cálculo para normalização

4.3.3 CÁLCULOS DE AFINIDADE ENTRE ELEMENTOS

Para verificar a afinidade entre um antígeno e o linfócito serão utilizadas métricas de distância. Para dados numéricos o usuário poderá definir se prefere utilizar o cálculo da distância euclidiana ou o cálculo da distância euclidiana quadrática.

No caso de se ter dois vetores, um de antígenos definidos por Ag = {Ag_1, Ag_2,

..., Ag_i}, e um vetor de linfócitos definidos por Al = {Al_1, Al_2, ..., Al_x}, a distância entre os elemento poderá ser calculada conforme as equações 1 ou representadas na tabela 9.

Tabela 9 - Equações para cálculo de afinidade em SIA

Equação 11 - Distância euclidiana para SIA	**Equação 12 - Distância euclidiana quadrática para SAI**
$D = \sqrt{\sum_{i=1}^{L}(Al_x - Ag_i)^2}$	$D = \sum_{i=1}^{L}(Al_x - Ag_i)^2$

Se a afinidade *D* entre as moléculas for maior ou igual a um determinado limiar de afinidade pré-definido, então ocorreu uma ligação entre as moléculas. Neste caso, vai haver um lugar geométrico de pontos no espaço de formas, tal que todo antígeno que lá se encontra apresenta a mesma afinidade a um dado anticorpo.

4.3.4 ÍNDICES DE VALIDAÇÃO

O principal objetivo do agrupamento, conforme visto no capítulo 2 é gerar uma grande homogeneidade entre elementos do mesmo grupo e uma grande heterogeneidade entre elementos de grupos distintos. Pensando nesta teoria podemos aplicar índices de validação para verificar quando substituir os grupos e, inclusive, quando o algoritmo deve parar de ser executado. No caso do algoritmo imunológico foram utilizados dois principais índices de validação: a homogeneidade e a separação.

Segundo Chen et al (2002) a homogeneidade é calculada como a distância média entre cada perfil de expressão gênica e do centro do grupo a que pertence, onde $C(g_i)$ é o centro do grupo, N_{gene} é o número total de instâncias e D é a função da distância a ser aplicada. Quanto menor o resultado mais homogêneo será o grupo e melhor será o resultado. A fórmula para a homogeneidade pode ser vista na equação 14.

$$H_{ave} = \frac{1}{N_{gene}} \sum D(g_i, C(g_i))$$

Equação 13 - Fórmula da homogeneidade

A separação é calculada como a distância média ponderada entre os centros do grupo, onde C_i e C_j são os centros do grupo, e N_{ci} e N_{cj} são o número de instâncias nos grupos. Quanto maior o resultado melhor será a separação entre os grupos e melhor será o resultado. A fórmula para a separação pode ser vista na equação 15 (Chen et al, 2002).

$$S_{ave} = \frac{1}{\sum_{i\#j} N_{ci} N_{cj}} \sum_{i\#j} N_{ci} N_{cj} D(C_i, C_j)$$

Equação 14 - Fórmula da separação

4.4 EXECUÇÃO DO ALGORITMO

A execução do algoritmo imunológico pode ser realizada em dois ambientes. Na figura 33 é apresentado o primeiro ambiente que será utilizado pelo programador para realização dos testes no algoritmo. Neste ambiente o programador seleciona um dataset pré-classificado e informa os parâmetros de entrada do programa, o algoritmo é executado e os resultados serão a matriz de confusão e três arquivos de texto. O primeiro arquivo conterá um log com todos os passos principais do algoritmo, o segundo arquivo conterá os índices de validação para auxiliar na construção dos gráficos para análise de resultados e o último arquivo conterá os grupos formados.

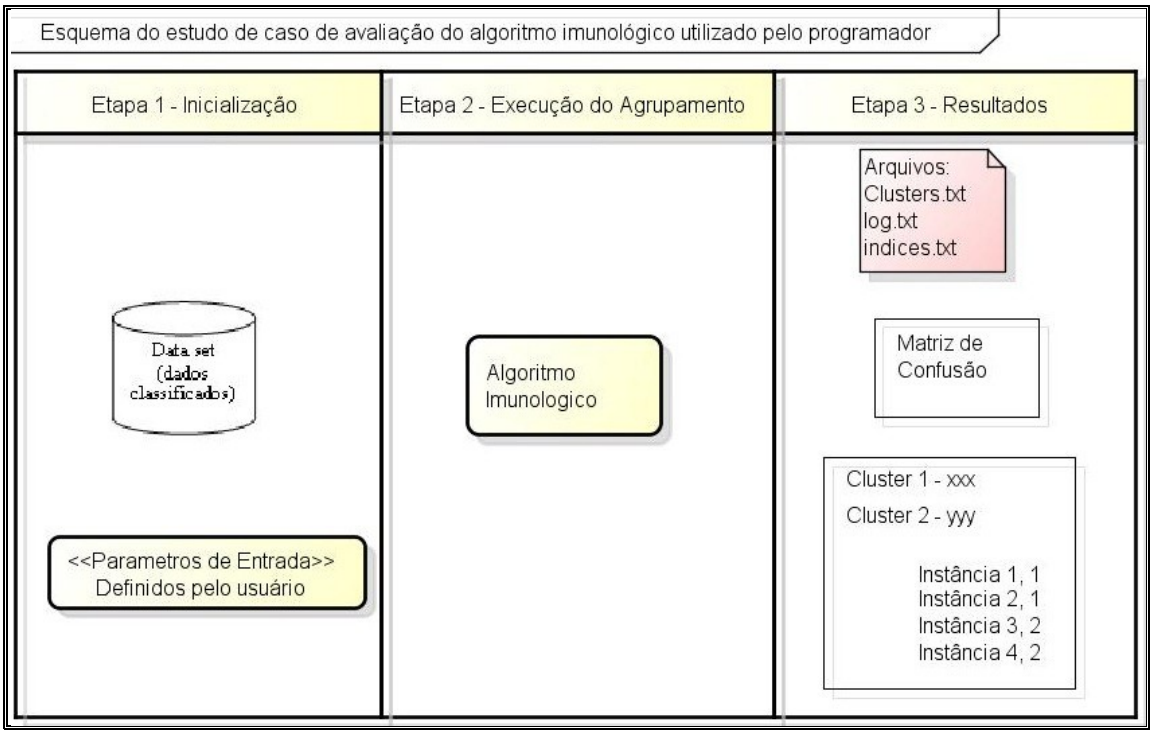


Figura 33 - Esquema do estudo de caso de avaliação do algoritmo imunológico utilizado pelo programador

A figura 34 ilustra o ambiente final utilizado pelo usuário, que informa o dataset que deseja agrupar e os parâmetros de entrada do programa, em seguida o algoritmo irá processar essas informações resultando em um arquivo texto chamado Clusters.txt contendo os grupos formados.

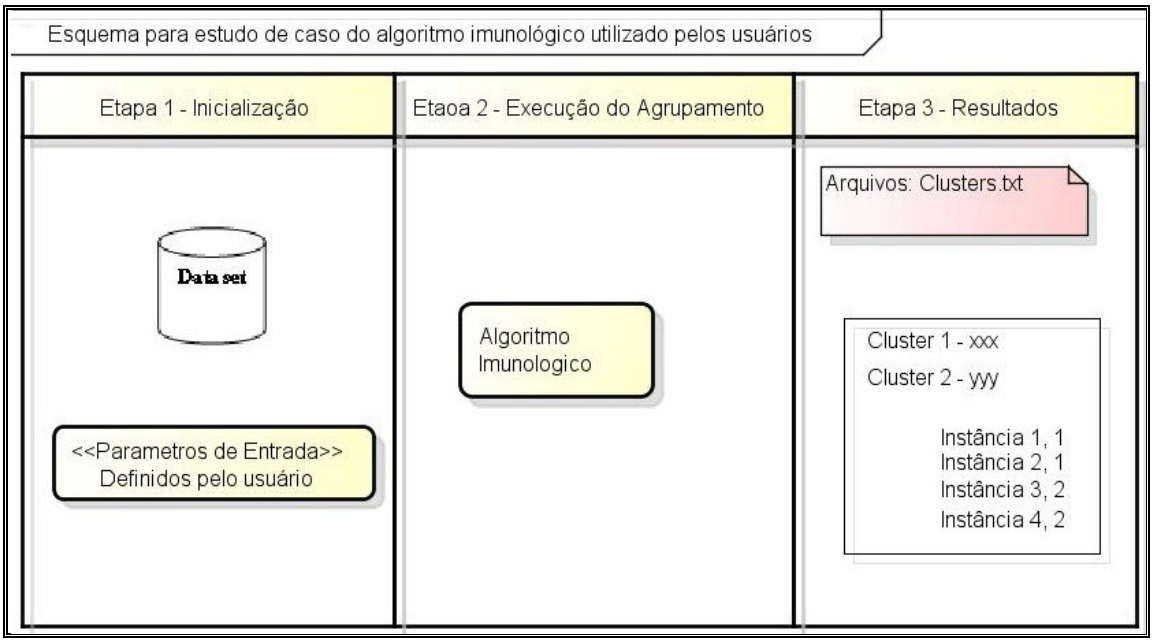


Figura 34 - Esquema do estudo de caso do algoritmo imunológico utilizado pelos usuários

4.5 CONSIDERAÇÕES FINAIS

Foi proposto um novo algoritmo imunológico para agrupamento de dados baseado na teoria da seleção clonal. O algoritmo foi construído conforme formalismos definidos por Holland (1975). O objetivo principal do algoritmo é formar grupos conforme suas similaridades através de datasets escolhidos, levando em consideração uma maior homogeneidade entre elementos do mesmo grupo e uma maior heterogeneidade entre elementos de grupos distintos. Foi definido um fluxograma e um pseudo-código para apresentação do algoritmo, bem como as principais teorias e fórmulas que serão utilizadas para implementação do algoritmo apresentado no próximo capítulo.

5 ALGORITMO IMUNOLÓGICO

O algoritmo imunológico foi desenvolvido com o objetivo de apresentar uma nova forma para agrupar dados, utilizando para isso, as teorias do SIA. Para seu desenvolvimento foi escolhida a linguagem Java. O sistema foi desenvolvido em camadas e sua interface é executada em Desktop sendo o resultado alguns arquivos textos. Os testes foram realizados em três datasets principais e foram feitas algumas análises a partir dos dados coletados.

5.1 PLATAFORMA DE DESENVOLVIMENTO

O algoritmo foi desenvolvido na linguagem Java, visto suas inúmeras vantagens como portabilidade, robustez e segurança. Para desenvolvimento foi escolhida a plataforma Eclipse 3.4. Os testes foram realizados em uma máquina Intel Core 2 Duo, Windows 7 com 4GB RAM e 320GB HD. Foi utilizada a biblioteca Swing + Substance para desenvolvimento da interface e o código foi desenvolvido em três camadas conforme a figura 35. A camada de apresentação será responsável por exibir as informações para o usuário, a camada de negócio irá conter as principais classes do sistema e será responsável pelo processamento do algoritmo e a camada de persistência será responsável pelas leituras e escritas em arquivo. Além disso, o sistema possui ainda a camada de transporte, onde irão transitar os objetos importantes para o sistema e uma pasta onde irão estar todos os arquivos resultantes e os arquivos que deverão ser processados.

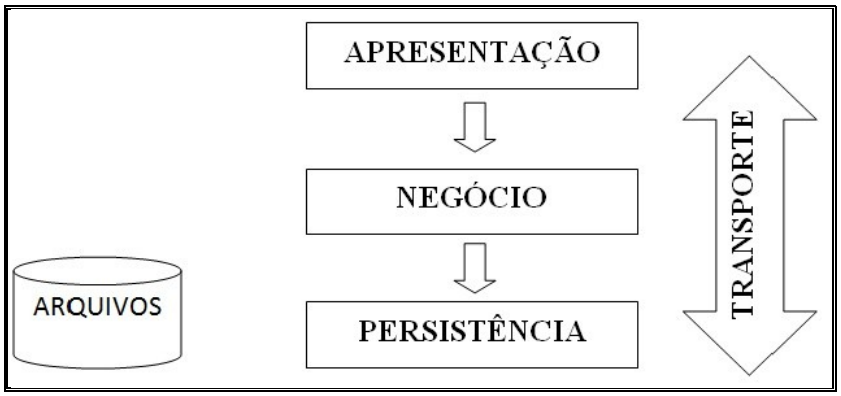


Figura 35 - Plataforma de desenvolvimento

5.2 PACOTE DE CLASSES DO SISTEMA

Na figura 36 pode-se visualizar o diagrama de pacotes onde é demonstrada a arquitetura geral do sistema. Um pacote pode representar um conjunto de classes e tem por finalidade organizar os elementos em módulos. Assim um dos pacotes representa a camada de apresentação, outro a camada de persistência, outro a camada de dados que possui subpacotes, e um último pacote que representa a camada de negócio com um subpacote para as fórmulas. Cada um deles possui suas classes, que têm responsabilidades distintas dentro do sistema.

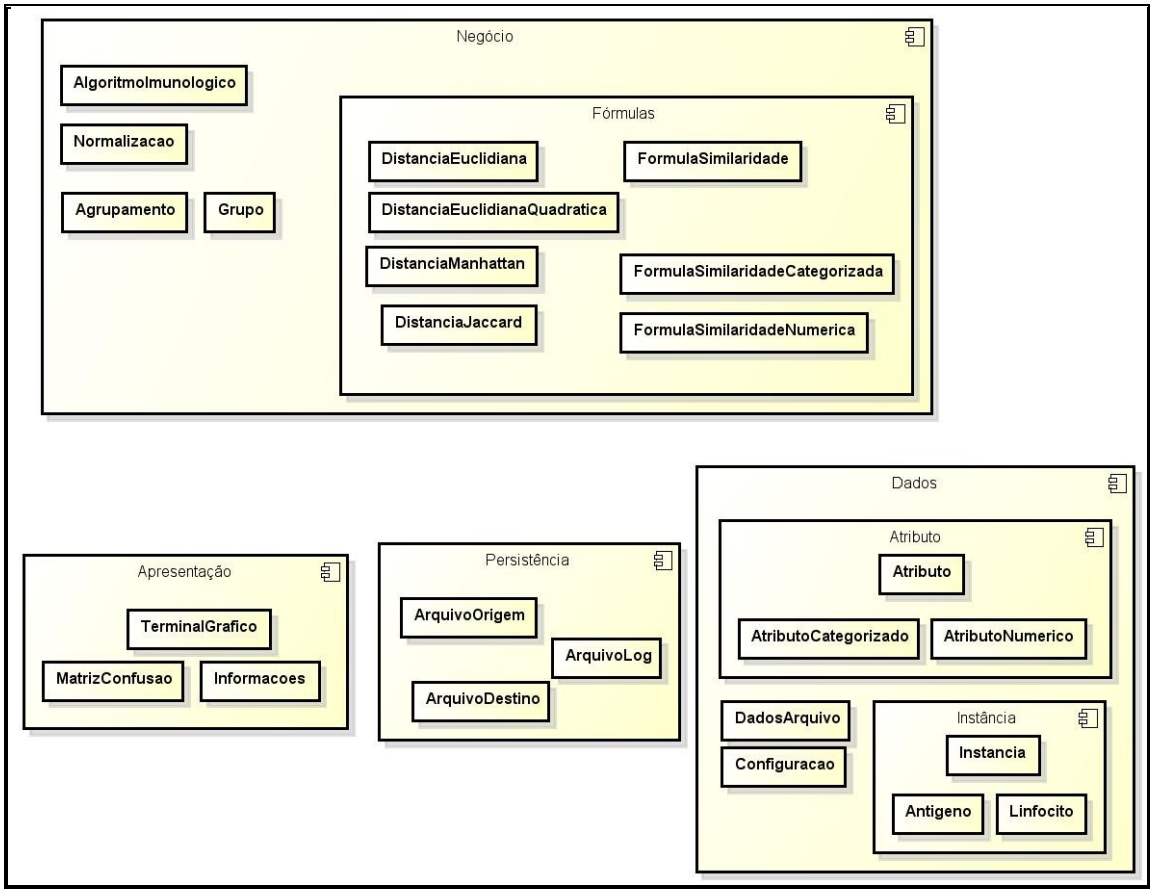


Figura 36 - Diagrama de pacotes

A camada de persistência possui três classes principais: *ArquivoLog*, *ArquivoDestino* e *ArquivoOrigem*. A classe *ArquivoLog* é responsável por criar dois arquivos que podem ser encontrados na raiz do projeto em uma pasta chamada "*ArquivoLog*". O primeiro arquivo chamado log.txt (visto no anexo A) possui todos os principais passos executados pelo algoritmo durante seu processamento. Já o segundo arquivo chamado indices.txt (visto no anexo B) conterá todos os índices de validação de homogeneidade e separação para cada uma das iterações executadas pelo algoritmo. Estes índices servirão

para construção dos gráficos na análise dos resultados. A classe *ArquivoDestino* é responsável por criar o arquivo final chamado cluster.txt (visto no anexo C), que irá conter todos os grupos formados. Este arquivo poderá ser encontrado na raiz do projeto em uma pasta chamada "*ArquivoDestino*". A classe *ArquivoOrigem* é a responsável pela leitura do arquivo original e inserção do seu conteúdo nos objetos que irão ser transportados para o restante do sistema.

A camada de transporte pode ser vista na pasta "*Dados"*. Esta camada contém os objetos que possuem as informações necessárias para formar os grupos. Nesta pasta teremos os dados do arquivo e os dados informados pelo usuário na classe de *Configuração*. Além disso, cada linha do dataset será representada por um objeto chamado *instância* e cada instância pode ser de dois tipos: ou será um *antígeno* (que é o dado do arquivo original) ou é um *linfócito* criado a partir do antígeno e utilizado para formar os grupos. Cada antígeno/linfócito irá possuir uma lista de atributos que poderão ser numéricos ou categorizados conforme o caso, e estes serão representados pela classe *Atributo*.

A camada de negócio contém as principais classes do programa. Dentro desta temos uma pasta chamada "*Formulas*" que possui as classes para cálculo das distâncias entre os elementos. Cada classe implementa uma fórmula diferente, e a escolha da fórmula a ser aplicada será realizada pelo usuário. Na camada de negócio temos também uma classe chamada *Normalizacao* que é responsável pela normalização dos dados utilizando a fórmula vista na seção 4.3.5, e esta será aplicada se o usuário assim desejar. Finalmente nesta camada temos três classes especiais: a classe de *Agrupamento* que contém os métodos relacionados ao agrupamento, como os cálculos de índices e o cálculo para formar os grupos propriamente ditos. Esta classe é representada por um objeto mapa [linfocito:grupo]. Assim, se tivermos três grupos, teremos um mapa[linfocito1:grupo1; linfocito2:grupo2;linfocito3:grupo3]. O grupo é um objeto representado pela classe *Grupo,* onde temos os principais métodos relacionados ao grupo como a classificação do mesmo e o cálculo da homogeneidade do grupo. Além disso, essa classe terá uma lista de antígenos que formam os grupos.

A principal classe do algoritmo é chamada *AlgoritmoImunologico***.** Ela possui todos os passos para o processamento do algoritmo, fazendo a chamada de cada uma das classes conforme o algoritmo for sendo executado.

A camada de apresentação é responsável pela exibição das telas para o usuário. A classe *TerminalGrafico* é responsável por montar a tela inicial com os parâmetros a

serem informados. A classe *Informacoes* é responsável por montar uma tela com algumas das principais informações deste trabalho como nome, data entre outros. E a classe *MatrizConfusao* é responsável por exibir o resultado do agrupamento para o usuário em forma da matriz de confusão entre outras informações relevantes, como os índices de homogeneidade e separação final, o caminho dos arquivos resultantes, etc.

