

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**ALGORITMOS GENÉTICOS MULTIOBJETIVO  
PARA CLASSIFICAÇÃO HIERÁRQUICA  
DE ELEMENTOS TRANSPONÍVEIS**

**GEAN TRINDADE PEREIRA**

**ORIENTADOR: PROF. DR. RICARDO CERRI**

São Carlos – SP

Novembro, 2018

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**ALGORITMOS GENÉTICOS MULTIOBJETIVO  
PARA CLASSIFICAÇÃO HIERÁRQUICA  
DE ELEMENTOS TRANSPONÍVEIS**

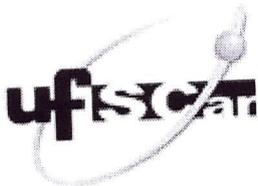
**GEAN TRINDADE PEREIRA**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Aprendizado de Máquina.

Orientador: Prof. Dr. Ricardo Cerri

São Carlos – SP

Novembro, 2018

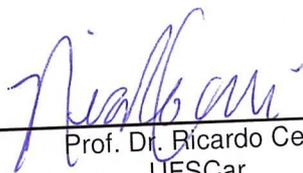


# UNIVERSIDADE FEDERAL DE SÃO CARLOS

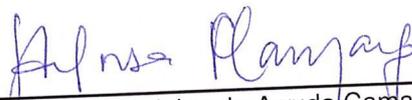
Centro de Ciências Exatas e de Tecnologia  
Programa de Pós-Graduação em Ciência da Computação

## Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Gean Trindade Pereira, realizada em 09/11/2018:



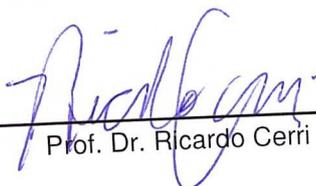
Prof. Dr. Ricardo Cerri  
UFSCar



Profa. Dra. Heloisa de Arruda Camargo  
UFSCar

Prof. Dr. Márcio Porto Basgalupp  
UNIFESP

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) Márcio Porto Basgalupp e, depois c arguições e deliberações realizadas, o(s) participante(s) à distância está(ao) de acordo com o conteúdo do parecer banca examinadora redigido neste relatório de defesa.



Prof. Dr. Ricardo Cerri

*"If I have seen further, it is by standing upon the shoulders of giants"*  
— *Sir Isaac Newton*

## AGRADECIMENTOS

---

---

Mais do que tudo, sou grato a minha mãe Tânia, por ser a pessoa maravilhosa que é e por tudo o que me proporcionou. Me considero muito sortudo de ser seu filho, tu me deu mais amor do que eu acho que mereço, fez e faz de tudo para que eu tenha paz e felicidade, sem você, com certeza eu não teria conseguido concluir essa pesquisa, obrigado. Quero dedicar essa pesquisa a minha falecida avó Catarina, ao qual passei junto seus últimos momentos. Você sempre estará em minha mente e coração. Agradeço aos demais familiares que estiveram, mesmo que só em memória, sempre presentes. Meu pai Glênio, que nunca poupou esforços para me ajudar, minhas queridas irmãs dona Gêssica e Gabriele bbzao, minha priminha recém nascida Isabele, meus avós Mario e Ismênia, e por fim, minha querida prima Tainara.

Sem nenhuma dúvida, não conseguiria sobreviver a essa trajetória sem os melhores amigos que esse mestrado poderia me fornecer. Tiago, Jorel (não o do desenho), Eduardinho, Eric, Gerson, Felipe, Danileira, Bruno, Igã, Diogo e Fredex. Juntos a gente cresceu muito, vocês me ajudaram a segurar a barra da ansiedade e da depressão, vocês foram a luz que eu havia perdido. Espero que a gente possa contar uns com os outros sempre, e que todos vocês tenham sucesso no que buscam. Agradeço muito também aos meus ex-colegas de apê, Loki e Delegado. Muitas vezes vocês foram o psicólogo que eu não podia pagar, os irmãos que nunca tive e a família que não estava ali presente, obrigado. Sou muito grato, também, aos amigos de sempre, Duda, Coto, Luciano e Vitor, por serem eternamente os brothers de fé que são. Mesmo sem nos falarmos por um grande período, a proximidade e o conforto não se alteram. Por fim, Alex, Douglas e Miguelito, mesmo que esse 2018 tenha abalado um pouco nossa amizade, serei eternamente grato a vocês, pois foram, muitas vezes, o único suporte que eu tive. Espero que logo a gente possa voltar a ser o que éramos antes.

Por último, mas não menos importante, agradeço imensamente ao meu Orientador prof. Dr. Ricardo Cerri. Obrigado por dar um voto de confiança a um mero recém formado sem experiência nenhuma na área mas que tinha a esperança de evoluir e se encontrar. Tudo o que sou hoje como futuro pesquisador, devo a você.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

# RESUMO

Em Aprendizado de Máquina (no inglês, *Machine Learning (ML)*), comumente um problema de classificação consiste em associar uma instância a apenas uma classe dentre um número normalmente pequeno de classes. Entretanto, existem problemas mais complexos que envolvem dezenas e até centenas de classes arranjadas numa estrutura hierárquica, sendo conhecidos na literatura como problemas de Classificação Hierárquica (no inglês, *Hierarchical Classification (HC)*). Nesses, uma instância é assinalada não só a uma classe, mas também as suas superclasses, e duas abordagens chamadas Global e Local são frequentemente utilizadas para tratá-los. Na Abordagem Local, múltiplos classificadores são treinados usando informações locais das classes, enquanto que na Abordagem Global um único classificador é induzido para lidar com toda a hierarquia de classes, o que a torna mais interpretável. Uma das áreas de maior aplicação da HC é a Bioinformática, onde ferramentas que exploram as relações hierárquicas nos dados e/ou que fazem uso de ML ainda são escassas. Além disso, uma questão importante em ambos os domínios e que ainda não recebe a devida atenção, é a interpretabilidade dos modelos. No contexto da Bioinformática, um tema que vem ganhando relevância é o estudo e classificação dos Elementos Transponíveis (no inglês, *Transposable Elements (TEs)*), fragmentos de DNA que se movem dentro do genoma de seus hospedeiros. Segundo pesquisas recentes, os TEs são responsáveis por mutações em diversos organismos, inclusive no genoma humano, o que lhes garantiu a alcunha de grandes responsáveis pela variabilidade genética das espécies. Nesse trabalho são propostos e investigados três métodos globais baseados em Algoritmos Genéticos que evoluem regras de classificação aplicadas a HC de TEs, sendo que em dois deles foram implementadas Abordagens Multiobjetivo com o intuito de lidar melhor com os objetivos de desempenho preditivo e interpretabilidade. O primeiro trata-se de um método de otimização simples chamado Hierarchical Classification with a Genetic Algorithm (HC-GA), o qual serviu de base para o desenvolvimento dos demais. O segundo método é denominado Hierarchical Classification with a Weighted Genetic Algorithm (HC-WGA), e implementa a Abordagem de Soma Ponderada. O terceiro método é chamado Hierarchical Classification with a Lexicographic Genetic Algorithm (HC-LGA), e segue a Abordagem Lexicográfica. Experimentos com a taxonomia de classes de TEs desenvolvida por Wicker et al. (2007), mostraram que os métodos propostos atingiram resultados superiores ou competitivos com os métodos de HC estado-da-arte da literatura, com a vantagem de que induzem modelos interpretáveis de regras. Quando comparados ao popular método global Clus-HMC, os métodos propostos apresentaram melhor desempenho além de produzirem regras menores e em menor quantidade. Nas comparações com as ferramentas de homologia BLASTn e RepeatMasker, os métodos hierárquicos alcançaram resultados superiores em ambos os conjuntos de dados e foram capazes de classificar todas as instâncias, diferente do que ocorreu nessas ferramentas. Ademais, foi verificado que os dois métodos multiobjetivo não só geraram os melhores resultados para os dois conjuntos de dados como também superaram com significância estatística o método de otimização simples.

**Palavras-chave:** Aprendizado de Máquina, Classificação Hierárquica, Algoritmos Genéticos, Otimização Multiobjetivo, Elementos Transponíveis.

# ABSTRACT

In Machine Learning (ML), commonly a classification problem consists of associating an instance to only one class within an usually small number of classes. However, there are more complex problems involving dozens and even hundreds of classes arranged in a hierarchical structure, which are known in the literature as Hierarchical Classification (HC) problems. In these problems, an instance is assigned not only to one class but also to its superclasses, and two approaches called Global and Local are often used in HC. In the Local Approach, multiple classifiers are trained using local information from classes, while in the Global Approach a single classifier is induced to deal with the entire class hierarchy, which makes it more interpretable. One of the fields of greatest application of HC is Bioinformatics, where tools that explore hierarchical relationships in data and/or made use of ML are still scarce. In addition, an issue that is important for both Bioinformatics and HC fields, and which is not always given due attention, is the interpretability of the models. In the context of Bioinformatics, a topic that is gaining attention is the study and classification of Transposable Elements (TEs), which are DNA fragments capable of moving inside the genome of their hosts. According to recent research, TEs are responsible for mutations in several organisms, including the human genome, which guaranteed them the nickname of great responsible for the genetic variability of species. In this work, three global methods based on Genetic Algorithms that evolve classification rules applied to HC of TEs were proposed and investigated, and in two of them, Multi-Objective Approaches were implemented in order to better deal with the predictive performance and interpretability objectives. The first one is a traditional optimization method called Hierarchical Classification with a Genetic Algorithm (HC-GA), which was used as the basis for the development of the others. The second method is called Hierarchical Classification with a Weighted Genetic Algorithm (HC-WGA), and implements the Weighted Sum Approach. The third method is called Hierarchical Classification with a Lexicographic Genetic Algorithm (HC-LGA), which follows the Lexicographic Approach. Experiments with the TEs class taxonomy developed by [Wicker et al. \(2007\)](#), have shown that the proposed methods have achieved better or competitive results with the state-of-the-art HC methods from the literature, with the advantage of generating interpretable rule models. When compared to the popular global method Clus-HMC, the proposed methods presented better predictive performance in addition to producing less rules with a fewer tests. In the comparisons with the homology tools BLASTn and RepeatMasker, the hierarchical methods achieved superior results in both datasets and were able to classify all the instances, different from what occurred in those tools. Moreover, it was verified that the two multi-objective methods not only obtained the best results for the two datasets used but they also surpassed the simple optimization method with statistical significance.

**Keywords:** Machine Learning, Hierarchical Classification, Genetic Algorithms, Multi-Objective Optimization, Transposable Elements.

# SUMÁRIO

---

---

<b>CAPÍTULO 1–INTRODUÇÃO</b> . . . . .	<b>17</b>
1.1 Contexto . . . . .	17
1.2 Motivação . . . . .	18
1.3 Objetivo Geral e Específicos . . . . .	20
1.4 Organização do Documento . . . . .	21
<b>CAPÍTULO 2–ELEMENTOS TRANSPONÍVEIS</b> . . . . .	<b>22</b>
2.1 Considerações Iniciais . . . . .	22
2.2 Sistema de Classificação dos TEs . . . . .	24
2.3 Ferramentas de Classificação Automática . . . . .	26
2.4 Considerações Finais do Capítulo . . . . .	27
<b>CAPÍTULO 3–CLASSIFICAÇÃO E CLASSIFICAÇÃO HIERÁRQUICA</b> . . . . .	<b>28</b>
3.1 Considerações Iniciais . . . . .	28
3.2 Abordagem de Classificação Plana . . . . .	31
3.3 Abordagem Local de Classificação Hierárquica . . . . .	33
3.3.1 Classificador Local por Nó (LCN) . . . . .	34
3.3.2 Classificador Local por Nó Pai (LCPN) . . . . .	35
3.3.3 Classificador Local por Nível (LCL) . . . . .	36
3.4 Abordagem Global de Classificação Hierárquica . . . . .	38
3.5 Considerações Finais do Capítulo . . . . .	41
<b>CAPÍTULO 4–ALGORITMOS GENÉTICOS</b> . . . . .	<b>43</b>
4.1 Considerações Iniciais . . . . .	43
4.2 Representação das Soluções . . . . .	44
4.3 Codificação de Indivíduos . . . . .	45
4.4 Processo Evolutivo . . . . .	46
4.5 Operadores de Seleção . . . . .	47
4.6 Operadores de Cruzamento . . . . .	49
4.7 Operadores de Mutação . . . . .	50
4.8 Considerações Finais do Capítulo . . . . .	51
<b>CAPÍTULO 5–OTIMIZAÇÃO MULTIOBJETIVO</b> . . . . .	<b>52</b>
5.1 Considerações Iniciais . . . . .	52
5.2 Abordagens Multiobjetivo . . . . .	54
5.3 Algoritmos de Otimização Multiobjetivo . . . . .	55

5.4	Considerações Finais do Capítulo . . . . .	56
<b>CAPÍTULO 6–MÉTODOS PROPOSTOS . . . . .</b>		<b>58</b>
6.1	Classificação Hierárquica com um Algoritmo Genético (HC-GA) . . . . .	58
6.1.1	Codificação dos Indivíduos . . . . .	59
6.1.2	Construção do Consequente . . . . .	60
6.1.3	Inicialização da População . . . . .	61
6.1.4	Funções de Aptidão . . . . .	61
6.1.5	Operadores Genéticos . . . . .	63
6.1.6	Busca Local . . . . .	65
6.2	Classificação Hierárquica com um Algoritmo Genético de Soma Ponderada (HC-WGA) . . . . .	66
6.3	Classificação Hierárquica com um Algoritmo Genético Lexicográfico (HC-LGA) . . . . .	67
<b>CAPÍTULO 7–METODOLOGIA . . . . .</b>		<b>70</b>
7.1	Conjunto de Dados . . . . .	70
7.2	Métodos Comparados . . . . .	72
7.3	Treinamento e Teste . . . . .	72
7.4	Medidas de Avaliação . . . . .	74
7.5	Testes Estatísticos . . . . .	74
7.6	Testes de Correlação . . . . .	75
<b>CAPÍTULO 8–EXPERIMENTOS E DISCUSSÕES . . . . .</b>		<b>77</b>
8.1	Experimentos com o HC-GA . . . . .	77
8.2	Experimentos com o HC-WGA . . . . .	80
8.3	Experimentos com o HC-LGA . . . . .	84
8.4	Correlações entre Objetivos . . . . .	87
8.5	Comparações entre os Métodos Propostos . . . . .	101
8.6	Comparações com os Métodos da Literatura . . . . .	103
<b>CAPÍTULO 9–CONSIDERAÇÕES FINAIS . . . . .</b>		<b>108</b>
9.1	Contribuições da Pesquisa . . . . .	109
9.2	Trabalhos Futuros . . . . .	111
<b>REFERÊNCIAS . . . . .</b>		<b>113</b>
<b>A–CORRELAÇÕES ENTRE OBJETIVOS QUE NÃO ATINGIRAM SIGNIFICÂNCIA ESTATÍSTICA . . . . .</b>		<b>124</b>

## **LISTA DE ABREVIATURAS**

---

---

**ML** Machine Learning

**GA** Genetic Algorithm

**DT** Decision Tree

**HMM** Hidden Markov Model

**SVM** Support Vector Machine

**KNN** K-Nearest Neighbors

**NB** Naive Bayes

**NN** Neural Network

**RF** Random Forest

**MLP** Multi-Layer Perceptron

**TE** Transposable Element

**FC** Flat Classification

**CER** Classification Error Rate

**HC** Hierarchical Classification

**DAG** Directed Acyclic Graph

**MLNP** Mandatory Leaf-Node Prediction

**NMLNP** Non-Mandatory Leaf-Node Prediction

**LCN** Local Classifier per Node

**LCPN** Local Classifier per Parent Node

**LCL** Local Classifier per Level

**hP** hierarchical Precision

**hR** hierarchical Recall

**hF** hierarchical F-measure

**PC** Percentage Coverage

**VG** Variance Gain

**MOOP** Multi-Objective Optimization Problem

**PF** Pareto-Front

**PO** Pareto-Optimal

**PF** Pareto Front

**HMC-GA** Hierarchical Multi-label Classification with a Genetic Algorithm

**HC-GA** Hierarchical Classification with a Genetic Algorithm

**HC-WGA** Hierarchical Classification with a Weighted Genetic Algorithm

**HC-LGA** Hierarchical Classification with a Lexicographic Genetic Algorithm

## LISTA DE ALGORITMOS

---

---

1	Algoritmo Genético tradicional (EIBEN et al., 2003). . . . .	44
2	Procedimento geral dos GAs hierárquicos. . . . .	59
3	Cruzamento especializado das regras. . . . .	64
4	Mutação e generalização/restrição de operadores. . . . .	64
5	Busca local min-max. . . . .	65
6	Processo lexicográfico de otimização multiobjetivo. . . . .	68

## LISTA DE FIGURAS

---

---

Figura 1 – Ilustração dos mecanismos de mobilização dos Elementos Transponíveis. . .	24
Figura 2 – Taxonomia hierárquica de classificação de TEs. Adaptado de Wicker et al. (2007). . . . .	25
Figura 3 – Exemplo de uma hierarquia de classes. . . . .	29
Figura 4 – Estruturas hierárquicas em problemas de HC. Adaptado de Cerri (2014). . .	30
Figura 5 – Classificador multi-classe que prediz apenas classes folhas (SILLA; FREITAS, 2010). . . . .	32
Figura 6 – Classificador binário local por nó da hierarquia (SILLA; FREITAS, 2010). . .	34
Figura 7 – Classificador multi-classe local por nó pai (SILLA; FREITAS, 2010). . . . .	35
Figura 8 – Classificador multi-classe local por nível (SILLA; FREITAS, 2010). . . . .	37
Figura 9 – Cromossomo na codificação binária. Adaptado de Cerri (2014). . . . .	45
Figura 10 – Exemplo prático de codificação binária. . . . .	46
Figura 11 – Cálculo do fitness e seleção de indivíduos. . . . .	46
Figura 12 – Cruzamento entre pais e surgimento dos filhos (nova geração). . . . .	46
Figura 13 – Mutação da nova geração. . . . .	46
Figura 14 – Seleção por Roleta. . . . .	47
Figura 15 – Seleção por Amostragem Universal Estocástica. . . . .	48
Figura 16 – Seleção por Torneio. Adaptado de Razali et al. (2011). . . . .	48
Figura 17 – Método do Cruzamento em $K$ -Pontos. Adaptado de Cerri (2014). . . . .	49
Figura 18 – Método do Cruzamento Uniforme. . . . .	50
Figura 19 – Método de mutação <i>Bit-flip</i> . Adaptado de Castro (2006). . . . .	51
Figura 20 – Exemplo prático de um MOOP. Adaptado de Deb et al. (2016). . . . .	53
Figura 21 – Codificação de um indivíduo nos GAs hierárquicos. Adaptado de Cerri (2014). . . . .	60
Figura 22 – Exemplo de um vetor de classes nos GAs hierárquicos. . . . .	61
Figura 23 – Exemplificação do processo de busca e otimização realizado pelo HC-WGA. . . . .	66
Figura 24 – Exemplificação do processo de busca e otimização realizado pelo HC-LGA. . . . .	69
Figura 25 – Padrões de funções monotônicas. . . . .	76
Figura 26 – Distribuição dos dados de $AU(\overline{PRC})$ . . . . .	88
Figura 27 – Distribuição dos dados de hF. . . . .	88
Figura 28 – Distribuição dos dados de PC. . . . .	89
Figura 29 – Distribuição dos dados de VG. . . . .	90

Figura 30 – Correlações dos objetivos $AU(\overline{PRC})$ e PC utilizando o HC-WGA. . . . .	91
Figura 31 – Correlações dos objetivos $AU(\overline{PRC})$ e VG utilizando o HC-WGA. . . . .	92
Figura 32 – Correlações dos objetivos hF e PC utilizando o HC-WGA. . . . .	93
Figura 33 – Correlações dos objetivos hF e VG utilizando o HC-WGA. . . . .	94
Figura 34 – Correlações dos objetivos $AU(\overline{PRC})$ e hF utilizando o HC-LGA. . . . .	96
Figura 35 – Correlações dos objetivos hF e $AU(\overline{PRC})$ utilizando o HC-LGA. . . . .	97
Figura 36 – Correlações dos objetivos $AU(\overline{PRC})$ e PC utilizando o HC-LGA. . . . .	97
Figura 37 – Correlações dos objetivos $AU(\overline{PRC})$ e VG utilizando o HC-LGA. . . . .	97
Figura 38 – Correlações dos objetivos VG e $AU(\overline{PRC})$ utilizando o HC-LGA. . . . .	98
Figura 39 – Correlações dos objetivos hF e PC utilizando o HC-LGA. . . . .	98
Figura 40 – Correlações dos objetivos hF e VG utilizando o HC-LGA. . . . .	98
Figura 41 – Correlações dos objetivos VG e hF utilizando o HC-LGA. . . . .	99
Figura 42 – Correlações dos objetivos hP e hR utilizando o HC-LGA. . . . .	99
Figura 43 – Correlações dos objetivos hR e hP utilizando o HC-LGA. . . . .	99
Figura 44 – Correlação de Spearman dos objetivos $AU(\overline{PRC})$ e PC no PGSB com HC-LGA. . . . .	124
Figura 45 – Correlações dos objetivos PC e $AU(\overline{PRC})$ no PGSB com HC-LGA. . . . .	125
Figura 46 – Correlações dos objetivos PC e hF no PGSB com HC-LGA. . . . .	125
Figura 47 – Correlação de Pearson dos objetivos PC e VG no PGSB com HC-LGA. . . . .	126
Figura 48 – Correlações dos objetivos VG e PC no PGSB com HC-LGA. . . . .	126
Figura 49 – Correlações de $AU(\overline{PRC})$ (20%) + PC (80%) no PGSB com HC-WGA. . . . .	127
Figura 50 – Correlação de Spearman em $AU(\overline{PRC})$ (40%) + PC (60%) no PGSB com HC-WGA. . . . .	127
Figura 51 – Correlação de Pearson em $AU(\overline{PRC})$ (50%) + PC (50%) no PGSB com HC-WGA. . . . .	128
Figura 52 – Correlação de Spearman em $AU(\overline{PRC})$ (80%) + PC (20%) no PGSB com HC-WGA. . . . .	128
Figura 53 – Correlação de Pearson em hF (20%) + PC (80%) no PGSB com HC-WGA. . . . .	129
Figura 54 – Correlação de Pearson em hF (40%) + PC (60%) no PGSB com HC-WGA. . . . .	129
Figura 55 – Correlações de hF (50%) + PC (50%) no PGSB com HC-WGA. . . . .	130
Figura 56 – Correlação de Pearson em hF (60%) + PC (40%) no PGSB com HC-WGA. . . . .	130
Figura 57 – Correlação de Spearman dos objetivos $AU(\overline{PRC})$ e VG no REPBASE com HC-LGA. . . . .	131
Figura 58 – Correlações dos objetivos PC e $AU(\overline{PRC})$ no REPBASE com HC-LGA. . . . .	131
Figura 59 – Correlações dos objetivos PC e hF no REPBASE com HC-LGA. . . . .	132
Figura 60 – Correlações dos objetivos PC e VG no REPBASE com HC-LGA. . . . .	132
Figura 61 – Correlação de Spearman dos objetivos VG e $AU(\overline{PRC})$ no REPBASE com HC-LGA. . . . .	133
Figura 62 – Correlação de Spearman dos objetivos VG e hF no REPBASE com HC-LGA. . . . .	133

Figura 63 – Correlações de $AU(\overline{PRC})$ (20%) + PC (80%) no REPBASE com HC-WGA.	134
Figura 64 – Correlações de $AU(\overline{PRC})$ (40%) + PC (60%) no REPBASE com HC-WGA.	134
Figura 65 – Correlações de $AU(\overline{PRC})$ (50%) + PC (50%) no REPBASE com HC-WGA.	134
Figura 66 – Correlação de Spearman em $AU(\overline{PRC})$ (40%) + VG (60%) no REPBASE com HC-WGA. . . . .	135
Figura 67 – Correlações de hF (20%) + PC (80%) no REPBASE com HC-WGA. . . . .	135
Figura 68 – Correlação de Spearman em hF (50%) + PC (50%) no REPBASE com HC-WGA. . . . .	136

## LISTA DE TABELAS

---

---

Tabela 1 – Ferramentas usadas para classificar TEs. . . . .	26
Tabela 2 – Métodos locais de HC e suas respectivas estratégias. . . . .	38
Tabela 3 – Métodos globais de HC. . . . .	39
Tabela 4 – Estatísticas a respeito dos conjuntos de dados. . . . .	71
Tabela 5 – Esquema numérico de etiquetas aplicado a hierarquia de classes. . . . .	71
Tabela 6 – Relação de métodos da literatura usados nos comparativos. . . . .	72
Tabela 7 – Hiperparâmetros utilizados pelos métodos HC-GA, HC-WGA e HC-LGA. . . . .	73
Tabela 8 – Análise das predições do HC-GA considerando a Medida-F Hierárquica. . . . .	78
Tabela 9 – Análise de interpretabilidade das regras geradas pelo HC-GA. . . . .	79
Tabela 10 – Tempos de treinamento do HC-GA em minutos. . . . .	79
Tabela 11 – Análise das predições do HC-WGA considerando a Medida-F Hierárquica. . . . .	81
Tabela 12 – Análise de interpretabilidade das regras geradas pelo HC-WGA. . . . .	82
Tabela 13 – Tempos de treinamento do HC-WGA em minutos. . . . .	83
Tabela 14 – Análise das predições do HC-LGA considerando a Medida-F Hierárquica. . . . .	84
Tabela 15 – Análise de interpretabilidade das regras geradas pelo HC-LGA. . . . .	86
Tabela 16 – Tempos de treinamento do HC-LGA em minutos. . . . .	87
Tabela 17 – Correlações dos objetivos testados no HC-WGA. . . . .	95
Tabela 18 – Correlações dos objetivos testados no HC-LGA. . . . .	100
Tabela 19 – Comparação de desempenho entre os métodos propostos. . . . .	103
Tabela 20 – Resultados dos métodos de Homologia considerando a Medida-F Hierárquica. . . . .	104
Tabela 21 – Comparações entre métodos hierárquicos considerando medidas hierárquicas. . . . .	104
Tabela 22 – Resultados usando medidas não-hierárquicas. Adaptado de Pereira e Cerri (2017). . . . .	106
Tabela 23 – Análise de interpretabilidade das regras geradas por métodos globais. . . . .	106

# Capítulo 1

## INTRODUÇÃO

---

---

### 1.1 Contexto

Em Aprendizado de Máquina (no inglês, *Machine Learning (ML)*), muitos dos trabalhos da literatura que abordam a tarefa de classificação lidam com a chamada Classificação Plana (no inglês, *Flat Classification (FC)*), onde uma instância de um conjunto de dados é associada a apenas uma classe dentre um conjunto normalmente pequeno de classes (SILLA; FREITAS, 2010). Entretanto, existe um tipo de classificação mais complexa conhecida na literatura de ML como Classificação Hierárquica (no inglês, *Hierarchical Classification (HC)*), em que uma instância é assinalada não apenas a uma classe, mas também às suas superclasses (FREITAS; CARVALHO, 2007). Um problema de HC envolve dezenas e até centenas de classes que são arranjadas de acordo com uma estrutura hierárquica, a qual possui certas restrições.

Outras peculiaridades fazem parte de um problema hierárquico, como as estruturas de dados utilizadas, sendo comumente uma *Árvore* ou um *Grafo Acíclico Direcionado* (no inglês, *Directed Acyclic Graph (DAG)*). O que as difere é o número de caminhos que um classificador pode explorar na estrutura, sendo a *Árvore* o modelo mais simples. Outro ponto diz respeito às abordagens para tratar problemas hierárquicos, onde as principais são as *Abordagens Globais e Locais*. Na *Abordagem Global* um único classificador é induzido para lidar com todas as classes da hierarquia e a classificação de uma nova instância ocorre em um único passo (VENS et al., 2008). Já a *Abordagem Local* emprega diversos classificadores que utilizam de informações locais das classes para realizar predições que acontecem em etapas, explorando a hierarquia nível a nível, possuindo assim uma modularidade de treinamento (CERRI et al., 2014).

Uma das áreas de maior aplicação da HC é a *Bioinformática*, que em conjunto com a utilização de ML costuma resultar em melhores predições quando comparado a outras abordagens, conforme já apontado na literatura (LOUREIRO et al., 2013). Nesse contexto, um tema que vem ganhando atenção recentemente é a classificação automática dos chamados *Elementos Transponíveis* (no inglês, *TEs*), fragmentos de DNA capazes de se mover dentro do genoma de seus hospedeiros (MCCLINTOCK, 1993). Segundo pesquisas recentes, identificou-se que esses elementos são responsáveis por mutações em diversos organismos, inclusive no genoma

humano, o que lhes garantiu a alcunha de grandes responsáveis pela variabilidade genética das espécies (LOPES et al., 2008; FESCHOTTE, 2008; KAZAZIAN, 2004). A correta classificação desses elementos traz benefícios para o entendimento de várias espécies, bem como se deu a evolução das mesmas, aspectos importantes tanto para o campo da Biologia quanto da Bioinformática (NEKRUTENKO; LI, 2001; LEVY et al., 2008; VOLFF, 2006).

As classes de TEs podem ser estruturadas de forma hierárquica uma vez que existem sistemas que se propõem a organizá-las em taxonomias hierárquicas utilizando a estrutura de Árvore. Dois exemplos são os sistemas de classificação propostos por Finnegan e Fawcett (1986) e Wicker et al. (2007), que estruturam as classes de TEs em múltiplos níveis, as agrupando de acordo com as características em comum observadas. A taxonomia de Wicker et al. (2007) foi proposta com o intuito de atualizar e aprimorar o sistema apresentado por Finnegan e Fawcett (1986), e mesmo sendo recente e ainda em desenvolvimento, vem sendo cada vez mais adotada pela comunidade científica (NAKANO et al., 2017).

Existem diversas ferramentas de Bioinformática que podem ser empregadas na classificação de TEs, e entre essas, existem métodos criados unicamente para a classificação desses elementos (ALTSCHUL et al., 1990; ABRUSÁN et al., 2009). Entretanto, a maioria dessas ferramentas são destinadas a classificar alguma classe ou um sub-grupo específico dentre as classes de TEs existentes, e em sua grande maioria, utilizam estratégias simples como a homologia para classificar as sequências (STEINBISS et al., 2009; HOEDE et al., 2014).

## 1.2 Motivação

Diversas motivações conduzem esse estudo que engloba a área de Inteligência Computacional e ML assim como o campo da Bioinformática. A primeira delas diz respeito a HC, mais precisamente a Abordagem Global de classificação. Após uma revisão da literatura, observou-se a pouca variedade de métodos exemplo dessa abordagem, e que fica mais evidente quando comparado com a quantidade de trabalhos que seguem a Abordagem Local (SILLA; FREITAS, 2010). Além disso, são escassas as pesquisas que fazem uma comparação do desempenho preditivo entre as duas abordagens supracitadas (CERRI, 2014; PEREIRA et al., 2018). Portanto, uma das preocupações dessa pesquisa é justamente propor novos métodos globais visando explorar de forma mais aprofundada as características dessa abordagem, além de fornecer análises comparativas entre as abordagens Global e Local.

A segunda motivação tem relação com a interpretabilidade dos classificadores. Gerar regras de classificação para problemas hierárquicos não é uma tarefa trivial, já que uma regra deve ser capaz de classificar instâncias em um conjunto de classes de acordo com as restrições impostas pela hierarquia, e que variam dependendo do tipo de estrutura utilizada. A interpretabilidade das regras é diretamente afetada pela abordagem de classificação adotada. Enquanto métodos locais geram diversos conjuntos ou listas de regras, uma vez que usam de vários classificadores

que resultam em múltiplos modelos, métodos globais produzem um número menor de regras já que utilizam de um único classificador, e portanto, tendem a ser mais interpretáveis. No que tange a esse assunto, existe uma carência de modelos interpretáveis em ambas as abordagens. Ainda assim, a Abordagem Global é a que apresenta mais exemplos bem sucedidos, como o Clus-HMC (BLOCKEEL et al., 2002; VENS et al., 2008), HC4.5 (CLARE; KING, 2003) e o hAnt-Miner (OTERO et al., 2009), mas que ainda apresentam um desempenho preditivo ruim. Logo, um dos desafios desta pesquisa é desenvolver classificadores com bom desempenho e que ao mesmo tempo sejam capazes de gerar modelos interpretáveis.

A terceira motivação está relacionada com uma das áreas de maior aplicação da HC, a Bioinformática. Nela, algoritmos de ML têm sido amplamente utilizados e bons resultados vem sendo relatados, principalmente no que se remete a predição de estruturas e funções de proteínas (LOUREIRO et al., 2013). Muitos dos trabalhos nessa área envolve o uso de ferramentas que operam por meio de homologia, onde se compara uma sequência alvo com um banco de sequências de referência. Embora sejam úteis e apresentem bons resultados para muitos casos, estes métodos apresentam uma série de limitações, como ignorar muitas das propriedades bioquímicas das sequências ou a assunção de pressupostos negligentes, como classificar duas sequências de proteínas similares em uma mesma classe sendo que estas desempenham funções totalmente diferentes. Além disso, métodos de homologia frequentemente exigem certo trabalho manual uma vez que não criam modelos de inferência automaticamente a partir dos dados, diferente do que vários algoritmos de ML fazem. Por essas razões, e pelo fato de muitas das bases de dados do domínio serem organizadas hierarquicamente com classes e subclasses, a aplicação de métodos hierárquicos baseados em ML torna-se uma necessidade. Ademais, diversas limitações foram observadas em trabalhos da literatura que abordam a classificação de TEs, como a não exploração dos relacionamentos hierárquicos entre classes, a carência de métodos que classifiquem os elementos de acordo com uma taxonomia própria para eles, como a proposta em Wicker et al. (2007), e a inexistência de predições em várias superfamílias de TEs.

A quarta e última motivação diz respeito as abordagens de otimização multiobjetivo e sua aplicação no contexto desse trabalho. Como a preocupação de construir classificadores interpretáveis sem que haja perda de desempenho preditivo é algo cada vez mais presente tanto em ML quanto em Bioinformática, explorar métodos multiobjetivo capazes de lidar com esses dois objetivos naturalmente conflitantes é uma alternativa interessante a se considerar. Ademais, a partir de uma revisão da literatura, não foram encontrados trabalhos que relatem alguma aplicação de abordagens multiobjetivo na tarefa de HC ou na classificação de TEs. Logo, é pertinente verificar se essas abordagens trazem melhorias na otimização dos diversos critérios presentes em problemas hierárquicos, algo que ainda não fora inspecionado na literatura.

### 1.3 Objetivo Geral e Específicos

Com base no contexto e motivações apresentadas, essa pesquisa de mestrado tem como objetivo o desenvolvimento de classificadores hierárquicos capazes de realizar predições nas superfamílias de TEs presentes na taxonomia de Wicker et al. (2007), apresentando bom desempenho preditivo e interpretabilidade. São encontradas diversas definições de interpretabilidade na literatura, variando de acordo com o domínio de aplicação, visão do autor, entre outros. Nesta dissertação consideramos que, quanto menor, mais simples e compreensível for o modelo, mais interpretável ele é. Portanto, a interpretabilidade é aqui avaliada em questão de complexidade, e como os modelos implementados e comparados geram listas de regras, ela tem relação com o quão grande ou pequena é a lista de regras e o quão extensa são essas regras.

Para a construção dos métodos, Algoritmos Genéticos (no inglês, *Genetic Algorithms* (GAs)) são utilizados como base dos classificadores hierárquicos que operam segundo a Abordagem Global. Esses algoritmos facilitam a codificação de modelos interpretáveis como regras de classificação, e portanto, foram adotados a fim de induzir uma lista de regras de HC para TEs. Mais precisamente, abordagens multiobjetivo são exploradas junto aos GAs a fim de lidar de forma mais adequada com os objetivos naturalmente conflitantes que são o desempenho preditivo e a interpretabilidade. Como objetivos específicos, listam-se:

- Desenvolver GAs Multiobjetivo que gerem modelos de regras com bom desempenho preditivo e interpretabilidade;
- Analisar as predições realizadas nos múltiplos níveis hierárquicos das classes de TEs;
- Verificar se há ganhos relevantes na utilização de abordagens multiobjetivo e qual é a mais adequada para o problema em questão;
- Realizar uma análise da correlação dos objetivos considerados;
- Realizar análises comparativas entre as Abordagens Global, Local e Plana;
- Comparar o desempenho dos GAs com ferramentas de Bioinformática utilizadas na classificação de sequências de TEs;
- Comparar a interpretabilidade dos modelos gerados pelos GAs entre si e com um método exemplo da Abordagem Global;
- Coletar e comparar tempos de treinamento dos métodos hierárquicos implementados.

## 1.4 Organização do Documento

O restante deste documento está organizado da seguinte forma: no Capítulo 2 são introduzidos os conceitos acerca dos Elementos Transponíveis, cujo os dados são a fonte que move essa pesquisa de mestrado; no Capítulo 3 são apresentados os conceitos fundamentais de Classificação e Classificação Hierárquica, além de ser feita uma revisão da literatura relativa a trabalhos que abordam esse segundo tipo de classificação; os GAs, técnicas evolutivas utilizadas como a base dos classificadores desenvolvidos nesse trabalho são introduzidos no Capítulo 4; conceitos acerca da Otimização Multiobjetivo como sua definição formal, particularidades dos problemas e as abordagens existentes são vistas no Capítulo 5; em seguida, no Capítulo 6 são apresentados em detalhes os métodos hierárquicos globais propostos que são baseados tanto em GAs simples quanto em Multiobjetivo; a Metodologia dessa pesquisa é apresentada no Capítulo 7, que inclui a especificação dos conjuntos de dados, os métodos da literatura comparados, o processo de treinamento e teste, as correlações e testes estatísticos além das medidas de avaliação empregadas; o Capítulo 8 reporta os resultados dos métodos propostos e demais implementados, assim como as discussões pertinentes; por fim, as considerações finais da pesquisa e os trabalhos futuros são discutidos no Capítulo 9.

# Capítulo 2

## ELEMENTOS TRANSPONÍVEIS

---

---

### 2.1 Considerações Iniciais

Anterior a descoberta da estrutura do DNA, Barbara McClintock ao realizar experimentos com o genoma do milho em meados de 1940, identificou o que viriam a ser chamados de Elementos Transponíveis (no inglês, *Transposable Elements (TEs)*), ou Elementos Móveis como também são conhecidos, sendo esses fragmentos de DNA que possuem a capacidade de se mover no genoma (BIÉMONT, 2010; JURKA et al., 2005; MCCLINTOCK, 1993). Tal estudo estabeleceu um novo paradigma onde o genoma não seria estático como se pensava até então, gerando certo ceticismo em parte da comunidade científica da época (PRAY; ZHAUROVA, 2008; REBOLLO et al., 2012; JANICKI et al., 2011). A existência dos TEs só ganhou credibilidade na década de 80, graças a publicação de trabalhos que relatavam sua identificação e mutações causadas por eles em vários organismos, como no milho (10% das mutações) e em *Drosophila*<sup>1</sup> (50% das mutações) (KAZAZIAN, 1998; LINDSLEY; ZIMM, 2012).

Como reconhecimento da descoberta dos TEs e considerando o impacto no estudo e entendimento do genoma, Barbara McClintock foi premiada com o Nobel de Fisiologia ou Medicina em 1983. Em detrimento do prestígio gerado juntamente dos estudos cada vez mais frequentes por parte da comunidade científica, a teoria que coloca os TEs como grandes responsáveis pela variabilidade genética das espécies é bem aceita atualmente (JURKA et al., 2005; REBOLLO et al., 2012; NEKRUTENKO; LI, 2001; LAGEMAAT et al., 2003; KAZAZIAN, 2004; VOLFF, 2006; FESCHOTTE, 2008; LEVY et al., 2008; LOPES et al., 2008).

Desde sua descoberta os TEs têm sido foco de pesquisas que investigam seu mecanismo de transposição. Estudos recentes apontam que sua transposição pode provocar mudanças significativas e de larga escala no genoma, como rearranjos cromossômicos, eliminação e criação de novos genes e até mesmo inovações biológicas (FESCHOTTE; PRITHAM, 2007). Em Slotkin e Martienssen (2007), Böhne et al. (2008) e Sela et al. (2010), é apontado que os TEs possuem certo papel na manutenção, estrutura e funcionamento do genoma em diferentes espécies.

<sup>1</sup> Conhecida como a mosca da banana ou mosca das frutas.

Outras pesquisas apontam ainda que os TEs são influenciados diretamente pelo ambiente, sendo incentivados à ação quando em situação de estresse (LEVIN; MORAN, 2011; DAI et al., 2007; BIÉMONT; VIEIRA, 2006). Em Lagemaat et al. (2003), foram analisados genes associados a respostas ao estresse e a estímulos externos em camundongos e humanos. Dessa análise foi possível observar um maior acúmulo de TEs em comparação a outras classes de genes, o que reforça a hipótese de que esses elementos podem estar relacionados a respostas adaptativas dos hospedeiros, sendo eles uma fonte de inovações mediante situações adversas.

Os TEs são frequentemente apontados como grandes responsáveis pelas mutações nos genomas de diversas espécies. Em Alberts et al. (2009), foi estudado o genoma de humanos e camundongos e constatou-se que TEs geram novas mutações. Em camundongos os TEs são responsáveis por cerca de 10% das mutações, enquanto que em humanos a taxa é menor, situando-se entre 0.1% e 1%. No entanto, em termos de composição, estima-se que aproximadamente 45% do genoma humano deriva de TEs (BIÉMONT; VIEIRA, 2006; BIÉMONT, 2010).

A relação entre TEs e o câncer também tem sido apontada na literatura (LEVIN; MORAN, 2011; CHÉNAIS, 2013). Em Miki et al. (1992), foi analisado o gene APC, considerado um supressor do câncer colorretal. Segundo os autores, não existia qualquer relato na literatura de uma interrupção de sequência codificante de um gene supressor de tumor mediada por um TE.

Como evidência da importância do estudo dos TEs, é possível citar algumas iniciativas de instituições de renome que têm movido esforços na investigação desses elementos. O projeto Genoma da cana-de-açúcar, também conhecido como SUCEST (do inglês, *Sugarcane EST Project*) (CALSA et al., 2004; VETTORE et al., 2003), e apoiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), é um bom exemplo de sucesso. Esse projeto já revelou que existem altas taxas de TEs expressos na cana-de-açúcar, planta que tem grande importância no cenário econômico do país. Segundo Calsa et al. (2004) e Vettore et al. (2003), o estudo dos TEs pode fornecer meios de controlar e manipular os genes responsáveis por processos adaptativos ao ambiente na cana-de-açúcar, o que abre possibilidades para a criação de variedades da cana resistentes a determinados climas.

Outro investimento nacional de iniciativa da FAPESP em conjunto com diversas universidades do estado de São Paulo é o programa BIOEN<sup>2</sup>, que visa integrar os estudos sobre a cana-de-açúcar a outras plantas do bioma brasileiro a fim de impulsionar a produção de biocombustíveis no país. Tanto o BIOEN como o SUCEST evidenciam a relevância da pesquisa dos TEs na área de Bioenergia e consequentemente, na economia do Brasil.

---

<sup>2</sup> <http://bioenfapesp.org/> (acessado em 21 de junho de 2016).

## 2.2 Sistema de Classificação dos TEs

A presença de TEs é abundante na natureza, principalmente nos organismos de animais e plantas. No trigo, cerca de 90% do genoma é constituído de sequências repetidas, e destas, 68% correspondem a TEs (LI et al., 2004). Em *Drosophila melanogaster*, *Homo sapiens*, *Rana esculenta*<sup>3</sup> e *Zea mays*<sup>4</sup>, os TEs correspondem a 15-22%, 45%, 77% e 60% do genoma dessas espécies, respectivamente (BIÉMONT; VIEIRA, 2006). Devido ao grande número de TEs descobertos a partir de meados de 1980, um sistema para sua classificação se fez necessário. O primeiro sistema de classificação relatado na literatura foi proposto por Finnegan (1989), onde os elementos foram discriminados de acordo com seu mecanismo de transposição: *Retrotransposons* (Classe I) e *Transposons* (Classe II). A Figura 1 ilustra esses dois modos de transposição.

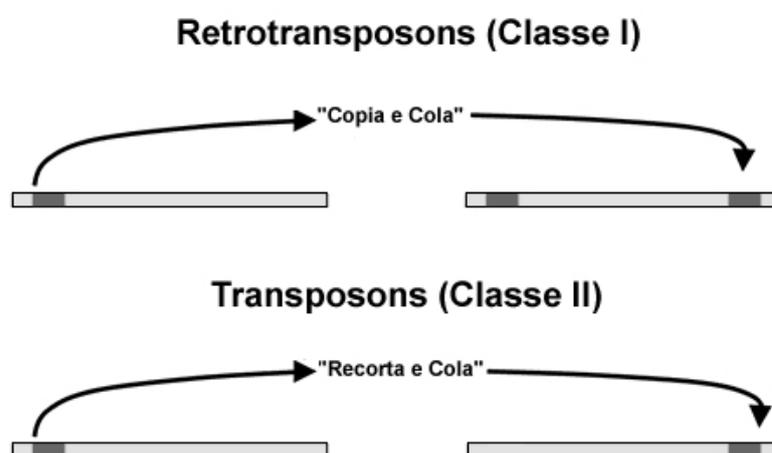


Figura 1 – Ilustração dos mecanismos de mobilização dos Elementos Transponíveis.

Os TEs pertencentes a Classe I utilizam um mecanismo de mobilização baseado em um intermediário de RNA. Tal modo de transposição ficou conhecido como “copiar e colar” (FINNEGAN; FAWCETT, 1986). Já os elementos da Classe II não utilizam um intermediário de RNA. Em vez disso, realizam a mobilização de forma direta por meio de uma cópia de DNA. Esse modo de transposição ficou conhecido como “recortar e colar” (FINNEGAN; FAWCETT, 1986).

Com os decorrentes sequenciamentos de genomas ao longo dos anos, TEs com novas características foram descobertos. Logo, o sistema proposto por Finnegan e Fawcett (1986) tornou-se obsoleto e um novo sistema de classificação foi proposto por Wicker et al. (2007), conforme exibido na Figura 2. Esse sistema manteve as duas classes propostas inicialmente por Finnegan e Fawcett (1986) (*Retrotransposons* e *Transposons*), e ainda, adicionou sub-classes a fim de abranger as novas características observadas nos elementos descobertos.

<sup>3</sup> Rã europeia.

<sup>4</sup> Milho.

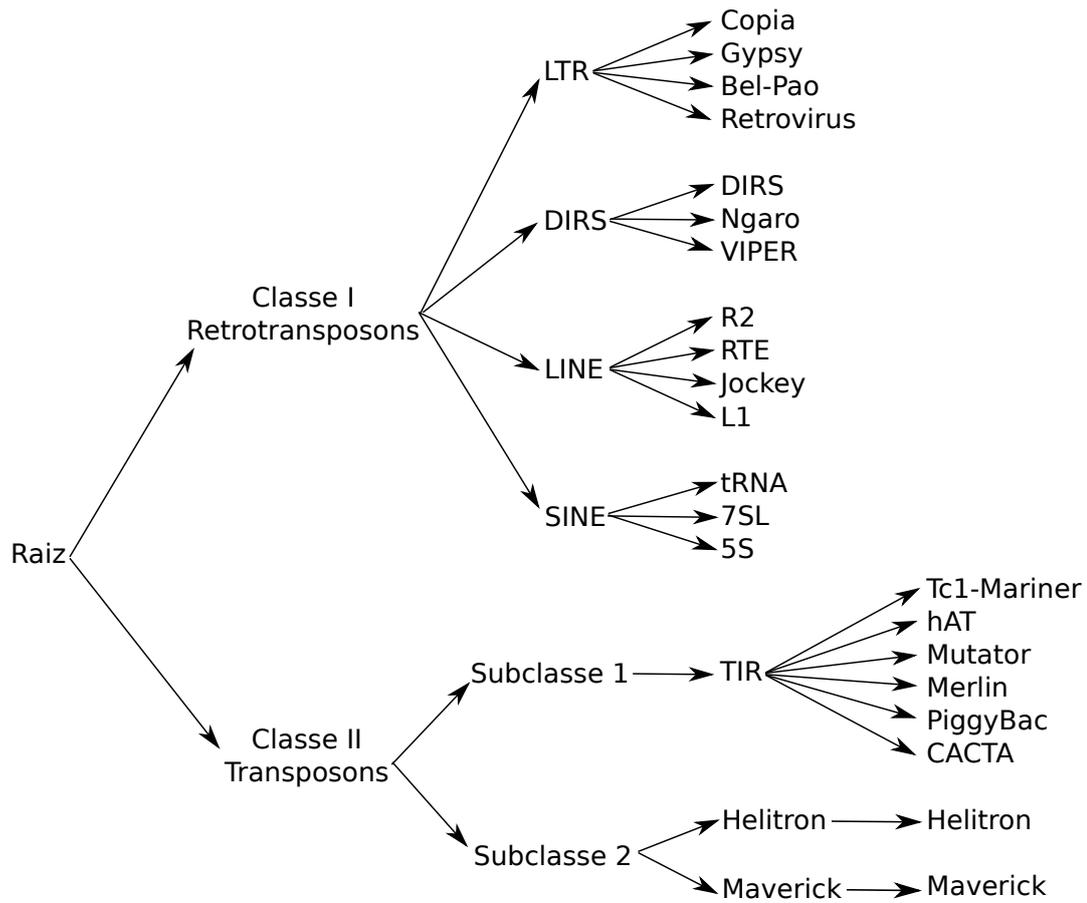


Figura 2 – Taxonomia hierárquica de classificação de **TEs**. Adaptado de [Wicker et al. \(2007\)](#).

O sistema de classificação de [Wicker et al. \(2007\)](#) consiste em uma taxonomia hierárquica de **TEs** organizada da seguinte maneira: **Classe** > **Subclasse** > **Ordem** > **Superfamília**. No topo da hierarquia estão as duas *Classes* primordiais *Retrotransposons* e *Transposons* que distinguem um **TE** de acordo com o seu mecanismo de transposição.

Os *Transposons* apresentam duas *Subclasses* (Subclasse 1 e Subclasse 2). Na *Subclasse 1* se encontram os elementos que realizam auto cópias para uma nova inserção cortando apenas uma das fitas do DNA. Já na *Subclasse 2* estão aqueles que se deslocam totalmente do sítio onde se encontravam para se inserirem em outro local do genoma cortando as duas fitas do DNA. A partir dos *Retrotransposons* e das *Subclasses* dos *Transposons* existem as subcategorias denominadas *Ordens* (LTR, DIRS, Helitron, etc). Essa subdivisão discrimina os elementos com base nas diferenças do mecanismo de inserção, organização geral e enzimologia dos **TEs**. Por fim, na categoria conhecida como *Superfamílias* (Copia, L1, Maverick, etc), são agrupados os elementos que compartilham a estratégia de replicação, mas diferem-se uns dos outros, na estrutura de proteínas, presença e/ou tamanho de uma repetição genética ou sítio de duplicação.

Cabe salientar que a taxonomia apresentada na Figura 2, não endereça todas as categorias de **TEs** do sistema proposto por [Wicker et al. \(2007\)](#). Isso se dá pelo fato dos conjuntos de dados utilizados nesse trabalho não cobrirem toda a taxonomia de **TEs**, não agregando as categorias de

*Superfamília, Família e Subfamília*. Tal situação já era esperada uma vez que esse sistema ainda está em desenvolvimento e vem aos poucos sendo adotado pela comunidade científica.

## 2.3 Ferramentas de Classificação Automática

Uma variedade de ferramentas para a classificação automática de TEs pode ser encontrada na literatura, que variam de acordo com os atributos utilizados, níveis da hierarquia dos TEs exploradas e tipos de estratégias empregadas. A maioria dessas ferramentas utilizam como atributos os domínios funcionais conservados,  $k$ -mers e características estruturais das sequências. Dentre as categorias da hierarquia dos TEs, essas ferramentas comumente se concentram no nível de *Ordem*, classificando os elementos em LTR, TIR, LINE, ou SINE. Já a respeito das estratégias de classificação, a mais frequentemente utilizada pelos métodos encontrados é a homologia, onde são realizadas buscas por repetições específicas em sequências de DNA. No restante dessa seção serão apresentadas algumas ferramentas que podem ser utilizadas ou são propriamente específicas para a classificação de TEs. A Tabela 1 apresenta uma visão geral dessas ferramentas.

Tabela 1 – Ferramentas usadas para classificar TEs.

<b>Método</b>	<b>Artigo da Literatura</b>
BLAST	<a href="#">Altschul et al. (1990)</a>
Repeat-Masker	<a href="#">Smit et al. (1996)</a>
LTRDigest	<a href="#">Steinbiss et al. (2009)</a>
TEClass	<a href="#">Abrusán et al. (2009)</a>
RepClass	<a href="#">Feschotte et al. (2009)</a>
Pastec	<a href="#">Hoede et al. (2014)</a>

Em [Altschul et al. \(1990\)](#), é proposto o BLAST, ferramenta bastante popular baseada em homologia e amplamente utilizada em aplicações de Bioinformática ([WICKER et al., 2007](#)). O BLAST busca por repetições específicas em sequências de DNA, classificando instâncias dada a sua similaridade com sequências já anotadas em um banco de dados de referência. Esse possui uma variação denominada BLASTn que recebe como entrada uma cadeia de nucleotídeos. Em [Smit et al. \(1996\)](#), é proposto o Repeat-Masker, método que opera de forma semelhante ao BLAST, procurando e mascarando repetições em sequências que possuam similaridades com sequências previamente anotadas. Em [Steinbiss et al. \(2009\)](#), é proposto o LTRDigest, um método específico para a identificação de LTRs que utiliza uma lista desses como referencial para anotar sequências com domínios proteicos e outros atributos estruturais através do uso de HMMs (do inglês, *Hidden Markov Models*). Em [Abrusán et al. \(2009\)](#), é apresentado o TEClass, um método que usa ML para classificar sequências de TEs. O TEClass classifica sequências por meio do uso de uma hierarquia de classificadores SVM, onde frequências de oligômeros e  $k$ -mers são adotados como atributos. Em [Feschotte et al. \(2009\)](#), é introduzido o

RepClass, ferramenta que combina um módulo baseado em homologia, um módulo de busca por características estruturais e um módulo de busca por duplicações em sítios alvo em uma única saída preditiva. Em [Hoede et al. \(2014\)](#), é apresentado o Pastec, um método para a classificação de **TEs** que utiliza atributos estruturais, como o tamanho das sequências dos elementos, presença de um LTR ou TIR e repetições específicas em sequências, além de considerar similaridades com sequências já anotadas e domínios funcionais conservados encontrados com **HMMs**.

## 2.4 Considerações Finais do Capítulo

Este capítulo focou-se na introdução dos chamados Elementos Transponíveis, sendo que seu conceito, particularidades, presença na natureza, relevância e pesquisas concentradas em seu estudo foram elencadas. Além disso, fora apresentado um sistema de classificação unificado constituído de uma taxonomia hierárquica de **TEs** proposto por [Wicker et al. \(2007\)](#), a qual será tomada como base nessa pesquisa. Adicionalmente, foi feita uma revisão da literatura focada em ferramentas de Bioinformática utilizadas na classificação de **TEs**.

Apesar da variedade de ferramentas passíveis de utilização para a classificação automática de **TEs**, a grande maioria apresenta limitações relevantes. A mais recorrente é assumir pressupostos com base apenas nas similaridades entre sequências, o que pode acarretar a geração e propagação de classificações incorretas. Ademais, nenhuma das ferramentas listadas, salvo o trabalho de [Abrusán et al. \(2009\)](#), aprende modelos de classificação a partir dos dados.

Conforme descreve [Loureiro et al. \(2013\)](#), a utilização de métodos baseados em **ML** melhora significativamente o desempenho preditivo da classificação dos **TEs** em comparação com outros métodos. Além disso, o uso de **ML** possui duas vantagens em relação as demais ferramentas listadas: (1) Uma vez que um classificador é treinado e um modelo indutivo é gerado, a classificação de novas instâncias acontece de maneira automática e direta, sem que seja necessário demais processamentos; e (2) A utilização de **ML** permite explorar tanto os relacionamentos hierárquicos quanto as características bioquímicas dos **TEs**, propriedades estas que são ignoradas pelos métodos previamente apresentados.

# Capítulo 3

## CLASSIFICAÇÃO E CLASSIFICAÇÃO HIERÁRQUICA

---

---

### 3.1 Considerações Iniciais

Dentre as muitas tarefas em **ML**, a classificação está entre uma das mais importantes e utilizadas tanto na indústria quanto na academia (BARROS et al., 2013). Essa atividade pode ser descrita como a busca por um modelo capaz de abstrair informações a partir de um conjunto de dados de treinamento, que com base na experiência obtida, consiga categorizar uma instância ainda não observada (conjunto de teste) em uma determinada classe dentre um conjunto de classes pertencentes ao domínio do problema (RUSSELL et al., 1995; MITCHELL, 1997).

Uma definição formal do problema de classificação é: dado um conjunto de instâncias de treinamento composto por pares  $(\mathbf{x}_i, c_j)$ , no qual  $\mathbf{x}_i$  corresponde a um vetor de atributos que descrevem uma instância  $i$  e  $c_j$  sua classe associada, visa-se encontrar uma função que mapeie cada  $\mathbf{x}_i$  para sua classe correspondente no vetor de classes  $c$ , tal que  $i = \{1, 2, \dots, n\}$  onde  $n$  é o número de instâncias e  $j = \{1, 2, \dots, m\}$  onde  $m$  é o número classes do problema (MITCHELL, 1997; TAN et al., 2006). Conforme aponta Tan et al. (2006), essa função a ser encontrada é obtida através de ajustes dos parâmetros livres do algoritmo de indução utilizado.

A tarefa de classificação faz parte de um tipo de aprendizado em **ML** conhecido como *Aprendizado Supervisionado*, em que os algoritmos utilizados realizam induções a partir de instâncias previamente rotuladas (RUSSELL et al., 1995; MITCHELL, 1997). Esses classificadores são treinados com instâncias que contêm a informação de sua saída esperada (classe) a fim de aprender um modelo indutivo. Deste modo, após passado a fase de treinamento, esses classificadores são capazes de receber dados ainda não observados, e através da experiência prévia obtida, realizar previsões. Essa segunda etapa é conhecida como fase de *teste* ou *validação*.

Segundo a literatura de **ML**, a tarefa de classificação como é tradicionalmente conhecida é chamada de Classificação Plana (**FC**), onde uma instância é associada a uma única classe em um conjunto finito e geralmente pequeno de classes (COSTA et al., 2007). Entretanto, existe um outro

tipo de classificação mais complexo conhecida na literatura como Classificação Hierárquica (HC), em que uma instância pode ser associada não só a uma classe mas a superclasses e subclasses arranjadas de maneira hierárquica (SILLA; FREITAS, 2010). Embora exista um número maior de pesquisas focadas em FC, problemas de HC vem recebendo cada vez mais atenção nos últimos anos, especialmente no campo da Bioinformática. A escassez de pesquisa ainda presente na área pode ser um reflexo da complexidade desses problemas, uma vez que as classes a serem preditas estão arranjadas de acordo com uma estrutura hierárquica e cada instância pode estar associada a múltiplas classes simultaneamente, sendo muitas vezes dezenas ou centenas de classes (SILLA; FREITAS, 2010). As tarefas de HC são relevantes em vários domínios e estão comumente relacionadas as áreas de Processamento de Linguagem Natural com a classificação textual de documentos, e na Bioinformática com a predição de funções de proteínas e estruturas proteicas (CERRI et al., 2016; FREITAS; CARVALHO, 2007).

Como exemplo do funcionamento da HC, considere a hierarquia apresentada na Figura 3. Tomando como base o nó 1.1.2, é possível constatar que este é subclasse de 1.1, e que consequentemente é subclasse de 1, formando assim todo o caminho pela hierarquia até a raiz. No caso da predição da classe 1.1.2, deve-se manter a consistência hierarquia das classes, de forma que todas as superclasses devem ser automaticamente preditas também.

Em problemas hierárquicos, um conjunto de classes pode ser representado como uma matriz  $V = \{v_{x_1}, v_{x_2}, \dots, v_{x_N}\}$ , em que cada  $v_{x_i}$  é um vetor de classes binário  $c$ -dimensional que representa as classes da hierarquia associadas a uma instância  $x_i$ , sendo  $c$  o número de classes da hierarquia. Cada posição no vetor  $v_{x_i}$  representa uma classe, que recebe o valor 1 se  $x_i$  pertence a ela, ou 0 caso contrário. No caso de uma abordagem probabilística,  $v_{x_i}$  pode conter valores entre [0-1] representando a probabilidade de associação de cada classe a dada instância.

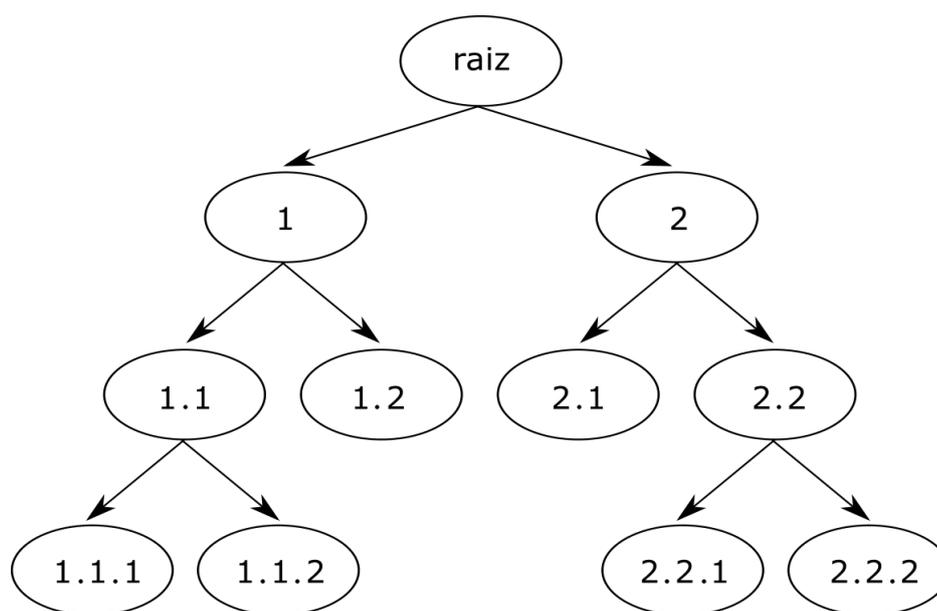


Figura 3 – Exemplo de uma hierarquia de classes.