O anexo D apresenta o diagrama de classes do algoritmo imunológico, ilustrando todas as classes do sistema e suas dependências.

5.3 INTERFACE

O algoritmo imunológico foi desenvolvido para Desktop, utilizando a biblioteca Swing do Java, juntamente com uma biblioteca chamada Substance, a qual possui algumas funcionalidades a mais como, por exemplo, deixar o usuário selecionar o layout que gostaria de utilizar. Esta biblioteca possui 27 layouts distintos. Na figura 37 temos um exemplo da janela e de dois tipos de layout que podem ser selecionados pelo usuário: Autumn e Business. Para selecionar um novo layout basta clicar com o botão direto em cima do símbolo do Java, que abrirá o menu de opções. Em seguida selecione "*Aparência do Substance*" e o layout desejado, a janela irá atualizar automaticamente.

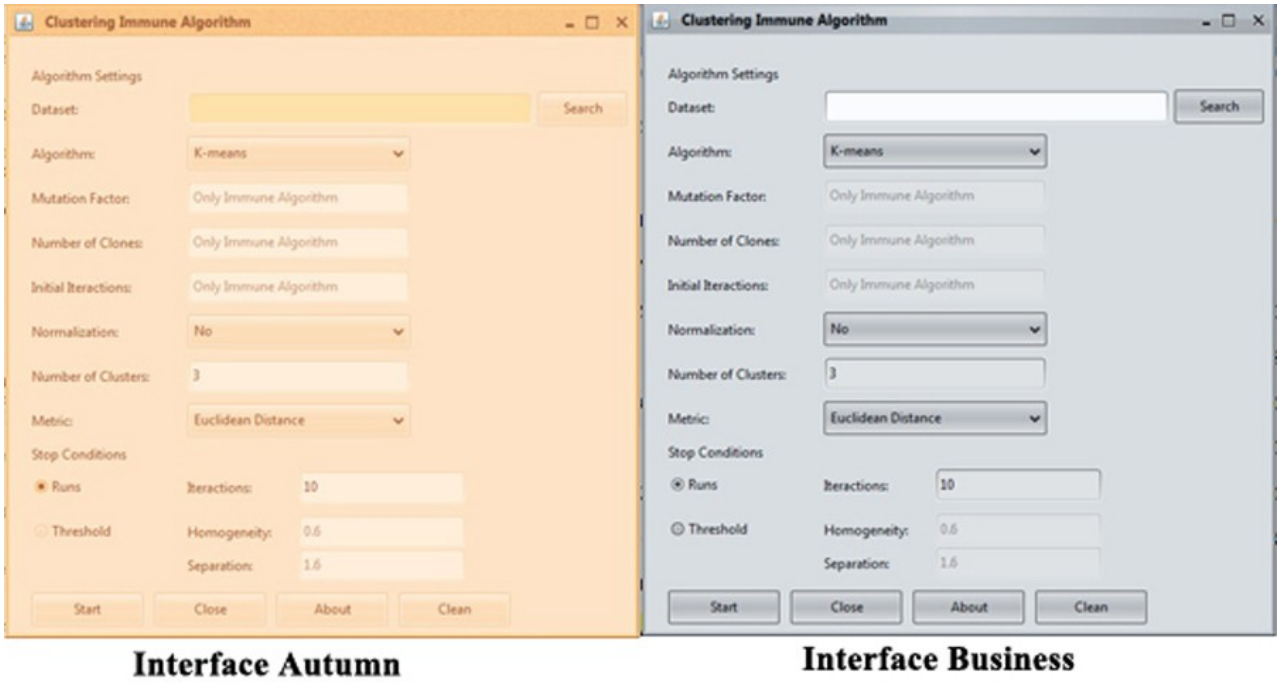


Figura 37 - Interfaces da biblioteca substance

Na figura 38 temos a primeira tela do sistema na qual o usuário irá informar os

parâmetros de entrada para que o algoritmo seja processado. O primeiro parâmetro a ser informado é o dataset, ao qual o usuário deseja que o algoritmo rode. O segundo parâmetro que poderá ser informado pelo usuário, é se ele deseja rodar o algoritmo variante de k-means ou o imunológico. Se ele selecionar o algoritmo imunológico três opções a mais serão habilitadas para serem preenchidas. O Mutation Factor que será o percentual de mutação a ser aplicado em cima dos atributos dos linfócitos, o Number of Clones que define o número de clones a ser feito de cada um dos linfócitos e Initial Iteractions que é o número inicial de iterações para ser realizado com o algoritmo variante do k-means.

Após, o usuário poderá selecionar se deseja ou não realizar a normalização nos dados originais. Um outro parâmetro é o Number of Clusters, que indica quantos linfócitos serão criados a partir do arquivo original. É necessário informar também o cálculo de distância a ser aplicado na formação dos grupos, através do parâmetro Metric. E por último temos as Stop Conditions, ou seja, as condições de parada, que podem ser de dois tipos: ou por número de iterações ou por limiar. Se for selecionado Runs, deve ser informado o número de iterações que o algoritmo irá executar. Caso seja selecionado threshold deve ser informado um índice de homogeneidade e separação, para que o algoritmo processe até atingir estes índices.

Esta tela também apresenta quatro botões. O botão Start inicia o processamento do algoritmo, o botão Close irá parar a execução e fechar o programa, o botão About abre uma tela com mais informações sobre o programa e o botão Clean limpa todos os dados informados voltando ao estado inicial do programa.

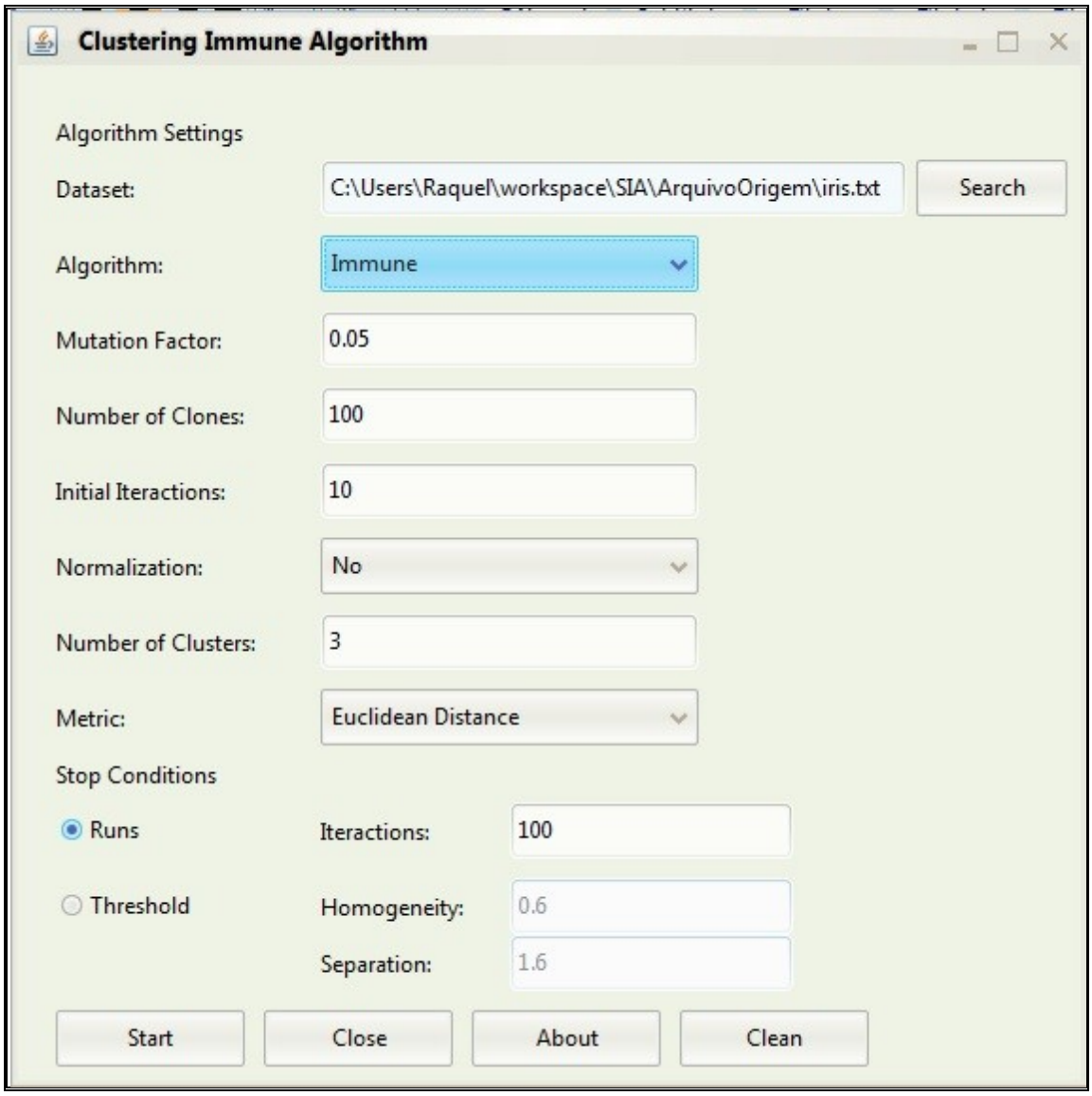


Figura 38 - Tela inicial do algoritmo imunológico

Na figura 39 temos a tela de informação onde são apresentadas algumas informações adicionais do programa como local, nome dos autores, data e contato.

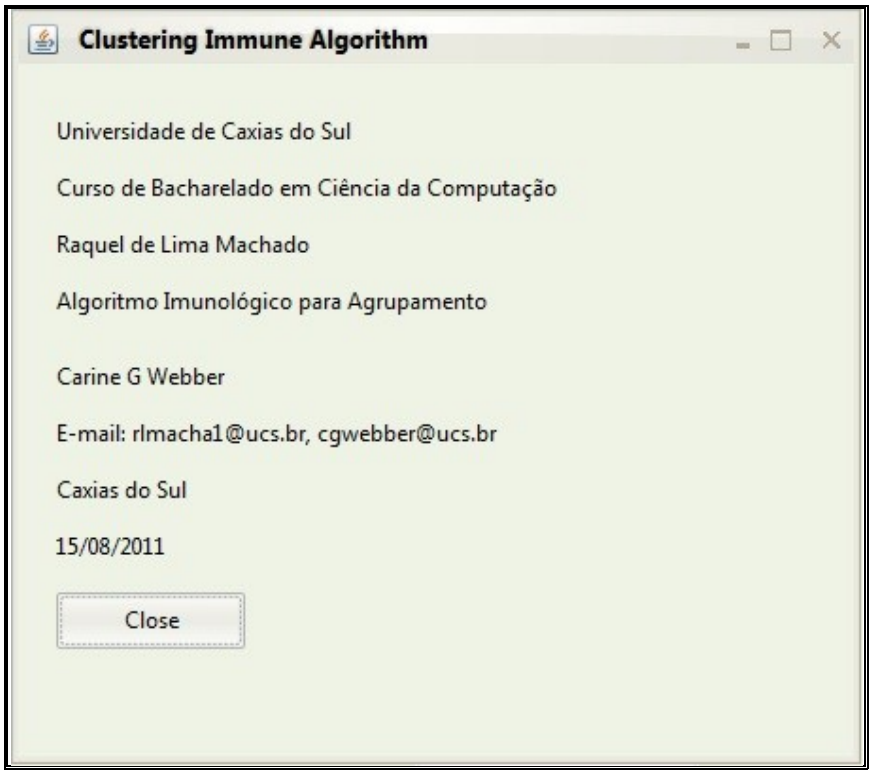


Figura 39 - Tela de informações do algoritmo imunológico

Na figura 40 temos a tela final do programa, onde são apresentadas algumas das opções selecionadas pelo usuário, o tempo que o algoritmo demorou a ser processado, o número de iterações e índices de validação final de homogeneidade e separação. Esta tela também apresenta a matriz de confusão que pode ser utilizada para análise dos resultados. Ela irá exibir o número de instâncias que cada um dos grupos formou. Abaixo da matriz de confusão é exibido o caminho onde se encontram os arquivos resultantes: o arquivo de log e o arquivo de clusters. Podemos ver também nesta tela o número de instâncias que cada um dos grupos possui originalmente através do Original Dataset.

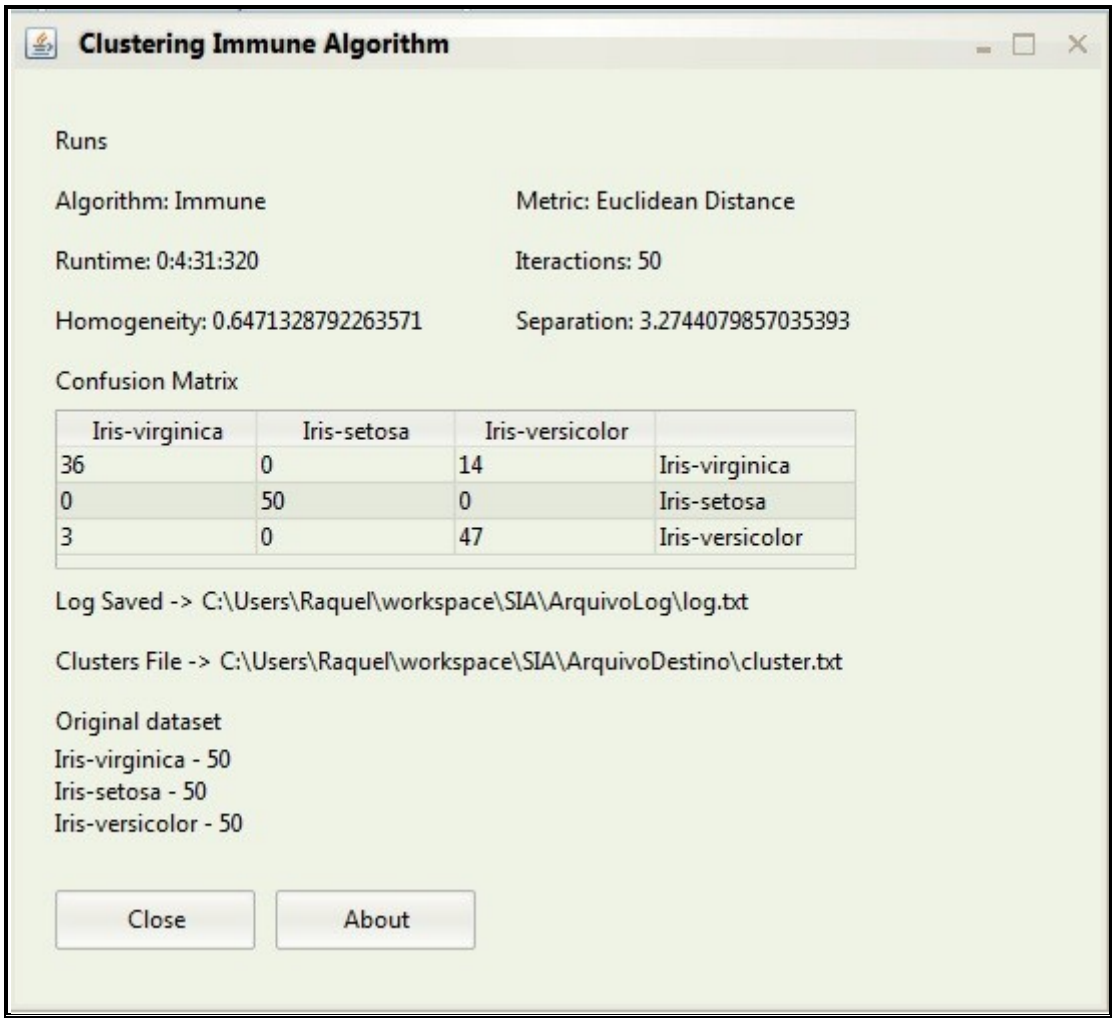


Figura 40 - Tela de resultados do algoritmo imunológico

5.4 CENÁRIO DE USO

Para análise dos resultados foram selecionados três datasets distintos: Iris Plants Database, Wisconsin Breast Cancer Database e Diabetes Data Set. Todos são de acesso público e estão disponíveis no repositório da Universidade da Califórnia em Irvine "*UCI Machine Learning Repository*" (NEWMAN et al, 1998). Todos os datasets testados são nominais, ou seja, possuem atributos reais e um atributo classificador categórico. A partir destes datasets podemos analisar o funcionamento do algoritmo e avaliar seu desempenho. Os dados coletados para todos os datasets foram os melhores resultados apresentados em uma série de execuções de cada cenário.

5.4.1 PREPARAÇÃO DOS DADOS

O algoritmo imunológico processa arquivos de texto que tenham dados a serem agrupados. O arquivo deve ter todos seus atributos reais separados por vírgula. O formato do arquivo deve ser conforme a figura a seguir:

```
Atributo1, Atributo2, Atributo3, AtributoX, Classe
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
6.2,2.2,4.5,1.5,Iris-versicolor
5.6,2.5,3.9,1.1,Iris-versicolor
5.9,3.2,4.8,1.8,Iris-versicolor
6.1,2.8,4.0,1.3,Iris-versicolor
6.3,2.5,4.9,1.5,Iris-versicolor
6.4,3.2,5.3,2.3,Iris-virginica
6.5,3.0,5.5,1.8,Iris-virginica
7.7,3.8,6.7,2.2,Iris-virginica
7.7,2.6,6.9,2.3,Iris-virginica
6.0,2.2,5.0,1.5,Iris-virginica
```

É importante lembrar que todas as linhas devem ter a mesma quantidade de atributos e todos eles devem ser do mesmo tipo senão será lançada uma exceção em tela para o usuário.

5.4.2 BASE DE DADOS ÍRIS

A base Iris Plants Database é bastante utilizada na análise dos resultados e foi criada por R. A. Fisher em 1988. Esta base é composta por 150 amostras e possui 4 atributos: tamanho e largura da sépala e tamanho e largura da pétala. A tabela 10 apresenta a distribuição dos dados por classe que podem ser de três tipos: Iris Setosa, Iris Versicolour e Iris Virginica.

Tabela 10 - Distribuição de dados da Iris por classe

Instâncias	Classe
50	Iris Setosa
50	Iris Virginica
50	Iris Versicolour

A primeira análise em relação à base da Iris foi uma comparação entre o algoritmo k-means e o algoritmo imunológico apresentada na tabela 11. Os algoritmos

foram executados para 100, 300 e 500 iterações e temos duas métricas de distância para cada um deles: a distância euclidiana e a distância euclidiana quadrática. A tabela apresenta o total de instâncias corretas classificadas para cada um dos três grupos. Considere “Vi” como Iris Virginica, “Se” como Iris Setosa, “Ve” como Iris Versicolour e “To” como o total de instâncias classificadas corretamente por cada um deles. Os dados representam os melhores resultados para uma série de execuções.