Segundo [Freitas e Carvalho \(2007\)](#) e [Sun e Lim \(2001\)](#), métodos de **HC** diferem-se em vários quesitos, mais precisamente, são elencados três pontos chave. O primeiro deles diz respeito a estrutura hierarquia utilizada. Em problemas hierárquicos as classes podem estar estruturadas como uma *Árvore* ou *Grafo Acíclico Direcionado (DAG)*, sendo que a principal diferença está na relação entre os nós da hierarquia. Conforme mostra a [Figura 4](#), em uma estrutura em *Árvore* cada *nó filho* possui **apenas um nó pai**, enquanto que em um **DAG** cada *nó filho* pode ter **mais de um nó pai**, tornando-a mais complexa. Em ambas as estruturas os nós representam as classes de um problema, sendo o *nó raiz* correspondente a “qualquer ou todas as classe”, denotando a ausência de conhecimento sobre a classe de uma instância.

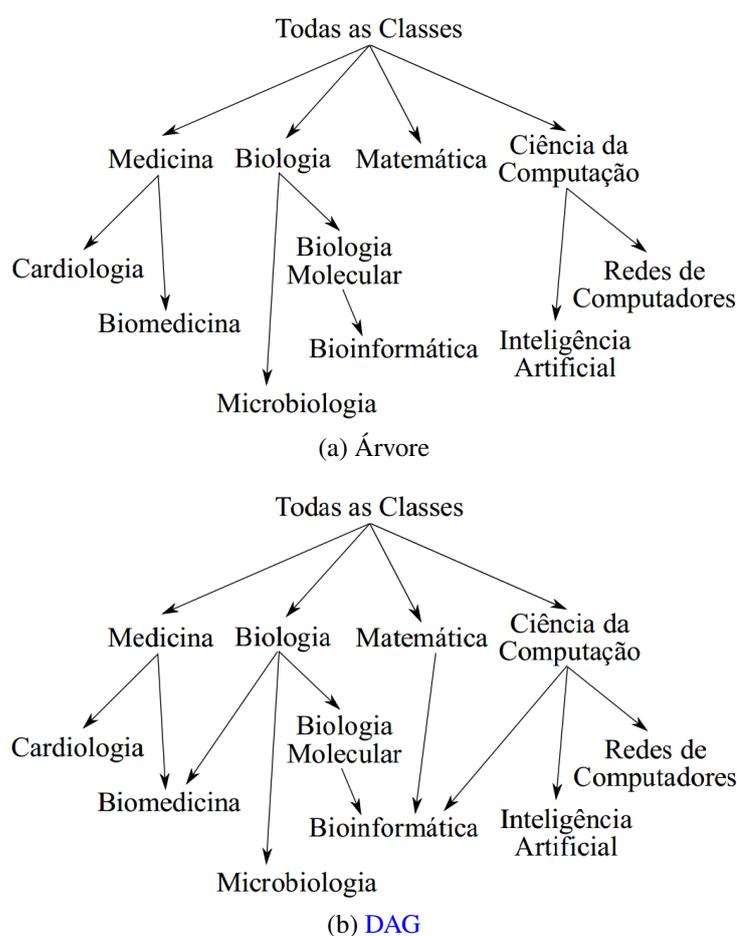


Figura 4 – Estruturas hierárquicas em problemas de **HC**. Adaptado de [Cerri \(2014\)](#).

Ademais, em uma estrutura de *Árvore* um único caminho pode ser traçado do nó raiz até qualquer outro nó da hierarquia. Já no caso de hierarquias estruturadas como **DAGs**, a partir do nó raiz, pode haver mais de um caminho até um dado nó da hierarquia. Tais características devem ser levadas em consideração na avaliação e desenvolvimento de classificadores hierárquicos.

O segundo critério está relacionado ao quão profunda é a exploração dos nós da hierarquia. Problemas de **HC** frequentemente têm como objetivo a classificação de instâncias em níveis inferiores na hierarquia ([FREITAS; CARVALHO, 2007](#)). Quanto mais profunda é a classe, mais específico é o conhecimento associado. Entretanto, pode haver casos em que não se tem a

confiabilidade desejada para classificar uma instância em nós próximos ou nas folhas, já que esses tendem a ter menos instâncias que as classes dos níveis superiores, tornando mais árdua sua predição. Nesses casos é preferível realizar a classificação em níveis mais altos na hierarquia. Em contraste, quanto mais próximo a classe predita é da raiz, menos útil é o conhecimento associado, entretanto, menor tende a ser sua Taxa de Erro de Classificação (no inglês, *Classification Error Rate (CER)*). Portanto, um classificador hierárquico deve alcançar um equilíbrio entre a especificidade das classes e o CER (SILLA; FREITAS, 2010). Em alguns problemas de HC, todos as instâncias devem ser associados a classes folhas da hierarquia, e esses são conhecidos como problemas de Predição Obrigatória em Nós Folha (no inglês, *Mandatory Leaf-Node Prediction (MLNP)*). Quando não existe tal obrigação, são denominados como problemas de Predição Não Obrigatória (ou Opcional) em Nós Folha (no inglês, *Non-Mandatory Leaf-Node Prediction (NMLNP)*) (FREITAS; CARVALHO, 2007; SILLA; FREITAS, 2010).

O terceiro e último ponto chave em HC diz respeito a como as estruturas hierárquicas são exploradas. Quando é utilizado um conjunto de classificadores que exploram informações locais durante o treinamento e suas predições são combinadas na fase de testes, esta é referenciada na literatura como uma **Abordagem Local** (CERRI et al., 2016). Nos casos em que um único classificador lida com toda a hierarquia de uma só vez, trata-se da **Abordagem Global** (em alguns bibliografias pode aparecer como *Big Bang* ou *One-Shot*) (VENS et al., 2008). Existem ainda trabalhos que transformam um problema de HC em FC (FREITAS; CARVALHO, 2007).

Nos últimos anos devido ao acréscimo no número de publicações acerca de HC, gerou-se muitas terminologias diferentes. Em Silla e Freitas (2010), é feito um apanhado dos trabalhos de HC através de vários domínios com o intuito de analisar e chegar a um consenso a respeito das terminologias existentes. A abordagem *top-down*, por exemplo, frequentemente apontada como sinônimo de Abordagem Local, não é uma abordagem de classificação hierárquica por completo. Segundo os autores, trata-se de um método usado na fase de testes (e não no treinamento) a fim de evitar ou corrigir inconsistências preditivas conforme os níveis de uma hierarquia são explorados. Portanto, nessa dissertação de mestrado é considerado apenas as abordagens local, global e plana como abordagens válidas para tratar problemas hierárquicos. Essas abordagens, bem como os trabalhos da literatura que as utilizam são melhor explorados a seguir.

## 3.2 Abordagem de Classificação Plana

Essa abordagem representa a maneira mais simples de lidar com problemas hierárquicos, em que são utilizados algoritmos tradicionais de ML que ignoram os relacionamentos hierárquicos entre classes (SILLA; FREITAS, 2010). A Abordagem de Classificação Plana é caracterizada pela transformação ou redução de um problema hierárquico a um problema de FC multi-classe. Logo, parte-se da ideia de que a FC pode ser interpretado como um caso particular de HC, em que não é interessante predizer classes intermediárias, conforme ilustra a Figura 5. Essa abordagem

consegue trabalhar com ambas as estruturas hierárquicas previamente apresentadas (*Árvore* e *DAG*), e também em ambos os tipos de restrição preditiva (*NMLNP* e *MLNP*), dependendo é claro, de como o problema é modelado. Comumente, algoritmos clássicos de *ML* como *Árvore de Decisão* (no inglês, *Decision Tree (DT)*), *Classificador Bayesiano Ingênuo* (no inglês, *Naive Bayes (NB)*) e o método dos *K* vizinhos mais próximos (no inglês, *K-Nearest Neighbors (KNN)*) podem ser aplicados. Cabe salientar que tanto a fase de treinamento quanto a de teste ocorrem de maneira convencional nessa abordagem (CERRI, 2014).

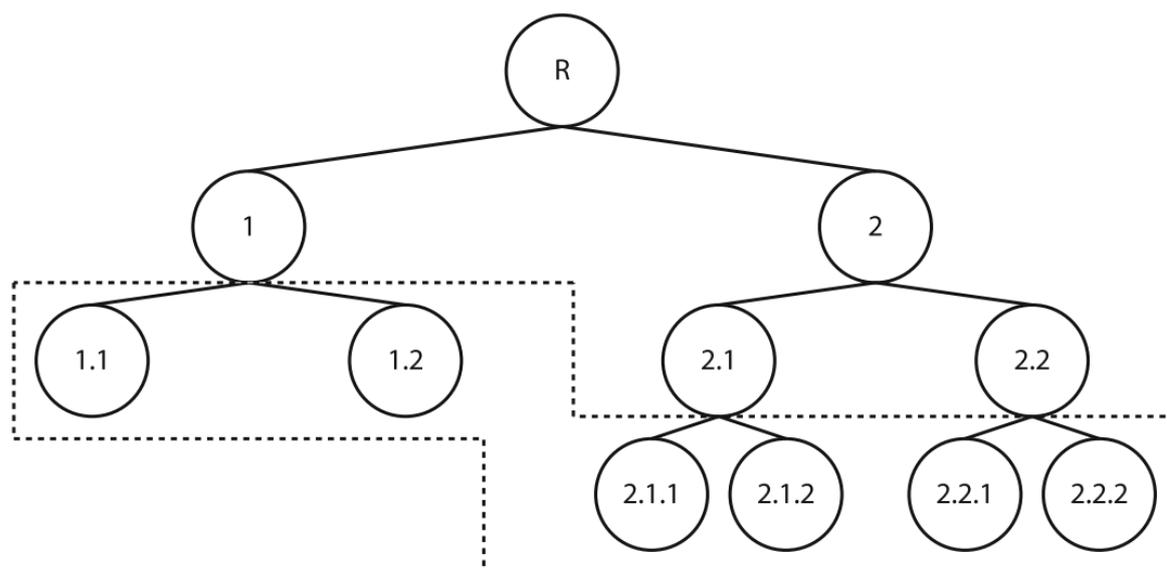


Figura 5 – Classificador multi-classe que prediz apenas classes folhas (SILLA; FREITAS, 2010).

Frequentemente, trabalhos que propõem ou investigam métodos hierárquicos acabam validando-os, naturalmente, em problemas hierárquicos e/ou comparando-os com outros métodos hierárquicos, sendo que normalmente as abordagens locais ou globais são empregadas. Entretanto, ainda que em menor quantidade, alguns estudos comparam abordagens puramente hierárquicas com a *FC*, a fim de verificar se a exploração de relacionamentos hierárquicos realmente contribui na qualidade das predições em determinados problemas.

Em Zimek et al. (2010), são realizadas comparações com classificadores que exploram informações hierárquicas e classificadores tradicionais, entre esses *SVM*, *DT* e *ensembles*. São utilizados dados reais e sintéticos relacionados a classificação de enzimas de proteínas. Apesar da natureza hierárquica dos problemas, os autores concluem que nem sempre a exploração de informações hierárquicas resulta em melhores resultados. Segundo eles, a exploração das hierarquias de classes melhora as predições no caso de dados sintéticos apenas.

Em Ghazi et al. (2010), a *HC* é explorada na classificação automática de textos através de emoções. Os autores propõem um método que organiza hierarquicamente a neutralidade e a polaridade de emoções presentes em textos, comparando-o com um algoritmo de classificação tradicional em dois conjuntos de dados. De acordo com os experimentos, é verificado que o método proposto não apenas supera a abordagem tradicional comparada, mas também consegue

suavizar o problema do alto desbalanceamento presente nos conjuntos de dados, característica a qual normalmente apresenta impacto negativo no desempenho dos classificadores.

Em [Silla e Kaestner \(2013\)](#), é realizada a **HC** de espécies de aves usando dados de seus sons gravados em áudio. Os autores exploram três abordagens nesse trabalho, a Abordagem de Classificação Plana e as duas abordagens hierárquicas Local e Global. Para as abordagens Plana e Local, um classificador **NB** foi aplicado, enquanto que para a Abordagem Global foi utilizada uma versão modificada do **NB** clássico. Os autores empregam uma taxonomia padrão de espécie de aves como estrutura hierárquica para organizar os dados. Os experimentos realizados foram avaliados usando a medida-F hierárquica, e os resultados mostraram que o uso da Abordagem Global supera tanto a Abordagens Plana quanto a Local. Por fim, os autores recomendam o uso de modelos globais, alegando que estes representam uma maneira viável de melhorar o desempenho da classificação para problemas com um grande número de classes.

### 3.3 Abordagem Local de Classificação Hierárquica

Essa abordagem utiliza de informações locais a respeito das classes da hierarquia durante o processo de treinamento ([SILLA; FREITAS, 2010](#); [FREITAS; CARVALHO, 2007](#)). Assim como é o caso da Abordagem de Classificação Plana, a Abordagem Local pode ser vista como mais uma estratégia reducionista para tratar um problema complexo, mas que ainda considera os relacionamentos hierárquicos entre classes ([BEYGELZIMER et al., 2008](#); [CERRI et al., 2016](#)). Consequentemente, essa abordagem também possibilita que algoritmos de **ML** tradicionais sejam utilizados sem que haja qualquer modificação no modo em que operam.

Outro ponto importante no treinamento de classificadores locais é o número de instâncias positivas e negativas de cada classe fornecidas a um classificador. Conforme descreve [Silla e Freitas \(2010\)](#), há uma variedade de estratégias de definição para as abordagens locais, dentre elas: Exclusiva, Menos Exclusiva, Inclusiva, Menos Inclusiva, Irmãos e Irmãos Exclusivos ([EISNER et al., 2005](#); [FAGNI; SEBASTIANI, 2007](#); [CECI; MALERBA, 2007](#)).

Existem ainda diferentes formas de utilizar informações locais das classes em classificadores hierárquicos. Em [Silla e Freitas \(2010\)](#), são apresentadas três estratégias para a Abordagem Local utilizando **ML**: (1) Classificador Local por Nó (no inglês, *Local Classifier per Node (LCN)*); (2) Classificador Local por Nó Pai (no inglês, *Local Classifier per Parent Node (LCPN)*); e (3) Classificador Local por Nível (no inglês, *Local Classifier per Level (LCL)*). A seguir, essas estratégias são exploradas mais a fundo.

### 3.3.1 Classificador Local por Nó (LCN)

Essa estratégia consiste em treinar um classificador binário para cada nó da hierarquia de classes, exceto o nó raiz, conforme mostra a Figura 6. Dentre as estratégias para a Abordagem Local, a LCN é a mais frequentemente utilizada da literatura (SILLA; FREITAS, 2010). Alguns exemplos de trabalhos que utilizam essa estratégia são apresentados a seguir.

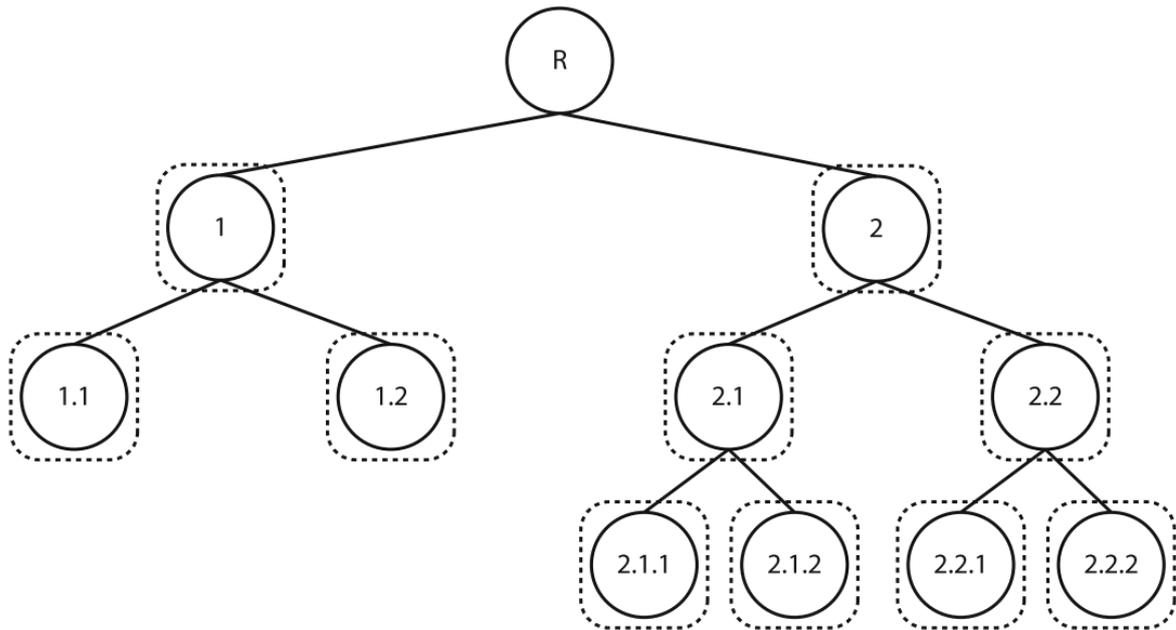


Figura 6 – Classificador binário local por nó da hierarquia (SILLA; FREITAS, 2010).

Em Blockeel et al. (2002), são propostos dois métodos locais baseados na estratégia LCN denominados Clus-SC e Clus-HSC, sendo posteriormente explorados em Vens et al. (2008). O método Clus-SC treina uma DT para cada classe do problema mas ignora os relacionamentos entre classes. Já o método Clus-HSC também induz uma DT para cada classe, com a diferença de que explora os relacionamentos hierárquicos entre elas.

Em Valentini (2009), é utilizado um *Ensemble*<sup>1</sup> de métodos LCN treinados de forma a estimar a probabilidade local de uma instância pertencer a dada classe. Posteriormente, uma etapa de combinação dos resultados é realizada e a estimativa global das probabilidades é obtida. Os autores deram continuidade ao trabalho modificando o método para que o mesmo modularizasse a predição de uma superclasse e suas subclasses (VALENTINI; RE, 2009; VALENTINI, 2011).

Em Cesa-Bianchi et al. (2012), foi investigado a sinergia<sup>2</sup> entre diferentes estratégias locais LCN aplicadas a tarefa de predição de função de genes. Os experimentos se deram através da integração de ferramentas baseadas em métodos de *Kernel* e *Ensemble* sensíveis ao custo (CESA-BIANCHI; VALENTINI, 2009; VALENTINI, 2011). Segundo os autores, quando

<sup>1</sup> Técnica que consiste em utilizar um conjunto de métodos (classificadores, nesse caso) a fim de combinar suas saídas e assim produzir melhores resultados do que quando operam isoladamente.

<sup>2</sup> Melhoria na precisão preditiva de um classificador causada pela utilização simultânea de estratégias de aprendizagem e levando em consideração qualquer medida de avaliação.

a ação combinada de duas ou mais estratégias resulta em melhores taxas de acurácia que a média obtida pelas mesmas estratégias executadas separadamente, detecta-se sinergia.

Em Santos et al. (2018), foram exploradas estratégias de seleção de instâncias positivas e negativas para classificadores locais LCN aplicados a classificação de TEs. Os autores treinaram uma gama de classificadores tradicionais segundo a estratégia LCN, foram eles: C4.5, Random Forest, NB, KNN, MLP e SVM. Para os experimentos foram comparados os resultados desses classificadores em dois conjuntos de dados de TEs, o PGSB e o REPBASE, de forma que utilizaram-se quatro estratégias de seleção: Exclusiva, Menos Exclusiva, Inclusiva e Menos Inclusiva. A ideia da utilização de classificadores com diferentes vieses submetidos a várias estratégias de seleção e em conjuntos de dados distintos, fora a de chegar a um consenso empírico de qual estratégia de seleção é a mais adequada, no geral, a ser utilizada em conjunto com a estratégia LCN. No entanto, é preciso levar em conta o escopo fechado que o estudo apresenta.

### 3.3.2 Classificador Local por Nó Pai (LCPN)

Diferenciando-se um pouco da anterior, a estratégia LCPN emprega um classificador em cada superclasse da hierarquia a fim de predizer suas subclasses, o que conseqüentemente exige um número menor de classificadores ao final (NAKANO et al., 2017). A Figura 7 ilustra essa estratégia, seguida de alguns trabalhos que a utilizam.

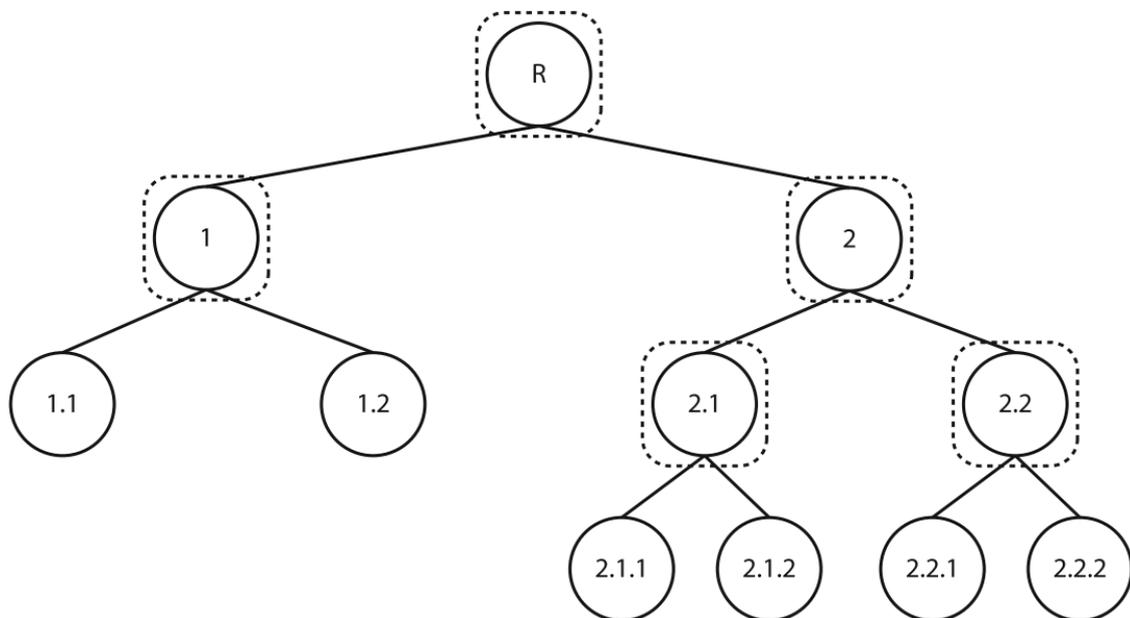


Figura 7 – Classificador multi-classe local por nó pai (SILLA; FREITAS, 2010).

Em Cerri e Carvalho (2010a), é proposto um método local e multirrótulo baseado na estratégia LCPN, denominado HMC-LP (do inglês, *Hierarchical Multi-label Classification with Label-Powerset*), que utiliza uma Rede Neural (no inglês, *Neural Network (NN)*) como classificador base. Esse método é oriundo de uma adaptação do *Label-Powerset* (BOUTELL et

al., 2004; TSOUMAKAS; KATAKIS, 2007), cujo classificador é do tipo multirrótulo porém não hierárquico. O HMC-LP opera de maneira que, em cada nível de uma hierarquia, as classes atribuídas a uma instância são combinadas em uma nova e única classe, transformando o problema hierárquico multirrótulo original em um problema hierárquico simples-rótulo.

Em Cerri e Carvalho (2010b), o HMC-LP é utilizado com uma SVM como seu classificador base. Ademais, os autores propõem um novo método multirrótulo baseado em LCPN denominado HMC-CT (do inglês, *Hierarchical Multi-label Classification with Cross-Training*). O HMC-CT usa uma estratégia de decomposição de classes para transformar o problema hierárquico multirrótulo em um conjunto de problemas hierárquicos simples-rótulo. Ambos os métodos foram aplicadas em dez conjuntos de dados biológicos a fim de prever funções de genes. Posteriormente, este trabalho foi estendido em Cerri et al. (2011), onde foram utilizados diferentes classificadores tradicionais como base, entre eles o C4.5 (QUINLAN, 1993), Redes Bayesianas (FRIEDMAN et al., 1997), KNN (AHA et al., 1991), e o Ripper (COHEN, 1995).

Em Nakano et al. (2017), os autores propõem duas abordagens baseadas em LCPN usando NNs como classificadores base, o nLLCPN (do inglês, *non-Leaf Local Classifier per Parent Node*), e o LCPNB (do inglês, *Local Classifier per Parent Node and Branch*). O método nLLCPN tenta evitar o problema de propagação de erros preditivos presente na estratégia *top-down* modificando a hierarquia de classes para permitir uma classificação do tipo NMLNP. Já o método LCPNB, além de modificar a hierarquia assim como faz o nLLCPN, considera todas as probabilidades de predição em todos os caminhos da hierarquia quando classifica uma nova instância. Portanto, o LCPNB utiliza das predições feitas nos níveis mais profundos a fim de corrigir os erros originados em níveis superiores (mais próximos a raiz).

### 3.3.3 Classificador Local por Nível (LCL)

Na estratégia LCL é empregado um classificador em cada nível da hierarquia, conforme ilustra a Figura 8. Novamente, algoritmos tradicionais de classificação podem ser utilizados para prever as classes de cada nível. Entre as três estratégias aqui apresentadas, esta é a que possui menos estudos reportados (SILLA; FREITAS, 2010; FREITAS; CARVALHO, 2007). Alguns trabalhos que a empregam são apresentados a seguir.

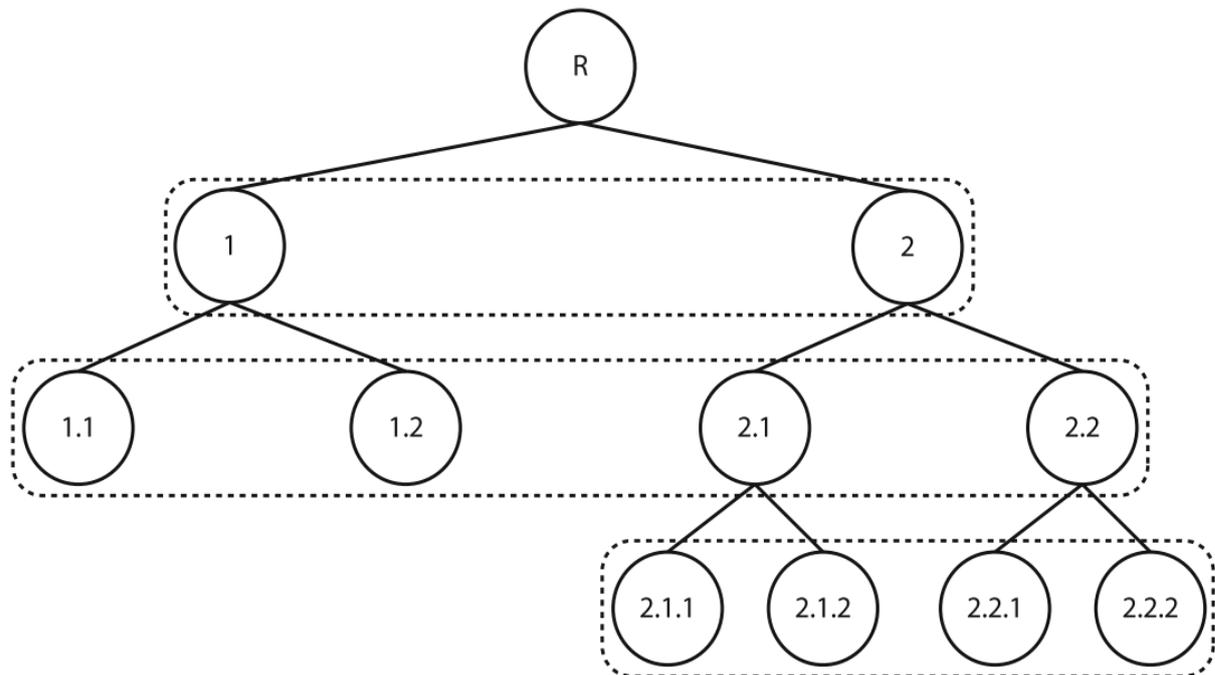


Figura 8 – Classificador multi-classe local por nível (SILLA; FREITAS, 2010).

No trabalho de Cerri et al. (2014), é proposto o HMC-LMLP (do inglês, *Hierarchical Multi-label Classification with Local Multi-Layer Perceptrons*), um método baseado em NNs no qual é associado uma MLP (do inglês, *Multi-Layer Perceptron (MLP)*), a cada nível da hierarquia de classes. Nesse método as instâncias de treinamentos são utilizados como entrada para a MLP associada ao primeiro nível da hierarquia. A partir do segundo nível adiante, cada MLP de um nível é alimentada somente com a saída provida pela MLP do nível anterior.

Em Cerri et al. (2016), o método HMC-LMLP apresentado em Cerri et al. (2014) é estendido e aplicado a dados de predição de função de proteínas. Diferente do que foi proposto anteriormente, em Cerri et al. (2016) o HMC-LMLP usa a saída de uma MLP treinada no nível  $l$  como parte da entrada para a MLP do nível  $l + 1$ . Portanto, as saídas da MLP associada ao nível  $l$  são usadas para aumentar os vetores de atributos que são empregados para treinar a MLP do nível  $l + 1$ . A ideia dessa modificação é garantir que as dependências entre classes sejam levadas em consideração durante o processo de treinamento. Assim, espera-se que as MLPs aprendem ou “descubram” essas dependências automaticamente. Conforme apontam os autores, as modificações trouxeram melhorias nas predições do método.

Em Nakano et al. (2017), são apresentados dois métodos baseados na estratégia LCL que têm NNs como classificadores base. Nesses, foram aplicados dois procedimentos de correção de inconsistências: o SWV (do inglês, *Sum of Weighted Votes*) (PAES et al., 2012) e o *Simple Prune* (CERRI et al., 2016). O SWV realiza uma soma das probabilidades de predições a fim de decidir qual classificação deve ser substituída. Já o *Simple Prune* apenas retira da classificação final todas aquelas classes preditas as quais não foram também preditas suas superclasses.

Por fim, um apanhado geral dos métodos vistos nessa seção é apresentado na Tabela 2, com distinção de suas respectivas estratégias utilizadas.

Tabela 2 – Métodos locais de HC e suas respectivas estratégias.

Método/Descrição	Artigo da Literatura	Estratégia
Clus-SC	<a href="#">Blockeel et al. (2002)</a>	LCN
Clus-HSC	<a href="#">Blockeel et al. (2002)</a>	LCN
Ensemble de Classificadores	<a href="#">Valentini (2009)</a>	LCN
Métodos <i>Kernel</i> e Ensemble	<a href="#">Cesa-Bianchi et al. (2012)</a>	LCN
HMC-LP	<a href="#">Cerri e Carvalho (2010a)</a>	LCPN
HMC-CT	<a href="#">Cerri e Carvalho (2010b)</a>	LCPN
nLLCPN	<a href="#">Nakano et al. (2017)</a>	LCPN
LCPNB	<a href="#">Nakano et al. (2017)</a>	LCPN
HMC-LMLP	<a href="#">Cerri et al. (2014)</a> , <a href="#">Cerri et al. (2016)</a>	LCL
MLP + SWV	<a href="#">Nakano et al. (2017)</a>	LCL
MLP + <i>Simple Prune</i>	<a href="#">Nakano et al. (2017)</a>	LCL

### 3.4 Abordagem Global de Classificação Hierárquica

Em contraste com as abordagens supracitadas, na Abordagem Global um único classificador é treinado utilizando todo o conjunto de instâncias de treinamento e considerando todas as classes da hierarquia de uma só vez ([SILLA; FREITAS, 2010](#); [FREITAS; CARVALHO, 2007](#)). Outra diferença ocorre na fase de teste ou inferência, que acontece em um único passo utilizando o modelo induzido, e não de maneira *top-down* como nas abordagens locais ([VENS et al., 2008](#)).