Tabela 11 - Comparação entre o algoritmo k-means e imunológico através do agrupamento

Algoritmo	**Métrica**	**Iterações**											
		100				**300**				**500**			
		Vi	**Se**	**Ve**	**To**	**Vi**	**Se**	**Ve**	**To**	**Vi**	**Se**	**Ve**	**To**
K-means	Distância Euclidiana	49	50	36	*135*	45	50	47	*142*	50	11	0	*61*
	Distância Euclidiana Quadrática	49	50	35	*134*	50	38	0	*88*	18	50	46	*114*
Imunológico	Distância Euclidiana	28	50	49	*127*	40	50	36	*135*	49	50	22	*121*
	Distância Euclidiana Quadrática	36	50	48	*134*	34	50	46	*130*	48	50	43	*141*

Analisando os dados da tabela 11, verifica-se que a maior taxa de acerto foi de 142 instâncias e foi realizada pelo algoritmo k-means, com 300 iterações para a métrica de distância euclidiana. E a segunda maior taxa foi realizada pelo algoritmo imunológico de 141 instâncias para 500 iterações com a métrica de distancia euclidiana quadrática. Enquanto a pior taxa foi de 61 instâncias, realizada pelo K-means, com 500 iterações para distância euclidiana. É possível verificar que ao aumentar o número de iterações o algoritmo imunológico se mostra mais eficiente em formar os grupos e que se for feita uma média dos números de todas as iterações o algoritmo imunológico se saiu melhor que o k-means.

Observa-se também que em alguns casos os algoritmos de agrupamento podem não reconhecer uma classe (veja a execução de 300 e 500 iterações do k-means). Isso pode acontecer, pois se trata de uma técnica não-determinística onde a inicialização das centróides é feita aleatoriamente. Se ela inicia bem é mais provável que os grupos sejam bem formados, do contrário é provável que eles não consigam encontrar todos os grupos conforme solicitado.

Uma segunda análise que pode ser feita em relação aos dados coletados é através dos índices de homogeneidade e separação. As figuras 41 e 42 apresentam gráficos comparando o algoritmo k-means com o imunológico, utilizando a distância euclidiana e euclidiana quadrática respectivamente, com 500 iterações.

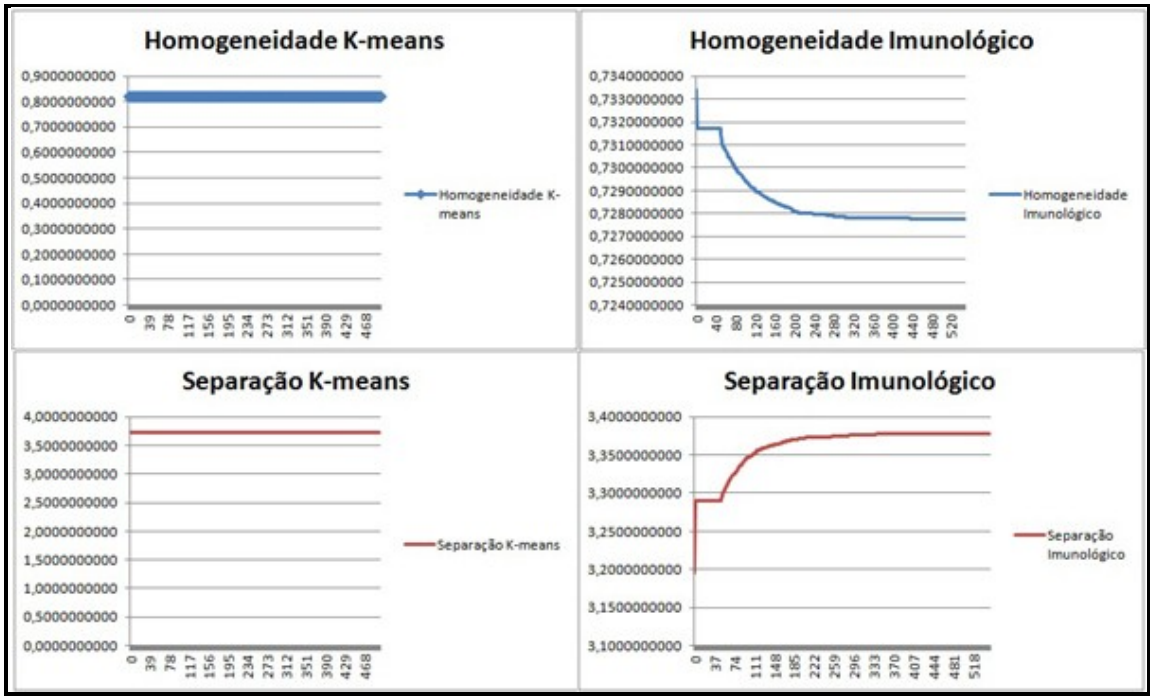


Figura 41 - Índice de homogeneidade e separação para distância euclidiana na base da Íris

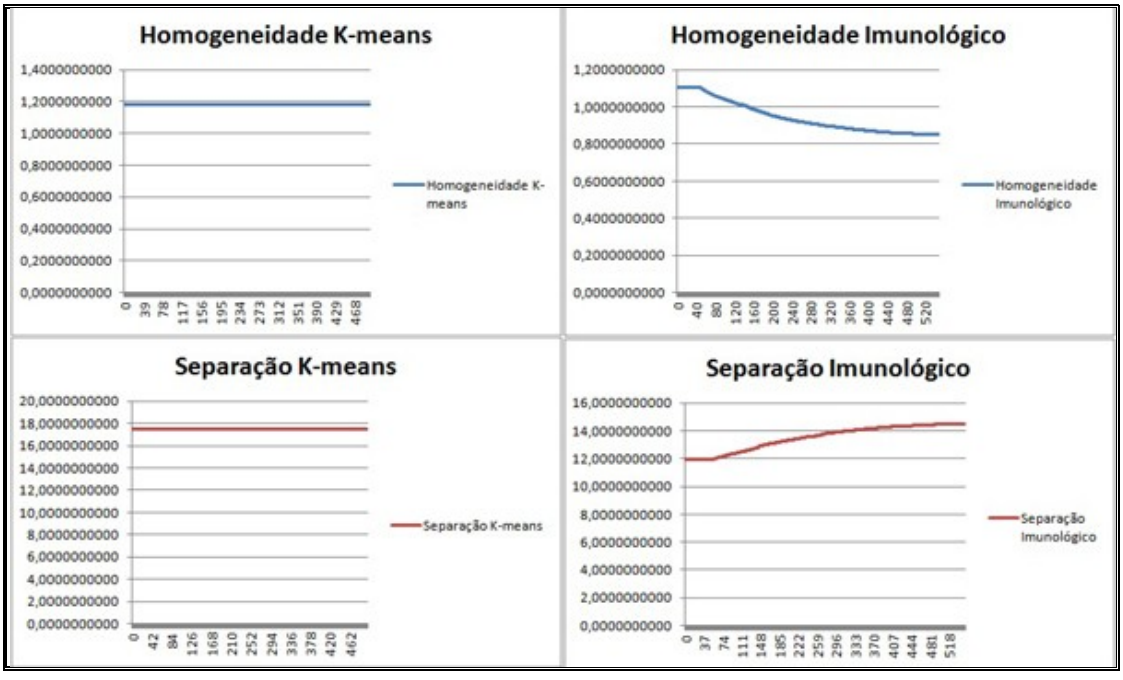


Figura 42 - Índice de homogeneidade e separação para distância euclidiana quadrática na base da Íris

Em relação à homogeneidade o algoritmo k-means se apresenta estável, ou seja, seu índice não muda visto que esse algoritmo tende a convergir rapidamente. Já o algoritmo imunológico apresenta variação no índice, visto que no decorrer das iterações o mesmo vai diminuindo, o que é um ótimo resultado, pois quanto menor a homogeneidade, mais similares são os elementos do grupo. Analisando o gráfico da separação pode-se ver que o k-means se apresenta estável novamente, enquanto o imunológico possui uma variação significativa. Seu índice vai aumentando conforme o algoritmo está sendo processado, o que é um ótimo sinal, pois quanto maior a separação, maior a heterogeneidade dos grupos.

Uma terceira análise que pode ser realizada comparando os dois algoritmos é através do tempo. A tabela 12 possui os tempos para os dois algoritmos, levando em conta as distâncias euclidiana e euclidiana quadrática para 100, 300 e 500 iterações.

Tabela 12 - Comparação entre o algoritmo k-means e imunológico através do tempo

Algoritmo	**Métrica**	**Iterações por tempo *(minuto:segundo:milissegundo)***		
		100	**300**	**500**
K-means	Distância Euclidiana	*0:00:283*	0:01:140	*0:01:506*
	Distância Euclidiana Quadrática	*0:00:338*	0:00:881	*0:01:410*
Imunológico	Distância Euclidiana	*0:20:370*	4:23:120	*16:55:826*
	Distância Euclidiana Quadrática	*0:20:400*	4:33:610	*17:27:449*

Analisando os tempos verifica-se que o algoritmo imunológico é mais lento que o k-means. Isto ocorre porque o algoritmo imunológico trabalha com iterações por clone e o k-means trabalha apenas com iterações. Portanto para cada clone do algoritmo imunológico é feito uma iteração, se foi selecionada 100 iterações e 100 clones, resulta em um total de 10.000 iterações.

Como dito no capítulo 4.3.3, a normalização dos dados poderá ou não ser realizada pelo usuário, visto que, apesar de ser uma boa prática não é muito utilizada. Uma análise interessante a ser realizada é verificar se a normalização dos dados traz algum impacto direto para o algoritmo imunológico. A tabela 13 apresenta uma comparação dos dados com e sem normalização para o algoritmo imunológico, com os parâmetros de 100 clones para 200 iterações. Esses dados foram coletados para a distância euclidiana e euclidiana quadrática.

Tabela 13 - Comparação para o algoritmo imunológico através da normalização

Normalização	**Métrica**	**Vi**	**Se**	**Ve**	**Total**

Sim	Distância Euclidiana	46	50	39	*135*
	Distância Euclidiana Quadrática	50	49	7	*106*
Não	Distância Euclidiana	36	50	47	*133*
	Distância Euclidiana Quadrática	36	50	47	*133*

Utilizando a normalização é possível chegar a dados mais bem classificados, como no caso da distância euclidiana, porém, isso não é uma regra visto que, para dados não normalizados a diferença não foi muita.

Foram feitas mais algumas análises relevantes para o algoritmo imunológico. É difícil saber o fator de mutação e o número de clones a se utilizar em cada caso, assim foram testados alguns fatores de mutação e alguns números de clones diferentes para o algoritmo, para verificar a diferença de um para o outro no caso da base da Iris.

Em um primeiro momento será feita análise do fator de mutação. A tabela 14 apresenta diferentes fatores de mutação aplicados no algoritmo imunológico para distância euclidiana e euclidiana quadrática, para 100, 300 e 500 iterações. Nesta tabela pode-se visualizar o total de instâncias corretas classificadas para cada um dos três grupos. Considere "Vi" como Iris Virginica, "Se" como Iris Setosa, "Ve" como Iris Versicolour e "To" como o total de instâncias classificadas corretamente por cada um deles.

Tabela 14 - Comparação do fator de mutação no algoritmo imunológico

Fator da Mutação	**Métrica**	**Iterações**											
		100				**300**				**500**			
		Vi	**Se**	**Ve**	**To**	**Vi**	**Se**	**Ve**	**To**	**Vi**	**Se**	**Ve**	**To**
0.05	Distância Euclidiana	49	50	25	*124*	50	50	16	*116*	22	50	47	*119*
	Distância Euclidiana Quadrática	36	50	48	*134*	47	50	38	*135*	32	50	49	*131*
0.01	Distância Euclidiana	36	50	47	*133*	36	50	48	*134*	49	50	31	*130*
	Distância Euclidiana Quadrática	13	50	49	*102*	49	50	43	*142*	36	50	47	*133*
Sem mutação	Distância Euclidiana	36	50	48	*134*	47	50	42	*139*	50	27	0	*77*
	Distância Euclidiana Quadrática	36	50	48	*134*	32	50	49	*131*	47	50	37	*134*

Para a base de dados da Iris, pode-se verificar que o fator de mutação 0.01 é mais eficiente do que o fator de mutação 0.05, e que mesmo aplicando o algoritmo sem a mutação os dados foram relativamente satisfatórios. A maior taxa de acerto foi de 142 instâncias ao utilizar um fator de 0.01 na distância euclidiana quadrática, e a pior taxa foi de 77 instâncias para distância euclidiana no caso de não utilizar-se nenhuma mutação.

A segunda análise será sobre o impacto do número de clones no algoritmo imunológico. Na tabela 15 constam os dados coletados para 100, 200 e 300 clones nas distâncias euclidiana e euclidiana quadrática, com 100, 300 e 500 iterações. A tabela apresenta o total de instâncias corretas classificadas para cada um dos três grupos. Considere "Vi" como Iris Virginica, "Se" como Iris Setosa, "Ve" como Iris Versicolour e "To" como o total de instâncias classificadas corretamente por cada um deles.

Tabela 15 - Comparação do número de clones no algoritmo imunológico

Número de Clones	**Métrica**	**Iterações**											
		100				**300**				**500**			
		Vi	**Se**	**Ve**	**To**	**Vi**	**Se**	**Ve**	**To**	**Vi**	**Se**	**Ve**	**To**
100	Distância Euclidiana	49	50	25	*124*	50	50	16	*116*	22	50	47	*119*
	Distância Euclidiana Quadrática	36	50	48	*134*	47	50	38	*135*	32	50	49	*131*
200	Distância Euclidiana	49	50	36	*135*	11	50	47	*108*	49	50	27	*126*
	Distância Euclidiana Quadrática	49	50	36	*135*	36	50	47	*133*	32	50	49	*131*
300	Distância Euclidiana	28	50	49	*127*	40	50	36	*135*	49	50	22	*121*
	Distância	36	50	48	*134*	34	50	46	*130*	48	50	43	*141*

	Euclidiana Quadrática												

Analisando a tabela pode-se ver que quanto maior o número de clones, melhor será o agrupamento. A maior taxa foi de 141 instâncias para 300 clones em 500 iterações e, a menor taxa foi de 108 instâncias para 200 clones em 300 iterações.

O algoritmo imunológico para base de dados da Iris se comportou bem e foi verificado que, para um melhor agrupamento, foi necessário um maior número de clones e uma taxa de mutação de 0.01. Ao analisar os gráficos com índices de homogeneidade e separação, pode-se verificar que o algoritmo tem um crescimento bem pequeno a cada iteração, então quanto mais iterações melhor. No entanto, cada iteração tem um custo elevado devido ao número de clones informado, o que prejudica no tempo que o algoritmo demora para ser processado.

5.4.3 BASE DE DADOS DO CÂNCER

A base Wisconsin Breast Cancer foi criada pelo Dr. W. H. Wolberg em 1991, no Hospital Universitário de Wisconsin e consiste em informações médicas ligadas ao câncer de mama. Esta base é composta por 699 amostras e possui 9 atributos: *Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell, Bare Nuclei, Bland Chromatin, Normal Nucleoli e Mitoses*. A tabela 16 apresenta a distribuição dos dados por classe que podem ser de dois tipos: Maligno ou Benigno.

Tabela 16 - Distribuição de dados da base de Câncer por classe

Instâncias	Classe
241	Maligno
458	Benigno

A análise em relação a base do Câncer foi uma comparação entre o algoritmo k-means e o algoritmo imunológico e pode ser vista na tabela 17. Os algoritmos foram rodados para 100, 200 e 300 iterações e temos duas métricas de distância para cada um deles: a distância euclidiana e a distância euclidiana quadrática. A tabela apresenta o total de instâncias corretas classificadas para cada um dos dois grupos. Considere “Ma” como Maligno e “Be” como Benigno. Os dados representam os melhores resultados

para uma série de execuções.

Tabela 17 - Comparação entre o algoritmo k-means e imunológico através do agrupamento

Algoritmo	Métrica	Iterações								
		100			200			300		
		Ma	Be	Total	Ma	Be	Total	Ma	Be	Total
K-means	Distância Euclidiana	175	455	*630*	226	447	*673*	217	448	*665*
	Distância Euclidiana Quadrática	218	448	*666*	222	447	*669*	216	448	*664*
Imunológico	Distância Euclidiana	186	455	*641*	223	447	*670*	225	447	*672*
	Distância Euclidiana Quadrática	223	447	*670*	221	447	*668*	221	450	*671*

A maior taxa foi realizada pelo algoritmo k-means que classificou 673 instancias corretamente seguido do algoritmo imunológico com 672 instâncias para 300 iterações, e a pior taxa foi de 630 instâncias realizada pelo algoritmo k-means na distância euclidiana com 100 iterações.

A segunda análise realizada na base do Câncer foi através dos índices de homogeneidade e separação. As figuras 43 e 44 apresentam os gráficos comparando o algoritmo k-means com o imunológico, utilizando a distância euclidiana e euclidiana quadrática respectivamente, com 300 iterações.

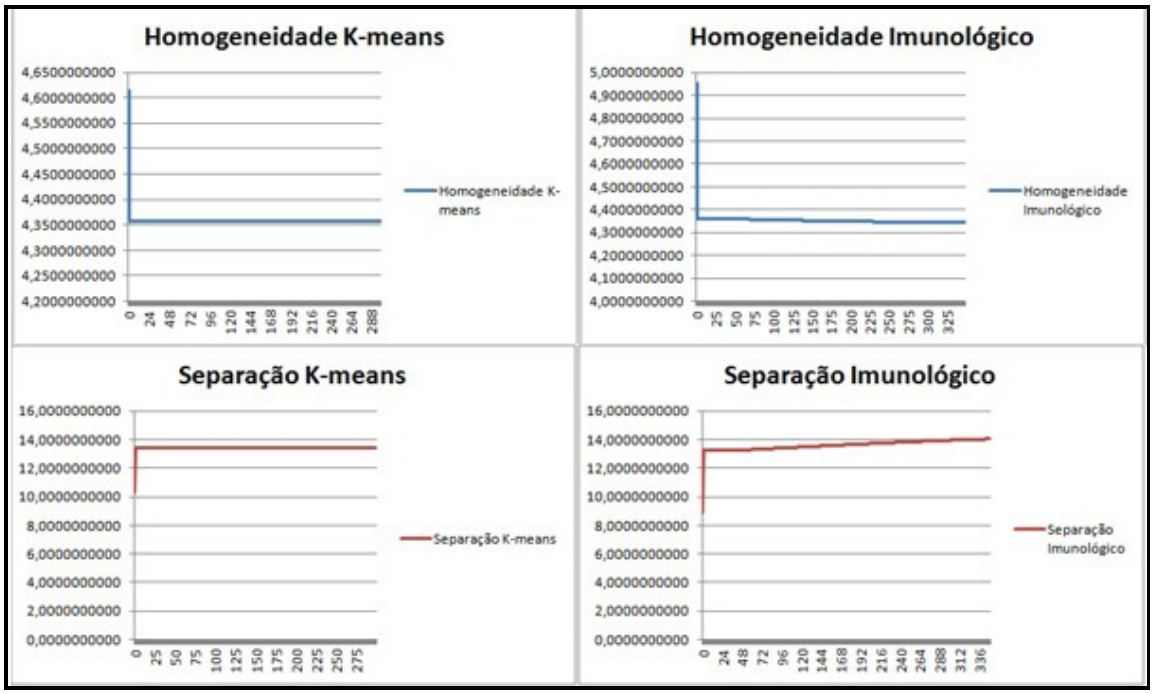


Figura 43- Índice de homogeneidade e separação para distância euclidiana na base do Câncer

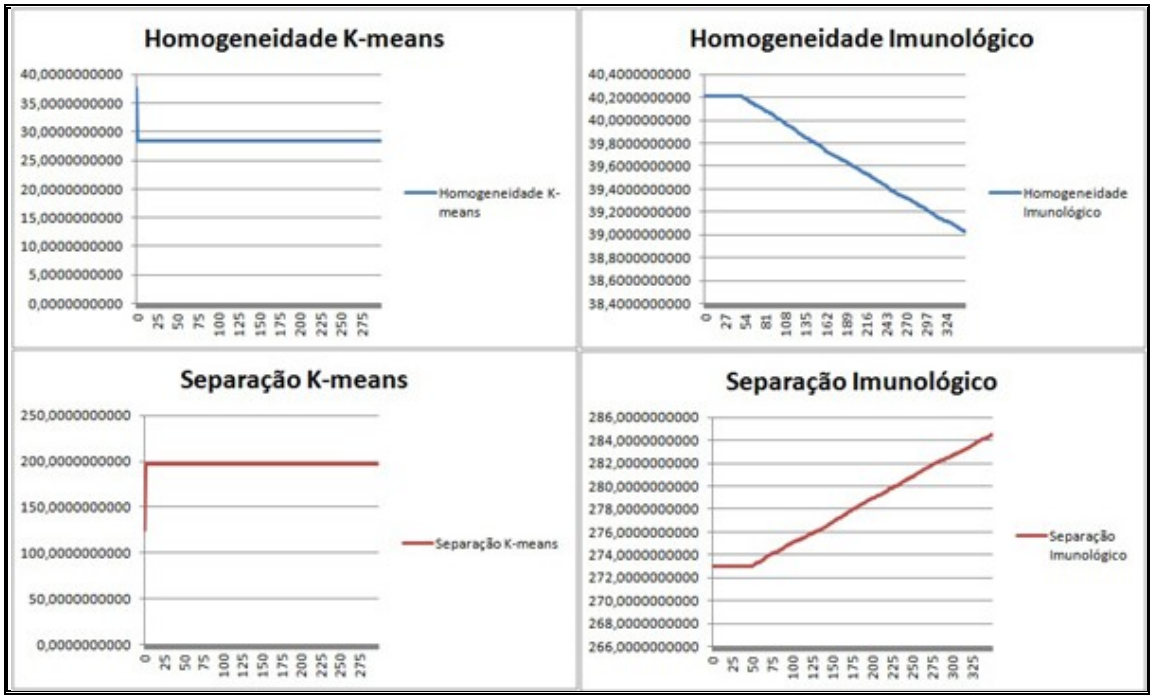


Figura 44 - Índice de homogeneidade e separação para distância euclidiana quadrática na base do Câncer

Em relação a distância euclidiana tanto para homogeneidade como para separação visualiza-se que o algoritmo k-means apresenta uma pequena variação nas primeiras iterações depois se mantém estável, ou seja, seu índice não muda pois ele converge rapidamente. Já o algoritmo imunológico apresenta uma pequena variação no

índice durante as iterações podemos ver que a homogeneidade tende a ir diminuindo e a separação aumentando, o que é bom. Na distância euclidiana quadrática o algorimto k-means novamente apresenta uma pequena variação no início e depois se mantém estável. Já o algoritmo imunológico a medida que vai sendo executado a sua homogeneidade vai caindo e sua separação aumentando, resultando melhores grupos.