Em [Silla e Freitas \(2010\)](#), é enfatizada a falta de consenso sobre as definições de Abordagem Local e Global. Conforme apontado pelos autores, embora pareça não existir um padrão entre as abordagens globais, em geral, classificadores ditos globais têm mostrado duas características em comum: (1) Consideram toda a hierarquia de uma só vez e (2) Não possuem a modularidade de treinamento apresentada pelas abordagens locais. Portanto, é possível chegar a algumas conclusões. Como os modelos globais são treinados com todas as instâncias da hierarquia, as políticas de definição de instâncias positivas e negativas mencionadas na Seção 3.3 não são aplicáveis. A partir da segunda característica elencada, infere-se que para uma abordagem ser considerada global, esta deve utilizar apenas um classificador em vez de um conjunto de classificadores, diferente do que acontece nas abordagens locais.

No restante dessa seção serão apresentados alguns exemplos de classificadores globais encontrados na literatura, aos quais se dará maior atenção aqueles que geram modelos interpretáveis. A Tabela 3 apresenta um apanhado geral dos métodos citados nessa seção.

Tabela 3 – Métodos globais de HC.

Método/Descrição	Artigo da Literatura	Interpretável
Agrupamento de classes	Rocchio (1971), Labrou e Finin (1999)	Não
Coverage	Wang et al. (2001)	Sim
Clus-HMC	Blockeel et al. (2002)	Sim
HC4.5	Clare e King (2003)	Sim
MHC-AIS	Alves et al. (2008)	Sim
hAnt-Miner	Otero et al. (2009)	Sim
Naive Bayes adaptado	Jr e Freitas (2009)	Não
NN competitiva	Borges e Nievola (2012)	Não
HMC-GA	Cerri et al. (2012), Cerri (2014)	Sim
NHMC	Stojanova et al. (2013)	Sim
OPP	Sun et al. (2017)	Não

Ao que tudo indica, a primeira Abordagem Global foi proposta por Rocchio (1971), que utiliza a ideia de agrupamento de classes a fim de assinalar novas instâncias a classes mais próximas. Isso é feito através do cálculo da distância entre uma nova instância e cada classe, sendo que vários tipos de distância podem ser considerados. Um exemplo dessa abordagem é também utilizada em Labrou e Finin (1999), em que foi proposto um sistema que classifica páginas *web* em um subconjunto de categorias hierárquicas do site Yahoo!. Durante a fase de teste, cada novo documento tem sua similaridade computada com respeito a cada tópico presente em texto, sendo que a classificação final é dada por algum limiar.

No trabalho de Wang et al. (2001), um algoritmo de mineração de regras de associação foi altamente modificado para lidar com a categorização hierárquica de documentos. Sua modificação principal foi fazer com que o algoritmo trabalhasse com um conjunto de etiquetas de uma só vez, em outras palavras, transformando-o de simples rótulo para multirrótulo.

Proposto em Blockeel et al. (2002) e posteriormente explorado em Vens et al. (2008), o Clus-HMC é um método baseado em *Predictive Cluster Trees* capaz de gerar uma única DT considerando toda a hierarquia de classes de um problema. No Clus-HMC as DTs são vistas como uma hierarquia de grupos e que predizem um conjunto de classes. O nó raiz contém todos as instâncias de treinamento, que são recursivamente divididos em grupos menores conforme a hierarquia é percorrida em direção as folhas. A classificação de uma instância ocorre segundo uma métrica baseada em distância, que calcula o quão similar a instância é a alguma árvore. Ao final, os grupos formados são combinados a fim de que seja induzida uma única DT. A saída do Clus-HMC é um vetor binário correspondente as possíveis classes de uma instância. Esse método pode ser aplicado tanto para classificação quanto para agrupamento.

Em Clare e King (2003), é proposto o HC4.5, uma versão modificada do algoritmo C4.5 (QUINLAN, 1993) para lidar com uma hierarquia de classes. Ainda que poucos detalhes de sua implementação sejam especificados, sabe-se que o cálculo da entropia foi modificado de

forma a atribuir pesos aos níveis da hierarquia. No HC4.5, os níveis mais profundos da hierarquia possuem mais peso pois esses fornecem informação mais específica e útil acerca dos dados. Segundo [Clare e King \(2003\)](#), essas informações têm maior relevância para as predições, apesar de alguns autores refutarem tal pressuposto dado a abrangência de domínios e particularidades dos problemas hierárquicos, o que pode invalidar essa premissa ([LORD et al., 2003](#); [CERRI et al., 2016](#)). Aparentemente, os outros aspectos do algoritmo C4.5 permanecem idênticos.

No trabalho de [Alves et al. \(2008\)](#), é apresentado o método MHC-AIS (do inglês, *Multi-label Hierarchical Classification with an Artificial Immune System*), baseado em um Sistema Imune Artificial e que gera regras de HC multirrótulo. Esse método utiliza uma abordagem de cobertura sequencial de instâncias ou, como também é conhecida, uma abordagem iterativa ([PI-MENTA et al., 2014](#); [ABADEH; HABIBI, 2007](#)), que é executada até que todas ou quase todas as instâncias de treinamento (parâmetro definido pelo usuário) sejam cobertas pelas regras encontradas. As regras geradas são do tipo “Se-Então”, cujo os antecedentes são representados por vetores contendo valores de atributos utilizados como testes e os consequentes como conjuntos de classes preditas. Posteriormente, os autores estenderam esse trabalho em [Alves et al. \(2010\)](#), propondo novos mecanismos de otimização a fim de aumentar a acurácia do método.

Em [Otero et al. \(2009\)](#), os autores propõem um método global baseado em Otimização por Colônias de Formigas ([DORIGO et al., 1991](#)), denominado hAnt-Miner. Esse método realiza um processo de busca construtivo de regras de classificação hierárquica do tipo “Se-Então”. Para a construção de uma regra são utilizadas duas colônias de formigas que trabalham de forma cooperativa, sendo uma colônia *a* responsável por construir o antecedente e uma colônia *c* que é incumbida da construção do consequente. A cada iteração, uma regra é construída por meio do pareamento de uma formiga oriunda de *a* com uma formiga oriunda de *c*. O hAnt-Miner também utiliza um processo de cobertura sequencial de instâncias durante a criação das regras, para que estas cubram todas, ou quase todas as instâncias de treinamento. Enquanto o número de instâncias não cobertas for maior que dado parâmetro (definido pelo usuário), as novas regras construídas vão sendo adicionadas a uma lista final de regras (inicializada vazia).

No trabalho de [Jr e Freitas \(2009\)](#), um classificador plano NB é adaptado para operar de acordo com a Abordagem Global, sendo então aplicado na tarefa de predição de funções de proteínas. Os autores adaptaram o cálculo das probabilidades a priori e da verossimilhança do classificador Bayesiano tradicional, para que estas levassem em consideração uma estrutura hierárquica, o que permitiu ao método predizer classes em qualquer nível da hierarquia.

Em [Borges e Nievola \(2012\)](#), foi proposto uma NN competitiva composta por uma camada de entrada e uma de saída. O treinamento da rede é realizado via cálculo das distâncias entre os nós da hierarquia e as instâncias de treino. Dessa forma, os neurônios que apresentam as menores distâncias são considerados vencedores, influenciando suas vizinhanças. Os pesos da NN são então ajustados de acordo com as classes associadas aos neurônios vencedores.

No trabalho de Cerri et al. (2012), é proposto o HMC-GA (do inglês, *Hierarchical Multi-label Classification with a Genetic Algorithm*), um GA que induz regras de HC multirrótulo. Esse GA utiliza um processo evolutivo nos antecedentes das regras de classificação com o intuito de melhorar a cobertura de instâncias, utilizando para isso uma função *fitness* que beneficia regras que cobrem um número maior de instâncias. Esse método usa um processo determinístico que considera as instâncias de treinamento cobertas por uma regra para gerar seu consequente. Cada consequente é representado por um vetor de números reais entre [0-1] que pode ser interpretado como as probabilidades de atribuição das classes a dada instância. Esse trabalho foi estendido em Cerri (2014), onde outras funções *fitness* foram apresentadas juntamente com outros mecanismos de otimização visando aprimorar as predições das regras geradas.

Em Stojanova et al. (2013), é proposto uma Abordagem Global que utiliza redes PPI (do inglês, *Protein-Protein Interaction*) e considera auto-correlação entre dados de funções de proteínas no contexto da HC multirrótulo. O método chamado NHMC (do inglês, *Network Hierarchical Multi-label Classification*), leva em consideração as relações estatísticas entre uma mesma variável em instâncias diferentes, porém relacionadas. Como saída, o NHMC gera uma forma generalizada de DT semelhante ao Clus-HMC (VENS et al., 2008).

Em Sun et al. (2017), o problema de HC é tratado como um problema de seleção de um caminho ótimo na hierarquia de classes, que se inicia na raiz e segue até um nó intermediário ou folha. Os autores utilizam a técnica de otimização dos mínimos quadrados, que busca o melhor ajuste de parâmetros para um conjunto de dados, minimizando a soma dos quadrados das diferenças entre o valor estimado e os dados observados, criando assim um novo método denominado OPP (do inglês, *Optimal Path Prediction*). Esse método realiza a predição dos caminhos hierárquicos na forma de grafos formados pelos nós da hierarquia. A saída final do OPP é um *caminho ótimo* constituído pela união dos grafos previamente explorados.

### 3.5 Considerações Finais do Capítulo

Este capítulo apresentou alguns conceitos fundamentais acerca dos problemas de Classificação e Classificação Hierárquica, como suas respectivas definições formais, as estruturas de dados empregadas e as principais abordagens e estratégias utilizadas para tratar problemas hierárquicos. Naturalmente, todas as abordagens aqui apresentadas possuem vantagens e desvantagens, sendo algumas mais adequadas que outras dependendo do contexto ao qual estão inseridas.

A Abordagem de Classificação Plana, apesar de ser a mais simplória e de fácil implementação das três apresentadas, obtém resultados ruins para o contexto de HC uma vez que ignora as relações hierárquicas. Já a Abordagem Local considera o relacionamento das classes numa hierarquia e explora informações locais a respeito dessas, além de compartilhar da simplicidade de implementação e modularidade de treinamento da Abordagem Plana, permitindo o uso de classificadores tradicionais sem quaisquer adaptações. Em contraponto, possui a desvantagem

de que os erros gerados nos níveis superiores são propagados aos níveis mais profundos da hierarquia. A Abordagem Global por outro lado evita a propagação de erros ao longo da hierarquia já que a considera por inteiro. Além disso, tipicamente gera modelos finais menores (mais compreensíveis) e as dependências associativas entre classes podem ser tratadas de maneira mais simples e direta. Uma desvantagem é que abordagens globais não fazem uso de informações locais, e essas muitas vezes podem ser úteis a fim de explorar peculiaridades de dado nível ou classe da hierarquia, já que a informação utilizada para classificar uma instância nos níveis superiores pode ser diferente da usada em níveis mais profundos. A Abordagem Global acaba sendo mais complexa, no sentido de que geralmente exige um maior esforço de implementação pois deve-se adaptar algoritmos convencionais para tratar da hierarquia de classes inteira, ou ainda, utilizar de técnicas específicas como é o caso da *Computação Evolutiva*.

Pelo fato da Bioinformática, domínio no qual este trabalho também está incluído, possuir a demanda por modelos computacionais mais interpretáveis e assim ser possível extrair conhecimento útil, optamos por adotar a Abordagem Global na construção dos classificadores e explorar sua capacidade de gerar modelos menores e conseqüentemente, mais compreensíveis.

# Capítulo 4

## ALGORITMOS GENÉTICOS

---

---

### 4.1 Considerações Iniciais

Inspirados pelas ideias da Teoria da Evolução, os Algoritmos Genéticos (no inglês, *Genetic Algorithms (GAs)*) propostos por [Holland \(1992\)](#), utilizam de uma população de indivíduos onde os mais aptos são selecionados e submetidos a um processo de cruzamento, sendo que alguns destes sofrem mutações. Desta forma, espera-se que as futuras gerações de indivíduos sejam mais adaptadas ao ambiente e conseqüentemente, tenham mais chances de sobreviver. Segundo [Castro \(2006\)](#), um GA tradicional tem as seguintes particularidades:

- **Codificação de soluções:** um GA não trabalha diretamente com o problema a ser otimizado, e sim com uma representação codificada das soluções candidatas (indivíduos) que representam o espaço de busca;
- **População de indivíduos:** utiliza-se uma população de indivíduos que se reproduzem entre si e transmitem suas características às futuras gerações;
- **Seleção natural:** uma função de aptidão (*fitness*) avalia a qualidade ou adaptabilidade de um indivíduo ao ambiente (problema). A competição/seleção numa população é realizada por meio da comparação do valor de *fitness* de cada indivíduo, logo, aqueles que apresentam valores mais elevados possuem vantagem seletiva;
- **Varição genética:** os indivíduos de uma população podem ter suas estruturas alteradas de acordo com uma taxa de mutação  $mr$ , que é geralmente baixa e permite o surgimento de novas características. Esse processo traz variabilidade genética a população e possibilita a exploração de novas regiões no espaço de busca.

Um GA é iterativo, estocástico e genérico o suficiente para ser aplicado a uma gama de problemas. Sua execução se inicia com a geração de uma população  $P$  de  $n$  indivíduos. Na iteração  $t$ , a população  $P$  é avaliada por meio de uma função que atribui um valor de *fitness* a cada indivíduo visando determinar o quão adaptado este é ao ambiente (problema). Na iteração

$t + 1$ , uma nova população é gerada a partir da seleção e cruzamento dos indivíduos mais aptos. Ademais, variações genéticas podem ser alcançadas por meio de mutações aplicadas a alguns membros da população. Tanto o cruzamento quanto a mutação ocorrem de acordo com probabilidades  $cr$  e  $mr$ , respectivamente. O processo evolutivo se encerra quando algum critério de parada é satisfeito, geralmente um número máximo de gerações ou o alcance de algum objetivo especificado são utilizados. Adicionalmente, é comum ser utilizado um mecanismo de Elitismo, onde uma parcela dos melhores indivíduos da população atual são copiados ou movidos diretamente para a nova população (BALUJA; CARUANA, 1995). O Elitismo possibilita aumentar rapidamente o desempenho do GA e conseqüentemente acelera sua convergência, pois previne a perda das melhores soluções já encontradas (GOLDBERG; SASTRY, 2001). Em linhas gerais, um GA tradicional segue o procedimento apresentado no Algoritmo 1.

---

**Algoritmo 1:** Algoritmo Genético tradicional (EIBEN et al., 2003).

---

**Entrada:** tamanho da população  $n$ , taxa de cruzamento  $cr$ , taxa de mutação  $mr$ .  
**Saída:** população  $P$

```

1 início
2    $P \leftarrow$  inicialização( $n$ )
3    $f \leftarrow$  avaliação( $P$ )
4    $t \leftarrow 0$ 
5   enquanto critério de parada não for satisfeito faça
6      $novaP \leftarrow \emptyset$ 
7      $pais \leftarrow$  seleção( $P, f$ )
8      $filhos \leftarrow$  cruzamento( $pais, cr$ )
9      $novaP \leftarrow$  mutação( $filhos, mr$ )
10     $f \leftarrow$  avaliação( $novaP$ )
11     $t \leftarrow t + 1$ 

```

---

Contudo, a literatura apresenta uma variedade de trabalhos que utilizam variações desse procedimento, como outros operadores de seleção, cruzamento, mutação, diferentes formas de representação e codificação de indivíduos, além de também haver alterações na ordem dos operadores genéticos. Essas questões são melhor abordadas nas seções posteriores.

## 4.2 Representação das Soluções

No que tange a indução de regras utilizando GAs, a literatura de *Data Mining* frequentemente apresenta duas abordagens para representação dos indivíduos (SRINIVASAN; RAMAKRISHNAN, 2011; OTERO et al., 2013): Pittsburgh (SMITH, 1980) e Michigan (HOLLAND, 1986). Na abordagem de Pittsburgh cada indivíduo corresponde a um conjunto de regras enquanto que na abordagem de Michigan cada indivíduo representa uma única regra. Logo, uma população de indivíduos é composta por vários conjuntos de regras na abordagem de Pittsburgh, à medida que na abordagem de Michigan uma população é formada por uma só lista (em vez de um

conjunto) de regras. Essas abordagens possuem vantagens e desvantagens quando comparadas e sua principal diferença está em como as regras são avaliadas.

Em Pittsburgh um conjunto de regras é avaliado em vez de uma só regra. Avaliar um conjunto inteiro em vez de avaliar regras separadamente pode levar a conjuntos mais concisos e coerentes, porém individualmente essas regras podem não capturar aspectos importantes do problema. Em contraponto, avaliar a qualidade de regras individualmente em vez de um conjunto de regras ignora a interação entre essas, tornando-as uma lista em vez de um conjunto de regras que se complementa. Além disso, nem sempre a lista das melhores regras é necessariamente a melhor lista (ou conjunto) de regras. A abordagem de Pittsburgh, por representar um conjunto de regras em um único indivíduo, também pode sofrer com problemas de cromossomos com tamanho excessivo, o que é menos provável de ocorrer na Michigan. Em linhas gerais, uma abordagem tende a explorar melhor (Pittsburgh) e a outra a explorar (Michigan).

### 4.3 Codificação de Indivíduos

Grande parte dos termos empregados por um GA são provenientes da Biologia, mais precisamente da Genética. Conforme aponta Holland (1992), os indivíduos de um GA são codificados em **genótipos**, que por sua vez são representados por estruturas de dados denominados **cromossomos**. Cada posição/local de um cromossomo é conhecido como **locus**, e dentro desse, a informação armazenada é denominada **gene**. Ademais, os possíveis valores que um gene pode assumir são chamados de **alelos**. Esses genótipos podem ser codificados de diversas maneiras, de acordo com o sistema binário, inteiro ou ponto flutuante. Esses conceitos são exemplificados na Figura 9, que representa um cromossomo na codificação binária.

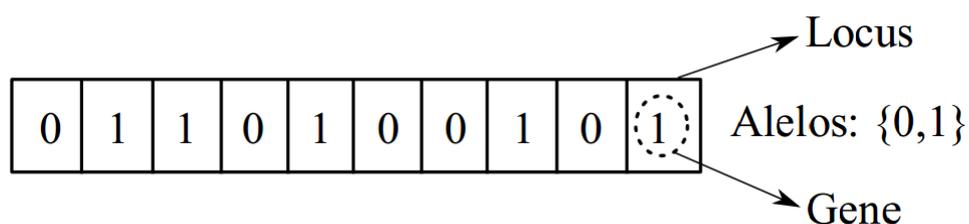


Figura 9 – Cromossomo na codificação binária. Adaptado de Cerri (2014).

O termo genótipo refere-se à constituição genética de um indivíduo. Já o fruto da interação do genótipo com o ambiente é chamado de **fenótipo**. Esse é empregado para denotar as características visíveis do indivíduo, sendo elas morfológicas, fisiológicas ou comportamentais. Como exemplo, pode-se citar a cor dos olhos de uma pessoa, a textura do cabelo e sua altura. Em outras palavras, diz-se que o significado do genótipo é seu fenótipo. No exemplo da Figura 10, onde a população é composta por espécies de cachorros com características diferentes, os genótipos desses indivíduos são justamente seus cromossomos codificados como vetores binários, enquanto que as suas características observáveis são os seus fenótipos (GOLDBERG, 1989).

## 4.4 Processo Evolutivo

A fim de exemplificar de maneira simples o funcionamento básico de um GA que opera sob a codificação binária, as Figuras 10, 11, 12 e 13 podem ser levadas em consideração. Iniciando-se pela geração dos indivíduos de uma população, a Figura 10 exemplifica uma representação binária de quatro diferentes indivíduos que constituem uma população do GA.

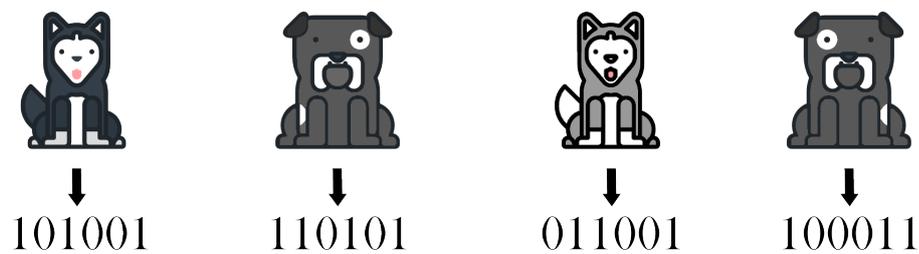


Figura 10 – Exemplo prático de codificação binária.

Na Figura 11, é ilustrado o processo de avaliação da população gerada, o qual atribui valores de *fitness* aos indivíduos. Nessa mesma figura, é ilustrado o processo de seleção de dois indivíduos da população (indivíduos destacados em vermelho), cujo os valores de *fitness* são os maiores da população. Na Figura 12 é visto o processo de cruzamento, em que esses dois indivíduos selecionados no processo anterior têm seus genes cruzados a fim de gerar dois novos indivíduos filhos, que irão posteriormente compor a nova geração da população. Por fim, na Figura 13 um processo de mutação conhecido como *Bit-flip* é exemplificado, selecionando um dos indivíduos filhos gerados e trocando um de seus genes de 0 para 1.

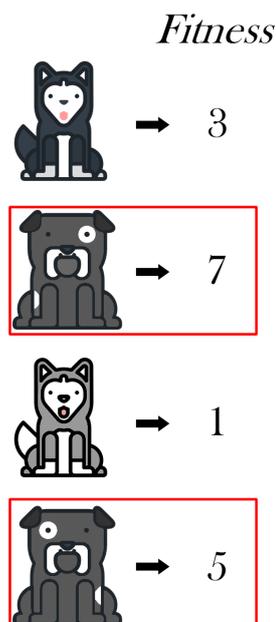


Figura 11 – Cálculo do fitness e seleção de indivíduos.

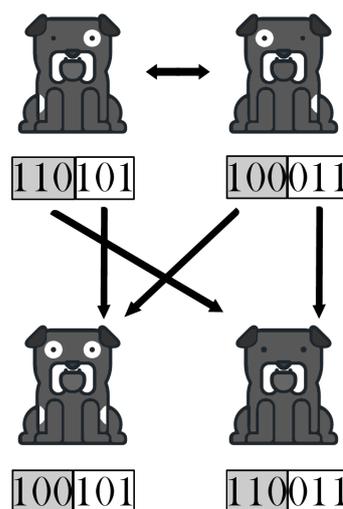


Figura 12 – Cruzamento entre pais e surgimento dos filhos (nova geração).

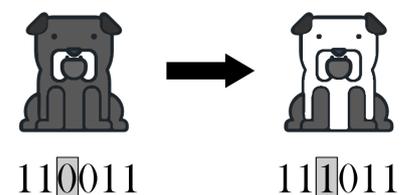


Figura 13 – Mutação da nova geração.

## 4.5 Operadores de Seleção

O operador de seleção de um **GA** tem como papel determinar quais indivíduos irão compor a próxima geração de uma população diretamente (Elitismo) ou quais irão sofrer operações de cruzamento e, possivelmente, mutação (**GOLDBERG; HOLLAND, 1988**). Vários operadores de seleção são encontrados na literatura, alguns exemplos populares são a Seleção por Roleta (**HOLLAND, 1992; GOLBERG, 1989**), Seleção por Amostragem Universal Estocástica (**BAKER, 1987**) e a Seleção por Torneio (**WETZEL, 1983**).

A Seleção por Roleta é um dos métodos de seleção mais simples e utilizados da literatura, e que consiste em selecionar indivíduos por meio de sorteio utilizando a lógica de uma roleta (**GOLDBERG; HOLLAND, 1988**). Cada indivíduo da população é representado na roleta por uma fatia proporcional ao seu valor de *fitness* e, conseqüentemente, aqueles com maior *fitness* possuem vantagem seletiva. Uma ilustração desse operador é visto na Figura 14.

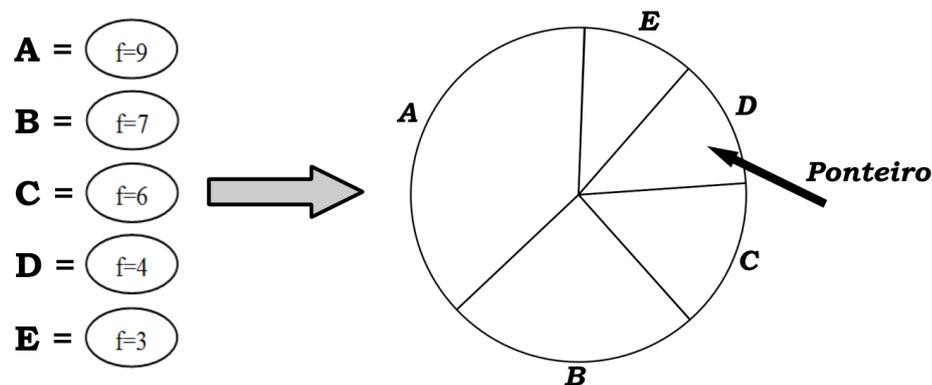


Figura 14 – Seleção por Roleta.

Por ser um método de seleção proporcional ao *fitness* dos indivíduos, a Seleção por Roleta pode acarretar numa convergência prematura do algoritmo. Isso ocorre principalmente quando existem os chamados “Super Indivíduos”, isto é, indivíduos com valores de *fitness* muito mais elevados do que o restante da população. Essa situação é mais agravada quando a população é discrepante, possuindo também vários indivíduos com valores de *fitness* muito baixos. Nesse caso, como os melhores indivíduos possuem maior probabilidade de serem selecionados, a população tenderá a convergir rapidamente para apenas descendentes desses indivíduos, produzindo soluções sub-ótimas e com pouca diversidade.

Semelhante a Seleção por Roleta, a Seleção por Amostragem Universal Estocástica não utiliza apenas um ponteiro na roleta, e sim  $n$  ponteiros igualmente espaçados, onde  $n$  corresponde ao número de indivíduos a serem selecionados (**BAKER, 1987**). Outra peculiaridade do operador é que ao invés de realizar as seleções de indivíduos várias vezes (um a um), esse método seleciona os  $n$  indivíduos de uma só vez. A Figura 15 exemplifica a ideia do método.

A Seleção por Amostragem Universal Estocástica suaviza o problema de convergência prematura e a falta de diversidade na população apresentada pela Seleção por Roleta. Ainda

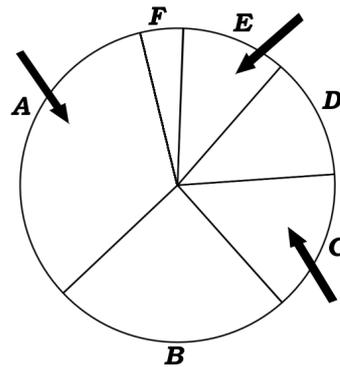


Figura 15 – Seleção por Amostragem Universal Estocástica.

assim, sua convergência é prejudicada quando existem grupos polarizados de indivíduos na população, isto é, grupos de indivíduos com altos valores de *fitness* e grupos com valores baixos. Além disso, para populações com muitos indivíduos, esse método pode não ser adequado já que irá demandar uma complexidade computacional mais elevada.

Já na Seleção por Torneio,  $n$  indivíduos da população corrente são escolhidos aleatoriamente para compor uma subpopulação, de acordo com uma probabilidade  $sr$ , e então competem entre si a fim de participar de operações de cruzamento e possivelmente, mutação (WETZEL, 1983). Os indivíduos ditos vencedores do torneio são selecionados de acordo com seu valor de *fitness*. Este processo se repete até que uma nova população intermediária seja totalmente ocupada ou algum outro critério seja satisfeito. Tal processo é exemplificado na Figura 16.

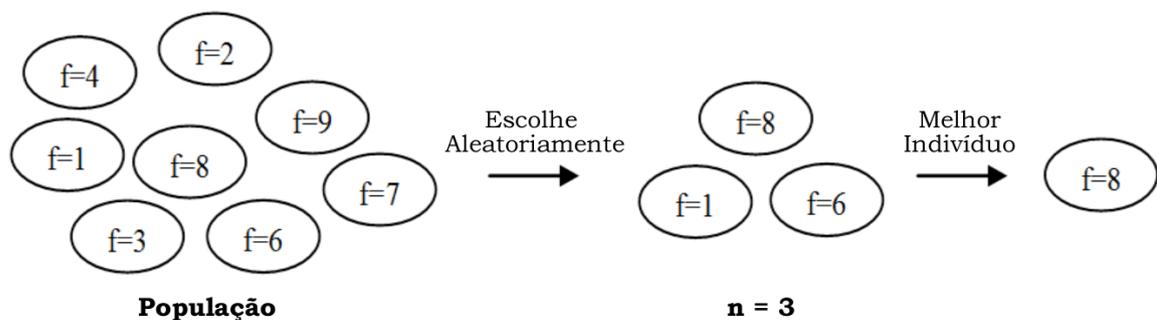


Figura 16 – Seleção por Torneio. Adaptado de Razali et al. (2011).

Graças ao parâmetro  $n$  que determina o tamanho do torneio, é possível controlar explicitamente a pressão seletiva durante o processo, sendo que quanto maior o número de indivíduos no torneio, maior a pressão seletiva. Em outras palavras, maior é o grau de favorecimento dos melhores indivíduos na competição, o que influencia diretamente a convergência do algoritmo.

A Seleção por Torneio ameniza o problema da convergência prematura e baixa diversidade apresentado pela Seleção por Roleta, pois todos os indivíduos possuem a mesma chance de serem selecionados para participar do torneio. Porém, dependendo da pressão seletiva definida, esse operador pode apresentar problemas. Uma pressão muito alta, por exemplo, irá gerar convergência prematura e conseqüentemente, uma diversidade pobre. Por outro lado, uma pressão

seletiva baixa irá gerar uma convergência lenta. Formas de prevenir tais problemas são: (1) Controlar o número de oportunidades de reprodução de cada indivíduo; (2) Aumentar o tamanho da população do torneio (parâmetro  $n$ ); e (3) Aumento da taxa de mutação.

## 4.6 Operadores de Cruzamento

Operadores de cruzamento são utilizados a fim de combinar genes dos cromossomos selecionados (pais) e assim gerar novos cromossomos (filhos) (GOLBERG, 1989). Logo, espera-se que esses novos indivíduos possuam características que reflitam numa melhor adaptação ao ambiente e, conseqüentemente, tenham mais chances de sobreviver. Em outras palavras, espera-se que tenham mais êxito na resolução do problema.

Diversos operadores de cruzamento são encontrados na literatura, sendo estes aplicáveis a diferentes tipos de codificação. Nessa seção, os operadores são exemplificados utilizando a codificação binária, entretanto, estes também podem ser usados com as codificações de ponto flutuante ou inteira. Dois operadores frequentemente adotados são o *Cruzamento de  $k$  Pontos* e o *Cruzamento Uniforme* (GOLBERG, 1989; SPEARS, 1997). Geralmente os cromossomos de um GA são cruzados dois a dois, e no caso do cruzamento de  $k$  pontos, são selecionadas aleatoriamente  $k$  regiões dos cromossomos (as mesmas regiões para ambos), e então troca-se os valores dos genes entre eles. As variações mais utilizadas são a de um e dois pontos, conforme mostra a Figura 17 (a) e a Figura 17 (b), respectivamente.