Uma terceira análise que pode ser realizada comparando os dois algoritmos é através do tempo. A tabela 18 possui os tempos para os dois algoritmos levando em conta as distâncias euclidiana e euclidiana quadrática para 100, 200 e 300 iterações.

Tabela 18 - Comparação entre o algoritmo k-means e imunológico através do tempo

Algoritmo	**Métrica**	**Iterações por tempo *(minuto:segundo:milissegundo)***		
		100	**200**	**300**
K-means	Distância Euclidiana	*0:1:177*	0:2:366	*0:3:321*
	Distância Euclidiana Quadrática	*0:1:152*	0:2:311	*0:3:249*
Imunológico	Distância Euclidiana	*1:21:222*	5:36:368	*13:6:249*
	Distância Euclidiana Quadrática	*1:24:749*	5:26:416	*12:47:113*

Analisando os tempos pode-se verificar que o algoritmo imunológico é mais lento que o k-means, sendo o maior tempo de 12:47:113 para 300 iterações.

Através destas análises pode-se dizer que o algoritmo imunológico se comportou bem para base de dados do câncer e que seu desempenho foi um pouco melhor que o k-means, dado o número de instâncias corretas classificadas. Novamente o algoritmo levou mais tempo para ser executado, porém seus resultados foram satisfatórios.

5.4.4 BASE DE DADOS DA DIABETE

A base de dados da Diabete foi criada pelo Vicente Sigillito em 1990, e pertence ao *National Institute of Diabetes and Digestive and Kidney Diseases*. Esta base é composta por 768 amostras e possui 8 atributos: *Number of times pregnant, Plasma glucose concentration a 2 hours in an oral glucose tolerance test,Diastolic blood pressure, Triceps skin fold thickness, 2-Hour serum insulin, Body mass index, Diabetes pedigree function, Age*. A tabela 19 apresenta a distribuição dos dados por classe que podem ser de dois tipos: Positivo ou Negativo.

Tabela 19 - Distribuição de dados da base de Diabete por classe

Instâncias	Classe
268	Positivo
500	Negativo

A análise em relação a base da Diabete foi uma comparação entre o algoritmo k-means e o algoritmo imunológico, e pode ser vista na tabela 20. Os algoritmos foram rodados para 100, 200 e 300 iterações e temos duas métricas de distância para cada um deles: a distância euclidiana e a distância euclidiana quadrática. A tabela apresenta o total de instâncias corretas classificadas para cada um dos dois grupos. Considere "Pos" como Positivo e "Neg" como Negativo. Os dados representam os melhores resultados para uma série de execuções.

Tabela 20 - Comparação entre o algoritmo k-means e imunológico através do agrupamento

Algoritmo	Métrica	Iterações								
		100			200			300		
		Pos	Neg	Total	Pos	Neg	Total	Pos	Neg	Total
K-means	Distância Euclidiana	104	397	*501*	99	405	*504*	13	490	*503*
	Distância Euclidiana Quadrática	86	421	*507*	86	421	*507*	69	435	*504*
Imunológico	Distância Euclidiana	88	415	*503*	83	421	*504*	57	448	*505*
	Distância Euclidiana Quadrática	86	421	*507*	86	421	*507*	86	421	*507*

A maior taxa foi de 507 instâncias, realizada pelos algoritmos k-means e imunológico, e a pior taxa foi realizada pelo algoritmo k-means com a distância euclidiana para 100 iterações, que encontrou 501 instâncias corretas.

As figuras 45 e 46 apresentam uma análise para os índices de homogeneidade e separação na base da Diabetes. É comparado o algoritmo k-means com o imunológico, utilizando a distância euclidiana e euclidiana quadrática respectivamente, com 300 iterações.

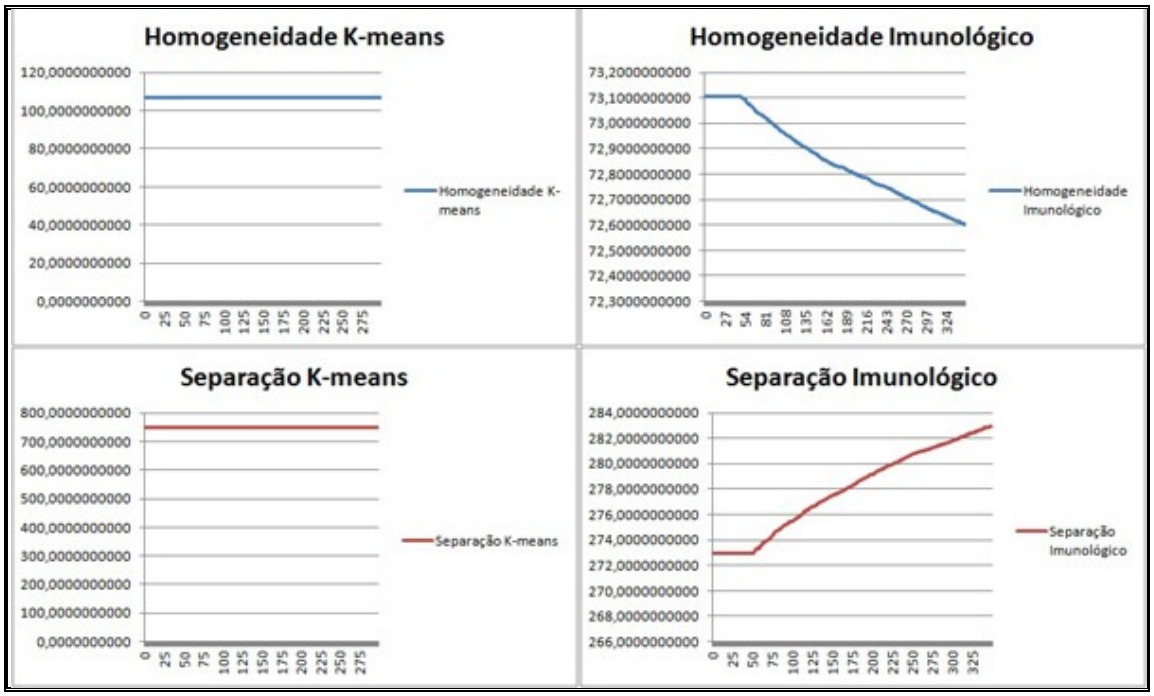


Figura 45 - Índice de homogeneidade e separação para distância euclidiana na base da Diabete

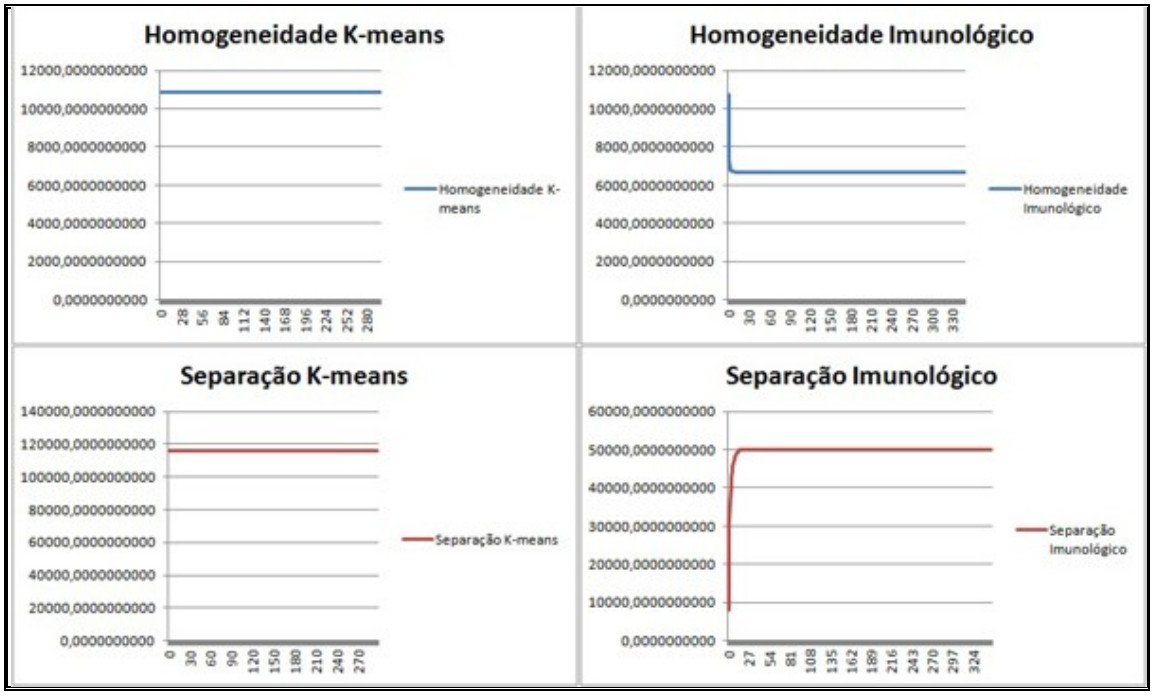


Figura 46 - Índice de homogeneidade e separação para distância euclidiana quadrática na base da Diabete

Para distância euclidiana, analisando o gráfico de homogeneidade, percebe-se que o algoritmo k-means se mantém estável desde o início, enquanto o algoritmo imunológico possui uma grande variação nestes índices. Já utilizando a distância euclidiana quadrática, tanto o algoritmo k-means quanto o imunológico se mantém

estáveis, não apresentando grandes melhorias.

Na tabela 21 temos uma comparação de tempo entre o algoritmo k-means e o imunológico para as distâncias euclidiana e euclidiana quadrática com 100, 200 e 300 iterações.

Tabela 21 - Comparação entre o algoritmo k-means e imunológico através do tempo

Algoritmo	**Métrica**	**Iterações por tempo *(minuto:segundo:milissegundo)***		
		100	**200**	**300**
K-means	Distância Euclidiana	*0:1:239*	0:2:423	*0:4:455*
	Distância Euclidiana Quadrática	*0:1:506*	0:2:297	*0:3:289*
Imunológico	Distância Euclidiana	*1:33:379*	6:29:143	*17:51:249*
	Distância Euclidiana Quadrática	*1:32:690*	6:23:579	*16:14:770*

Analisando os tempos de execução pode-se verificar que o algoritmo imunológico demorou novamente mais que o algoritmo k-means, levando até 17:51:249. Este tempo foi maior devido aos clones realizados e o número de amostras do arquivo ser maior.

Para base de dados da Diabetes o algoritmo imunológico se comportou relativamente bem, porém teve um desempenho próximo ao k-means.

5.5 CONSIDERAÇÕES FINAIS

O algoritmo imunológico foi desenvolvido baseado nas teorias do SIA, a linguagem escolhida foi Java e o código foi desenvolvido em três camadas. Foram realizados testes em três datasets distintos e seus resultados foram analisados. O algoritmo imunológico se comportou satisfatoriamente nos três casos, porém demonstrou ser mais lento que o algoritmo k-means.

6 CONCLUSÕES

6.1 SÍNTESE DO TCC

No mundo atual, onde há uma grande gama de informações a serem manipuladas, o agrupamento de dados se mostra uma técnica muito útil. A tarefa de agrupar dados consiste em formar grupos conforme a similaridade dos elementos de cada grupo, com o objetivo de manter uma alta homogeneidade entre elementos do mesmo grupo e alta heterogeneidade entre grupos. O processo de agrupamento contém cinco etapas básicas: a preparação de padrões onde os dados serão selecionados e normalizados, a escolha de uma medida de similaridade conforme a característica dos dados em questão, a escolha de um algoritmo para o agrupamento conforme o objetivo que se pretende alcançar, a validação dos grupos formados conforme índices pré-definidos e a interpretação dos resultados, muitas vezes realizada por um especialista do problema em questão. O agrupamento esta sendo utilizado nas mais diversas áreas do conhecimento e as técnicas de agrupamento tem sido cada vez mais exploradas para atender os requisitos dos usuários. Como motivação a essa questão, este trabalho propõe um novo algoritmo para agrupamento de dados baseado nos SIA.

Os SIA são responsáveis por criar algoritmos baseados no sistema imunológico dos vertebrados, mais especificamente no sistema adaptativo. O sistema imunológico é um complexo sistema que possui várias características desejáveis em qualquer sistema computacional como, por exemplo, segurança e flexibilidade.

Atualmente existem três principais teorias baseadas nos SIA, através das quais são criados novos algoritmos: a teoria da seleção clonal, o princípio da seleção negativa e a teoria das redes imunológicas. O algoritmo imunológico tem como inspiração as duas primeiras teorias. Os SIA estão sendo utilizados para resolver problemas em diversas áreas, como por exemplo, no reconhecimento de padrões, controle de robôs, sistemas classificadores, na área de agrupamento de dados, dentre outras.

O algoritmo imunológico foi desenvolvido utilizando a linguagem Java e foi feito em três camadas: apresentação, negócio e persistência. O sistema recebe um dataset com as instâncias a agrupar e alguns parâmetros informados pelo usuário e o resultado é um arquivo texto com os grupos formados.

Foram analisadas três bases de dados: Iris Plants Database, Wisconsin Breast Cancer Database e Diabetes Data Set através das quais foram construídas tabelas e gráficos de comparação do algoritmo imunológico com o k-means. Através dos resultados obtidos foi verificado que o algoritmo imunológico foi mais ou igualmente eficiente ao k-means.

6.2 CONTRIBUIÇÕES DO TRABALHO

O presente trabalho atingiu os objetivos especificados e contribui para área da Ciência da Computação. Foi apresentado um novo algoritmo para agrupamento de dados baseado nas teorias do SIA. Apesar de já existirem algoritmos imunológicos para agrupamento, o algoritmo proposto neste trabalho não se baseou fortemente em nenhum outro existente, mas seguiu os princípios da teoria da seleção negativa e o princípio da seleção clonal. Estes dois conceitos não haviam ainda sido integrados para desenvolver novos algoritmos na área de agrupamento.

Foram estudadas novas formas de agrupar os dados levando em consideração os índices de homogeneidade e separação. O objetivo do agrupamento é alcançar uma maior homogeneidade para os elementos do mesmo grupo e uma maior heterogeneidade entre os grupos. Esses dois índices serviram como parâmetro para verificar se os grupos deveriam ser substituídos ou não. Em pesquisas realizadas diz-se que os grupos sempre devem ser substituídos, mesmo que não sejam melhores. Entretanto, com esses dois índices pode-se verificar se os grupos melhoraram e substituir somente quando necessário resultando em um melhor agrupamento.

A área de agrupamento ganhou um novo algoritmo baseado nas teorias do SIA que tem auxiliado na resolução de diversos problemas complexos. O algoritmo demonstrou-se eficiente na construção dos grupos, apresentando resultados satisfatórios para os casos analisados.

6.2 TRABALHOS FUTUROS

Através dos testes realizados e durante a implementação foram anotadas algumas melhorias que podem ser realizadas no algoritmo imunológico. A otimização do algoritmo seria uma delas, pois para algumas iterações o algoritmo se mostrou lento ao

formar os grupos devido ao número de iterações por clone que foi informada. É recomendável estudar uma nova forma para selecionar os linfócitos iniciais, que hoje são selecionados aleatoriamente e podem causar alguns resultados indesejados. A respeito dos parâmetros do algoritmo, principalmente o fator de mutação e o número de clones é uma questão relevante a ser estudada, pois para cada base de dados o usuário terá que informar esses parâmetros e até chegar a um número desejável pode ser que demore, esses parâmetros poderiam ser calculados automaticamente pelo algoritmo.

Devem ser realizados mais testes para o número de iterações que o algoritmo possa executar, para assim verificarmos sua real complexidade. Os testes devem ser feitos seguindo metodologias estatísticas e a definição de uma amostra de dados. É necessário comparar o algoritmo com outros algoritmos de agrupamento além do k-means para analisar melhor o seu desempenho**.**

7 REFÊRENCIAS

ALVES, L**.** *Eficiência de Métodos de Agrupamento de Dados na Modelagem Nebulosa de TAKAGI-SUGENO.* 2007. 143f. Dissertação (Mestrado em Engenharia de Produção e Sistemas) - Pontifícia Universidade Católica do Paraná, Curitiba, Paraná. 2007.

ALMEIDA, A. A. M. *Aplicação de Algoritmos de Agrupamento a Dados Biológicos.* 2004. 72f. Monografia (Ciência da Computação) - Universidade Federal de Alagoas – UFAL. Maceió, Alagoas. 2004.

AMARAL, J. L. M. do. *Sistemas Imunológicos Artificiais Aplicados a Detecção de Falhas.* 2006. Tese (Doutorado em Engenharia Elétrica) - Universidade Católica do Rio de Janeiro, Rio de Janeiro, Rio de Janeiro. 2006.

ANKERST, M.; et al. *OPTICS:* Ordering Points to Identify the Clustering Structure. In: Proceedings of the ACM SIGMOD Conference on Management of Data, New York, USA, v. 28, n. 2, p. 49-60, jun. 1999.

AZEVEDO, P. J. *Clustering***.** Universidade do Minho. 2010. 46 slides, color.

BACKER, E**.** *Computer-assisted reasoning in cluster analysis.* Reino Unido: Prentice Hall International, 1995. 367 p.

BARANAUSKAS, J. A. *Clustering*. São Paulo: Departamento de Física e Matemática-Universidade de São Paulo. 2003. 66 slides, color.

BORGIDON, F. L. *Técnicas Inteligentes para Análise de Agrupamento de Dados.* 2010. 61f. Monografia (Ciência da Computação) - Universidade Federal de Santa Catarina, Santa Catarina. 2010.

BROWNLEE, J. *The Shape Space and affinity landscape Immunological Paradigms.* Australia: Complex Intelligent Systems Laboratory, Centre for Information Technology Research, Faculty of Information Communication Technology, Swinburne. 2007.

BROWNLEE, J. *Clever Algorithms:* Nature-Inspired Programming Recipes. Austrália: LuLu, 2011. 436 p.

BOSCARIOLI, C. *Análise de Agrupamentos baseada na Topologia dos Dados e em Mapas Auto-organizáveis.* 2008. 138f. Tese (Doutorado em Engenharia Elétrica) - Universidade de São Paulo. São Paulo. 2008.

BURNET, F. M. *The Clonal Selection Theory of Acquired Immunity.* Nashville: Vanderbilt University Press, 1959. 488 p.

CARLANTONIO, L. M. *Novas Metodologias para Clusterização de Dados.* 2001. 157 f. Tese (Doutorado em Engenharia Civil). Universidade Federal do Rio de Janeiro, Rio de Janeiro. 2001.

CARVALHO, F. A. T. *Introdução a Análise de Agrupamentos:* Abordagem Numérica e Conceitual. Universidade Federal de Pernambuco. 2002. 84 slides, color.

CASTRO, B. M.; et al. *Raio de Influência*: *um método de agrupamento para análise de cluster.* São Pedro: 19º Simpósio Brasileiro de Probabilidade e Estatística. São Paulo, jul. 2010.

CASTRO, L. N.; ZUBEN, F. J. V. *Articial Immune Systems:* Part I – Basic Theory and

Applications. Campinas: Unicamp, 1999.

CASTRO, L. N.; ZUBEN, F. J. V. *Articial Immune Systems:* Part II – A Survey of Applications. Campinas: Unicamp, 2000.

CASTRO, L. N.; ZUBEN, F. J. V. *aiNet: An Artificial Immune Network for Data Analysis.* USA: Idea Group Publishing. p 231-261. mar. 2001.

CASTRO L.; et al. *Um Sistema Imunológico Artificial Aplicado à Identificação de Spams.* Laboratório de Sistemas Inteligentes, Universidade Católica de Santos. São Paulo. 2006.

CARDOSO, L. O. et al. *Uso do método Grade of Membership na identificação de perfis de consumo e comportamento alimentar de adolescentes do Rio de Janeiro, Brasil.* Cad. Saúde Pública, Rio de Janeiro, v. 27, n 35, p 336-346, fev. 2011.

CHEN, G. et al. *Evaluation and Comparison of Clustering Algorithms in Analyzing ES Cell Gene Expression Data*. Laboratories of Genetics, National Institute on Aging, National Institute of Health, Baltimore, NY, USA. 2002.

COLE, R. M. *Clustering with Genetic Algorithms.* 1998. Thesis (Master of Science). Department of Computer Science. University of Western Austrália, Austrália. 1998

COSTA, K. C. O. *Análise de DFA e de Agrupamento do Perfil de densidade de poços de petróleo.* 2009. 89f. Dissertação (Mestrado em Ciências e Engenharia de Petróleo) - Universidade Federal do Rio Grande do Norte. Natal, Rio Grande do Norte. 2009

DAMASO, J. C. G. *Otimização do Processo de Seleção de Atributos para Agrupamento de Dados.* 2008. 113f. Dissertação (Mestrado em Ciências em Engenharia Civil) - Universidade Federal do Rio de Janeiro. Rio de Janeiro. 2008.

DEMPSTER, A.P. et al. *Maximum Likelihood from Incomplete Data via the EM Algorithm***.** Journal of the Royal Statistical Society, London, UK, v. 39, p. 1-38, dez. 1976.

ESTER, M.; et al. *A density-based algorithm for discovering clusters in large spatial databases with noise***.** In Second International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, p. 226-231, 1996.

FACELI K. *Um framework para análise de agrupamento baseado na combinação multi-objetivo de algoritmos de agrupamento.* 2007. 185f. Tese (Doutorado em Ciência da Computação e Matemática Computacional) - Universidade de São Paulo. São Paulo. 2007.

FISHER, D. H. *Manufactured in The Netherlands Knowledge Acquisition Via Incremental Conceptual Clustering***.** Irvine Computational Intelligence Project, Department of Information and Computer Science, University of California, Irvine, p. 139-172, jul. 1987.

FORREST, S. et al. *Self-Nonself Discrimination in a Computer*. In IEEE Symposium on Research in Security and Privacy, Oakland, USA. p. 202-212, mai.1994.

KASNAR, I. K. ; GONÇALVES, B. M. L. *Técnicas de Agrupamento Clustering.* Rio de Janeiro. EletroRevista Revista Científica e Tecnológica. Instituto Business Consultoria Internacional. 2007.

GARCIA, R. J.; HERNÁNDEZ, J. E. 2010. 107 fls. *Clustering of unhealthy lifestyle behaviors is associated with nonadherence to clinical preventive recommendations among adults with diabetes***.** Preventive Medicine and Public Health Teaching and Research Unit, Health Sciences Faculty, Rey Juan Carlos University, Madrid, Spain.

GUHA, S.; RASTOGI, R.; SHIM, K. *CURE: an efficient clustering algorithm for large*

databases. ACM SIGMOD Record, New York, v. 27, n. 2, jun. 1998.

HAN, J. et al. *Intelligent query answering by knowledge discovery techniques.* IEEE Transactions on Knowledge and Data Engineering, Washington, v.8, n.3, p.373-390, jul. 1996.

HAN, J.; KAMBER, M. *Data Mining: Concepts and Techniques.* San Francisco: Morgan Kaufmann Publishers, 2001. 550 p.

HINNEBURG, A.; GABRIEL, H. H. *DENCLUE 2.0: Fast Clustering based on Kernel Density Estimation*. In Proceedings of the 7th international conference on Intelligent data analysis, Heidelberg, Berlin. 2007.

HRUSCHKA, E. R.; EBECKEN, N. F. F. *A Genetic algorithm for cluster analysis*. Journal Intelligent Data Analysis, Netherlands, v. 7, n. 1, p.15-25, jan. 2003.

HARTIGAN, J. A. *Clustering Algorithms*. New York: Publisher John Wiley & Sons, 351 p. 1975.

HOLLAND, J. H. *Adaptation in natural and artificial systems:* An introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press, 1975.

JAIN, A. K.; DUBES, R. C. *Algorithms for Clustering Data*. New Jersey: Prentice Hall, 1988. 334 p.

JAIN, A.; et al. *Data Clustering: A Review*. Journal: ACM Computing Surveys, Redwood , v. 31, n. 3, p. 264-323, set. 1999.

JANGARELI, M.; et al. *Níveis de significância na identificação de marcadores moleculares no mapeamento genômico***.** Revista Brasileira de Zootecnia, Minas Gerais, v. 40, n. 2, p. 308-313, 2011.

JIA, L.; et al. *Study of Artificial Immune Clustering Algorithm and Its Applications to Urban Traffic Control.* International Journal of Information Technology, Shandong Province, China, v. 12, n. 3, 2006.

JUNIOR, D. L. S.; MANTOVANI, D. M. N. *Comunicação nas redes sociais: Um estudo com usuários das comunidades do Orkut.* Revista Acadêmica da FACE, Porto Alegre, v. 21, n. 1, p. 30-41, jan./jun. 2010.

KOHONEN, T.; et al. *Sompak: the selforganization map program package.* Relatório técnico, Helsinki University of tecnology, Laboratory of computer and information science, Espoo. 1996.

LENZ, A. R. *Utilizando Técnicas de Aprendizado de Máquina para Apoiar o teste de regressão.* 2009. 150 fls. Dissertação (Mestrado em Informática) - Universidade Federal do Paraná. Curitiba. 2009.

LINDEN, R. *Técnicas de Agrupamento***.** Revista de Sistema de Informação da FSMA, Santa Maria, n. 4, p. 18-36. 2009.

LUNA, J. E. O. *Algoritmo EM para Aprendizagem de Redes Bayesianas a partir de dados incompletos.* 2004. 132 fls. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal do Mato Grosso do Sul. Mato Grosso do Sul. 2004.

MAXIMILIANO, A. S. CORDEIRO, M. T. A. 2008. 45 fls. *Partição de grupos e validação da análise de agrupamento para equipamentos de fiscalização eletrônica de trânsito.* Monografia - Universidade Estadual do Paraná. Paraná. 2008

METZ, J. *Interpretação de clusters gerados por algoritmos de clustering hierárquico.* 2006. 152 fls. Dissertação (Mestrado em Ciência da Computação e Matemática Computacional) - Universidade de São Paulo. São Paulo. 2006.

MELLO, C. E. R. *Agrupamento de Regiões: Uma abordagem utilizando acessibilidade.* 2008. 96f. Dissertação (Mestrado em Ciência em Engenharia de Sistema de Computação) - Universidade Federal do Rio de Janeiro. Rio de Janeiro. 2008.

MEIRELES, A. C. M.; et al. *Sustentabilidade do modelo agrícola da bacia do riacho Faé***.** Revista Ciência Agronômica**,** Ceará, Fortaleza, v. 42, n. 1, p. 84-91, jan./mar. 2011.

MEDINA, M.; FERTIG, C. *Algoritmos e Programação:* Teoria e Prática. São Paulo: Editora Novatec, 2005. p 384.

NALDI, M. C. *Técnicas de combinação para agrupamento centralizado e distribuído de dados.* 2010. 276f. Tese (Doutorado em Ciências Matemáticas e de Computação) - Universidade de São Paulo. São Paulo. 2010.

NEWMAN, D.J.; HETTICH,S.; BLAKE, C.; MERZ, C.J. UCI- *Repository of Machine Learning Databases*. 1998. Disponível em: http://archive.ics.uci.edu/ml/datasets.html

OLIVEIRA, T. B. S. *Clusterização de dados utilizando técnicas de redes complexas e computação bioinspirada.* 2008. 112 f. Dissertação (Mestrado em Ciência da Computação e Matemática Computacional) - Universidade de São Paulo, São Paulo. 2008.

OLIVEIRA, L E S. *Inteligência Computacional:* Tipos de Aprendizagem. 2005. Pontifícia Universidade Católica do Paraná. 9 slides, color.

OLIVERIA, D. S.; CORREIA, A. R. *Estudo do desempenho operacional dos aeroportos brasileiros relativo ao movimento de cargas.* Revista de Literatura e Transportes, v. 5, n. 3, p. 141-162, mar./nov. 2010.

PALACIO, H. A. Q.; et al. *Similaridade e fatores determinantes na salinidade das águas superficiais do Ceará, por técnicas multivariadas.* Campina Grande: Revista Brasileira de Engenharia Agrícola e Ambiental, v. 15, n. 4, p. 395-402, jan. 2011.

PRASS, F. S; OGLIARI, P. J. *Avaliação Comparativa de Algoritmo de Análise de Agrupamentos Utilzados em Data Mining.* Santa Maria: Jornada de Pesquisa. 2007.

PEREIRA, D. L. *Estudo do PSO na Clusterização de dados.* 2007. 52f. Monografia (Ciência da Computação). Universidade Federal de Lavras. Minas Gerais. 2007.

PEREIRA, R. R.; et al. *Algoritmos e Programação*. São Paulo: Editora RDS. 2009. 85 p.

Perelson, A. S.; OSTER, G. F. *Theoretical studies of clonal selection: Minimal antibody repertoire size and reliability of self-non-self discrimination*. Journal of Theoretical Biology, Berkeley, California, v. 81, n. 4, p. 645-670, dez. 1979..

PIRES, M. M. *Agrupamento Incremental e Hierárquico de Documentos.* 2008. 91f. Dissertação (Mestrado em Engenharia Civil) - Universidade Federal do Rio de Janeiro. Rio de Janeiro. 2008.

PREUSS, E. *Algoritmo e Estrutura de Dados.* Universidade Regional Integrada do Alto Uruguai e das Missões. Departamento de Engenharias e Ciência da Computação. Santo Ângelo, Rio Grande do Sul. 2002.

PUNTAR, S. G. *Métodos e Visualização de Agrupamento de Dados.* 2003. 131f. Dissertação (Mestrado em Ciência e Engenharia Civil) - Universidade Federal do Rio de Janeiro. Rio de Janeiro. 2003.

ROCHA, H. V.; LACHI, R. L. *Aspectos básicos de clustering:* conceitos e técnicas. Núcleo de Informática Aplicada à Educação. Instituto de Computação – Universidade Estadual de Campinas. Campinas, São Paulo. 2005.

RODRIGUES, F. S. *Métodos de Agrupamento na análise de dados de expressão gênica.* 2009. 95f. Dissertação (Mestrado em Estatística) - Universidade Federal de São Carlos. São Paulo. 2009.

SANTOS, D. S. *Bee clustering:* um Algoritmo para Agrupamento de Dados Inspirado em Inteligência de Enxames. 2009. 89f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal do Rio Grande do Sul. Porto Alegre, Rio Grande do Sul. 2009.

SAMPAIO, U. A. T. *Um estudo quantitativo de indicadores criminais da Secretaria de Segurança Pública do estado do Rio Grande do Sul.* 2008. 102f. Monografia (Bacharel em Estatística) - Universidade Federal do Rio Grande do Sul. Porto Alegre, Rio Grande do Sul. 2008.

SHEIKHOLESLAMI, G.; et al. *WaveCluster: A multi-resolution clustering approach for very large spatial databases.* In Proceedings of the 24th VLDB Conference, The International Journal on Very Large Data Bases, New York, USA, v.8, n. 3-4, p. 428-439. Fev. 2000.

SILVA. E. B. *Agrupamento Semi-Supervisionado de Documentos XML.* 2006. 128f. Tese (Doutorado em Engenharia de Sistema de Computação) - Universidade Federal do Rio de Janeiro. Rio de Janeiro. 2006.

SILVA, L. A. M. *Material de Apoio a Disciplina de Algoritmos.* União Educacional do Norte. Faculdade Barão do Rio Branco – FAB. Curso de Sistema de Informação. Rio Branco, Acre. 2006

SILVA, L. N. C. *Engenharia Imunológica: Desenvolvimento e Aplicação de Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artificiais.* 250 fls. Tese (Doutorado em Engenharia Elétrica) - Universidade Estadual de Campinas. São Paulo. 2001.

XU, L.; et al. *Immune Algorithm for Supervised Clustering.* In 5th IEEE International Conference On Cognitive Informatics, Beijing, China, v. 2, n.1, p. 953-958, jul. 2006.

ZHANG, T.; et al. *BIRCH: An Efficient Data Clustering Method for Very Large Databases.* In Proceedings of the ACM SIGMOD Conference on Management of Data, New York, USA, v. 25, n. 2, p. 103-114, jun. 1996.

ZUBEN, V.; MOSCATO. *Uma Visão Geral de Clusterização de Dados.* São Paulo: Universidade Estadual de Campinas. 2002. 11 slides, color.

WANG, W.; et al. *Sting: A statical information grid approach to spatial data mining***.** In: Proceedings of the 23rd International Conference on Very Large Data Bases, Athens, Greece, p. 186-195. 1997.

WATKINS, A.; BOGGES, L. *Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm***.** Genetic Programming and Evolvable Machines, p.1, 3 set. 2004.

WIVES, L. K. *Um estudo sobre Agrupamento de Documentos Textuais em Processamento de Informações não Estruturadas Usando Técnicas de Clustering.* 1999. 84 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul. Porto Alegre, Rio Grande do Sul. 1999.

8 ANEXOS

8.1 ANEXO A – ARQUIVO DE LOG

O anexo A apresenta o arquivo de log coletado do algoritmo imunológico para base da Íris, com os seguintes parâmetros: 100 clones, fator de mutação de 0.05, para 50 iterações iniciais e 100 iterações do imunológico, sem normalização dos dados, para distância euclidiana. O arquivo esta resumido sendo apresentada somente a primeira e última iteração inicial e a primeira e última iteração do imunológico, pois a estrutura restante do arquivo é parecida com elas.