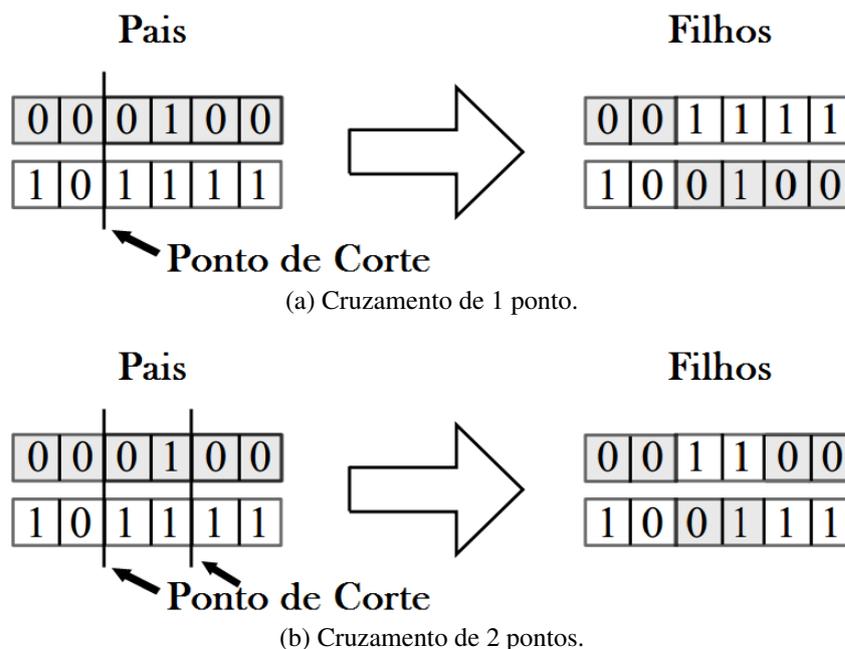


Figura 17 – Método do Cruzamento em  $K$ -Pontos. Adaptado de Cerri (2014).

Já no cruzamento uniforme cada gene de um cromossomo pai é selecionado de forma independente (do outro cromossomo pai) de acordo com uma probabilidade  $cr$  definida pelo

usuário. Diferente do que ocorre no cruzamento de  $k$ -pontos, o cruzamento uniforme combina genes individuais e não segmentos de genes, eliminando o viés que faz com que genes próximos no cromossomo sejam selecionados para passar suas características adiante, o que pode ser prejudicial ao processo evolutivo (MITCHELL, 1998). A Figura 18 ilustra o procedimento desse cruzamento, onde normalmente o valor atribuído a  $cr$  é 0.5 (GOLBERG, 1989).

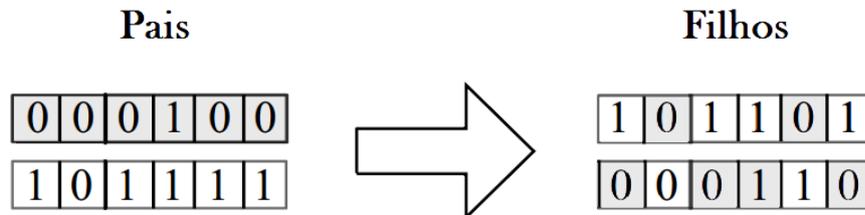


Figura 18 – Método do Cruzamento Uniforme.

## 4.7 Operadores de Mutação

O operador de mutação de um GA modifica os alelos de acordo com uma probabilidade de mutação, o que cria novos cromossomos. Além de evitar que se estabeleça uma população incapaz de evoluir, essa operação é realizada com o intuito de adicionar diversidade aos indivíduos, sendo essa uma importante característica para a sobrevivência de uma população (GOLDBERG; HOLLAND, 1988). Além disso, no contexto de um processo de otimização, a mutação permite a exploração de novas regiões no espaço de busca que poderiam não ser alcançadas caso apenas os operadores de seleção e cruzamento fossem aplicados (GOLDBERG, 1989).

Normalmente, o que ocorre no processo evolutivo de um GA é o operador de mutação ser aplicado aos genes de acordo com uma probabilidade de mutação  $pm$  geralmente baixa (SASTRY et al., 2005). É importante que  $pm$  tenha um valor baixo pois valores altos farão com que boas soluções sejam perdidas, estagnando o processo de otimização e direcionando a busca para regiões esparsas e sem direção a potenciais ótimos globais e até mesmo locais.

Existem inúmeros operadores de mutação aplicáveis a codificações binárias, reais e inteiras. Um exemplo é a mutação de "Intercâmbio", onde dois genes de um cromossomo são aleatoriamente selecionados e seus valores correspondentes são trocadas. Outro operador comum e independente da codificação é a mutação de "Reversão", onde um segmento de tamanho  $n$  é selecionado no cromossomo e então os genes correspondentes são invertidos. Um dos operadores de mutação mais utilizados junto a codificação binária é denominado *bit-flip*. Este operador simplesmente troca o *bit* de um cromossomo por outro, portanto, dado um gene selecionado segundo uma probabilidade  $mr$ , este tem seu alelo alterado de 0 para 1 e vice-versa (CASTRO, 2006; MITCHELL, 1998). A Figura 19 ilustra a aplicação desse operador de mutação.

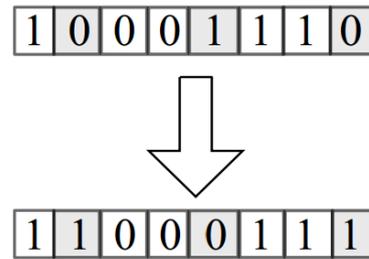


Figura 19 – Método de mutação *Bit-flip*. Adaptado de Castro (2006).

## 4.8 Considerações Finais do Capítulo

Neste capítulo foram apresentados os conceitos básicos acerca de Algoritmos Genéticos, técnicas evolutivas inspiradas na Teoria da Evolução dos seres vivos. Foram apresentadas as terminologias utilizadas por um GA, assim como o seu funcionamento padrão exemplificado no Algoritmo 1, seguido dos dois tipos populares de representação de soluções, as Abordagens de Pittsburgh e Michigan. Além disso, usando como exemplo uma codificação binária, o funcionamento dos operadores genéticos de um GA foram apresentados, bem como os exemplos de métodos populares para cada um desses operadores.

Dentre os operadores de seleção, foram apresentadas a Seleção por Roleta, Seleção por Amostragem Universal Estocástica e a Seleção por Torneio. Quanto aos modos de cruzamento, duas variações foram mostradas, o Cruzamento Uniforme e o Cruzamento de  $k$ -Pontos. Além disso, operadores de mutação como o de Reversão, Intercâmbio e *Bit-flip* foram introduzidos.

Como o desenvolvimento dos GAs dessa pesquisa são inspirados no método HMC-GA proposto em Cerri (2014), optamos por também utilizar da mesma forma de representação (Michigan) e codificação (ponto flutuante) dos indivíduos. O uso da Abordagem de Michigan para representar as soluções tende a facilitar a exploração de soluções e, dado as particularidades do problema hierárquico, consideramos está uma hipótese promissora. Quanto a implementação dos operadores genéticos, optamos pela utilização da Seleção por Torneio dadas as vantagens em relação as demais, juntamente com o método de Cruzamento Uniforme e a Mutação *Bit-flip* aplicada aos genes responsáveis pela ativação/desativação de testes nos indivíduos (regras).

# Capítulo 5

## OTIMIZAÇÃO MULTIOBJETIVO

---

### 5.1 Considerações Iniciais

É do senso comum que diversos problemas do cotidiano envolvem múltiplos fatores, como é o caso da fabricação de um produto (custo vs qualidade), a compra de um bem (quão desejável vs valor vs durabilidade) ou a escolha de uma rota de viagem (distância vs tempo vs combustível vs pontos de pedágio) (DEB et al., 2016). Ainda que alguns desses possam ser simplificados a um “único objetivo”, compactar as características de um problema pode acarretar em perda de informações cruciais e, conseqüentemente, levar a resultados não satisfatórios. A Otimização Multiobjetivo é uma abordagem que permite lidar com vários aspectos de um problema de forma a otimizá-los e sem a necessidade de omitir informações acerca deste, visando encontrar uma variedade de soluções que satisfaçam os objetivos especificados (COELLO, 2006).

Um processo de otimização pode ser definido como sendo a busca por um valor máximo ou mínimo de uma função, a qual depende de um número finito de variáveis e que deve respeitar determinadas restrições para que seja considerada factível. Conforme aponta Deb (2001), no contexto de um Problema de Otimização Multiobjetivo (no inglês, *Multi-Objective Optimization Problem (MOOP)*), uma solução é factível se respeitar as seguintes restrições:

$$\left. \begin{array}{ll} \text{maximizar/minimizar} & f_m(x), \quad m = 1, 2, \dots, N_{obj} \\ \text{restrita a} & g_j(x) \geq 0, \quad j = 1, 2, \dots, NR_{des} \\ & h_k(x) = 0, \quad k = 1, 2, \dots, NR_{igu} \\ & x_i^{(inf)} \leq x_i \leq x_i^{(sup)}, \quad i = 1, 2, \dots, N_{var} \end{array} \right\} \quad (5.1)$$

Portanto,  $x$  é um vetor de  $N_{var}$  variáveis de decisão, tal que  $x = (x_1, x_2, \dots, x_{N_{var}})$ . Os valores  $x_i^{inf}$  e  $x_i^{sup}$  representam os limites inferior e superior de  $x$ , respectivamente. Tais limites definem o espaço de decisão  $S_{dec}$ . Existem ainda as funções de restrição para as desigualdades  $g_j(x) \leq 0$  e igualdades  $h_k(x) = 0$ . Uma solução  $x$  é dita factível se satisfaz ambas as funções de restrição e os limites  $x_i^{inf}$  e  $x_i^{sup}$ . O conjunto de soluções factíveis geram a região factível do espaço de decisão  $S_{fact}$ . Já a função  $f_m$  corresponde a uma função objetivo de um MOOP,

tal que  $m = (m_1, m_2, \dots, m_{N_{obj}})$ , sendo  $N_{obj}$  o número de objetivos existentes. Esse vetor de funções objetivo representa um espaço  $N_{obj}$ -dimensional denominado espaço de objetivos  $S_{obj}$ . A existência de um espaço de objetivos multidimensional é uma das principais diferenças entre a otimização multi-objetivo e a mono-objetivo ou simples, onde esse espaço de busca é unidimensional já que utiliza-se apenas uma função objetivo (DEB et al., 2016).

Um exemplo de MOOP é a análise do custo benefício na compra de um carro, conforme ilustrado na Figura 20. Neste, busca-se adquirir o veículo com maior conforto pelo menor preço, em outras palavras, o intuito é **minimizar** o custo e **maximizar** o conforto. Na Figura 20, existem sete possíveis opções de compra (Pontos de A até G). Intuitivamente, descarta-se a solução A, já que B oferece mais conforto por igual preço. Entre as soluções D, E e F, fica claro que E e F devem ser descartadas, pois a solução D oferece igual conforto por um preço menor. Assim, têm-se quatro soluções interessantes que são boas alternativas de compra (B, C, D e G). Em termos quantitativos, nenhuma solução é superior pois enquanto uma é melhor no conforto, é pior no preço e vice-versa. Diz-se então que existe um equilíbrio entre os dois objetivos desejáveis porém conflitantes, isto é, um *trade-off*, que é justamente um dos aspectos desejados nesse tipo de otimização. Ademais, esse é um bom exemplo de soluções bem diversificadas.

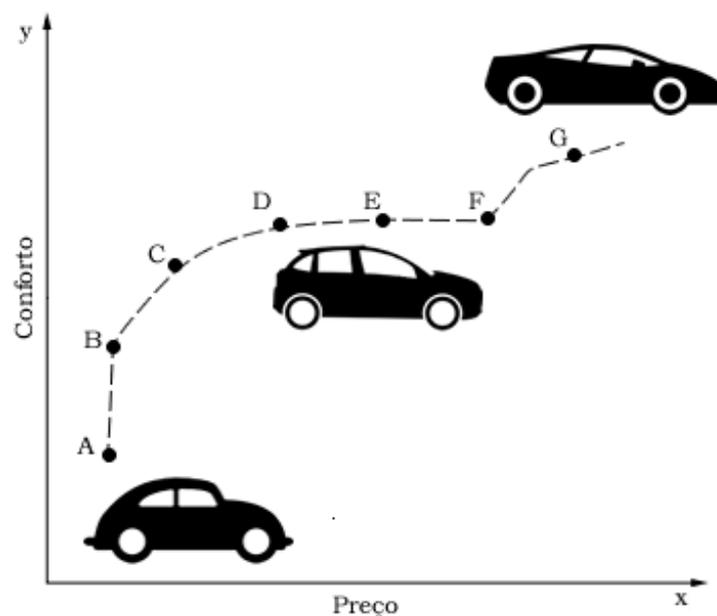


Figura 20 – Exemplo prático de um MOOP. Adaptado de Deb et al. (2016).

O problema supracitado representa a comum existência de objetivos conflitantes entre si em MOOPs, o que dificulta ainda mais a tarefa de otimização. Diz-se que dois objetivos são conflitantes quando não há garantia de melhora simultânea no valor de uma função objetivo  $f_1$  e  $f_2$ , em outras palavras, se o valor de  $f_1$  é otimizado, o valor da função  $f_2$  é deteriorado.

## 5.2 Abordagens Multiobjetivo

Embora muitos dos trabalhos da literatura que abordam a Otimização Multiobjetivo considerem o uso da Fronteira de Pareto (em inglês, *Pareto Front (PF)*), esta não é a única alternativa, pois de fato existem outras abordagens multiobjetivo viáveis para o tratamento de **MOOPs**. Segundo Freitas (2004), existem três principais abordagens multiobjetivo: (1) Abordagem de Pareto, (2) Abordagem de Soma Ponderada e a (3) Abordagem Lexicográfica.

A Abordagem de Pareto visa encontrar um conjunto de soluções não-dominadas próximas da **PF**, e para isso, utiliza do conceito de dominância de Pareto. O conceito de dominância de soluções, introduzido por Edgeworth (1881) e generalizado por Pareto (1964), é utilizado para comparar duas soluções factíveis de um problema. Dada as soluções  $X$  e  $Y$ , pode se dizer que  $X$  domina  $Y$  (denota-se como  $X \preceq Y$ ), caso todos os valores de objetivos de  $X$  são ao menos iguais aos de  $Y$  e, pelos menos um valor de objetivo de  $X$  é superior a um valor de objetivo de  $Y$ . Em linhas gerais, nessa abordagem o que se deseja é encontrar um conjunto de soluções ótimas (não-dominadas), denominado *conjunto Pareto-Ótimo* (do inglês, *Pareto-Optimal*). Essas soluções são demonstradas graficamente pela **PF**, um plano  $n$ -dimensional em que são exibidos os valores das soluções do conjunto Pareto-Ótimo e que representa o maior limite possível de equilíbrio entre múltiplos objetivos. A Abordagem de Pareto permite lidar com vários aspectos de um problema simultaneamente onde objetiva-se encontrar soluções equiparáveis em qualidade, além de ser desejável que sejam mais esparsas quanto possível (PARETO, 1964; DEB, 2001).

Das três abordagens multiobjetivo, a Abordagem de Pareto é a que apresenta a forma mais natural de otimização de um **MOOP**, partindo do princípio que é necessário adaptar a maneira de solucionar o problema e não o contrário. Portanto, esta não simplifica o problema multiobjetivo, de maneira que consegue encontrar soluções não-dominadas ao longo da **PF** e de forma esparsa, o que possibilita a análise dos diferentes *trade-offs* entre os objetivos. Além disso, das abordagens supracitadas é a mais sofisticada, o que lhe permite resolver problemas mais complexos como problemas de Otimização de Muitos Objetivos (no inglês, *Many-Objective Optimization*), termo usado na literatura quando um **MOOP** apresenta mais de três objetivos e para qual é considerada a abordagem estado-da-arte para lidar com esses.

Já a Abordagem de Soma Ponderada consiste em transformar um **MOOP** em um problema tradicional de otimização usando uma fórmula ponderada, onde os objetivos definidos são expressos em conjunto e os pesos associados representam suas respectivas relevâncias para o problema (BANDARU et al., 2017). A Equação 5.2 exemplifica uma função objetivo nessa abordagem, em que  $w_1$ ,  $w_2$  e  $w_3$  são os pesos atribuídos a cada objetivo. Nessa fórmula, os valores dos objetivos são combinados linearmente em um único valor que representa o compromisso de todos os objetivos na resolução do **MOOP**. Das três abordagens, essa é a mais utilizada na literatura de Data Mining e a única considerada *ad-hoc*<sup>1</sup>, enquanto que as demais seguem uma

<sup>1</sup> Algo sem processo bem definido. Genérico, podendo ser aplicado a múltiplas situações. Faz oposição a uma abordagem sistemática para resolver problemas.

lógica de funcionamento bem estabelecida (FREITAS, 2004).

$$Fitness = w_1 \cdot objetivo_1 + w_2 \cdot objetivo_2 + w_3 \cdot objetivo_3 \quad (5.2)$$

Essa é a abordagem multiobjetivo mais simples tanto no funcionamento quanto na implementação, sendo essa sua principal vantagem em relação as demais, pois possibilita uma rápida adaptação de algoritmos de otimização tradicionais para multiobjetivo uma vez que é necessário apenas mudar a função objetivo do método (FREITAS, 2004; DUTTA, 2009).

Na Abordagem Lexicográfica os objetivos têm prioridades associadas, geralmente definidas *a priori* por um especialista de domínio ou com base em observações, onde são então ranqueados conforme essas e otimizados em ordem (FREITAS, 2004). Quando duas soluções são comparadas visando determinar a melhor, compara-se primeiramente o valor do objetivo de maior prioridade. Caso uma das soluções seja suficientemente melhor que a outra no que se remete a esse objetivo, essa solução em questão é definida como a melhor. Caso não haja uma diferença significativa entre as soluções, compara-se os valores para o segundo objetivo mais relevante. Esse processo se repete até que uma solução seja declarada a melhor solução, e em caso de empate em todos os objetivos, a solução que apresente maior valor no objetivo prioritário é escolhida como a melhor solução. Cabe salientar que essa “significância” na comparação entre objetivos é simplesmente um limiar que deve ser satisfeito (JUNGGIT; FREITAS, 2015).

Essa abordagem tem como principal diferencial o uso de conhecimento *a priori* acerca da importância dos objetivos definidos na otimização de um MOOP. Apesar de ter uma lógica de funcionamento mais elaborada que a Abordagem de Soma Ponderada, ainda assim, é considerada conceitualmente simples e de fácil implementação.

Naturalmente, essas abordagens possuem vantagens e desvantagens quando comparadas umas com as outras, além de terem aplicabilidades diferentes dependendo do problema para o qual estão sendo aplicadas. Logo, nenhuma abordagem é um aprimoramento direto de outra, sendo comparáveis em diversos critérios de qualidade.

### 5.3 Algoritmos de Otimização Multiobjetivo

Uma das etapas primordiais em uma metodologia de Otimização Multiobjetivo é a identificação da estratégia de otimização adequada (COELLO et al., 2007). Uma variedade de algoritmos com diferentes vieses podem ser utilizados na pesquisa por um ótimo global em um MOOP, e entre as estratégias mais comuns, estão os métodos **baseados no gradiente** e os de **processo estocástico** (COELLO, 2006). Os algoritmos baseados no gradiente avaliam o gradiente da função objetivo com o intuito de encontrar seu sentido crescente<sup>2</sup>, e dessa forma, selecionar e ajustar um conjunto de variáveis dentro do domínio da função. Já os algoritmos

<sup>2</sup> Quando se realiza a otimização em direção ao sentido decrescente da função, trata-se do gradiente descendente

estocásticos realizam uma busca exploratória através de múltiplas avaliações, muitas vezes simultâneas, em amplos espaços de busca no domínio da função (DEB et al., 2016).

Métodos de otimização têm sido propostos desde meados de 1950, sendo muitos destes, métodos de programação matemática (ex: método do gradiente) (COELLO, 2006; GOLDBERG, 1989). Apesar da ampla variedade encontrada na literatura, essa abordagem possui certas limitações em relação a algumas particularidades dos MOOPs. Uma delas é a sensibilidade quanto a distribuição dos valores de objetivos, pois os métodos de programação matemática têm dificuldade em lidar com padrões côncavos ou descontínuos, alguns ainda, requerem diferencialidade nas funções objetivo e suas restrições. Adicionalmente, a maioria dessas técnicas gera apenas uma única solução por execução, sendo então necessária várias execuções utilizando diversos pontos de partidas diferentes (variáveis de decisão) a fim de encontrar pontos diversificados no espaço de objetivos, algo que é desejável na resolução de MOOPs.

Em contraste, métodos estocásticos como as técnicas evolutivas conseguem trabalhar com um conjunto de soluções candidatas, o que lhes permite encontrar diversas soluções de qualidade em uma única execução do algoritmo. Adicionalmente, algoritmos evolutivos são menos sensíveis aos problemas de distribuição dos dados, conseguindo alcançar boas soluções mesmo quando existem padrões descontínuos ou côncavos (DEB, 2001; DEB et al., 2016).

Graças aos vários casos de aplicação bem sucedidos, algoritmos evolutivos tornaram-se populares em tarefas de otimização simples, sendo os GAs um dos principais exemplos dessa família de algoritmos (GOLDBERG; SASTRY, 2001; GOLBERG, 1989). Nos últimos anos, os GAs também têm se mostrado boas alternativas a Otimização Multiobjetivo, já que conseguem encontrar boas soluções mesmo para problemas complexos com objetivos conflitantes e com espaço de busca grande (COELLO, 2006; DEB et al., 2016).

A aplicação de GAs em MOOPs apresenta três grandes vantagens em relação a métodos clássicos de otimização, conforme aponta Deb (2001), sendo eles:

- Não introduzem parâmetros adicionais ao problema;
- Conseguem trabalhar com várias funções objetivos;
- Podem encontrar um conjunto de boas soluções em apenas uma execução do algoritmo.

## 5.4 Considerações Finais do Capítulo

Neste capítulo foi apresentada uma revisão da literatura acerca da Otimização multiobjetivo que incluiu desde sua definição formal até seus conceitos básicos como a caracterização do que são objetivos conflitantes no processo de otimização, a importância do equilíbrio entre objetivos para determinados problemas, os diferentes espaços de busca em que a otimização multiobjetivo deve lidar, sendo esses o plano das variáveis de decisão e dos objetivos, que

constituem a principal diferença entre as otimizações simples e multiobjetivo, bem como as principais abordagens existentes para o tratamento de um **MOOP** como a Abordagem de Soma Ponderada, Lexicográfica e a baseada em Pareto, além de conceitos relacionados a estas como a Dominância de Soluções e a Fronteira de Pareto.

# Capítulo 6

## MÉTODOS PROPOSTOS

---

---

Nesse capítulo são apresentados as soluções propostas nessa pesquisa de mestrado. Os métodos hierárquicos aqui discutidos seguem a Abordagem Global de **HC** e utilizam Algoritmos Genéticos como base para gerar e evoluir regras de classificação. Esses métodos foram implementados usando a linguagem Java sem qualquer *framework* evolutivo. Cabe salientar que o método **HC-GA** trata-se de uma extensão do também método de otimização simples **HMC-GA**, proposto na tese de doutorado de Cerri (2014). Como o **HC-GA** serviu de base para a construção dos demais métodos, o detalhamento acerca dos processos básicos de seu funcionamento são os mesmos utilizados nos métodos multiobjetivo **HC-WGA** e **HC-LGA**, e portanto, não serão duplamente comentados. Dessa forma, será abordado posteriormente apenas as modificações feitas no método base que permitiram a criação dos novos métodos multiobjetivo.

### 6.1 Classificação Hierárquica com um Algoritmo Genético (HC-GA)

Essa seção apresenta o método global de otimização simples denominado **HC-GA** (do inglês, *Hierarchical Classification with a Genetic Algorithm*), capaz de gerar uma lista de regras do tipo “Se-Então” que classificam instâncias em múltiplos níveis de uma hierarquia de classes. O processo evolutivo que realiza a indução dessas regras é mostrado no Algoritmo 2.

Juntamente com o processo evolutivo, é empregado uma estratégia de cobertura sequencial na indução de regras de classificação. Logo que uma regra é induzida (linha 21) ao final da aplicação dos operadores genéticos, essa é usada para cobrir instâncias, as quais são imediatamente removidas do conjunto de treinamento (linha 22), permitindo assim a geração de novas regras que cobrirão as instâncias remanescentes. Se ao final do treinamento ainda existirem instâncias não cobertas, é utilizado uma regra padrão que as classifica utilizando um vetor médio obtido a partir dos vetores de classe de **todas as instâncias** do conjunto de treinamento. A definição de um vetor de classes é apresentada na Seção 6.1.2.

**Algoritmo 2:** Procedimento geral dos GAs hierárquicos.

---

**Entrada:** Dados de treinamento  $D$ , número de gerações  $G$ , tamanho da população  $p$ , número máximo de instâncias cobertas por regra  $maxCov$ , número mínimo de instâncias cobertas por regra  $minCov$ , número máximo de instâncias não cobertas  $maxUncov$ , taxa de cruzamento  $cr$ , taxa de mutação  $mr$ , tamanho do torneiro  $t$ , número de indivíduos selecionados por elitismo  $e$ , probabilidade de usar um termo numa regra  $pt$ .

**Saída:** Lista de *Regras*

```

1 início
2    $Regras \leftarrow \emptyset$ 
3   enquanto ( $|D| > maxUncov$ ) faça
4      $atualPop \leftarrow geraPopulacao(D, p, pt)$ 
5      $atualPop \leftarrow buscaLocal(minCov, maxCov, atualPop)$ 
6      $calculaFitness(atualPop, D)$ 
7      $melhorRegra \leftarrow obtemMelhorRegra(atualPop)$ 
8      $j \leftarrow 0$ 
9     enquanto  $j < G$  ou  $regraConverge()$  faça
10       $novaPop \leftarrow \emptyset$ 
11       $novaPop \leftarrow novaPop \cup elitismo(atualPop, e)$ 
12       $pais \leftarrow selecaoTorneio(atualPop, t)$ 
13       $filhos \leftarrow cruzamentoEspecializado(pais, cr)$ 
14       $filhos \leftarrow mutacao(filhos, mr)$ 
15       $novaPop \leftarrow novaPop \cup filhos$ 
16       $novaPop \leftarrow buscaLocal(minCov, maxCov, novaPop)$ 
17       $atualPop \leftarrow novaPop$ 
18       $calculaFitness(atualPop, D)$ 
19       $melhorRegra \leftarrow obtemMelhorRegra(atualPop, melhorRegra)$ 
20       $j \leftarrow j + 1$ 
21       $Regras \leftarrow Regras \cup melhorRegra$ 
22      Remove instâncias de  $D$  cobertos pela  $melhorRegra$ 
23   se  $|D| > 0$  então
24     Remove instâncias de  $D$  usando a  $regraPadrao$ 

```

---

### 6.1.1 Codificação dos Indivíduos

Um indivíduo no HC-GA é um vetor codificado de tamanho fixo contendo valores reais que representam o antecedente de uma regra de classificação. Cada conjunto de quatro posições nesse vetor corresponde a um teste relacionado a um atributo do conjunto de dados, que é combinado com os demais testes usando cláusulas conjuntivas **AND** que devem ser satisfeitas para que uma instância seja considerada coberta pelo antecedente. Esses testes são codificados utilizando uma tupla de quatro posições, e seus componentes são uma FLAG, um operador (OP) e os dois genes  $\Delta_1$  e  $\Delta_2$  que recebem índices ou valores de atributo.

O gene FLAG pode receber os valores 1 ou 0, indicando se um teste está ativo ou não no antecedente da regra. A vantagem de seu uso é permitir o fácil controle do número de testes numa

regra. Já o gene OP é responsável por armazenar o índice correspondente a um dos possíveis operadores de comparação, que variam de acordo com o tipo de atributo testado, podendo ser numérico, ordinal ou categórico. Como nesse trabalho apenas dados numéricos são utilizados, os operadores possíveis limitam-se a  $\leq$  e  $\geq$ . Por sua vez, os genes  $\Delta_1$  e  $\Delta_2$  são definidos com os valores reais utilizados nos testes, podendo inclusive serem usados em conjunto, representando assim os limites inferior e superior de dado atributo. A Figura 21 ilustra um indivíduo codificado (antecedente de regra) e seus respectivos testes ativos.

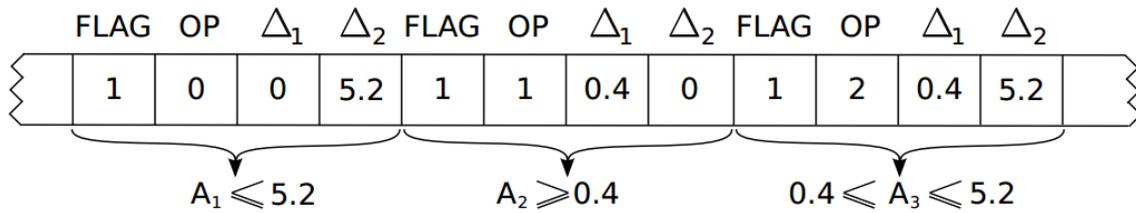


Figura 21 – Codificação de um indivíduo nos GAs hierárquicos. Adaptado de Cerri (2014).

Diferente do valor para o gene FLAG que é definido de acordo com uma probabilidade  $pt$ , o gene OP recebe valores de forma aleatória. Caso o índice 0 seja escolhido aleatoriamente, o operador  $\leq$  é usado no teste, fazendo com que  $\Delta_1$  seja definido como 0 e  $\Delta_2$  receba o valor do atributo que está sendo testado. Em caso da escolha do índice 1, o operador  $\geq$  é usado e o oposto ocorre com os genes  $\Delta$ . Além disso, é possível verificar se um valor de atributo numérico pertence a um determinado intervalo quando o índice 2 é escolhido. Tomando como exemplo  $\Delta_1 \leq A_k \leq \Delta_2$ , os valores  $\Delta$  representam os limites inferior e superior para o atributo, respectivamente. Nesse caso, os genes  $\Delta$  são escolhidos aleatoriamente para que o valor da condição de teste satisfaça o atributo da instância. É importante enfatizar que todos os operadores possíveis são indexados previamente com índices fixos, permitindo que um teste seja facilmente construído através da recuperação dos índices e operadores correspondentes.

### 6.1.2 Construção do Consequente

Uma vez que o antecedente de uma regra  $r$  é construído, é possível calcular seu vetor médio de classes através da Equação 6.1, onde  $S_r$  corresponde ao conjunto de todas as instâncias de treinamento cobertas pela regra  $r$ , enquanto que  $S_{r,i}$  equivale ao conjunto de instâncias cobertas por  $r$  que são classificadas na classe  $i$ . Portanto, cada posição  $i$  desse vetor de tamanho  $C$ , em que  $C$  é o conjunto de classes do problema, recebe um valor real entre  $[0, 1]$  correspondente a proporção de instâncias cobertas e que são classificadas na classe  $i$ . Esses valores são então interpretados como as probabilidades de atribuição de classes as instâncias.

$$\bar{v}_{r,i} = \frac{|S_{r,i}|}{|S_r|} \tag{6.1}$$

O vetor médio de classes de uma regra é considerado o seu consequente. Deste modo, para instâncias que satisfaçam o antecedente de  $r$ , esse vetor é utilizado para assinalar classes as instâncias cobertas. A Figura 22 exibe um exemplo do vetor consequente em que cada posição contém o valor real ou probabilidade para as classes dadas acima. Vale ressaltar que é garantido a consistência hierárquica desse vetor, deste modo, sempre formasse um caminho hierárquico válido. Ademais, um limiar é aplicado nos vetores consequentes a fim de se obter as predições finais. Nesse trabalho foi utilizado um limiar igual à 0.5, desta forma, se o valor da  $i_{th}$  posição no vetor consequente for maior ou igual a esse limiar, recebe o valor 1, caso contrário receberá 0.

1	1.1	1.1.1	1.1.2	1.4	1.5	2	2.1	2.1.1	2.1.1.1	2.1.1.2	2.1.1.3	2.1.1.8	2.1.1.9
0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.9	0.41901	0.81292	0.21409	0.24489	0.16666

Figura 22 – Exemplo de um vetor de classes nos GAs hierárquicos.

### 6.1.3 Inicialização da População

Uma população de indivíduos (regras) é criada utilizando um processo de semeadura, no qual as instâncias de treinamento são selecionadas aleatoriamente e seus atributos são usados para compor os indivíduos. Cada atributo de uma instância selecionada tem uma probabilidade  $pt$  de ser usado em um indivíduo. Como os indivíduos são previamente indexados, essa probabilidade  $pt$  é usada para ativar ou desativar os genes FLAG, habilitando ou desabilitando um teste no antecedente da regra. Normalmente, um valor baixo é escolhido para  $pt$  tanto no processo de semeadura quanto para a mutação das FLAGS, já que um valor alto resultaria em regras com muitos testes ativos e que cobririam somente algumas instâncias ou apenas a instância semente.

Esse processo de semeadura garante que cada indivíduo cubra pelo menos uma instância de treinamento, sendo essa a sua própria instância semente. Uma instância é considerada coberta por uma regra se todos os testes ativos nessa regra forem satisfeitos pelos atributos da instância. Esse processo é repetido até que uma população com o tamanho desejado seja gerada.

### 6.1.4 Funções de Aptidão

Uma variedade de funções de aptidão foram implementadas e testadas inicialmente no HC-GA e, posteriormente, utilizadas e combinadas no HC-WGA e HC-LGA. Todas essas funções foram otimizadas de maneira a maximizá-las, sendo melhor abordadas a seguir:

1) Medidas de Avaliação Hierárquica **hP**, **hR** e **hF**: Fruto de adaptações nas medidas de avaliação tradicionais Precisão, Revocação e *F1 Score*, as versões hierárquicas foram propostas para que fosse levado em consideração as predições realizadas em múltiplos níveis hierárquicos (KIRITCHENKO et al., 2005). Uma vez que tais medidas são aplicadas aos resultados dos métodos hierárquicos aqui propostos, optou-se pela utilização direta dessas medidas como funções objetivo. Essas medidas são melhores discutidas na Seção 7.4.

2) Área sob a curva de Precisão-Revocação Média ( $AU(\overline{PRC})$ ): Para cada instância de treinamento, os **GAs** hierárquicos calculam um vetor de classes contendo valores no intervalo  $[0, 1]$  e que podem ser interpretados como a probabilidade de associação de dada classe a uma instância. Através desses vetores é possível aplicar diversos limiares que resultarão em diferentes valores de Precisão e Revocação (um para cada limiar), em outras palavras, pontos dentro do espaço Precisão-Revocação (PR). Com a união desses pontos se obtém a curva PR, o que possibilita o cálculo da área sob a curva ( $AUC$ ) (DAVIS; GOADRICH, 2006).

Para calcular a  $AU(\overline{PRC})$  é necessária uma interpolação dos pontos de PR, pois conectar os pontos sem interpolação aumentaria artificialmente a  $AUC$ . Dado um limiar, um ponto PR é obtido através das Equações 6.2 e 6.3, que correspondem às micro médias de Precisão e Revocação e, assim como as medidas **hP** e **hR**, levam em consideração os relacionamentos hierárquicos entre classes. Nessas equações,  $i$  varia de 1 até o número de classes do problema, enquanto que o número de Verdadeiros Positivos, Falsos Positivos e Falsos Negativos são representados por  $TP$ ,  $FP$  e  $FN$ , respectivamente. Os valores de  $AU(\overline{PRC})$  variam entre  $[0, 1]$  e quanto mais próximo de 1, melhor (VENS et al., 2008).

$$\overline{P} = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FP_i} \quad (6.2)$$

$$\overline{R} = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FN_i} \quad (6.3)$$

3) Porcentagem de Cobertura (no inglês, *Percentage Coverage (PC)*): Essa função de aptidão mensura a porcentagem de instâncias no conjunto de treinamento que são cobertas por uma determinada regra. Assim, seu valor é calculado utilizando a Equação 6.4, em que  $S_r$  é o conjunto de instâncias cobertas por uma regra  $r$  e  $S$  é o conjunto total de instâncias.

$$PC(r) = \frac{|S_r|}{|S|} \quad (6.4)$$

4) Ganho de Variância (no inglês, *Variance Gain (VG)*): Essa função mensura o decréscimo da variância alcançada entre instâncias cobertas por uma regra e instâncias que não o são, em outras palavras, tende a particionar um conjunto inteiro de instâncias em conjuntos de instâncias mais semelhantes (VENS et al., 2008). A fórmula do **VG** é mostrada na Equação 6.5, sendo calculada para uma regra  $r$  em relação a um conjunto de instâncias de treinamento  $S$ . Nesse equação, o conjunto  $S$  é dividido em dois subconjuntos: (1) o conjunto  $S_r$  de instâncias cobertas por uma regra  $r$  e (2) o conjunto  $S_{\neg r}$  de instâncias não cobertas por  $r$ . Para se obter o valor de **VG**, deve-se calcular a Variância ( $var$ ) dos três conjuntos de instâncias levados em consideração, cujo a fórmula é apresentada na Equação 6.6.

$$VG(r, S) = var(S) - \frac{|S_r|}{|S|} \cdot var(S_r) - \frac{|S_{\neg r}|}{|S|} \cdot var(S_{\neg r}) \quad (6.5)$$

$$var(S) = \frac{\sum_{k=1}^{|S|} distanciaEuclidianaPond(\mathbf{v}_k, \bar{\mathbf{v}})^2}{|S|} \quad (6.6)$$

A Variância é definida como a distância média quadrática entre dois vetores médio de classes (vetor  $\mathbf{v}$  de dada regra e vetor  $\bar{\mathbf{v}}$  de todas as instâncias do conjunto  $S$ ) dividido pelo número de instâncias presentes no conjunto sendo considerado ( $S$ ,  $S_r$  ou  $S_{-r}$ ). Nesse cálculo é utilizada a Distância Euclidiana ponderada que adota um esquema de pesos para lidar com as particularidades de uma estrutura hierárquica, sendo apresentada na Equação 6.7. Nesta equação,  $w_i$  corresponde ao peso associado à classe  $i^{th}$  na hierarquia, enquanto  $v_{1i}$  e  $v_{2i}$  são os valores associados a posição  $i^{th}$  nos vetores médio de classes  $\bar{\mathbf{v}}_1$  e  $\bar{\mathbf{v}}_2$ , respectivamente. Esses pesos estão associados a todas as classes da estrutura e diferem em cada nível hierárquico, porque no contexto da HC as similaridades entre classes em níveis mais altos são frequentemente mais relevantes do que aquelas em níveis mais profundos de uma hierarquia (VENS et al., 2008).

$$distanciaEuclidianaPond(\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2) = \sqrt{\sum_{i=1}^{|C|} w_i \cdot (\bar{v}_{1,i} - \bar{v}_{2,i})^2} \quad (6.7)$$

$$w_i = w_0 \times \sum_{j=1}^{P_i} w(p_j) / P_i \quad (6.8)$$

Quanto a atribuição dos pesos, foi utilizado o mesmo processo adotado em Vens et al. (2008), onde após experimentos os autores relataram que o melhor esquema encontrado é definido de acordo com a Equação 6.8. O peso  $w_0$  associado a uma classe no primeiro nível foi fixado em 0.75, enquanto que o peso de uma classe  $i$  é recursivamente definido como a multiplicação de  $w_0$  pelo peso médio de todas as classes ancestrais  $P_i$ .

### 6.1.5 Operadores Genéticos

Após o cálculo de aptidão da população (Algoritmo 2, linha 6) e o início do processo evolutivo (Algoritmo 2, linhas 9 à 20), uma operação de Elitismo é aplicada a população atual (Algoritmo 2, linha 11), salvando as  $e$  melhores regras e passando-as diretamente para a nova população. Posteriormente, um conjunto de regras pai  $p - e$  (onde  $p$  é o tamanho da população) é selecionado usando uma *Seleção por Torneio* (Algoritmo 2, linha 12), o que permite que um *Cruzamento Especializado* que combina regras similares seja aplicado às regras pai (Algoritmo 2, linha 13), gerando as regras filho. Este cruzamento é apresentado no Algoritmo 3.

A ideia por trás desse operador é a de que regras similares provavelmente cobrem instâncias próximas umas das outras no espaço de busca, o que significa que possivelmente também são classificadas nas mesmas classes e, portanto, combiná-las levaria a regras com alta coesão e precisão. Esse operador calcula as distâncias euclidianas entre os vetores médio de classes de duas regras para determinar se são semelhantes (Algoritmo 3, linha 6), sendo

**Algoritmo 3:** Cruzamento especializado das regras.

---

**Entrada:** Regras pais  $R$ , taxa de cruzamento  $cr$ .  
**Saída:** Conjunto de regras  $filhas$

```

1 início
2    $filhas \leftarrow \emptyset$ 
3   enquanto ( $|R| \geq 2$ ) faça
4      $pai_1 \leftarrow R[0]$ 
5      $R \leftarrow R - pai_1$ 
6      $pai_2 \leftarrow obterRegraMaisSimilar(pai_1, R)$ 
7      $R \leftarrow R - pai_2$ 
8      $filhas \leftarrow filhas \cup cruzamentoUniforme(pai_1, pai_2, cr)$ 

```

---

que quanto menor a distância, mais semelhantes elas são. Uma vez selecionadas as regras, um Cruzamento Uniforme convencional é aplicado de acordo com uma taxa de cruzamento  $cr$  (Algoritmo 3, linha 8). Esse processo é repetido até que uma nova geração de regras seja obtida.

Uma vez realizado o cruzamento das regras, a nova população é submetida a um operador de mutação (Algoritmo 2, linha 14). O operador aqui aplicado trata-se da chamada mutação *Bit-flip*, muito utilizada em indivíduos com codificações binárias, mas que no caso desse trabalho fora aplicado aos *bits* binários que ativam e desativam testes em indivíduos codificados de forma real/ponto flutuante. Este operador consiste em alterar os genes binários de acordo com uma taxa de mutação  $mr$ , alterando um valor “0” para “1” e vice-versa. O Algoritmo 4 mostra esse processo de mutação juntamente com as operações de generalização e restrição relacionadas.

**Algoritmo 4:** Mutação e generalização/restrrição de operadores.

---

**Entrada:** População  $P$ , taxa de mutação  $mr$ , probabilidade de usar um termo  $pt$ .  
**Saída:** Conjunto de  $regrasMutadas$

```

1 início
2    $regrasMutadas \leftarrow selecaoMutacao(P, mr)$ 
3    $populacaoRestante \leftarrow P - regrasMutadas$ 
4   para cada  $individuo$  das  $regrasMutadas$  faça
5      $aleatorio \leftarrow numeroRealAleatorio(0, 1)$ 
6     se  $aleatorio \leq 0.5$  então
7        $individuo \leftarrow mutacaoFlag(individuo, pt)$ 
8     senão
9       para cada  $testeAtivo$  do  $individuo.testes$  faça
10         $operacao \leftarrow numeroInteiroAleatorio(0, 1)$ 
11        se  $operacao = 0$  então
12           $testeAtivo \leftarrow generalizacao(testeAtivo)$ 
13        senão
14           $testeAtivo \leftarrow restricao(testeAtivo)$ 
15    $regrasMutadas \leftarrow regrasMutadas \cup populacaoRestante$ 

```

---

### 6.1.6 Busca Local

Depois que uma população é gerada inicialmente (Algoritmo 2, linha 4) e quando uma nova população é obtida através de mutação (Algoritmo 2, linha 14), essas populações são submetidas a uma busca Local (Algoritmo 5, linhas 5 e 16), conforme apresentado no Algoritmo 2. Essa busca local visa assegurar que as regras geradas cubram entre um número mínimo e máximo de instâncias, e assim, evitando que essas regras sejam muito gerais ou específicas, o que poderia comprometer o desempenho preditivo e a interpretabilidade do modelo. Além disso, esse procedimento contribui para que a busca não permaneça estagnada em ótimos locais, pois direciona a exploração para regiões mais promissoras no espaço de busca.

---

#### Algoritmo 5: Busca local min-max.

---

**Entrada:** Dados de treinamento  $D$ , População  $P$ , número máximo de instâncias cobertas por regra  $maxCov$ , número mínimo de instâncias cobertas por regra  $minCov$ , número máximo de tentativas  $tMax$ .

**Saída:** Conjunto de *novasRegras*

```

1 início
2   para cada individuo de P faça
3     tentativa ← 0
4     convergencia ← 0
5     enquanto convergencia = 0 E tentativa < tMax faça
6       tentativa ← tentativa + 1
7       se individuo.instanciasCov < minCov então
8         se |individuo.testes| > 1 então
9           individuo ← desativaTesteRandom(individuo)
10        senão se |individuo.testes| = 1 então
11          novoTeste ← generalizacaoOperador(individuo.testes[0])
12          individuo.testes[0] ← novoTeste
13        senão se individuo.instanciasCov > maxCov então
14          se |individuo.testes| < |D.atributos| então
15            individuo ← ativaTesteRandom(individuo)
16        senão
17          convergencia ← 1
18   novasRegras ← P

```

---

## 6.2 Classificação Hierárquica com um Algoritmo Genético de Soma Ponderada (HC-WGA)

Nessa seção é apresentado o método global de otimização multiobjetivo que usa de uma Abordagem de Soma Ponderada denominado **HC-WGA** (do inglês, *Hierarchical Classification with a Weighted Genetic Algorithm*). Como já mencionado, a mesma lógica de funcionamento do **HC-GA** assim como os operadores genéticos, a cobertura sequencial de instâncias e demais especificações discutidas na Seção 6.1, também se aplicam ao **HC-WGA** sem quaisquer alterações, com exceção dos cálculos das funções de aptidão.

O **HC-WGA** foi criado com a intenção de explorar a simplicidade da Abordagem de Soma Ponderada, o que possibilitou a rápida adaptação de um método de otimização simples como o **HC-GA** em um método de otimização multiobjetivo, que apesar de simplificar um **MOOP**, ainda assim é capaz de lidar com os múltiplos objetivos de um problema a um baixo custo. A Figura 23 exemplifica como se comporta o processo de busca e otimização realizado pelo método **HC-WGA**, em que cada ponto expresso no plano cartesiano representa uma solução (regra) e seus respectivos valores para os objetivos *Obj1* e *Obj2*.

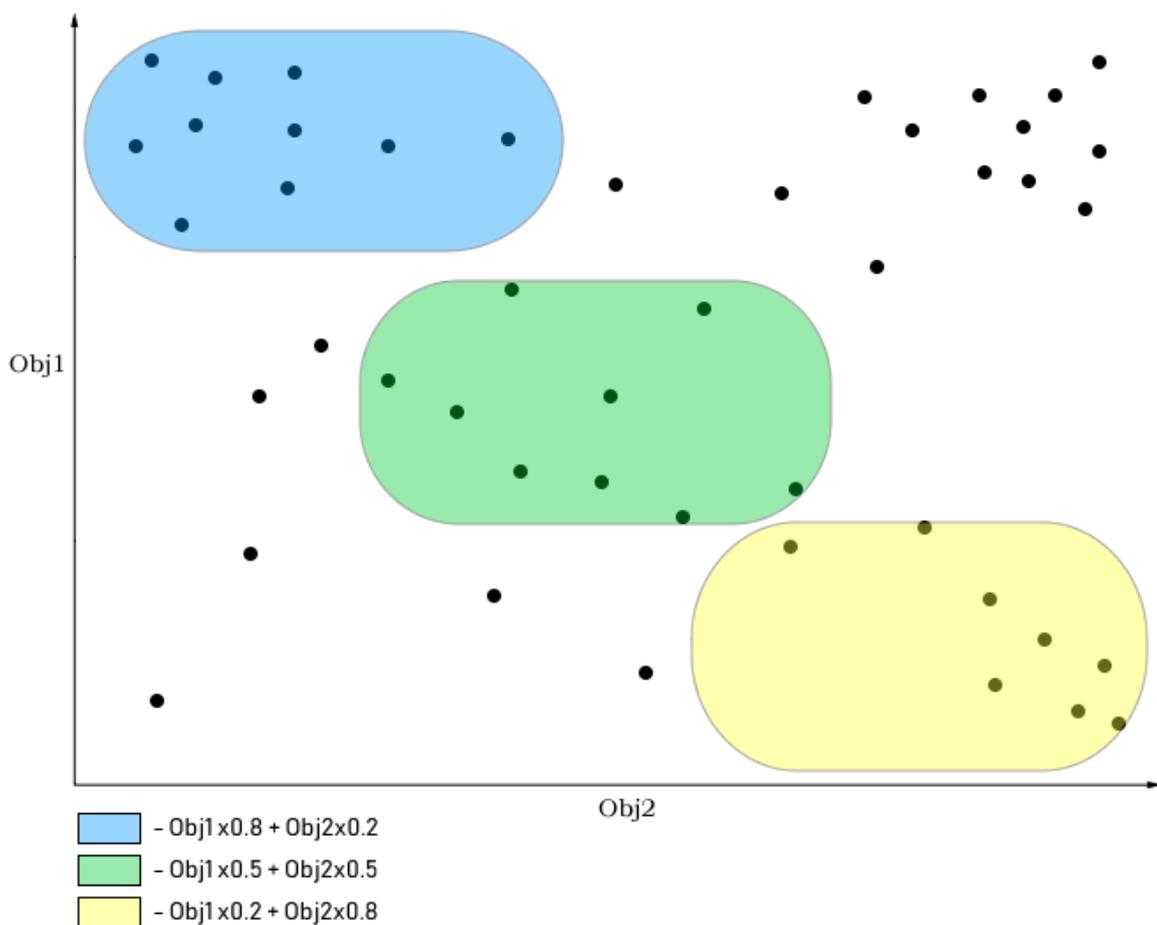


Figura 23 – Exemplificação do processo de busca e otimização realizado pelo **HC-WGA**.

Numa otimização usando a Abordagem de Soma Ponderada, são utilizados conjuntos de pesos que representam as “importâncias” atribuídas aos valores de objetivo, de maneira que esses são combinados de forma linear em um único valor de aptidão que guia a busca por soluções. Deste modo, a alteração do conjunto de pesos modifica o espaço a ser explorado dentro do espaço de soluções possíveis, direcionando a busca para determinadas regiões desse plano.

Considerando o cenário mostrado na Figura 23, cada região do espaço de busca marcada com uma cor representa o limite hipotético explorado por uma fórmula ponderada usando uma certa combinação de pesos. Nesse exemplo, a região mais à esquerda (azul) contém soluções que apresentam maiores valores para o *Obj1*, já a região na extrema direita (amarelo) contém soluções que favorecem mais o *Obj2*, enquanto que a região ao centro (verde) apresenta soluções com valores mais equilibrados entre esses dois objetivos. Cada uma dessas regiões é explorada por diferentes combinações de pesos usados na fórmula ponderada, o que gera certa flexibilidade ao método e esta pode ser vista como sua principal vantagem, pois pode ser facilmente customizado para encontrar soluções em regiões consideradas potencialmente promissoras.

### 6.3 Classificação Hierárquica com um Algoritmo Genético Lexicográfico (HC-LGA)

Nessa seção é apresentado o método global de otimização multiobjetivo que utiliza de uma Abordagem Lexicográfica denominado **HC-LGA** (do inglês, *Hierarchical Classification with a Lexicographic Genetic Algorithm*), e que assim como o **HC-GA** e o **HC-WGA**, compartilha dos mesmos mecanismos básicos de funcionamento apresentados na Seção 6.1, diferindo apenas na maneira como avalia os indivíduos e determina a melhor solução.

O método **HC-LGA** visa encontrar as soluções que otimizam os objetivos definidos através de uma checagem lexicográfica. Seu processo se inicia pela busca da solução que melhor otimiza o objetivo prioritário dentre uma população de soluções candidatas, comparando-as em pares e levando em consideração um limiar de tolerância. Cada objetivo possui um limiar associado que é usado para verificar se duas soluções são equiparáveis em qualidade. Caso existam mais soluções ótimas para o objetivo corrente, parte-se para checagem do próximo objetivo mais relevante. Esse processo é repetido até que todos os objetivos e respectivos limiares tenham sido verificados e uma solução ótima tenha sido definida, ou até que se conclua que não existem soluções ótimas em potencial na população. Cabe ressaltar que se em alguma iteração apenas uma solução for considerada ótima para o objetivo corrente, mesmo que hajam mais objetivos na sequência o processo é interrompido. Isso é feito pois significa que não houveram mais soluções que atenderam ao limiar de tolerância e portanto não são equiparáveis em qualidade. O Algoritmo 6 define na prática esse funcionamento.

O procedimento geral apresentado no Algoritmo 2 ocorre da mesma maneira no **HC-LGA**, com a exceção de que é aplicado um processo lexicográfico que encontra a melhor regra dentre

uma população de regras, conforme mostra o Algoritmo 6. Essa otimização lexicográfica acontece mais precisamente antes da execução do operador de Elitismo na linha 11 do Algoritmo 2.

---

**Algoritmo 6:** Processo lexicográfico de otimização multiobjetivo.

---

**Entrada:** População *atualPop*, lista de funções objetivos *objetivos*.

**Saída:** *melhorRegra*

```

1 início
2   atualPop ← ordenaPorObjetivoPrioritario(atualPop)
3   melhorRegra ← obtemMelhorRegra(atualPop)
4   para cada regra da atualPop faça
5     para cada obj de objetivos faça
6       diferenca ← melhorRegra.fitness(obj) – regra.fitness(obj)
7       se valorAbsoluto(diferenca) > limiaresObjetivos(obj) então
8         se diferenca < 0 então
9           melhorRegra ← regra
10          Interrompe o loop corrente
11         senão se obj = objetivos[0] então
12           Interrompe o processo lexicográfico pois nenhuma regra
13           apresentará melhores valores para o objetivo prioritário
14         senão
15           se toda a lista objetivos foi percorrida então
16             se melhorRegra.fitness(0) > regra.fitness(0) então
17               Processo lexicográfico é interrompido
18             senão
19               melhorRegra é atualizada ou não por regra aleatoriamente

```

---

Diferente do HC-WGA que compila os valores de objetivo em uma só função ponderada com a adição de pesos que modificam os espaços a serem explorados, o método lexicográfico HC-LGA utiliza de diversas funções objetivo em ordem de importância que guiam o processo de otimização conforme as preferências definidas *a priori*. A Figura 24 ilustra a ideia da busca e otimização realizada pelo HC-LGA, em que cada ponto no plano cartesiano é uma solução ou indivíduo e os eixos *Obj1* e *Obj2* representam os valores de objetivo para esse.

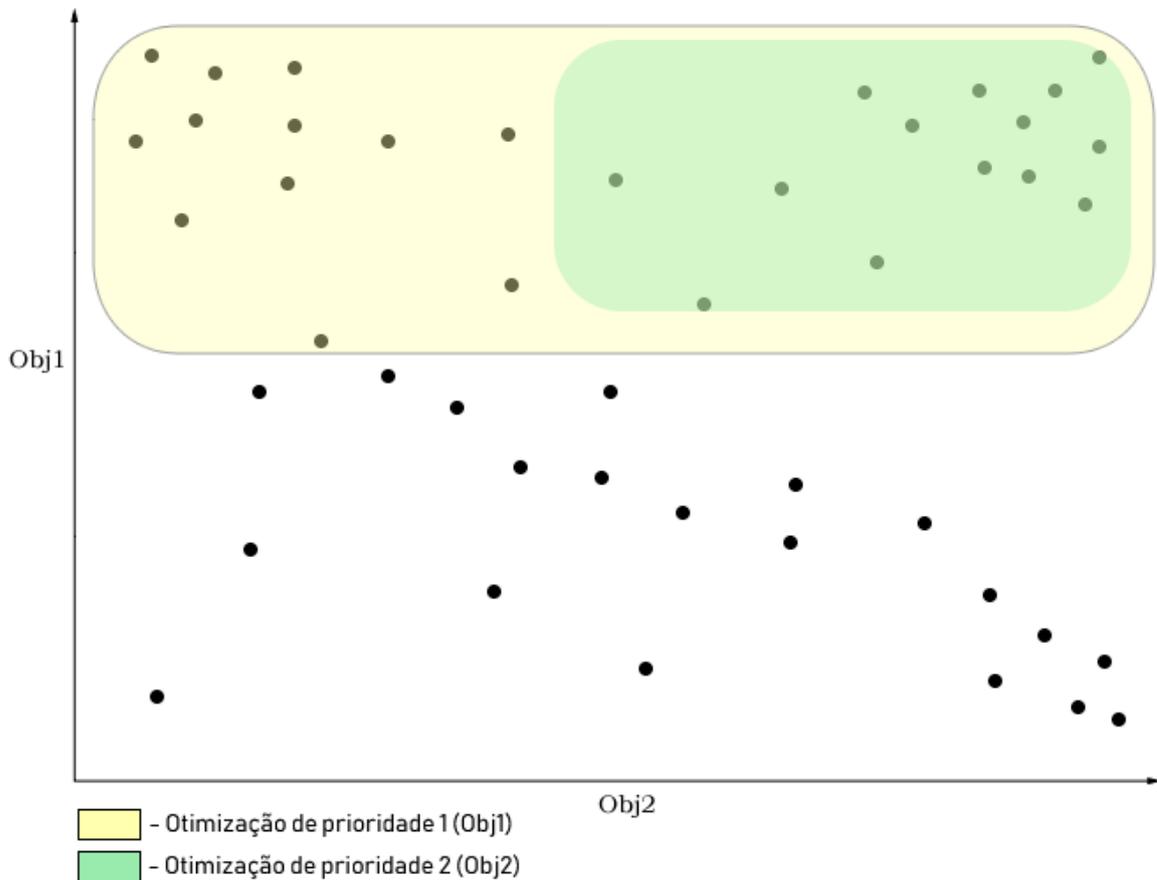


Figura 24 – Exemplificação do processo de busca e otimização realizado pelo [HC-LGA](#).

A ideia básica implementada pelo [HC-LGA](#) é a de encontrar uma região interessante no espaço de busca que apresente soluções ótimas para o objetivo de maior prioridade, e então dentro dessa região, sem que haja o deterioramento do objetivo já otimizado, encontrar uma região composta de soluções que também otimizem o próximo objetivo de maior prioridade, repetindo esse processo até que todos os objetivos definidos também sejam otimizados. Em outras palavras, primeiro realiza-se uma operação de *exploração* num espaço de busca mais amplo e, assim que uma região interessante é encontrada, inicia-se um processo de *exploração* dessa região. No exemplo da Figura 24, a área em amarelo representa justamente a região hipotética encontrada, e que contém soluções ótimas para o objetivo mais relevante, fruto de um processo de exploração num amplo espaço de soluções candidatas. Já a área em verde representa o processo de exploração da região previamente encontrada, contendo soluções que não só otimizam o segundo objetivo na ordem de importância, mas que otimizam ambos os objetivos.

# Capítulo 7

## METODOLOGIA

---

---

### 7.1 Conjunto de Dados

Atualmente, existe uma variedade pequena de conjuntos de dados contendo instâncias de **TEs**, e estas na maioria das vezes, não se encontram arranjados de acordo com alguma taxonomia hierárquica própria para sua classificação nem são formatados para serem usados por algoritmos de **ML**. Apesar de uma iniciativa dessa natureza ter sido feita pelo repositório PGSB, reunindo diferentes sequências oriundas de outras bases de dados como o TREP<sup>1</sup> e TIGR<sup>2</sup> e organizando-as de maneira hierárquica, a questão da formatação dos dados ainda permanece.

A primeira etapa dessa pesquisa foi justamente a extração e formatação dos dados de **TEs**, aos quais foram publicamente disponibilizados<sup>3</sup>. Os dois conjuntos de dados utilizados nesse trabalho tratam-se de sequências coletadas de duas bases de dados públicas: (1) o PGSB<sup>4</sup> que contém sequências de repetições de plantas; e o (2) REPBASE<sup>5</sup>, que armazena sequências de repetições de DNA oriundos de diferentes organismos eucariotos. Ambos os repositórios têm sido amplamente utilizados em projetos no mundo todo, o que os torna bases de dados de referência. Cabe ressaltar ainda que não há intersecção entre os dados desses dois repositórios.

Após coletadas as sequências de **TEs**, algoritmos de extração de características foram aplicados a fim de construir conjuntos de dados adequados a algoritmos de **ML**. As características em questão foram contagens de  $k$ -mers nas sequências, um atributo popular usado em ferramentas de Bioinformática (**MELSTED; PRITCHARD, 2011**). Nos conjuntos de dados criados, cada instância de **TE** é representada por um conjunto de  $k$ -mers obtidos por contagem de sub-sequências de tamanho  $k$  nas sequências originais de nucleotídeos. Foram definidos diferentes tamanhos para  $k$ , como sendo 2, 3 e 4, e então buscou-se por todas as combinações das bases nitrogenadas Adenina (*A*), Timina (*T*), Guanina (*G*) e Citosina (*C*), de acordo com o valor de  $k$ . Por exemplo, para  $k = 2$ , os atributos são as contagens de *AA*, *AC*, *AG*, *AT*, *CA*, *CG*, *CT*, *CC*,

<sup>1</sup> <http://wheat.pw.usda.gov/ITMI/Repeats/> (acessado em 15 de maio de 2016).

<sup>2</sup> <http://plantrepeats.plantbiology.msu.edu/> (acessado em 15 de maio de 2016).

<sup>3</sup> <https://github.com/geantrindade/TEsHierarchicalDatasets>

<sup>4</sup> <http://pgsb.helmholtz-muenchen.de/plant/> (acessado em 15 de maio de 2016).

<sup>5</sup> <http://www.girinst.org/replib/> (acessado em 15 de maio de 2016).

e assim por diante, até *TT*. A Tabela 4 apresenta algumas estatísticas a respeito dos dois conjuntos de dados construídos, cada um oriundo de uma base de dados. Ambos PGSB e REPBASE possuem o mesmo número de atributos e níveis hierárquicos (quatro níveis), porém o REPBASE é um conjunto de dados mais complexo contendo quase o dobro de instâncias, maior número de classes nos níveis inferiores e mais que o dobro de classes folha.

Tabela 4 – Estatísticas a respeito dos conjuntos de dados.

Conjunto de dados	Atributos	Instâncias	Classes por nível	Classes folha
PGSB	336	18678	2 / 4 / 3 / 5	11
REPBASE	336	34559	2 / 5 / 12 / 9	24

A fim de representar/codificar as classes de **TEs** presentes nos dados, foi utilizado um esquema que considera a taxonomia de [Wicker et al. \(2007\)](#) para anotá-las, conforme apresentado na Tabela 5. Cada classe recebeu uma etiqueta numerada que representa seu local na hierarquia. Dessa forma, cada **TE** tem como representação de sua classe a combinação de etiquetas correspondente, respeitando as restrições da hierarquia de classes. Como exemplo, se um dado elemento pertence a classe **hAT**, este recebe a etiqueta **2.1.1.2**, pois segundo esse esquema: **Transposon** (2) > **Subclasse 1**(1) > **TIR**(1) > **hAT**(2). Cabe ressaltar que os dados extraídos das bases PSGB e REPBASE não contemplam toda a taxonomia de [Wicker et al. \(2007\)](#), portanto nem todas as classes de **TEs** estão representadas nos conjuntos de dados.

Tabela 5 – Esquema numérico de etiquetas aplicado a hierarquia de classes.