========= ALGORITMO IMUNOLÓGICO PARA AGRUPAMENTO DE DADOS =========
================= AUTORA: RAQUEL DE L. MACHADO ======================
================= DATA INÍCIO: 09/11/2011 – 20:54:12 =================
================= LINFÓCITOS INICIAIS CRIADOS ========================
[[5.0, 2.0, 3.5, 1.0], [4.5, 2.3, 1.3, 0.3], [6.2, 3.4, 5.4, 2.3]]
=================== PRIMEIROS GRUPOS FORMADOS ========================
{[4.5, 2.3, 1.3, 0.3]=[[5.1, 3.5, 1.4, 0.2], [4.9, 3.0, 1.4, 0.2], [4.7, 3.2, 1.3, 0.2], [4.6, 3.1, 1.5, 0.2], [5.0, 3.6, 1.4, 0.2], [5.4, 3.9, 1.7, 0.4], [4.6, 3.4, 1.4, 0.3], [5.0, 3.4, 1.5, 0.2], [4.4, 2.9, 1.4, 0.2], [4.9, 3.1, 1.5, 0.1], [5.4, 3.7, 1.5, 0.2], [4.8, 3.4, 1.6, 0.2], [4.8, 3.0, 1.4, 0.1], [4.3, 3.0, 1.1, 0.1], [5.8, 4.0, 1.2, 0.2], [5.7, 4.4, 1.5, 0.4], [5.4, 3.9, 1.3, 0.4], [5.1, 3.5, 1.4, 0.3], [5.7, 3.8, 1.7, 0.3], [5.1, 3.8, 1.5, 0.3], [5.4, 3.4, 1.7, 0.2], [5.1, 3.7, 1.5, 0.4], [4.6, 3.6, 1.0, 0.2], [5.1, 3.3, 1.7, 0.5], [4.8, 3.4, 1.9, 0.2], [5.0, 3.0, 1.6, 0.2], [5.0, 3.4, 1.6, 0.4], [5.2, 3.5, 1.5, 0.2], [5.2, 3.4, 1.4, 0.2], [4.7, 3.2, 1.6, 0.2], [4.8, 3.1, 1.6, 0.2], [5.4, 3.4, 1.5, 0.4], [5.2, 4.1, 1.5, 0.1], [5.5, 4.2, 1.4, 0.2], [4.9, 3.1, 1.5, 0.1], [5.0, 3.2, 1.2, 0.2], [5.5, 3.5, 1.3, 0.2], [4.9, 3.1, 1.5, 0.1], [4.4, 3.0, 1.3, 0.2], [5.1, 3.4, 1.5, 0.2], [5.0, 3.5, 1.3, 0.3], [4.5, 2.3, 1.3, 0.3], [4.4, 3.2, 1.3, 0.2], [5.0, 3.5, 1.6, 0.6], [5.1, 3.8, 1.9, 0.4], [4.8, 3.0, 1.4, 0.3], [5.1, 3.8, 1.6, 0.2], [4.6, 3.2, 1.4, 0.2], [5.3, 3.7, 1.5, 0.2], [5.0, 3.3, 1.4, 0.2]], [5.0, 2.0, 3.5, 1.0]=[[5.5, 2.3, 4.0, 1.3], [5.7, 2.8, 4.5, 1.3], [4.9, 2.4, 3.3, 1.0], [5.2, 2.7, 3.9, 1.4], [5.0, 2.0, 3.5, 1.0], [6.0, 2.2, 4.0, 1.0], [5.6, 2.9, 3.6, 1.3], [5.8, 2.7, 4.1, 1.0], [6.2, 2.2, 4.5, 1.5], [5.6, 2.5, 3.9, 1.1], [6.1, 2.8, 4.0, 1.3], [5.7, 2.6, 3.5, 1.0], [5.5, 2.4, 3.8, 1.1], [5.5, 2.4, 3.7, 1.0], [5.8, 2.7, 3.9, 1.2], [6.3, 2.3, 4.4, 1.3], [5.6, 3.0, 4.1, 1.3], [5.5, 2.5, 4.0, 1.3], [5.5, 2.6, 4.4, 1.2], [5.8, 2.6, 4.0, 1.2], [5.0, 2.3, 3.3, 1.0], [5.6, 2.7, 4.2, 1.3], [5.7, 3.0, 4.2, 1.2], [5.7, 2.9, 4.2, 1.3], [5.1, 2.5, 3.0, 1.1], [5.7, 2.8, 4.1, 1.3], [4.9, 2.5, 4.5, 1.7]], [6.2, 3.4, 5.4, 2.3]=[[7.0, 3.2, 4.7, 1.4], [6.4, 3.2, 4.5, 1.5], [6.9, 3.1, 4.9, 1.5], [6.5, 2.8, 4.6, 1.5], [6.3, 3.3, 4.7, 1.6], [6.6, 2.9, 4.6, 1.3], [5.9, 3.0, 4.2, 1.5], [6.1, 2.9, 4.7, 1.4], [6.7, 3.1, 4.4, 1.4], [5.6, 3.0, 4.5, 1.5], [5.9, 3.2, 4.8, 1.8], [6.3, 2.5, 4.9, 1.5], [6.1, 2.8, 4.7, 1.2], [6.4, 2.9, 4.3, 1.3], [6.6, 3.0, 4.4, 1.4], [6.8, 2.8, 4.8, 1.4], [6.7, 3.0, 5.0, 1.7], [6.0, 2.9, 4.5, 1.5], [6.0, 2.7, 5.1, 1.6], [5.4, 3.0, 4.5, 1.5], [6.0, 3.4, 4.5, 1.6], [6.7, 3.1, 4.7, 1.5], [6.1, 3.0, 4.6, 1.4], [6.2, 2.9, 4.3, 1.3], [6.3, 3.3, 6.0, 2.5], [5.8, 2.7, 5.1, 1.9], [7.1, 3.0, 5.9, 2.1], [6.3, 2.9, 5.6, 1.8], [6.5, 3.0, 5.8, 2.2], [7.6, 3.0, 6.6, 2.1], [7.3, 2.9, 6.3, 1.8], [6.7, 2.5, 5.8, 1.8], [7.2, 3.6, 6.1, 2.5], [6.5, 3.2, 5.1, 2.0], [6.4, 2.7, 5.3, 1.9], [6.8, 3.0, 5.5, 2.1], [5.7, 2.5, 5.0, 2.0], [5.8, 2.8, 5.1, 2.4], [6.4, 3.2, 5.3, 2.3], [6.5, 3.0, 5.5, 1.8], [7.7, 3.8, 6.7, 2.2], [7.7, 2.6, 6.9, 2.3], [6.0, 2.2, 5.0, 1.5], [6.9, 3.2, 5.7, 2.3], [5.6, 2.8, 4.9, 2.0], [7.7, 2.8, 6.7, 2.0], [6.3, 2.7, 4.9, 1.8], [6.7, 3.3, 5.7, 2.1], [7.2, 3.2, 6.0, 1.8], [6.2, 2.8, 4.8, 1.8], [6.1, 3.0, 4.9, 1.8], [6.4, 2.8, 5.6, 2.1], [7.2, 3.0, 5.8, 1.6], [7.4, 2.8, 6.1, 1.9], [7.9, 3.8, 6.4, 2.0], [6.4, 2.8, 5.6, 2.2], [6.3, 2.8, 5.1, 1.5], [6.1, 2.6, 5.6, 1.4], [7.7, 3.0, 6.1, 2.3], [6.3, 3.4, 5.6, 2.4], [6.4, 3.1, 5.5, 1.8], [6.0, 3.0, 4.8, 1.8], [6.9, 3.1, 5.4, 2.1], [6.7, 3.1, 5.6, 2.4], [6.9, 3.1, 5.1, 2.3], [5.8, 2.7, 5.1, 1.9], [6.8, 3.2, 5.9, 2.3], [6.7, 3.3, 5.7, 2.5], [6.7, 3.0, 5.2, 2.3], [6.3, 2.5, 5.0, 1.9], [6.5, 3.0, 5.2, 2.0], [6.2, 3.4, 5.4, 2.3], [5.9, 3.0, 5.1, 1.8]]}

================= FORMANDO 100 GRUPOS ========================
================= GRUPO NÚMERO 0 =============================
================= NOVOS LINFÓCITOS CRIADOS ========================
[[5.005999999999999, 3.4180000000000006, 1.464, 0.2439999999999999], [5.574074074074072,

2.5666666666666664, 3.9481481481481486, 1.2111111111111112], [6.5164383561643815, 2.984931506849316, 5.260273972602741, 1.8479452054794518]]
========== NOVO AGRUPAMENTO ==========
{[6.5164383561643815, 2.984931506849316, 5.260273972602741, 1.8479452054794518]=[[7.0, 3.2, 4.7, 1.4], [6.4, 3.2, 4.5, 1.5], [6.9, 3.1, 4.9, 1.5], [6.5, 2.8, 4.6, 1.5], [6.3, 3.3, 4.7, 1.6], [6.6, 2.9, 4.6, 1.3], [6.1, 2.9, 4.7, 1.4], [6.7, 3.1, 4.4, 1.4], [5.9, 3.2, 4.8, 1.8], [6.3, 2.5, 4.9, 1.5], [6.6, 3.0, 4.4, 1.4], [6.8, 2.8, 4.8, 1.4], [6.7, 3.0, 5.0, 1.7], [6.0, 2.7, 5.1, 1.6], [6.0, 3.4, 4.5, 1.6], [6.7, 3.1, 4.7, 1.5], [6.1, 3.0, 4.6, 1.4], [6.3, 3.3, 6.0, 2.5], [5.8, 2.7, 5.1, 1.9], [7.1, 3.0, 5.9, 2.1], [6.3, 2.9, 5.6, 1.8], [6.5, 3.0, 5.8, 2.2], [7.6, 3.0, 6.6, 2.1], [7.3, 2.9, 6.3, 1.8], [6.7, 2.5, 5.8, 1.8], [7.2, 3.6, 6.1, 2.5], [6.5, 3.2, 5.1, 2.0], [6.4, 2.7, 5.3, 1.9], [6.8, 3.0, 5.5, 2.1], [5.7, 2.5, 5.0, 2.0], [5.8, 2.8, 5.1, 2.4], [6.4, 3.2, 5.3, 2.3], [6.5, 3.0, 5.5, 1.8], [7.7, 3.8, 6.7, 2.2], [7.7, 2.6, 6.9, 2.3], [6.0, 2.2, 5.0, 1.5], [6.9, 3.2, 5.7, 2.3], [5.6, 2.8, 4.9, 2.0], [7.7, 2.8 , 6.7, 2.0], [6.3, 2.7, 4.9, 1.8], [6.7, 3.3, 5.7, 2.1], [7.2, 3.2, 6.0, 1.8], [6.2, 2.8, 4.8, 1.8], [6.1, 3.0, 4.9, 1.8], [6.4, 2.8, 5.6, 2.1], [7.2, 3.0, 5.8, 1.6], [7.4, 2.8, 6.1, 1.9], [7.9, 3.8, 6.4, 2.0], [6.4, 2.8, 5.6, 2.2], [6.3, 2.8, 5.1, 1.5], [6.1, 2.6, 5.6, 1.4], [7.7, 3.0, 6.1, 2.3], [6.3, 3.4, 5.6, 2.4], [6.4, 3.1, 5.5, 1.8], [6.0, 3.0, 4.8, 1.8], [6.9, 3.1, 5.4, 2.1], [6.7, 3.1, 5.6, 2.4], [6.9, 3.1, 5.1, 2.3], [5.8, 2.7, 5.1, 1.9], [6.8, 3.2, 5.9, 2.3], [6.7, 3.3, 5.7, 2.5], [6.7, 3.0, 5.2, 2.3], [6.3, 2.5, 5.0, 1.9], [6.5, 3.0, 5.2, 2.0], [6.2, 3.4, 5.4, 2.3], [5.9, 3.0, 5.1, 1.8]], [5.005999999999999, 3.4180000000000006, 1.464, 0.2439999999999999]=[[5.1, 3.5, 1.4, 0.2], [4.9, 3.0, 1.4, 0.2], [4.7, 3.2, 1.3, 0.2], [4.6, 3.1, 1.5, 0.2], [5.0, 3.6, 1.4, 0.2], [5.4, 3.9, 1.7, 0.4], [4.6, 3.4, 1.4, 0.3], [5.0, 3.4, 1.5, 0.2], [4.4, 2.9, 1.4, 0.2], [4.9, 3.1, 1.5, 0.1], [5.4, 3.7, 1.5, 0.2], [4.8, 3.4, 1.6, 0.2], [4.8, 3.0, 1.4, 0.1], [4.3, 3.0, 1.1, 0.1], [5.8, 4.0, 1.2, 0.2], [5.7, 4.4, 1.5, 0.4], [5.4, 3.9, 1.3, 0.4], [5.1, 3.5, 1.4, 0.3], [5.7, 3.8, 1.7, 0.3], [5.1, 3.8, 1.5, 0.3], [5.4, 3.4, 1.7, 0.2], [5.1, 3.7, 1.5, 0.4], [4.6, 3.6, 1.0, 0.2], [5.1, 3.3, 1.7, 0.5], [4.8, 3.4, 1.9, 0.2], [5.0, 3.0, 1.6, 0.2], [5.0, 3.4, 1.6, 0.4], [5.2, 3.5, 1.5, 0.2], [5.2, 3.4, 1.4, 0.2], [4.7, 3.2, 1.6, 0.2], [4.8, 3.1, 1.6, 0.2], [5.4, 3.4, 1.5, 0.4], [5.2, 4.1, 1.5, 0.1], [5.5, 4.2, 1.4, 0.2], [4.9, 3.1, 1.5, 0.1], [5.0, 3.2, 1.2, 0.2], [5.5, 3.5, 1.3, 0.2], [4.9, 3.1, 1.5, 0.1], [4.4, 3.0, 1.3, 0.2], [5.1, 3.4, 1.5, 0.2], [5.0, 3.5, 1.3, 0.3], [4.5, 2.3, 1.3, 0.3], [4.4, 3.2, 1.3, 0.2], [5.0, 3.5, 1.6, 0.6], [5.1, 3.8, 1.9, 0.4], [4.8, 3.0, 1.4, 0.3], [5.1, 3.8, 1.6, 0.2], [4.6, 3.2, 1.4, 0.2], [5.3, 3.7, 1.5, 0.2], [5.0, 3.3, 1.4, 0.2]], [5.574074074074072, 2.5666666666666664, 3.9481481481481486, 1.2111111111111112]=[[5.5, 2.3, 4.0, 1.3], [5.7, 2.8, 4.5, 1.3], [4.9, 2.4, 3.3, 1.0], [5.2, 2.7, 3.9, 1.4], [5.0, 2.0, 3.5, 1.0], [5.9, 3.0, 4.2, 1.5], [6.0, 2.2, 4.0, 1.0], [5.6, 2.9, 3.6, 1.3], [5.6, 3.0, 4.5, 1.5], [5.8, 2.7, 4.1, 1.0], [6.2, 2.2, 4.5, 1.5], [5.6, 2.5, 3.9, 1.1], [6.1, 2.8, 4.0, 1.3], [6.1, 2.8, 4.7, 1.2], [6.4, 2.9, 4.3, 1.3], [6.0, 2.9, 4.5, 1.5], [5.7, 2.6, 3.5, 1.0], [5.5, 2.4, 3.8, 1.1], [5.5, 2.4, 3.7, 1.0], [5.8, 2.7, 3.9, 1.2], [5.4, 3.0, 4.5, 1.5], [6.3, 2.3, 4.4, 1.3], [5.6, 3.0, 4.1, 1.3], [5.5, 2.5, 4.0, 1.3], [5.5, 2.6, 4.4, 1.2], [5.8, 2.6, 4.0, 1.2], [5.0, 2.3, 3.3, 1.0], [5.6, 2.7, 4.2, 1.3], [5.7, 3.0, 4.2, 1.2], [5.7, 2.9, 4.2, 1.3], [6.2, 2.9, 4.3, 1.3], [5.1, 2.5, 3.0, 1.1], [5.7, 2.8, 4.1, 1.3], [4.9, 2.5, 4.5, 1.7]]}
========== INDÍCES DE VALIDAÇÃO ==========
Homogeneidade Anterior 1.1547147521825198 – Homogeneidade Nova 0.663968904951699
========== SEPARACAO ==========
Separação Anterior 3.9083808628784578 – Separação Nova 3.232650724047787
========== GRUPO NÃO MELHOROU ==========
========== GRUPO NÚMERO 49 ==========
========== NOVOS LINFÓCITOS CRIADOS ==========
[[5.005999999999999, 3.4180000000000006, 1.464, 0.2439999999999999], [5.574074074074072, 2.5666666666666664, 3.9481481481481486, 1.2111111111111112], [6.5164383561643815, 2.984931506849316, 5.260273972602741, 1.8479452054794518]]
========== NOVO AGRUPAMENTO ==========
{[5.005999999999999, 3.4180000000000006, 1.464, 0.2439999999999999]=[[5.1, 3.5, 1.4, 0.2], [4.9, 3.0, 1.4, 0.2], [4.7, 3.2, 1.3, 0.2], [4.6, 3.1, 1.5, 0.2], [5.0, 3.6, 1.4, 0.2], [5.4, 3.9, 1.7, 0.4], [4.6, 3.4, 1.4, 0.3], [5.0, 3.4, 1.5, 0.2], [4.4, 2.9, 1.4, 0.2], [4.9, 3.1, 1.5, 0.1], [5.4, 3.7, 1.5, 0.2], [4.8, 3.4, 1.6, 0.2], [4.8, 3.0, 1.4, 0.1], [4.3, 3.0, 1.1, 0.1], [5.8, 4.0, 1.2, 0.2], [5.7, 4.4, 1.5, 0.4], [5.4, 3.9, 1.3, 0.4], [5.1, 3.5, 1.4, 0.3], [5.7, 3.8, 1.7, 0.3], [5.1, 3.8, 1.5, 0.3], [5.4, 3.4, 1.7, 0.2], [5.1, 3.7, 1.5, 0.4], [4.6, 3.6, 1.0, 0.2], [5.1, 3.3, 1.7, 0.5], [4.8, 3.4, 1.9, 0.2], [5.0, 3.0, 1.6, 0.2], [5.0, 3.4, 1.6, 0.4], [5.2, 3.5, 1.5, 0.2], [5.2, 3.4, 1.4, 0.2], [4.7, 3.2, 1.6, 0.2], [4.8, 3.1, 1.6, 0.2], [5.4, 3.4, 1.5, 0.4], [5.2, 4.1, 1.5, 0.1], [5.5, 4.2, 1.4, 0.2], [4.9, 3.1, 1.5, 0.1], [5.0, 3.2, 1.2, 0.2], [5.5, 3.5, 1.3, 0.2], [4.9, 3.1, 1.5, 0.1], [4.4, 3.0, 1.3, 0.2], [5.1, 3.4, 1.5, 0.2], [5.0, 3.5, 1.3, 0.3], [4.5, 2.3, 1.3, 0.3], [4.4, 3.2, 1.3, 0.2], [5.0, 3.5, 1.6, 0.6], [5.1, 3.8, 1.9, 0.4], [4.8, 3.0, 1.4, 0.3], [5.1, 3.8, 1.6, 0.2], [4.6, 3.2, 1.4, 0.2], [5.3, 3.7, 1.5, 0.2], [5.0, 3.3, 1.4, 0.2]], [6.5164383561643815, 2.984931506849316, 5.260273972602741, 1.8479452054794518]=[[7.0, 3.2 , 4.7, 1.4], [6.4, 3.2, 4.5, 1.5], [6.9, 3.1, 4.9, 1.5], [6.5, 2.8, 4.6, 1.5], [6.3, 3.3, 4.7, 1.6], [6.6, 2.9, 4.6, 1.3], [6.1, 2.9, 4.7, 1.4], [6.7, 3.1, 4.4, 1.4], [5.9, 3.2, 4.8, 1.8], [6.3, 2.5, 4.9, 1.5], [6.6, 3.0, 4.4, 1.4], [6.8, 2.8, 4.8, 1.4], [6.7, 3.0, 5.0, 1.7], [6.0, 2.7, 5.1, 1.6], [6.0, 3.4, 4.5, 1.6], [6.7, 3.1, 4.7, 1.5], [6.1, 3.0, 4.6, 1.4], [6.3, 3.3, 6.0, 2.5], [5.8, 2.7, 5.1, 1.9], [7.1, 3.0, 5.9, 2.1], [6.3, 2.9, 5.6, 1.8], [6.5, 3.0, 5.8, 2.2], [7.6, 3.0, 6.6,