	Retrotransposons(1) [15998]	DNA Transposons(2) [2680]
PGSB	LTR(1.1) [15336]	Subclass1(2.1) [2680]
	Copia(1.1.1) [4279]	TIR(2.1.1) [1600]
	Gypsy(1.1.2) [7625]	Tc1-Mariner(2.1.1.1) [356]
	LINE (1.4) [471]	hAT(2.1.1.2) [63]
	SINE (1.5) [191]	Mutator(2.1.1.3) [320]
		PIF-Harbinger(2.1.1.8) [141]
	CACTA(2.1.1.9) [720]	
	Retrotransposons(1) [22414]	DNA Transposons(2) [12145]
REPBASE	LTR(1.1) [18768]	Subclass1(2.1) [12145]
	Copia(1.1.1) [6313]	TIR(2.1.1) [7903]
	Gypsy(1.1.2) [10068]	Tc1-Mariner(2.1.1.1) [2351]
	Bel-Pao(1.1.3) [1827]	hAT(2.1.1.2) [2437]
	DIRS(1.2) [374]	Mutator(2.1.1.3) [735]
	LINE (1.4) [2516]	Merlin(2.1.1.4) [73]
	R2(1.4.1) [78]	Transib(2.1.1.5) [123]
	RTE(1.4.2) [439]	P(2.1.1.6) [376]
	Jockey(1.4.3) [242]	Piggybac(2.1.1.7) [353]
	L1(1.4.4) [1566]	PIF-Harbinger(2.1.1.8) [874]
	I(1.4.5) [194]	CACTA(2.1.1.9) [581]
	SINE(1.5) [753]	
	tRNA(1.5.1) [505]	
	7SL(1.5.2) [95]	
5S(1.5.3) [29]		

## 7.2 Métodos Comparados

Visto que os conjuntos de dados utilizados nessa pesquisa foram criados pelo mesmo grupo de pesquisa em que o autor desse documento está incluído, o número de trabalhos da literatura que os utilizaram ainda é pequeno. Portanto, apenas os trabalhos já publicados que utilizaram tais dados foram usados nos comparativos, sendo estes apresentados na Tabela 6. A intenção da inclusão desses métodos fora a de possuir exemplos de cada tipo de abordagem e estratégia hierárquica, assim como uma ferramenta de Bioinformática que não utiliza de ML e, deste modo, ter-se uma maior cobertura e credibilidade nos experimentos realizados.

Tabela 6 – Relação de métodos da literatura usados nos comparativos.

Método	Artigo da Literatura	Abordagem/Estratégia
BLASTn	Altschul et al. (1990), Nakano (2018)	Homologia
RepeatMasker	Smit et al. (1996), Nakano (2018)	Homologia
LCN-DT	Santos et al. (2018)	LCN
LCN-RF	Santos et al. (2018)	LCN
LCN-NB	Santos et al. (2018)	LCN
LCN-KNN	Santos et al. (2018)	LCN
LCN-MLP	Santos et al. (2018)	LCN
LCN-SVM	Santos et al. (2018)	LCN
nLLCPN	Nakano et al. (2017)	LCPN
LCPNB	Nakano et al. (2017)	LCPN
LCL+SWV	Nakano et al. (2017)	LCL
LCL+SP	Nakano et al. (2017)	LCL
Clus-HMC	Vens et al. (2008), Pereira et al. (2018)	Global

## 7.3 Treinamento e Teste

No que se remete ao treinamento e teste de classificadores, este não difere muito nos trabalhos da literatura. O mais comum é a utilização de um processo de validação cruzada com  $k$ -folds para treino e teste, ou ainda, se opta pela divisão do conjunto de dados em porções para treino e teste, podendo haver também uma para validação. Essa segunda opção de particionamento é mais utilizada quando se têm poucos dados a disposição. Como dispõem-se de uma quantidade razoável de dados e, visando se ter resultados com maior significância estatística, optou-se pela utilização da validação cruzada com  $k = 10$  (valor mais comum) para treino e teste.

Quanto ao processo de ajuste dos hiperparâmetros de um classificador, as opções são mais variadas. A abordagem mais simples e tradicional é a realização de uma “pesquisa em grade”, em que um conjunto de valores dentro do espaço dos hiperparâmetros são testados de forma exaustiva a fim de encontrar uma combinação ótima. Uma variação dessa abordagem é a chamada “pesquisa randômica”, onde em vez de selecionar manualmente os valores a serem testados para cada hiperparâmetro, estes são escolhidos aleatoriamente, o que tende a gerar

melhores resultados (BERGSTRA; BENGIO, 2012). A otimização Bayesiana é outra opção para o ajuste, sendo está um método de otimização global baseado em um modelo probabilístico que realiza um mapeamento dos valores de hiperparâmetro a um objetivo, como a maximização da acurácia de um algoritmo (CLAESEN; MOOR, 2015). Apesar dessas duas últimas abordagens apresentarem melhores otimizações quando exploradas de forma adequada, demandam mais tempo de investigação. Portanto, considerando as limitações de tempo dessa pesquisa, optou-se pela utilização de uma *pesquisa em grade* para a otimização de alguns hiperparâmetros. Os valores finais assim como os valores testados são mostrado na Tabela 7, sendo estes utilizados, sem quaisquer alterações, nos três métodos propostos.

Tabela 7 – Hiperparâmetros utilizados pelos métodos HC-GA, HC-WGA e HC-LGA.

Parâmetro	Valor Final	Variações Testadas
Gerações	100	25 / 50 / 100
População	100	25 / 50 / 100
Min. Instâncias Cobertas por Regra	Instâncias da Classe Minoritária	10 / 20 / Instâncias da Classe Minoritária
Max. Instâncias Cobertas por Regra	Instâncias da Classe Majoritária	500 / 1000 / Instâncias da Classe Majoritária
Max. Instâncias Não Cobertas	10	10
Taxa de Cruzamento	90%	10% / 30% / 90%
Taxa de Mutação	30%	10% / 30% / 90%
Tamanho do Torneio	3	3
Elitismo	2	2 / 3 / 4
Prob. de usar um Teste na Regra	3%	1% / 3% / 9%

Mesmo que esses hiperparâmetros finais tenham sido obtidos via ajuste fino, alguns deles foram inicialmente definidos com base em algumas hipóteses que se mostraram verdadeiras após o ajuste. Um valor mais alto para o tamanho da população ajudou os métodos a encontrar regras melhores já que mais candidatos a boas regras estavam disponíveis. Com relação à alta taxa de cruzamento, foi observado que esta diminui a perda de boas regras e dificulta a permanência em ótimos locais, pois torna a busca mais esparsa e, com a combinação do elitismo, impede que se torne aleatória ou seja direcionada a regiões pouco promissoras. Também visando evitar ótimos locais, foram definidos intervalos maiores para os hiperparâmetros *mínimo* e *máximo* número de instâncias cobertas por regra utilizados na busca local (ver Algoritmo 5), caso contrário a busca seria limitada, o que poderia comprometer a generalização e/ou interpretabilidade do modelo gerado. A utilização de um intervalo curto entre os limites pode levar a regras muito específicas e, conseqüentemente, sem informações úteis acerca dos dados.

Em relação à taxa de elitismo e o tamanho do torneio, optou-se por mantê-los baixos a fim de evitar uma convergência prematura e a perda de variabilidade. Sobre a probabilidade de usar um teste em uma regra, um valor baixo foi definido já que muitos testes ativos tornariam uma regra muito específica ou, de modo contrário, muito geral. Essa mesma lógica foi aplicada à taxa de mutação. Assim, definimos valores moderados a esses hiperparâmetros o que fez com que os GAs encontrassem novas boas soluções no espaço de decisão em menores tempos de treinamento. Finalmente, o número de gerações foi definido de acordo com uma análise de ganho versus tempo de execução, considerando as limitações já mencionadas.

Outros hiperparâmetros otimizados têm relação exclusivamente com o método lexicográfico **HC-LGA**, sendo mais precisamente os limiares para comparação de valores das funções objetivo. A partir de observações prévias nas saídas das funções testadas (ver Seção 8.4) e tendo conhecimento de seus limites inferiores e superiores, definiu-se os valores de limiar igual à 0.05 para as funções objetivo **hP**, **hR**, **hF** e  $AU(\overline{PRC})$  e 0.005 para **PC** e **VG**.

## 7.4 Medidas de Avaliação

Problemas de **HC** requerem medidas de avaliação específicas para que uma análise adequada dos resultados seja feita. Isso se dá pelo fato de que as medidas tradicionais usadas em problemas de **FC** não consideram as predições feitas em vários níveis de uma hierarquia. Duas medidas populares em trabalhos de **HC**, oriundas de adaptações em medidas tradicionais, são a Precisão Hierárquica (no inglês, *hierarchical Precision* (**hP**)) e a Revocação Hierárquica (no inglês, *hierarchical Recall* (**hR**)) propostas por Kiritchenko et al. (2005), e apresentadas nas Equações 7.1 e 7.2. Ainda, existe uma medida que combina as citadas anteriormente, chamada Medida-F Hierárquica (no inglês, *hierarchical F-measure* (**hF**)), apresentada na Equação 7.3.

$$hP = \frac{\sum_i |\hat{Z}_i \cap \hat{C}_i|}{\sum_i |\hat{Z}_i|} \quad (7.1)$$

$$hR = \frac{\sum_i |\hat{Z}_i \cap \hat{C}_i|}{\sum_i |\hat{C}_i|} \quad (7.2)$$

$$hF = \frac{2 \cdot hP \cdot hR}{hP + hR} \quad (7.3)$$

Essas medidas de avaliação consideram que uma instâncias não pertence somente a sua classe mais específica, mas também a todos as suas super classes na hierarquia. Nessas equações,  $\hat{Z}_i$  e  $\hat{C}_i$  representam o conjunto de classes preditas e reais para uma instância  $i$ , respectivamente, incluindo todos as classes ancestrais nesses conjuntos.

## 7.5 Testes Estatísticos

Testes estatísticos vêm recebendo cada vez mais atenção da comunidade de **ML** como instrumentos sólidos para validação de experimentos (NAKANO et al., 2017; CERRI et al., 2012). No contexto de **ML**, um teste estatístico têm como ideia básica a comparação de resultados oriundos de diferentes métodos aplicados a um ou mais conjuntos de dados, de modo a verificar se esses resultados diferem uns dos outros com certa significância estatística (DEMŠAR, 2006).

Existe uma variedade de testes estatísticos tanto paramétricos quanto não paramétricos e que são adequados a diferentes propósitos, como comparar resultados de diversos métodos

aplicados a muitos conjuntos de dados, ou ainda, comparar apenas duas amostras de resultados levando em consideração um único conjunto de dados. Um exemplo de teste não paramétrico é o teste de Wilcoxon ou Teste dos Postos Sinalizados de Wilcoxon (do inglês, *Wilcoxon signed-rank test*) (WILCOXON, 1945), onde apenas um valor  $\alpha$  de significância é utilizado e que geralmente é definido como  $\alpha = 5\%$ . O teste de Wilcoxon foi escolhido para validar os resultados dessa pesquisa devido a sua simplicidade, já que não usa de um conjunto de parâmetros, o que demandaria um conhecimento estatístico mais aprofundado para configurá-lo de maneira adequada, e também pelo fato de ser recomendado para a comparação de resultados entre dois ou mais métodos aplicados a poucas bases de dados, como é o caso desse trabalho.

## 7.6 Testes de Correlação

Testes de correlação visam avaliar a associação entre duas ou mais variáveis, verificando se existe ou não uma correlação entre elas além de determinam a “força” e sentido da correlação. A maioria dos testes de correlação operam somente em determinadas distribuições de dados, como é o caso da correlação de Pearson que é aplicada a uma distribuição normal/gaussiana. Portanto, é importante verificar o tipo de distribuição antes de aplicar um teste de correlação.

Um teste que pode ser usado para verificar se uma amostra segue uma distribuição normal é o chamado teste de Shapiro-Wilk (SHAPIRO; WILK, 1965). Sua hipótese nula é que os dados são normalmente distribuídos, enquanto que sua hipótese alternativa é que os dados não o são. A hipótese nula é um fato comumente aceito, sendo o oposto da hipótese alternativa. Em uma pesquisa científica, pesquisadores trabalham para rejeitar a hipótese nula, de forma que apresentam uma hipótese alternativa que visa explicar o fenômeno no caso da hipótese nula ser rejeitada. Deste modo, caso o *valor p* ou *nível de significância estatística* for menor que o nível  $\alpha$  escolhido, a hipótese nula é rejeitada e há evidências de que os dados testados não são oriundos de uma população com distribuição normal. Por outro lado, se o valor *p* for maior ou igual à  $\alpha$ , a hipótese nula não pode ser rejeitada e portanto se assume a normalidade dos dados.

Dois testes populares para a análise da correlação de duas variáveis são os testes de Pearson e Spearman, que além de determinar se estas são correlacionadas também indicam a força do relacionamento (coeficiente de correlação). Assim como no teste de Shapiro-Wilk, para cada uma dessas correlações são utilizados valores  $\alpha$  de significância que devem ser satisfeitas para que as respectivas hipóteses nulas sejam rejeitadas. É importante ressaltar que a significância estatística não indica a força da correlação. De fato, o teste de significância na correlação não fornece nenhuma informação acerca da força do relacionamento. Assim, alcançar um valor de  $p = 0.001$  não significa que o relacionamento seja mais forte do que quando se tem  $p = 0.04$ . Isso ocorre porque o teste de significância está interessado em investigar se é possível rejeitar ou não a hipótese nula. O valor normalmente utilizado é  $\alpha = 0.05$ , que se satisfeito ( $p < 5\%$ ) indica que há confiança de 95% na existência da correlação.

Nas correlações de Pearson e Spearman seus coeficientes de correlação variam entre  $[-1, 1]$ . Logo, quanto mais próximo de  $-1$ , mais forte é a correlação negativa entre as variáveis o que significa que quando uma variável  $X$  cresce, a variável dependente  $Y$  decresce e vice-versa. Quanto mais próximo de  $1$  for o coeficiente de correlação, mais forte é a correlação positiva, indicando que quando  $X$  cresce  $Y$  também cresce e vice-versa. Há também a possibilidade do valor de correlação ser zero, sugerindo assim a não existência de associação entre as variáveis.

A correlação de Pearson mede a dependência linear entre duas variáveis  $X$  e  $Y$ , sendo considerado um teste de correlação paramétrico porque depende da distribuição dos dados e, portanto, pode ser utilizado somente quando  $X$  e  $Y$  são oriundos de uma distribuição normal (PEARSON, 1895). Quando as amostras não seguem uma distribuição normal, recomenda-se a utilização de uma correlação não paramétrica como a correlação de Spearman. O coeficiente de correlação de Spearman é uma medida estatística da força de uma relação monotônica entre duas variáveis (SPEARMAN, 1904). Uma função monotônica é aquela que jamais cresce ou decresce à medida que sua variável independente cresce. A Figura 25 ilustra alguns padrões de funções desse tipo. Em (a) tem-se um valor de função monotonicamente aumentando, visto que à medida em que a variável  $X$  aumenta, a variável  $Y$  jamais diminui. Já em (b) é apresentada uma função monotonicamente diminuindo, pois à medida em que  $X$  aumenta,  $Y$  jamais aumenta. Por fim, em (c) é mostrado um exemplo de padrão não monotônico, já que à medida em que a variável  $X$  aumenta, em certos pontos a variável  $Y$  diminui e em outros ela aumenta.

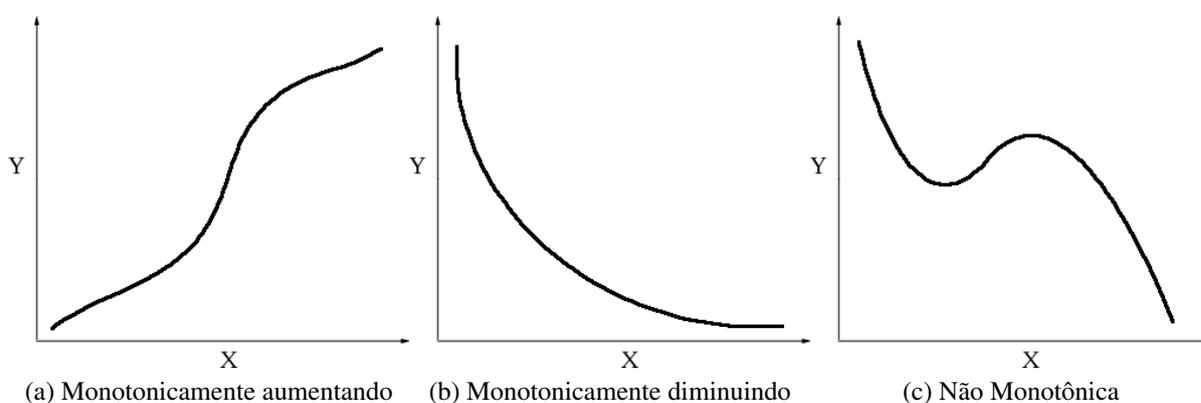


Figura 25 – Padrões de funções monotônicas.

A hipótese nula para a correlação de Pearson é: não existe associação linear entre as duas variáveis da amostra. Já a hipótese nula para a correlação de Spearman é: não há associação *monotônica* entre as duas variáveis da amostra. Nesse trabalho as correlações de Pearson e Spearman foram ambas utilizadas nas amostras dos valores de objetivo resultantes, variando-as de acordo com os padrões de distribuição observados.

# Capítulo 8

## EXPERIMENTOS E DISCUSSÕES

---

Neste capítulo são apresentados e discutidos os experimentos com os três métodos propostos utilizando a taxonomia hierárquica e dados de **TEs** previamente apresentados. Foram realizadas diferentes avaliações como a coleta e comparações dos tempos de treinamento dos métodos, análises acerca das predições tanto de forma geral usando toda a hierarquia de classes quanto análises mais elaboradas como por nível e nas classes folha. Além disso, como um dos focos dessa pesquisa é a construção de modelos mais interpretáveis, análises a esse respeito também foram feitas. Para as análises de desempenho preditivo por nível, foram consideradas as classes reais e preditas para dado nível incluindo, naturalmente, as superclasses. Já nas análises para as predições das classes folha, foram comparadas apenas as classes folhas reais e preditas, podendo essas serem classes mais específicas situadas em qualquer nível da hierarquia.

Quanto aos resultados presentes nas Tabelas 8, 11 e 14, esses correspondem as médias da Medida-F Hierárquica juntamente com os respectivos desvios padrão, oriundos das execuções usando validação cruzada com 10-folds. Vale ressaltar que os campos contendo “-”, indicam que o método não foi capaz de realizar predições e, no caso da análise das correlações presentes na Tabela 18, esses campos indicam que não foram feitos experimentos com tais objetivos.

### 8.1 Experimentos com o HC-GA

Nessa seção são apresentados os experimentos com o método hierárquico global **HC-GA**. Como trata-se de um **GA** de otimização simples, apenas funções de aptidão isoladas foram executadas. Basicamente, cada função de aptidão apresentada na Seção 6.1.4, endereça um dos objetivos definidos nessa pesquisa de mestrado, sendo esses o desempenho preditivo ou a interpretabilidade. O primeiro grupo de funções, as quais visam maximizar o desempenho, é composto por **hP**, **hR**, **hF** e  $AU(\overline{PRC})$ , enquanto que **Percentage Coverage (PC)** e **Variance Gain (VG)** compõem o segundo grupo que maximiza a interpretabilidade. A Tabela 8 apresenta os resultados gerados por cada uma dessas funções.

Conforme é visto na Tabela 8, as funções do primeiro grupo mostraram resultados muito superiores em ambos os conjuntos de dados quando comparados aos obtidos pelas funções

Tabela 8 – Análise das predições do HC-GA considerando a Medida-F Hierárquica.

Fitness	Nível 1	Nível 2	Nível 3	Nível 4	Classes Folha	Geral
PGSB						
<i>hP</i>	<b>0.95 ± 0.006</b>	<b>0.92 ± 0.009</b>	<b>0.78 ± 0.013</b>	<b>0.46 ± 0.055</b>	<b>0.84 ± 0.011</b>	<b>0.88 ± 0.008</b>
<i>hR</i>	0.94 ± 0.005	0.91 ± 0.007	0.77 ± 0.008	0.41 ± 0.047	0.83 ± 0.007	0.87 ± 0.005
<i>hF</i>	0.93 ± 0.005	0.91 ± 0.006	0.77 ± 0.011	0.43 ± 0.056	0.83 ± 0.009	0.87 ± 0.007
$AU(\overline{PRC})$	0.93 ± 0.005	0.90 ± 0.006	0.73 ± 0.012	0.24 ± 0.091	0.80 ± 0.008	0.85 ± 0.007
<i>PC</i>	0.54 ± 0.239	0.53 ± 0.230	0.08 ± 0.162	-	0.53 ± 0.085	0.43 ± 0.208
<i>VG</i>	0.90 ± 0.016	0.88 ± 0.011	0.63 ± 0.084	0.07 ± 0.043	0.74 ± 0.041	0.81 ± 0.016
REPBASE						
<i>hP</i>	<b>0.90 ± 0.005</b>	<b>0.86 ± 0.004</b>	<b>0.68 ± 0.008</b>	0.35 ± 0.015	<b>0.71 ± 0.007</b>	<b>0.79 ± 0.004</b>
<i>hR</i>	0.89 ± 0.007	0.85 ± 0.007	<b>0.68 ± 0.007</b>	<b>0.36 ± 0.026</b>	0.69 ± 0.007	0.78 ± 0.006
<i>hF</i>	0.89 ± 0.007	0.85 ± 0.008	<b>0.68 ± 0.008</b>	<b>0.36 ± 0.019</b>	0.69 ± 0.007	0.78 ± 0.007
$AU(\overline{PRC})$	0.86 ± 0.012	0.77 ± 0.020	0.40 ± 0.098	0.24 ± 0.024	0.45 ± 0.075	0.68 ± 0.031
<i>PC</i>	0.27 ± 0.153	0.22 ± 0.137	-	-	0.17 ± 0.051	0.18 ± 0.109
<i>VG</i>	0.81 ± 0.045	0.74 ± 0.043	0.43 ± 0.083	0.01 ± 0.011	0.50 ± 0.073	0.65 ± 0.033

do segundo grupo. Dentre as melhores performances então as das funções *hP*, *hR* e *hF*, que mostraram valores bem próximos umas das outras. Nos dois conjuntos de dados, a função *hP* obteve os melhores valores para todos os critérios avaliados, com exceção para as predições do *Nível 4* no *REPBASE*, no qual as funções *hR* e *hF* obtiveram melhores valores. A outra função desse grupo,  $AU(\overline{PRC})$ , mostrou resultados inferiores em comparação aos demais, apresentando inclusive valores de desvio padrão mais altos no *REPBASE*.

Quanto às funções *PC* e *VG*, embora superadas pelas funções do primeiro grupo, contribuíram para observações importantes acerca das dificuldades de se construir modelos interpretáveis com bom desempenho preditivo em *HC*. A função *VG* gerou resultados promissores e comparáveis ao de  $AU(\overline{PRC})$ , e considerando que é voltada a interpretabilidade, pode-se dizer que constitui uma opção interessante por apresentar um bom equilíbrio entre os dois objetivos. No entanto, o mesmo não ocorreu com *PC*, que obteve os piores resultados entre todas as funções testadas e foi a única incapaz de prever instâncias em alguns níveis mais baixos da hierarquia. Ademais, *PC* mostrou uma alta variação de desempenho quando levado em consideração seus valores de desvio padrão. Uma função simplória como *PC* tem dificuldade em capturar muitas das características chave que distinguem as instâncias de níveis mais baixos. Isso pode ser um indício que funções muito simples podem não ser boas alternativas no contexto hierárquico, entretanto, mais investigações devem ser feitas antes de assumir tal hipótese como verdadeira.

Outra análise realizada e que diz respeito à interpretabilidade dos modelos de regras gerados por cada função de aptidão, é mostrada na Tabela 9. Nela são comparadas algumas características observadas nos modelos finais, como o número de regras (**Regras**), o número de testes ativos em toda a lista de regras (**Testes Ativos**) e a média de testes por regra (**Testes por Regra**). Os valores apresentados correspondem as respectivas médias e desvios padrão.

No comparativo visto na Tabela 9, as funções *VG* e *PC* obtiveram os modelos mais

Tabela 9 – Análise de interpretabilidade das regras geradas pelo HC-GA.

Fitness	Regras	Testes Ativos	Testes por Regra	PGSB		REPBASE	
				Regras	Testes Ativos	Testes por Regra	Regras
<i>hP</i>	1180 ± 15.47	8749 ± 137.2	7 ± 0.080	2507 ± 27.88	19889 ± 300.8	8 ± 0.051	
<i>hR</i>	1154 ± 18.78	8376 ± 117.3	7 ± 0.089	2424 ± 22.72	18914 ± 165.0	8 ± 0.043	
<i>hF</i>	1156 ± 18.12	8366 ± 175.5	7 ± 0.075	2411 ± 34.80	18676 ± 315.0	8 ± 0.047	
$AU(\overline{PRC})$	367 ± 13.63	2257 ± 108.0	6 ± 0.175	204 ± 43.91	1251 ± 306.2	6 ± 0.298	
<i>PC</i>	<b>4 ± 0.699</b>	<b>15 ± 2.898</b>	<b>4 ± 0.894</b>	<b>4 ± 0.000</b>	<b>23 ± 4.442</b>	6 ± 1.111	
<i>VG</i>	23 ± 6.683	117 ± 29.05	5 ± 0.840	27 ± 8.044	143 ± 52.95	<b>5 ± 0.596</b>	

interpretáveis e com os menores desvios padrão, sendo **PC** a melhor em todos os critérios avaliados e em ambos os conjuntos de dados. Acreditamos que os maiores valores de desvio padrão dessas funções no critério *testes por regra*, que foi a exceção, deu-se em razão das amostras serem menores, já que nos demais critérios as funções voltadas ao desempenho preditivo mostraram desvios padrão significativamente maiores, principalmente para o REPBASE. Uma conclusão importante acerca dessa análise é que, juntamente com a apresentada na Tabela 8, observou-se que a função **VG** representa o melhor equilíbrio entre os objetivos desempenho e interpretabilidade, reforçando o que já havia sido levantado. Quanto às funções **hP**, **hR** e **hF** que alcançaram os melhores valores na análise anterior, em contraste, mostraram os piores resultados em relação a interpretabilidade dos modelos, obtendo valores muito inferiores aos demais. Isso contribui para a afirmação de que esses objetivos são conflitantes e do quão árduo é gerar modelos com bom desempenho e interpretabilidade. Por fim, na Tabela 10 é realizado um comparativo das médias do tempo de treinamento requerido por cada *fitness* para a geração de uma lista final de regras, juntamente com os respectivos valores de desvio padrão.

Tabela 10 – Tempos de treinamento do HC-GA em minutos.

Fitness	PGSB	REPBASE
<i>hP</i>	63 ± 8.458	1371 ± 141.9
<i>hR</i>	110 ± 24.64	1158 ± 237.2
<i>hF</i>	63 ± 17.19	735 ± 96.90
$AU(\overline{PRC})$	66 ± 4.513	114 ± 28.97
<i>PC</i>	<b>6 ± 2.063</b>	<b>33 ± 3.044</b>
<i>VG</i>	9 ± 2.240	59 ± 14.46

Conforme a Tabela 10, **PC** obteve os menores tempos e desvios padrão para ambos os conjuntos de dados, seguida por **VG** que mostrou resultados competitivos. Como essas funções são orientadas a interpretabilidade, o que as faz gerar menos regras como visto na Tabela 9 e, portanto, regras com mais abrangência, consequentemente o tempo de treinamento necessário para a cobertura de todo um conjunto de instâncias é menor. Já as funções **hP**, **hR**, **hF** e  $AU(\overline{PRC})$  voltadas a otimização do desempenho, obtiveram valores consideravelmente mais elevados justamente por gerarem mais regras. Considerando os dois conjuntos de dados, tem-se **hP** como o pior resultado, seguido de **hR**, **hF** e  $AU(\overline{PRC})$ . Outro ponto observado é que essas

funções além de apresentarem os maiores tempos de treinamento, têm os maiores desvios padrão, que aumentam drasticamente do PGSB para o REPBASE. Isso evidencia uma certa instabilidade principalmente quando considerado conjuntos de dados maiores.

## 8.2 Experimentos com o HC-WGA

Nessa seção são apresentados e discutidos os experimentos com o método hierárquico global de otimização multiobjetivo usando uma Abordagem de Soma Ponderada, denominado **HC-WGA**. Considerando as funções implementadas e testadas isoladamente no **HC-GA**, onde cada uma endereça um dos objetivos alvo dessa pesquisa (desempenho preditivo e interpretabilidade), partiu-se da ideia de combiná-las utilizando um conjunto de pesos distintos e, dessa forma, encontrar soluções que otimizam ambos os objetivos. Portanto, do grupo de funções focado no desempenho utilizou-se **hF** e  $AU(\overline{PRC})$  enquanto que do grupo da interpretabilidade **PC** e **VG** foram escolhidas. A Tabela 11 apresenta as médias da Medida-F Hierárquica obtidas por cada função ponderada assim como os seus respectivos desvios padrão.

Dado o alto número de combinações de funções e pesos associados (executados com validação cruzada de 10-folds), em prol de simplificar os experimentos, optou-se por utilizar apenas a função **hF** em vez de também incluir **hP** e **hR**, já que esta combina esses valores como uma média harmônica. Além disso, como não haveria sentido combinar duas funções do mesmo grupo, já que a intenção desses experimentos é a otimização dos dois objetivos conflitantes e não a exploração de um só, não realizou-se uma combinação de todos com todos. Em vez disso, testou-se apenas as quatro combinações possíveis entre as funções supracitadas, utilizando cinco variações de pesos para cada combinação, sendo essas:  $0.2 + 0.8$ ,  $0.4 + 0.6$ ,  $0.5 + 0.5$ ,  $0.6 + 0.4$  e  $0.8 + 0.2$ . Esses valores representam porcentagens, desse modo, no caso de  $0.8 \cdot hF + 0.2 \cdot PC$ , o valor final da função ponderada será composto pelo valor de **hF** que representa 80% do somatório, somado ao valor de **PC** que corresponde a 20%.

Se destacando das demais execuções vistas na Tabela 11, os experimentos com a função ponderada **hF** + **VG** obtiveram os melhores resultados em ambos os conjuntos de dados, seguida por **hF** (80%) + **PC** (20%) com resultados comparáveis. Um dado interessante visto nessa tabela e que também ocorre nos resultados da Tabela 8, é que as execuções consideradas como as melhores apresentaram os valores mais altos para os níveis superiores da hierarquia, especialmente para o Nível 1. Essa relação faz sentido e tem um embasamento teórico visto que, em **HC**, erros cometidos por classificadores em níveis mais próximos da raiz são propagados pela hierarquia, degradando a qualidade das predições. Logo, apresentar altos índices de desempenho preditivo nos primeiros níveis evita que os erros cresçam conforme as predições se tornam mais específicas, o que conseqüentemente reflete na predição geral e/ou em casos mais específicos como a predição de classes folha. Outras execuções que também poderiam ser consideradas como candidatas a soluções ótimas, já que atingiram resultados comparáveis no PGSB, foram  $AU(\overline{PRC})$  (80%)

Tabela 11 – Análise das previsões do HC-WGA considerando a Medida-F Hierárquica.