2.1], [7.3, 2.9, 6.3, 1.8], [6.7, 2.5, 5.8, 1.8], [7.2, 3.6, 6.1, 2.5], [6.5, 3.2, 5.1, 2.0], [6.4, 2.7, 5.3, 1.9], [6.8, 3.0, 5.5, 2.1], [5.7, 2.5, 5.0, 2.0], [5.8, 2.8, 5.1, 2.4], [6.4, 3.2, 5.3, 2.3], [6.5, 3.0, 5.5, 1.8], [7.7, 3.8, 6.7, 2.2], [7.7, 2.6, 6.9, 2.3], [6.0, 2.2, 5.0, 1.5], [6.9, 3.2, 5.7, 2.3], [5.6, 2.8, 4.9, 2.0], [7.7, 2.8, 6.7, 2.0], [6.3, 2.7, 4.9, 1.8], [6.7, 3.3, 5.7, 2.1], [7.2, 3.2, 6.0, 1.8], [6.2, 2.8, 4.8, 1.8], [6.1, 3.0, 4.9, 1.8], [6.4, 2.8, 5.6, 2.1], [7.2, 3.0, 5.8, 1.6], [7.4, 2.8, 6.1, 1.9], [7.9, 3.8, 6.4, 2.0], [6.4, 2.8, 5.6, 2.2], [6.3, 2.8, 5.1, 1.5], [6.1, 2.6, 5.6, 1.4], [7.7, 3.0, 6.1, 2.3], [6.3, 3.4, 5.6, 2.4], [6.4, 3.1, 5.5, 1.8], [6.0, 3.0, 4.8, 1.8], [6.9, 3.1, 5.4, 2.1], [6.7, 3.1, 5.6, 2.4], [6.9, 3.1, 5.1, 2.3], [5.8, 2.7, 5.1, 1.9], [6.8, 3.2, 5.9, 2.3], [6.7, 3.3, 5.7, 2.5], [6.7, 3.0, 5.2, 2.3], [6.3, 2.5, 5.0, 1.9], [6.5, 3.0, 5.2, 2.0], [6.2, 3.4, 5.4, 2.3], [5.9, 3.0, 5.1, 1.8]], [5.574074074074072, 2.5666666666666664, 3.9481481481481486, 1.2111111111111112]=[[5.5, 2.3, 4.0, 1.3], [5.7, 2.8, 4.5, 1.3], [4.9, 2.4, 3.3, 1.0], [5.2, 2.7, 3.9, 1.4], [5.0, 2.0, 3.5, 1.0], [5.9, 3.0, 4.2, 1.5], [6.0, 2.2, 4.0, 1.0], [5.6, 2.9, 3.6, 1.3], [5.6, 3.0, 4.5, 1.5], [5.8, 2.7, 4.1, 1.0], [6.2, 2.2, 4.5, 1.5], [5.6, 2.5, 3.9, 1.1], [6.1, 2.8, 4.0, 1.3], [6.1, 2.8, 4.7, 1.2], [6.4, 2.9, 4.3, 1.3], [6.0, 2.9, 4.5, 1.5], [5.7, 2.6, 3.5, 1.0], [5.5, 2.4, 3.8, 1.1], [5.5, 2.4, 3.7, 1.0], [5.8, 2.7, 3.9, 1.2], [5.4, 3.0, 4.5, 1.5], [6.3, 2.3, 4.4, 1.3], [5.6, 3.0, 4.1, 1.3], [5.5, 2.5, 4.0, 1.3], [5.5, 2.6, 4.4, 1.2], [5.8, 2.6, 4.0, 1.2], [5.0, 2.3, 3.3, 1.0], [5.6, 2.7, 4.2, 1.3], [5.7, 3.0, 4.2, 1.2], [5.7, 2.9, 4.2, 1.3], [6.2, 2.9, 4.3, 1.3], [5.1, 2.5, 3.0, 1.1], [5.7, 2.8, 4.1, 1.3], [4.9, 2.5, 4.5, 1.7]]}
========== INDÍCES DE VALIDAÇÃO ==========
Homogeneidade Anterior 1.1547147521825198 – Homogeneidade Nova 0.663968904951699
========== SEPARACAO ==========
Separação Anterior 3.9083808628784578 – Separação Nova 3.2326507240477875
========== GRUPO NÃO MELHOROU ==========
========== ALGORITMO IMUNOLÓGICO ==========
========== AGRUPAMENTO IMUNOLÓGICO ==========
{[4.997910935867413, 2.0000495190153926, 3.499512624784985, 0.9998486728945409]=[[5.5, 2.3, 4.0, 1.3], [5.7, 2.8, 4.5, 1.3], [4.9, 2.4, 3.3, 1.0], [5.2, 2.7, 3.9, 1.4], [5.0, 2.0, 3.5, 1.0], [6.0, 2.2, 4.0, 1.0], [5.6, 2.9, 3.6, 1.3], [5.8, 2.7, 4.1, 1.0], [6.2, 2.2, 4.5, 1.5], [5.6, 2.5, 3.9, 1.1], [6.1, 2.8, 4.0, 1.3], [5.7, 2.6, 3.5, 1.0], [5.5, 2.4, 3.8, 1.1], [5.5, 2.4, 3.7, 1.0], [5.8, 2.7, 3.9, 1.2], [6.3, 2.3, 4.4, 1.3], [5.6, 3.0, 4.1, 1.3], [5.5, 2.5, 4.0, 1.3], [5.5, 2.6, 4.4, 1.2], [5.8, 2.6, 4.0, 1.2], [5.0, 2.3, 3.3, 1.0], [5.6, 2.7, 4.2, 1.3], [5.7, 3.0, 4.2, 1.2], [5.7, 2.9, 4.2, 1.3], [5.1, 2.5, 3.0, 1.1], [5.7, 2.8, 4.1, 1.3], [4.9, 2.5, 4.5, 1.7]], [4.500366791818328, 2.3003776721929547, 1.3004807156242788, 0.2999075612667404]=[[5.1, 3.5, 1.4, 0.2], [4.9, 3.0, 1.4, 0.2], [4.7, 3.2, 1.3, 0.2], [4.6, 3.1, 1.5, 0.2], [5.0, 3.6, 1.4, 0.2], [5.4, 3.9, 1.7, 0.4], [4.6, 3.4, 1.4, 0.3], [5.0, 3.4, 1.5, 0.2], [4.4, 2.9, 1.4, 0.2], [4.9, 3.1, 1.5, 0.1], [5.4, 3.7, 1.5, 0.2], [4.8, 3.4, 1.6, 0.2], [4.8, 3.0, 1.4, 0.1], [4.3, 3.0, 1.1, 0.1], [5.8, 4.0, 1.2, 0.2], [5.7, 4.4, 1.5, 0.4], [5.4, 3.9, 1.3, 0.4], [5.1, 3.5, 1.4, 0.3], [5.7, 3.8, 1.7, 0.3], [5.1, 3.8, 1.5, 0.3], [5.4, 3.4, 1.7, 0.2], [5.1, 3.7, 1.5, 0.4], [4.6, 3.6, 1.0, 0.2], [5.1, 3.3, 1.7, 0.5], [4.8, 3.4, 1.9, 0.2], [5.0, 3.0, 1.6, 0.2], [5.0, 3.4, 1.6, 0.4], [5.2, 3.5, 1.5, 0.2], [5.2, 3.4, 1.4, 0.2], [4.7, 3.2, 1.6, 0.2], [4.8, 3.1, 1.6, 0.2], [5.4, 3.4, 1.5, 0.4], [5.2, 4.1, 1.5, 0.1], [5.5, 4.2, 1.4, 0.2], [4.9, 3.1, 1.5, 0.1], [5.0, 3.2, 1.2, 0.2], [5.5, 3.5, 1.3, 0.2], [4.9, 3.1, 1.5, 0.1], [4.4, 3.0, 1.3, 0.2], [5.1, 3.4, 1.5, 0.2], [5.0, 3.5, 1.3, 0.3], [4.5, 2.3, 1.3, 0.3], [4.4, 3.2, 1.3, 0.2], [5.0, 3.5, 1.6, 0.6], [5.1, 3.8, 1.9, 0.4], [4.8, 3.0, 1.4, 0.3], [5.1, 3.8, 1.6, 0.2], [4.6, 3.2, 1.4, 0.2], [5.3, 3.7, 1.5, 0.2], [5.0, 3.3, 1.4, 0.2]], [6.20110942683014, 3.3985780092432156, 5.402586352628883, 2.3008606055456466]=[[7.0, 3.2, 4.7, 1.4], [6.4, 3.2, 4.5, 1.5], [6.9, 3.1, 4.9, 1.5], [6.5, 2.8, 4.6, 1.5], [6.3, 3.3, 4.7, 1.6], [6.6, 2.9, 4.6, 1.3], [5.9, 3.0, 4.2, 1.5], [6.1, 2.9, 4.7, 1.4], [6.7, 3.1, 4.4, 1.4], [5.6, 3.0, 4.5, 1.5], [5.9, 3.2, 4.8, 1.8], [6.3, 2.5, 4.9, 1.5], [6.1, 2.8, 4.7, 1.2], [6.4, 2.9, 4.3, 1.3], [6.6, 3.0, 4.4, 1.4], [6.8, 2.8, 4.8, 1.4], [6.7, 3.0, 5.0, 1.7], [6.0, 2.9, 4.5, 1.5], [6.0, 2.7, 5.1, 1.6], [5.4, 3.0, 4.5, 1.5], [6.0, 3.4, 4.5, 1.6], [6.7, 3.1, 4.7, 1.5], [6.1, 3.0, 4.6, 1.4], [6.2, 2.9, 4.3, 1.3], [6.3, 3.3, 6.0, 2.5], [5.8, 2.7, 5.1, 1.9], [7.1, 3.0, 5.9, 2.1], [6.3, 2.9, 5.6, 1.8], [6.5, 3.0, 5.8, 2.2], [7.6, 3.0, 6.6, 2.1], [7.3, 2.9, 6.3, 1.8], [6.7, 2.5, 5.8, 1.8], [7.2, 3.6, 6.1, 2.5], [6.5, 3.2, 5.1, 2.0], [6.4, 2.7, 5.3, 1.9], [6.8, 3.0, 5.5, 2.1], [5.7, 2.5, 5.0, 2.0], [5.8, 2.8, 5.1, 2.4], [6.4, 3.2, 5.3, 2.3], [6.5, 3.0, 5.5, 1.8], [7.7, 3.8, 6.7, 2.2], [7.7, 2.6, 6.9, 2.3], [6.0, 2.2, 5.0, 1.5], [6.9, 3.2, 5.7, 2.3], [5.6, 2.8, 4.9, 2.0], [7.7, 2.8, 6.7, 2.0], [6.3, 2.7, 4.9, 1.8], [6.7, 3.3, 5.7, 2.1], [7.2, 3.2, 6.0, 1.8], [6.2, 2.8, 4.8, 1.8], [6.1, 3.0, 4.9, 1.8], [6.4, 2.8, 5.6, 2.1], [7.2, 3.0, 5.8, 1.6], [7.4, 2.8, 6.1, 1.9], [7.9, 3.8, 6.4, 2.0], [6.4, 2.8, 5.6, 2.2], [6.3, 2.8, 5.1, 1.5], [6.1, 2.6, 5.6, 1.4], [7.7, 3.0, 6.1, 2.3], [6.3, 3.4, 5.6, 2.4], [6.4, 3.1, 5.5, 1.8], [6.0, 3.0, 4.8, 1.8], [6.9, 3.1, 5.4, 2.1], [6.7, 3.1, 5.6, 2.4], [6.9, 3.1, 5.1, 2.3], [5.8, 2.7, 5.1, 1.9], [6.8, 3.2, 5.9, 2.3], [6.7, 3.3, 5.7, 2.5], [6.7, 3.0, 5.2, 2.3], [6.3, 2.5, 5.0, 1.9], [6.5, 3.0, 5.2, 2.0], [6.2, 3.4, 5.4, 2.3], [5.9, 3.0, 5.1, 1.8]]}
========== INDÍCES DE VALIDAÇÃO IMUNOLÓGIO ==========
Homogeneidade Anterior 1.1547147521825198 – Homogeneidade Nova 1.1546862069941257
========== SEPARACAO ==========
Separação Anterior 3.9083808628784578 – Separação Nova 3.9100003644497803
========== GRUPO MELHOROU ==========
========== INDÍCES DE VALIDAÇÃO IMUNOLÓGIO ==========

Homogeneidade Anterior 1.154538861508885 – Homogeneidade Nova 1.1543230124147568

========== SEPARACAO ==========

Separação Anterior 3.9101546726699996 – Separação Nova 3.910349915041953

========== GRUPO MELHOROU ==========

========== AGRUPAMENTO IMUNOLÓGICO ==========

{[6.305714355480047, 3.3725668395017205, 5.4391793058545685, 2.296892032413912]=[[7.0, 3.2, 4.7, 1.4], [6.4, 3.2, 4.5, 1.5], [6.9, 3.1, 4.9, 1.5], [6.5, 2.8, 4.6, 1.5], [6.3, 3.3, 4.7, 1.6], [6.6, 2.9, 4.6, 1.3], [5.9, 3.0, 4.2, 1.5], [6.1, 2.9, 4.7, 1.4], [6.7, 3.1, 4.4, 1.4], [5.6, 3.0, 4.5, 1.5], [5.9, 3.2, 4.8, 1.8], [6.3, 2.5, 4.9, 1.5], [6.1, 2.8, 4.7, 1.2], [6.4, 2.9, 4.3, 1.3], [6.6, 3.0, 4.4, 1.4], [6.8, 2.8, 4.8, 1.4], [6.7, 3.0, 5.0, 1.7], [6.0, 2.9, 4.5, 1.5], [6.0, 2.7, 5.1, 1.6], [5.4, 3.0, 4.5, 1.5], [6.0, 3.4, 4.5, 1.6], [6.7, 3.1, 4.7, 1.5], [6.1, 3.0, 4.6, 1.4], [6.2, 2.9, 4.3, 1.3], [6.3, 3.3, 6.0, 2.5], [5.8, 2.7, 5.1, 1.9], [7.1, 3.0, 5.9, 2.1], [6.3, 2.9, 5.6, 1.8], [6.5, 3.0, 5.8, 2.2], [7.6, 3.0, 6.6, 2.1], [7.3, 2.9, 6.3, 1.8], [6.7, 2.5, 5.8, 1.8], [7.2, 3.6, 6.1, 2.5], [6.5, 3.2, 5.1, 2.0], [6.4, 2.7, 5.3, 1.9], [6.8, 3.0, 5.5, 2.1], [5.7, 2.5, 5.0, 2.0], [5.8, 2.8, 5.1 , 2.4], [6.4, 3.2, 5.3, 2.3], [6.5, 3.0, 5.5, 1.8], [7.7, 3.8, 6.7, 2.2], [7.7, 2.6, 6.9, 2.3], [6.0, 2.2, 5.0, 1.5], [6.9, 3.2, 5.7, 2.3], [5.6, 2.8, 4.9, 2.0], [7.7, 2.8, 6.7, 2.0], [6.3, 2.7, 4.9, 1.8], [6.7, 3.3, 5.7, 2.1], [7.2, 3.2, 6.0, 1.8], [6.2, 2.8, 4.8, 1.8], [6.1, 3.0, 4.9, 1.8], [6.4, 2.8, 5.6, 2.1], [7.2, 3.0, 5.8, 1.6], [7.4, 2.8, 6.1, 1.9], [7.9, 3.8, 6.4, 2.0], [6.4, 2.8, 5.6, 2.2], [6.3, 2.8, 5.1, 1.5], [6.1, 2.6, 5.6, 1.4], [7.7, 3.0, 6.1, 2.3], [6.3, 3.4, 5.6, 2.4], [6.4, 3.1, 5.5, 1.8], [6.0, 3.0, 4.8, 1.8], [6.9, 3.1, 5.4, 2.1], [6.7, 3.1, 5.6, 2.4], [6.9, 3.1, 5.1, 2.3], [5.8, 2.7, 5.1, 1.9], [6.8, 3.2, 5.9, 2.3], [6.7, 3.3, 5.7, 2.5], [6.7, 3.0, 5.2, 2.3], [6.3, 2.5, 5.0, 1.9], [6.5, 3.0, 5.2, 2.0], [6.2, 3.4, 5.4, 2.3], [5.9, 3.0, 5.1, 1.8]], [4.999163158690948, 1.9938854265387171, 3.472993358231759, 0.9988107230702559]=[[5.5, 2.3, 4.0, 1.3], [5.7, 2.8, 4.5, 1.3], [4.9, 2.4, 3.3, 1.0], [5.2, 2.7, 3.9, 1.4], [5.0, 2.0, 3.5, 1.0], [6.0, 2.2, 4.0, 1.0], [5.6, 2.9, 3.6, 1.3], [5.8, 2.7, 4.1, 1.0], [6.2, 2.2, 4.5, 1.5], [5.6, 2.5, 3.9, 1.1], [6.1, 2.8, 4.0, 1.3], [5.7, 2.6, 3.5, 1.0], [5.5, 2.4, 3.8, 1.1], [5.5, 2.4, 3.7, 1.0], [5.8, 2.7, 3.9, 1.2], [6.3, 2.3, 4.4, 1.3], [5.6, 3.0, 4.1, 1.3], [5.5, 2.5, 4.0, 1.3], [5.5, 2.6, 4.4, 1.2], [5.8, 2.6, 4.0, 1.2], [5.0, 2.3, 3.3, 1.0], [5.6, 2.7, 4.2, 1.3], [5.7, 3.0, 4.2, 1.2], [5.7, 2.9, 4.2, 1.3], [5.1, 2.5, 3.0, 1.1], [5.7, 2.8, 4.1, 1.3], [4.9, 2.5, 4.5, 1.7]], [4.509015492392406, 2.3270724570662367, 1.2893705035505632, 0.30040646284038747]=[[5.1, 3.5, 1.4, 0.2], [4.9, 3.0, 1.4, 0.2], [4.7, 3.2, 1.3, 0.2], [4.6, 3.1, 1.5, 0.2], [5.0, 3.6, 1.4, 0.2], [5.4, 3.9, 1.7, 0.4], [4.6, 3.4, 1.4, 0.3], [5.0, 3.4, 1.5, 0.2], [4.4, 2.9, 1.4, 0.2], [4.9, 3.1, 1.5, 0.1], [5.4, 3.7, 1.5, 0.2], [4.8, 3.4, 1.6, 0.2], [4.8, 3.0, 1.4, 0.1], [4.3, 3.0, 1.1, 0.1], [5.8, 4.0, 1.2, 0.2], [5.7, 4.4, 1.5, 0.4], [5.4, 3.9, 1.3, 0.4], [5.1, 3.5, 1.4, 0.3], [5.7, 3.8, 1.7, 0.3], [5.1, 3.8, 1.5, 0.3], [5.4, 3.4, 1.7, 0.2], [5.1, 3.7, 1.5, 0.4], [4.6, 3.6, 1.0, 0.2], [5.1, 3.3, 1.7, 0.5], [4.8, 3.4, 1.9, 0.2], [5.0, 3.0, 1.6, 0.2], [5.0, 3.4, 1.6, 0.4], [5.2, 3.5, 1.5, 0.2], [5.2, 3.4, 1.4, 0.2], [4.7, 3.2, 1.6, 0.2], [4.8, 3.1, 1.6, 0.2], [5.4, 3.4, 1.5, 0.4], [5.2, 4.1, 1.5, 0.1], [5.5, 4.2, 1.4, 0.2], [4.9, 3.1, 1.5, 0.1], [5.0, 3.2, 1.2, 0.2], [5.5, 3.5, 1.3, 0.2], [4.9, 3.1, 1.5, 0.1], [4.4, 3.0, 1.3, 0.2], [5.1, 3.4, 1.5, 0.2], [5.0, 3.5, 1.3, 0.3], [4.5, 2.3, 1.3, 0.3], [4.4, 3.2, 1.3, 0.2], [5.0, 3.5, 1.6, 0.6], [5.1, 3.8, 1.9, 0.4], [4.8, 3.0, 1.4, 0.3], [5.1, 3.8, 1.6, 0.2], [4.6, 3.2, 1.4, 0.2], [5.3, 3.7, 1.5, 0.2], [5.0, 3.3, 1.4, 0.2]]}

========== INDÍCES DE VALIDAÇÃO IMUNOLÓGIO ==========

Homogeneidade Anterior 1.121910705899238 – Homogeneidade Nova 1.1236598565950566

========== SEPARACAO ==========

Separação Anterior 4.011746661623111 – Separação Nova 3.9823370233594235

```
====================== GRUPO NÃO MELHOROU ======================
========== MATRIZ DE CONFUSÃO ==========================
{Iris-virginica={Iris-virginica=49, Iris-setosa=null, Iris-versicolor=1}, Iris-setosa={Iris-virginica=null, Iris-setosa=50, Iris-versicolor=null}, Iris-versicolor={Iris-virginica=25, Iris-setosa=null, Iris-versicolor=25}}
```

8.2 ANEXO B – ARQUIVO DE ÍNDICES

O arquivo de índices foi utilizado para construção dos gráficos de homogeneidade e separação. O formato dele será o número da iteração, o valor da homogeneidade para aquela iteração e o valor de separação para aquela iteração, todos eles separados por vírgula para facilitar na exportação para o Excel. O arquivo de índice abaixo é do algoritmo imunológico para base da Íris, com os seguintes parâmetros: 100 clones, fator de mutação de 0.05, para 50 iterações iniciais e 100 iterações do imunológico, sem normalização dos dados, para distância euclidiana.

```
Iteração, Homogeneidade, Separação
0,1.1547147522,3.9083808629
1,1.1547147522,3.9083808629
2,1.1547147522,3.9083808629
3,1.1547147522,3.9083808629
4,1.1547147522,3.9083808629
5,1.1547147522,3.9083808629
6,1.1547147522,3.9083808629
7,1.1547147522,3.9083808629
8,1.1547147522,3.9083808629
9,1.1547147522,3.9083808629
10,1.1547147522,3.9083808629
11,1.1547147522,3.9083808629
12,1.1547147522,3.9083808629
13,1.1547147522,3.9083808629
14,1.1547147522,3.9083808629
15,1.1547147522,3.9083808629
16,1.1547147522,3.9083808629
17,1.1547147522,3.9083808629
18,1.1547147522,3.9083808629
19,1.1547147522,3.9083808629
20,1.1547147522,3.9083808629
21,1.1547147522,3.9083808629
22,1.1547147522,3.9083808629
23,1.1547147522,3.9083808629
24,1.1547147522,3.9083808629
25,1.1547147522,3.9083808629
26,1.1547147522,3.9083808629
27,1.1547147522,3.9083808629
28,1.1547147522,3.9083808629
29,1.1547147522,3.9083808629
30,1.1547147522,3.9083808629
31,1.1547147522,3.9083808629
32,1.1547147522,3.9083808629
33,1.1547147522,3.9083808629
34,1.1547147522,3.9083808629
```

35,1.1547147522,3.9083808629
36,1.1547147522,3.9083808629
37,1.1547147522,3.9083808629
38,1.1547147522,3.9083808629
39,1.1547147522,3.9083808629
40,1.1547147522,3.9083808629
41,1.1547147522,3.9083808629
42,1.1547147522,3.9083808629
43,1.1547147522,3.9083808629
44,1.1547147522,3.9083808629
45,1.1547147522,3.9083808629
46,1.1547147522,3.9083808629
47,1.1547147522,3.9083808629
48,1.1547147522,3.9083808629
49,1.1547147522,3.9083808629
50,1.1545388615,3.9101546727
51,1.1541008568,3.9130939412
52,1.1536648575,3.9148279876
53,1.1532352888,3.9160751688
54,1.1523694645,3.9168604490
55,1.1517243133,3.9175190326
56,1.1508124881,3.9191045114
57,1.1502988213,3.9203636178
58,1.1496888380,3.9218927870
59,1.1488434230,3.9226497100
60,1.1481888135,3.9237303235
61,1.1471086946,3.9252569855
62,1.1467255714,3.9271410247
63,1.1460159225,3.9280550302
64,1.1456275345,3.9295155609
65,1.1450626816,3.9315066434
66,1.1450476496,3.9336101469
67,1.1447952205,3.9356829112
68,1.1444112456,3.9378894667
69,1.1436539916,3.9388988452
70,1.1430162266,3.9403287093
71,1.1426728271,3.9409705306
72,1.1420280011,3.9415112982
73,1.1415654052,3.9420844932
74,1.1409927688,3.9436467996
75,1.1403174153,3.9439311074
76,1.1399431043,3.9450733985
77,1.1398137787,3.9458190009
78,1.1396299358,3.9463677078
79,1.1394979199,3.9470151878
80,1.1394657139,3.9485588300
81,1.1393892267,3.9499474623
82,1.1389911352,3.9503690115
83,1.1388997448,3.9513964290
84,1.1387837417,3.9519002380
85,1.1381239328,3.9526778783
86,1.1375566740,3.9532651144
87,1.1372745186,3.9540198970
88,1.1369943210,3.9548402672
89,1.1366049105,3.9552884406
90,1.1365001796,3.9556353864
91,1.1363293934,3.9573059100
92,1.1362365197,3.9581729091
93,1.1357125564,3.9585560304
94,1.1357125564,3.9585560304

95,1.1355420638,3.9590542700
96,1.1350828550,3.9593387203
97,1.1349514109,3.9593450657
98,1.1348573641,3.9604986220
99,1.1346411763,3.9610976694
100,1.1344759211,3.9611145337
101,1.1344759211,3.9611145337
102,1.1342027585,3.9611726679
103,1.1336476134,3.9618408181
104,1.1336103049,3.9636938379
105,1.1331784815,3.9640682041
106,1.1327436478,3.9644463528
107,1.1326523693,3.9654391236
108,1.1324102733,3.9664630353
109,1.1322833353,3.9670110841
110,1.1316531992,3.9671028380
111,1.1315578360,3.9680491729
112,1.1308657846,3.9683268836
113,1.1303361104,3.9693912888
114,1.1301735628,3.9697328734
115,1.1297373086,3.9701555728
116,1.1296131199,3.9708034331
117,1.1295912491,3.9711468832
118,1.1293180739,3.9716364884
119,1.1288691023,3.9718995075
120,1.1287236200,3.9728747915
121,1.1284086202,3.9730330903
122,1.1278454230,3.9730458919
123,1.1277185633,3.9741127501
124,1.1275835076,3.9752269269
125,1.1274043861,3.9757480501
126,1.1273482438,3.9759359256
127,1.1268124004,3.9765810822
128,1.1266662417,3.9772604906
129,1.1263923010,3.9781011883
130,1.1262408634,3.9787497725
131,1.1257723605,4.0043636860
132,1.1253462405,4.0048063015
133,1.1250867745,4.0049623473
134,1.1248270202,4.0052792889
135,1.1241876514,4.0053044576
136,1.1241876514,4.0053044576
137,1.1241576773,4.0059454678
138,1.1240284648,4.0071866261
139,1.1239524234,4.0077405837
140,1.1239268092,4.0084095654
141,1.1239268092,4.0084095654
142,1.1236036712,4.0088358176
143,1.1231707506,4.0093284631
144,1.1228793679,4.0096547926
145,1.1225798779,4.0097410749
146,1.1223743035,4.0101498782
147,1.1222382148,4.0106387495
148,1.1219107059,4.0117466616
149,1.1218030779,4.0130955305

8.3 ANEXO C – ARQUIVO DOS CLUSTERS

O anexo C apresenta o arquivo dos grupos formados para o algoritmo imunológico para base da Íris, com os seguintes parâmetros: 100 clones, fator de mutação de 0.05, para 50 iterações iniciais e 100 iterações do imunológico, sem normalização dos dados, para distância euclidiana. Seu formato apresenta os linfócitos utilizados para formar os grupos nas primeiras linhas denominadas Cluster 1,2,3 e em seguida os dados do arquivo onde o último parâmetro representa a que grupo cada instância pertence.

Centroid
Cluster 1 - 4.979646641731193, 1.979542770261063, 3.44573103793558, 0.9955760729948209, Iris-versicolor
Cluster 2 - 4.517128255383218, 2.3528859079464235, 1.2804707331893939, 0.29906449968311255, Iris-setosa
Cluster 3 - 6.359998424923794, 3.3438078920866734, 5.464363421375466, 2.2830212888876877, Iris-virginica

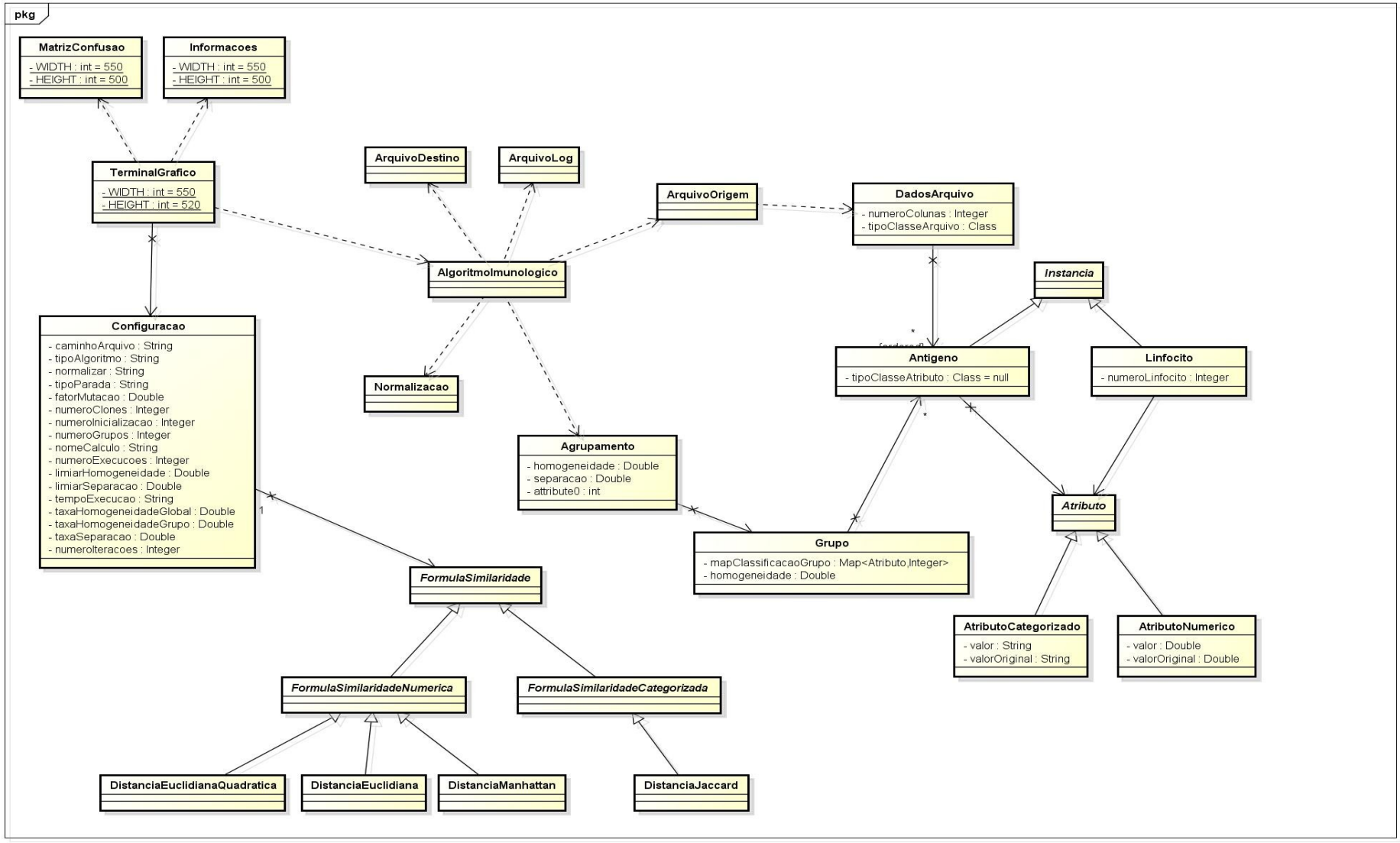
Instances

5.1,3.5,1.4,0.2,2	7.0,3.2,4.7,1.4,3	5.9,3.0,5.1,1.8,3
4.9,3.0,1.4,0.2,2	6.4,3.2,4.5,1.5,3	6.9,3.1,5.4,2.1,3
4.7,3.2,1.3,0.2,2	6.9,3.1,4.9,1.5,3	6.7,3.1,5.6,2.4,3
4.6,3.1,1.5,0.2,2	6.5,2.8,4.6,1.5,3	6.9,3.1,5.1,2.3,3
5.0,3.6,1.4,0.2,2	6.3,3.3,4.7,1.6,3	5.8,2.7,5.1,1.9,3
5.4,3.9,1.7,0.4,2	6.6,2.9,4.6,1.3,3	6.8,3.2,5.9,2.3,3
4.6,3.4,1.4,0.3,2	5.9,3.0,4.2,1.5,3	6.7,3.3,5.7,2.5,3
5.0,3.4,1.5,0.2,2	6.1,2.9,4.7,1.4,3	6.7,3.0,5.2,2.3,3
4.4,2.9,1.4,0.2,2	6.7,3.1,4.4,1.4,3	6.3,2.5,5.0,1.9,3
4.9,3.1,1.5,0.1,2	5.6,3.0,4.5,1.5,3	6.5,3.0,5.2,2.0,3
5.4,3.7,1.5,0.2,2	6.2,2.2,4.5,1.5,3	6.2,3.4,5.4,2.3,3
4.8,3.4,1.6,0.2,2	5.9,3.2,4.8,1.8,3	6.2,2.8,4.8,1.8,3
4.8,3.0,1.4,0.1,2	6.3,2.5,4.9,1.5,3	6.1,3.0,4.9,1.8,3
4.3,3.0,1.1,0.1,2	6.1,2.8,4.7,1.2,3	6.4,2.8,5.6,2.1,3
5.8,4.0,1.2,0.2,2	6.4,2.9,4.3,1.3,3	7.2,3.0,5.8,1.6,3
5.7,4.4,1.5,0.4,2	6.6,3.0,4.4,1.4,3	7.4,2.8,6.1,1.9,3
5.4,3.9,1.3,0.4,2	6.8,2.8,4.8,1.4,3	7.9,3.8,6.4,2.0,3
5.1,3.5,1.4,0.3,2	6.7,3.0,5.0,1.7,3	6.4,2.8,5.6,2.2,3
5.7,3.8,1.7,0.3,2	6.0,2.9,4.5,1.5,3	6.3,2.8,5.1,1.5,3
5.1,3.8,1.5,0.3,2	6.0,2.7,5.1,1.6,3	6.1,2.6,5.6,1.4,3
5.4,3.4,1.7,0.2,2	5.4,3.0,4.5,1.5,3	7.7,3.0,6.1,2.3,3
5.1,3.7,1.5,0.4,2	6.0,3.4,4.5,1.6,3	6.3,3.4,5.6,2.4,3
4.6,3.6,1.0,0.2,2	6.7,3.1,4.7,1.5,3	6.4,3.1,5.5,1.8,3
5.1,3.3,1.7,0.5,2	6.1,3.0,4.6,1.4,3	6.0,3.0,4.8,1.8,3
4.8,3.4,1.9,0.2,2	6.2,2.9,4.3,1.3,3	5.5,2.3,4.0,1.3,1
5.0,3.0,1.6,0.2,2	6.3,3.3,6.0,2.5,3	5.7,2.8,4.5,1.3,1
5.0,3.4,1.6,0.4,2	5.8,2.7,5.1,1.9,3	4.9,2.4,3.3,1.0,1
5.2,3.5,1.5,0.2,2	7.1,3.0,5.9,2.1,3	5.2,2.7,3.9,1.4,1
5.2,3.4,1.4,0.2,2	6.3,2.9,5.6,1.8,3	5.0,2.0,3.5,1.0,1
4.7,3.2,1.6,0.2,2	6.5,3.0,5.8,2.2,3	6.0,2.2,4.0,1.0,1
4.8,3.1,1.6,0.2,2	7.6,3.0,6.6,2.1,3	5.6,2.9,3.6,1.3,1
5.4,3.4,1.5,0.4,2	7.3,2.9,6.3,1.8,3	5.8,2.7,4.1,1.0,1
5.2,4.1,1.5,0.1,2	6.7,2.5,5.8,1.8,3	5.6,2.5,3.9,1.1,1
5.5,4.2,1.4,0.2,2	7.2,3.6,6.1,2.5,3	6.1,2.8,4.0,1.3,1
4.9,3.1,1.5,0.1,2	6.5,3.2,5.1,2.0,3	5.7,2.6,3.5,1.0,1

5.0,3.2,1.2,0.2,2	6.4,2.7,5.3,1.9,3	5.5,2.4,3.8,1.1,1
5.5,3.5,1.3,0.2,2	6.8,3.0,5.5,2.1,3	5.5,2.4,3.7,1.0,1
4.9,3.1,1.5,0.1,2	5.7,2.5,5.0,2.0,3	5.8,2.7,3.9,1.2,1
4.4,3.0,1.3,0.2,2	5.8,2.8,5.1,2.4,3	6.3,2.3,4.4,1.3,1
5.1,3.4,1.5,0.2,2	6.4,3.2,5.3,2.3,3	5.6,3.0,4.1,1.3,1
5.0,3.5,1.3,0.3,2	6.5,3.0,5.5,1.8,3	5.5,2.5,4.0,1.3,1
4.5,2.3,1.3,0.3,2	7.7,3.8,6.7,2.2,3	5.5,2.6,4.4,1.2,1
4.4,3.2,1.3,0.2,2	7.7,2.6,6.9,2.3,3	5.8,2.6,4.0,1.2,1
5.0,3.5,1.6,0.6,2	6.0,2.2,5.0,1.5,3	5.0,2.3,3.3,1.0,1
5.1,3.8,1.9,0.4,2	6.9,3.2,5.7,2.3,3	5.6,2.7,4.2,1.3,1
4.8,3.0,1.4,0.3,2	5.6,2.8,4.9,2.0,3	5.7,3.0,4.2,1.2,1
5.1,3.8,1.6,0.2,2	7.7,2.8,6.7,2.0,3	5.7,2.9,4.2,1.3,1
4.6,3.2,1.4,0.2,2	6.3,2.7,4.9,1.8,3	5.1,2.5,3.0,1.1,1
5.3,3.7,1.5,0.2,2	6.7,3.3,5.7,2.1,3	5.7,2.8,4.1,1.3,1
5.0,3.3,1.4,0.2,2	7.2,3.2,6.0,1.8,3	4.9,2.5,4.5,1.7,1

8.4 ANEXO D – DIAGRAMA DE CLASSES

O anexo E apresenta o diagrama de classes do sistema com suas classes e dependências.


pkg
MatrizConfusao
- WIDTH : int = 550
- HEIGHT : int = 500
Informacoes
- WIDTH : int = 550
- HEIGHT : int = 500
TerminalGrafico
- WIDTH : int = 550
- HEIGHT : int = 520
ArquivoDestino
ArquivoLog
ArquivoOrigem
DadosArquivo
- numeroColunas : Integer
- tipoClasseArquivo : Class
AlgoritmoImunologico
Instancia
Configuracao
- caminhoArquivo : String
- tipoAlgoritmo : String
- normalizar : String
- tipoParada : String
- fatorMutacao : Double
- numeroClones : Integer
- numeroInicializacao : Integer
- numeroGrupos : Integer
- nomeCalculo : String
- numeroExecucoes : Integer
- limiarHomogeneidade : Double
- limiarSeparacao : Double
- tempoExecucao : String
- taxaHomogeneidadeGlobal : Double
- taxaHomogeneidadeGrupo : Double
- taxaSeparacao : Double
- numeroIteracoes : Integer
Normalizacao
Antigeno
- tipoClasseAtributo : Class = null
Linfocito
- numeroLinfocito : Integer
Agrupamento
- homogeneidade : Double
- separacao : Double
- attribute0 : int
Atributo
Grupo
- mapClassificacaoGrupo : Map<Atributo,Integer>
- homogeneidade : Double
FormulaSimilaridade
AtributoCategorizado
- valor : String
- valorOriginal : String
AtributoNumerico
- valor : Double
- valorOriginal : Double
FormulaSimilaridadeNumerica
FormulaSimilaridadeCategorizada
DistanciaEuclidianaQuadratica
DistanciaEuclidiana
DistanciaManhattan
DistanciaJaccard
1
*
powered by astah*