Fitness	Ponderação	Nível 1	Nível 2	Nível 3	Nível 4	Classes Folha	Geral
PGSB							
<i>HF + PC</i>	20% + 80%	0.50 ± 0.169	0.49 ± 0.168	0.10 ± 0.100	-	0.61 ± 0.100	0.40 ± 0.135
	40% + 60%	0.61 ± 0.140	0.59 ± 0.145	0.14 ± 0.080	-	0.61 ± 0.075	0.50 ± 0.121
	50% + 50%	0.58 ± 0.174	0.55 ± 0.150	0.20 ± 0.096	-	0.64 ± 0.076	0.47 ± 0.134
	60% + 40%	0.62 ± 0.172	0.59 ± 0.164	0.18 ± 0.125	0.01 ± 0.032	0.58 ± 0.147	0.50 ± 0.143
	80% + 20%	<b>0.94 ± 0.006</b>	0.91 ± 0.006	<b>0.77 ± 0.012</b>	0.42 ± 0.050	<b>0.83 ± 0.008</b>	<b>0.87 ± 0.006</b>
<i>HF + VG</i>	20% + 80%	<b>0.94 ± 0.005</b>	0.91 ± 0.005	<b>0.77 ± 0.011</b>	0.41 ± 0.047	<b>0.83 ± 0.007</b>	<b>0.87 ± 0.006</b>
	40% + 60%	<b>0.94 ± 0.005</b>	0.91 ± 0.006	<b>0.77 ± 0.011</b>	0.41 ± 0.037	<b>0.83 ± 0.005</b>	<b>0.87 ± 0.006</b>
	50% + 50%	<b>0.94 ± 0.003</b>	<b>0.92 ± 0.006</b>	<b>0.77 ± 0.008</b>	<b>0.44 ± 0.037</b>	<b>0.83 ± 0.006</b>	<b>0.87 ± 0.005</b>
	60% + 40%	<b>0.94 ± 0.003</b>	0.91 ± 0.005	<b>0.77 ± 0.008</b>	0.43 ± 0.047	<b>0.83 ± 0.007</b>	<b>0.87 ± 0.004</b>
	80% + 20%	<b>0.94 ± 0.006</b>	<b>0.92 ± 0.007</b>	<b>0.77 ± 0.007</b>	0.43 ± 0.052	<b>0.83 ± 0.006</b>	<b>0.87 ± 0.006</b>
<i>AU(<math>\overline{PRC}</math>) + PC</i>	20% + 80%	0.64 ± 0.194	0.62 ± 0.182	0.24 ± 0.259	-	0.60 ± 0.047	0.53 ± 0.189
	40% + 60%	0.55 ± 0.201	0.52 ± 0.161	0.20 ± 0.085	-	0.75 ± 0.030	0.45 ± 0.152
	50% + 50%	0.69 ± 0.140	0.67 ± 0.127	0.45 ± 0.094	0.004 ± 0.012	0.70 ± 0.054	0.61 ± 0.117
	60% + 40%	0.84 ± 0.053	0.82 ± 0.048	0.65 ± 0.036	0.21 ± 0.071	0.75 ± 0.030	0.77 ± 0.047
	80% + 20%	0.93 ± 0.006	0.90 ± 0.005	0.73 ± 0.022	0.27 ± 0.072	0.80 ± 0.010	0.85 ± 0.009
<i>AU(<math>\overline{PRC}</math>) + VG</i>	20% + 80%	<b>0.94 ± 0.005</b>	0.90 ± 0.003	0.74 ± 0.010	0.32 ± 0.072	0.81 ± 0.007	0.86 ± 0.006
	40% + 60%	0.93 ± 0.006	0.90 ± 0.006	0.75 ± 0.018	0.31 ± 0.063	0.81 ± 0.012	0.86 ± 0.011
	50% + 50%	<b>0.94 ± 0.005</b>	0.90 ± 0.006	0.74 ± 0.017	0.31 ± 0.062	0.80 ± 0.011	0.86 ± 0.007
	60% + 40%	0.93 ± 0.008	0.90 ± 0.007	0.74 ± 0.018	0.31 ± 0.047	0.81 ± 0.010	0.86 ± 0.012
	80% + 20%	0.93 ± 0.007	0.90 ± 0.007	0.74 ± 0.023	0.31 ± 0.069	0.80 ± 0.013	0.85 ± 0.013
REPBASE							
<i>HF + PC</i>	20% + 80%	0.28 ± 0.234	0.23 ± 0.174	0.001 ± 0.001	-	0.19 ± 0.054	0.19 ± 0.162
	40% + 60%	0.38 ± 0.205	0.28 ± 0.153	0.01 ± 0.008	0.002 ± 0.003	0.25 ± 0.067	0.25 ± 0.135
	50% + 50%	0.60 ± 0.085	0.55 ± 0.070	0.31 ± 0.035	0.11 ± 0.024	0.47 ± 0.082	0.49 ± 0.060
	60% + 40%	0.75 ± 0.019	0.64 ± 0.010	0.52 ± 0.008	0.22 ± 0.024	0.58 ± 0.019	0.62 ± 0.011
	80% + 20%	<b>0.89 ± 0.005</b>	<b>0.85 ± 0.008</b>	<b>0.68 ± 0.008</b>	0.36 ± 0.016	0.69 ± 0.007	<b>0.78 ± 0.005</b>
<i>HF + VG</i>	20% + 80%	0.88 ± 0.006	0.84 ± 0.005	0.67 ± 0.006	0.30 ± 0.039	0.68 ± 0.008	0.77 ± 0.005
	40% + 60%	<b>0.89 ± 0.007</b>	<b>0.85 ± 0.008</b>	<b>0.68 ± 0.007</b>	0.35 ± 0.026	0.69 ± 0.008	<b>0.78 ± 0.005</b>
	50% + 50%	<b>0.89 ± 0.007</b>	<b>0.85 ± 0.007</b>	<b>0.68 ± 0.007</b>	0.36 ± 0.027	<b>0.70 ± 0.007</b>	<b>0.78 ± 0.006</b>
	60% + 40%	<b>0.89 ± 0.006</b>	<b>0.85 ± 0.005</b>	<b>0.68 ± 0.006</b>	0.36 ± 0.022	0.69 ± 0.005	<b>0.78 ± 0.004</b>
	80% + 20%	<b>0.89 ± 0.005</b>	<b>0.85 ± 0.004</b>	<b>0.68 ± 0.005</b>	<b>0.37 ± 0.027</b>	<b>0.70 ± 0.006</b>	<b>0.78 ± 0.004</b>
<i>AU(<math>\overline{PRC}</math>) + PC</i>	20% + 80%	0.34 ± 0.184	0.19 ± 0.146	-	-	0.17 ± 0.049	0.20 ± 0.116
	40% + 60%	0.35 ± 0.122	0.19 ± 0.114	0.003 ± 0.008	-	0.17 ± 0.045	0.20 ± 0.078
	50% + 50%	0.35 ± 0.127	0.12 ± 0.126	0.001 ± 0.001	-	0.14 ± 0.032	0.18 ± 0.088
	60% + 40%	0.84 ± 0.029	0.76 ± 0.027	0.35 ± 0.057	0.22 ± 0.031	0.42 ± 0.041	0.66 ± 0.028
	80% + 20%	0.86 ± 0.021	0.77 ± 0.023	0.36 ± 0.056	0.24 ± 0.026	0.42 ± 0.045	0.68 ± 0.026
<i>AU(<math>\overline{PRC}</math>) + VG</i>	20% + 80%	0.87 ± 0.010	0.79 ± 0.007	0.52 ± 0.054	0.28 ± 0.029	0.55 ± 0.040	0.72 ± 0.017
	40% + 60%	0.86 ± 0.019	0.77 ± 0.014	0.49 ± 0.058	0.27 ± 0.023	0.52 ± 0.045	0.70 ± 0.023
	50% + 50%	0.88 ± 0.008	0.78 ± 0.007	0.50 ± 0.057	0.29 ± 0.046	0.53 ± 0.050	0.71 ± 0.014
	60% + 40%	0.87 ± 0.019	0.78 ± 0.016	0.45 ± 0.061	0.28 ± 0.032	0.49 ± 0.047	0.70 ± 0.025
	80% + 20%	0.87 ± 0.013	0.78 ± 0.013	0.43 ± 0.066	0.28 ± 0.024	0.47 ± 0.053	0.69 ± 0.020

+ **PC** (20%) e as execuções de *AU( $\overline{PRC}$ ) + VG*, entretanto, essas não mantiveram o bom desempenho no conjunto REPBASE, decaindo demasiadamente e diferindo das demais com significância estatística. Considerando como critério de desempate os valores do Nível 4 nos dois conjuntos de dados, a execução **hF** (50%) + **VG** (50%) mostrou os melhores resultados no geral.

Na Tabela 12 são mostradas algumas estatísticas a respeito da interpretabilidade das listas de regras geradas nos experimentos com o **HC-WGA**. Conforme é possível observar nessas comparações, as alterações nas ponderações das fórmulas têm impacto significativo na interpretabilidade dos modelos, sendo em alguns casos ainda mais impactantes do que no desempenho. Como foram combinadas funções voltadas ao desempenho preditivo com funções que focam na interpretabilidade, quando uma ponderação favorece uma dessas, os resultados

são guiados a otimizar a função beneficiada. Esse cenário é visto claramente nas fórmulas que envolvem **PC**, o que acaba gerando modelos mais simples e menores, acontecendo o contrário quando a ponderação favorece as funções direcionadas ao desempenho. Diferente do que ocorreu nas funções ponderadas as quais **PC** está incluso e favorecido, nas execuções compostas por **VG** o mesmo comportamento não se repetiu com a mesma intensidade ou sequer se repetiu. Com **hF + VG** foi observado alterações bem discretas na complexidade dos modelos de regras, já em  $AU(\overline{PRC}) + VG$  houve uma inversão do comportamento esperado, pois quanto mais peso é atribuído a **VG**, mais complexo o modelo se torna. O que se esperava era que conforme as funções voltadas a interpretabilidade tivessem maiores pesos associados, mais interpretáveis seriam os modelos gerados. No entanto, nem sempre esse comportamento se reproduziu.

Tabela 12 – Análise de interpretabilidade das regras geradas pelo HC-WGA.

Fitness	Ponderação	Regras	Testes Ativos		Regras	Testes por Regra	
			PGSB	REPBASE			
$HF + PC$	20% + 80%	<b>4 ± 0.789</b>	20 ± 5.473	5 ± 0.667	<b>4 ± 0.699</b>	28 ± 7.642	6 ± 1.449
	40% + 60%	<b>4 ± 0.675</b>	19 ± 4.989	5 ± 1.174	14 ± 5.755	105 ± 44.73	7 ± 0.789
	50% + 50%	5 ± 1.197	30 ± 10.92	5 ± 1.174	509 ± 69.59	4220 ± 555.5	8 ± 0.084
	60% + 40%	10 ± 4.523	66 ± 37.76	6 ± 1.317	1240 ± 39.81	10211 ± 381.7	8 ± 0.106
	80% + 20%	1011 ± 17.37	7198 ± 143.1	7 ± 0.060	2415 ± 20.17	18823 ± 270.0	8 ± 0.055
$HF + VG$	20% + 80%	958 ± 21.00	6758 ± 185.2	7 ± 0.483	1902 ± 115.4	14608 ± 940.1	8 ± 0.055
	40% + 60%	1025 ± 23.99	7376 ± 187.7	7 ± 0.080	2396 ± 29.77	18538 ± 278.5	8 ± 0.062
	50% + 50%	1037 ± 15.15	7438 ± 173.5	7 ± 0.104	2397 ± 21.91	18640 ± 154.7	8 ± 0.052
	60% + 40%	1043 ± 14.81	7527 ± 147.1	7 ± 0.076	2405 ± 15.57	18675 ± 183.9	8 ± 0.049
	80% + 20%	1047 ± 14.61	7561 ± 131.1	7 ± 0.054	2404 ± 29.81	18654 ± 242.5	8 ± 0.033
$AU(\overline{PRC}) + PC$	20% + 80%	<b>4 ± 1.135</b>	<b>18 ± 5.855</b>	<b>4 ± 1.317</b>	<b>4 ± 0.000</b>	22 ± 5.559	5 ± 1.155
	40% + 60%	17 ± 3.204	106 ± 21.09	6 ± 0.817	<b>4 ± 0.483</b>	<b>20 ± 8.731</b>	<b>4 ± 1.595</b>
	50% + 50%	55 ± 5.801	360 ± 39.40	6 ± 0.471	5 ± 0.483	23 ± 5.865	5 ± 0.972
	60% + 40%	263 ± 36.40	1716 ± 174.5	6 ± 0.316	152 ± 58.66	902 ± 388.1	5 ± 0.675
	80% + 20%	368 ± 11.19	2224 ± 82.39	6 ± 0.422	188 ± 28.44	1133 ± 206.7	6 ± 0.483
$AU(\overline{PRC}) + VG$	20% + 80%	461 ± 7.729	2978 ± 89.60	6 ± 0.124	412 ± 25.71	2648 ± 168.7	6 ± 0.101
	40% + 60%	446 ± 10.74	2868 ± 79.59	6 ± 0.094	402 ± 25.20	2561 ± 205.7	6 ± 0.139
	50% + 50%	434 ± 11.71	2763 ± 110.6	6 ± 0.144	404 ± 16.63	2579 ± 131.5	6 ± 0.075
	60% + 40%	429 ± 14.57	2739 ± 115.3	6 ± 0.083	392 ± 16.89	2481 ± 132.3	6 ± 0.143
	80% + 20%	423 ± 11.81	2664 ± 128.1	6 ± 0.159	385 ± 23.27	2431 ± 166.4	6 ± 0.108

Semelhante ao que ocorreu na análise de interpretabilidade do **HC-GA**, as execuções que mais favoreceram a função **PC** nos experimentos com o **HC-WGA** acabaram obtendo os melhores resultados, sendo essas as otimizações de **hF + PC** e  $AU(\overline{PRC}) + PC$ , ambas com as ponderações de 20% + 80%, e assim, favorecendo consideravelmente mais a função voltada a interpretabilidade. No geral, considerando todos os critérios avaliados e os conjuntos de dados,  $AU(\overline{PRC})$  (20%) + **PC** (80%) mostrou os melhores e mais estáveis resultados, padrão este que não se repetiu para as demais execuções. Por estáveis, refere-se a gerar um número de Regras/Testes Ativos/Testes por Regra semelhantes para os dois conjunto de dados. Já os piores resultados foram obtidos por execuções que priorizaram mais as funções de desempenho, sendo essas todas as combinações de pesos para **hF + VG**, e também **hF** (80%) + **PC** (20%), todas essas apresentando valores bem semelhantes entre si.

Na Tabela 13 são comparados os tempos de treinamento coletados para cada fórmula ponderada testada. Assim como ocorreu com o **HC-GA**, os tempos variaram bastante de um

conjunto de dados para o outro, não crescendo de forma linear como poderia se esperar já que o REPBASE tem quase o dobro de instâncias do que possui o PGSB. Na realidade, apenas a execução  $AU(\overline{PRC})$  (80%) + **PC** (20%) correspondeu a essa projeção. Nas demais, como é o caso de  $AU(\overline{PRC})$  + **PC** com as ponderações de 20% + 80% e 50% + 50% e da execução **hF** (20%) + **PC** (80%), houveram acréscimos de 4 vezes nos tempos de treinamento. Em **hF** (50%) + **PC** (50%) e  $AU(\overline{PRC})$  + **VG** com as ponderações 20% + 80%, 40% + 60% e 80% + 20%, houve um aumento de 9 vezes aproximadamente, e por fim, observou-se o caso extremo de **hF** (80%) + **PC** (20%), que apresentou um aumento de 41 vezes no tempo requerido para induzir uma lista de regras no REPBASE em relação ao gerado no PGSB.

Tabela 13 – Tempos de treinamento do HC-WGA em minutos.

<b>Fitness</b>	<b>Ponderação</b>	<b>PGSB</b>	<b>REPBASE</b>
<i>HF + PC</i>	20% + 80%	20 ± 4.010	84 ± 8.240
	40% + 60%	<b>17 ± 4.985</b>	86 ± 8.089
	50% + 50%	18 ± 6.177	168 ± 49.29
	60% + 40%	<b>17 ± 3.744</b>	702 ± 220.4
	80% + 20%	44 ± 4.116	716 ± 55.06
<i>HF + VG</i>	20% + 80%	41 ± 7.431	867 ± 149.3
	40% + 60%	55 ± 2.650	1310 ± 66.03
	50% + 50%	60 ± 4.920	1321 ± 79.68
	60% + 40%	58 ± 3.090	1266 ± 39.81
	80% + 20%	53 ± 3.009	1004 ± 145.0
<i>AU(<math>\overline{PRC}</math>) + PC</i>	20% + 80%	20 ± 4.365	<b>71 ± 6.819</b>
	40% + 60%	23 ± 2.562	<b>71 ± 8.383</b>
	50% + 50%	20 ± 2.435	74 ± 5.671
	60% + 40%	33 ± 5.326	88 ± 13.88
	80% + 20%	36 ± 3.984	75 ± 9.262
<i>AU(<math>\overline{PRC}</math>) + VG</i>	20% + 80%	46 ± 3.992	391 ± 62.13
	40% + 60%	44 ± 5.328	382 ± 63.01
	50% + 50%	41 ± 4.531	330 ± 18.59
	60% + 40%	42 ± 5.246	322 ± 13.18
	80% + 20%	36 ± 9.065	316 ± 12.42

Os menores tempos de treinamento assim como os menores valores de desvio padrão foram atingidos pelas fórmulas que incluíram a função **PC**, onde  $AU(\overline{PRC})$  (20%) + **PC** (80%) apresentou os menores tempos no geral, seguida pelas execuções de mesma configuração, porém com as ponderações de 40% + 60% e 50% + 50%, e em seguida **hF** + **PC** com 20% + 80% e 40% + 60%. Inclusive, como é visto nesses comparativos, os desvios padrão aumentaram drasticamente no REPBASE, mas ainda assim, as execuções supracitadas mostraram valores baixos em comparação com as demais. Quanto aos maiores tempos de treinamento, esses foram obtidos pelas ponderações usadas com a fórmula **hF** + **VG**, onde mais precisamente, **hF** (50%) + **VG** (50%) atingiu os valores mais altos nos dois conjuntos de dados.

### 8.3 Experimentos com o HC-LGA

Nessa seção são apresentados e discutidos os experimentos com o método hierárquico global de otimização multiobjetivo usando uma Abordagem Lexicográfica, denominado **HC-LGA**. Diferente do que foi realizado com o **HC-WGA**, que combinou funções representantes de cada um dos objetivos definidos nessa pesquisa, a intenção dos experimentos aqui presentes fora a de testar diferentes funções em conjunto, independente de qual objetivo endereçam. Apesar de tal abordagem parecer redundante, é importante como prática de pesquisa a investigação de hipóteses nulas, como assumir que a combinação de funções voltadas ao desempenho preditivo não resultará, ao mesmo tempo, em modelos mais interpretáveis, ou ainda, que a utilização de funções orientadas a interpretabilidade não irão também otimizar o desempenho. Ademais, foram testadas diferentes ordens na otimização das funções visando analisar se uma dada ordem gera melhores resultados. A Tabela 14 apresenta as médias da Medida-F Hierárquica assim como os desvios padrão para cada otimização das funções objetivo utilizadas em ordem de importância.

Tabela 14 – Análise das predições do HC-LGA considerando a Medida-F Hierárquica.

Objetivo 1	Objetivo 2	Nível 1	Nível 2	Nível 3	Nível 4	Classes Folha	Geral
PGSB							
<i>hF</i>	<i>PC</i>	<b>0.94 ± 0.007</b>	0.91 ± 0.008	<b>0.77 ± 0.010</b>	0.42 ± 0.043	0.83 ± 0.007	0.87 ± 0.008
<i>PC</i>	<i>hF</i>	0.86 ± 0.000	0.82 ± 0.001	0.21 ± 0.225	-	0.52 ± 0.084	0.72 ± 0.025
<i>hF</i>	<i>AU(PRC)</i>	0.93 ± 0.005	0.91 ± 0.007	0.76 ± 0.009	0.40 ± 0.022	0.82 ± 0.008	0.86 ± 0.006
<i>AU(PRC)</i>	<i>hF</i>	0.93 ± 0.006	0.91 ± 0.008	0.76 ± 0.007	0.38 ± 0.049	0.83 ± 0.007	0.87 ± 0.007
<i>hF</i>	<i>VG</i>	<b>0.94 ± 0.005</b>	0.91 ± 0.006	<b>0.77 ± 0.008</b>	0.42 ± 0.033	0.83 ± 0.005	0.87 ± 0.005
<i>VG</i>	<i>hF</i>	0.92 ± 0.008	0.89 ± 0.008	0.65 ± 0.101	0.18 ± 0.106	0.76 ± 0.046	0.83 ± 0.021
<i>PC</i>	<i>AU(PRC)</i>	0.86 ± 0.000	0.82 ± 0.001	0.26 ± 0.279	-	0.55 ± 0.108	0.73 ± 0.032
<i>AU(PRC)</i>	<i>PC</i>	0.93 ± 0.005	0.90 ± 0.005	0.73 ± 0.016	0.26 ± 0.078	0.80 ± 0.009	0.85 ± 0.007
<i>PC</i>	<i>VG</i>	0.86 ± 0.009	0.82 ± 0.009	0.30 ± 0.285	0.04 ± 0.118	0.56 ± 0.113	0.73 ± 0.038
<i>VG</i>	<i>PC</i>	0.91 ± 0.015	0.88 ± 0.016	0.64 ± 0.072	0.10 ± 0.144	0.75 ± 0.037	0.82 ± 0.016
<i>AU(PRC)</i>	<i>VG</i>	<b>0.94 ± 0.008</b>	0.90 ± 0.007	0.73 ± 0.010	0.27 ± 0.072	0.80 ± 0.006	0.85 ± 0.006
<i>VG</i>	<i>AU(PRC)</i>	0.92 ± 0.012	0.89 ± 0.013	0.67 ± 0.060	0.18 ± 0.108	0.76 ± 0.034	0.83 ± 0.017
<i>hP</i>	<i>hR</i>	<b>0.94 ± 0.006</b>	<b>0.92 ± 0.006</b>	<b>0.77 ± 0.009</b>	<b>0.44 ± 0.054</b>	<b>0.84 ± 0.007</b>	<b>0.88 ± 0.006</b>
<i>hR</i>	<i>hP</i>	<b>0.94 ± 0.006</b>	<b>0.92 ± 0.006</b>	<b>0.77 ± 0.012</b>	0.42 ± 0.046	0.83 ± 0.009	0.87 ± 0.008
REPBASE							
<i>hF</i>	<i>PC</i>	<b>0.89 ± 0.004</b>	0.85 ± 0.005	<b>0.68 ± 0.008</b>	0.36 ± 0.026	0.69 ± 0.009	0.78 ± 0.005
<i>PC</i>	<i>hF</i>	0.75 ± 0.064	0.64 ± 0.079	-	-	0.24 ± 0.048	0.55 ± 0.055
<i>hF</i>	<i>AU(PRC)</i>	<b>0.89 ± 0.006</b>	0.85 ± 0.006	<b>0.68 ± 0.006</b>	0.36 ± 0.020	0.69 ± 0.006	0.78 ± 0.006
<i>AU(PRC)</i>	<i>hF</i>	0.88 ± 0.006	0.81 ± 0.013	0.59 ± 0.021	0.31 ± 0.026	0.61 ± 0.021	0.74 ± 0.010
<i>hF</i>	<i>VG</i>	<b>0.89 ± 0.006</b>	0.85 ± 0.006	<b>0.68 ± 0.008</b>	<b>0.37 ± 0.019</b>	0.69 ± 0.007	0.78 ± 0.006
<i>VG</i>	<i>hF</i>	0.85 ± 0.027	0.77 ± 0.025	0.39 ± 0.076	0.10 ± 0.085	0.48 ± 0.043	0.67 ± 0.024
<i>PC</i>	<i>AU(PRC)</i>	0.75 ± 0.085	0.63 ± 0.147	-	-	0.23 ± 0.068	0.54 ± 0.083
<i>AU(PRC)</i>	<i>PC</i>	0.86 ± 0.016	0.77 ± 0.019	0.34 ± 0.030	0.22 ± 0.028	0.39 ± 0.027	0.67 ± 0.014
<i>PC</i>	<i>VG</i>	0.71 ± 0.073	0.60 ± 0.078	-	-	0.21 ± 0.053	0.52 ± 0.058
<i>VG</i>	<i>PC</i>	0.85 ± 0.019	0.77 ± 0.021	0.43 ± 0.129	0.01 ± 0.013	0.51 ± 0.075	0.69 ± 0.027
<i>AU(PRC)</i>	<i>VG</i>	0.86 ± 0.019	0.77 ± 0.019	0.37 ± 0.035	0.24 ± 0.031	0.42 ± 0.028	0.67 ± 0.019
<i>VG</i>	<i>AU(PRC)</i>	0.84 ± 0.015	0.76 ± 0.014	0.45 ± 0.060	0.04 ± 0.037	0.52 ± 0.072	0.68 ± 0.015
<i>hP</i>	<i>hR</i>	<b>0.89 ± 0.007</b>	<b>0.86 ± 0.007</b>	<b>0.68 ± 0.008</b>	0.34 ± 0.019	<b>0.70 ± 0.006</b>	<b>0.79 ± 0.006</b>
<i>hR</i>	<i>hP</i>	<b>0.89 ± 0.005</b>	0.85 ± 0.006	<b>0.68 ± 0.006</b>	<b>0.37 ± 0.026</b>	0.69 ± 0.008	0.78 ± 0.005

A partir da análise dessas comparações, notou-se que a ordem das otimizações influencia consideravelmente nos resultados finais. Assim, todas as otimizações em que uma função orientada ao desempenho preditivo recebe prioridade, melhores desempenhos preditivos são observados, acontecendo o mesmo quanto a interpretabilidade do modelo quando uma função

voltada a ela é assinalada como objetivo prioritário. É interessante observar também que, quando comparados os resultados com os do **HC-WGA**, especificamente os quais são usadas ponderações mais altas para as funções de desempenho, percebe-se uma grande semelhança com os resultados gerados pelas mesmas combinações de funções no **HC-LGA**. Em outras palavras, pode-se dizer que nesses casos o **HC-LGA** está favorecendo com peso maior ou igual a 80% as funções de desempenho prioritárias. Portanto, comparando os valores das Tabelas 11 e 14, tem-se os resultados de **hF** (80%) + **PC** (20%) como idênticos aos alcançados pela otimização de **hF** e **PC** para ambos os conjuntos de dados. Correspondências semelhantes também ocorrem com **hF** + **VG** (todas as ponderações),  $AU(\overline{PRC})$  (80%) + **PC** (20%) e  $AU(\overline{PRC})$  (80%) + **VG** (20%).

Ainda sobre as ordens de otimização das funções objetivo, observou-se que em todos os casos e nos dois conjuntos de dados, utilizar uma função que favorece o desempenho gera de fato um desempenho preditivo melhor, apesar de haver casos como em **VG** e  $AU(\overline{PRC})$  onde os resultados são semelhantes a ordem inversa. Quanto as otimizações envolvendo apenas as funções de desempenho e considerando as predições nos dois conjuntos, conclui-se que **hF** e  $AU(\overline{PRC})$  é uma melhor opção que  $AU(\overline{PRC})$  e **hF**, assim como é o caso de **hP** e **hR** em vez da ordem contrária, que mostrou resultados ligeiramente superiores.

Alguns problemas mais pontuais também foram observados nessas otimizações. Nas execuções em que **PC** foi usado como objetivo prioritário, não foram realizadas predições no Nível 4 da hierarquia de **TEs**, com exceção da otimização de **PC** e **VG** no PGSB, mas que ainda assim, foi obtido um resultado de 0.04, sendo extremamente baixo e considerando que o desvio padrão é de 0.118, esse nem sequer pode ser considerado. Ademais, ainda ocorreu das otimizações com **PC** no REPBASE não serem capazes de predizer classes no Nível 3 e Nível 4.

Como apanhado final, considerando os valores obtidos pelas otimizações realizadas em cada critério avaliado, tem-se como os melhores resultados aqueles obtidos pela execução de **hP** como função objetivo prioritária e **hR** como secundária, tanto no PGSB quanto no REPBASE. Em seguida, vieram as execuções de **hR** e **hP**, **hF** e **PC**, **hF** e **VG** e, por fim, **hF** e  $AU(\overline{PRC})$ , todas comparáveis em qualidade. A seguir, os resultados da análise de interpretabilidade dos modelos de regras induzidos são compilados e apresentados na Tabela 15.

Como visto na Tabela 15, as inversões de importância das funções objetivo provocam grande impacto nos resultados, principalmente nas otimizações que endereçam ambos os objetivos. Nos casos de **PC** ou **VG** combinados com **hF** ou  $AU(\overline{PRC})$ , se nota uma inversão nítida na orientação da otimização que passa a priorizar com mais ênfase o objetivo prioritário. Isso é mais acentuado nas otimizações as quais **PC** se faz presente e menos nas que **VG** aparece. Na realidade, apesar das execuções com **VG** como objetivo principal induzirem modelos muitas vezes mais interpretáveis do que na ordem inversa, ainda assim, foram gerados bons resultados quanto ao desempenho, inclusive comparáveis aos casos em que uma função voltada a esse é utilizada como objetivo principal. Isso leva a conclusão de que adotar **VG** como prioritária é uma alternativa mais interessante dado o bom *trade-off* entre desempenho preditivo e interpretabilidade.

Tabela 15 – Análise de interpretabilidade das regras geradas pelo HC-LGA.

Objetivo 1	Objetivo 2	Regras	PGSB		Regras	REPBASE	
			Testes Ativos	Testes por Regra		Testes Ativos	Testes por Regra
<i>hF</i>	<i>PC</i>	690 ± 22.75	4675 ± 157.4	7 ± 0.080	2395 ± 22.32	18560 ± 241.6	8 ± 0.000
<i>PC</i>	<i>hF</i>	<b>3 ± 0.516</b>	12 ± 3.592	3 ± 0.882	<b>4 ± 0.000</b>	<b>13 ± 5.583</b>	<b>3 ± 1.396</b>
<i>hF</i>	$AU(\overline{PRC})$	1053 ± 14.16	7508 ± 136.0	7 ± 0.076	2434 ± 12.75	18871 ± 154.0	8 ± 0.034
$AU(\overline{PRC})$	<i>hF</i>	774 ± 18.75	5336 ± 164.9	7 ± 0.089	1123 ± 128.2	8583 ± 1005	8 ± 0.112
<i>hF</i>	<i>VG</i>	976 ± 33.72	6905 ± 255.0	7 ± 0.051	2419 ± 20.22	18701 ± 237.8	8 ± 0.054
<i>VG</i>	<i>hF</i>	76 ± 47.98	428 ± 306.1	5 ± 0.550	139 ± 82.56	981 ± 637.0	7 ± 1.012
<i>PC</i>	$AU(\overline{PRC})$	4 ± 1.033	12 ± 5.821	3 ± 0.677	<b>4 ± 0.000</b>	15 ± 5.442	4 ± 1.360
$AU(\overline{PRC})$	<i>PC</i>	367 ± 13.05	2277 ± 85.91	6 ± 0.071	209 ± 31.68	1282 ± 239.5	6 ± 0.260
<i>PC</i>	<i>VG</i>	4 ± 0.707	<b>8 ± 2.251</b>	<b>2 ± 0.641</b>	<b>4 ± 0.000</b>	<b>13 ± 4.459</b>	<b>3 ± 1.115</b>
<i>VG</i>	<i>PC</i>	13 ± 4.428	49 ± 18.53	4 ± 0.414	24 ± 7.720	127 ± 46.42	5 ± 0.525
$AU(\overline{PRC})$	<i>VG</i>	397 ± 16.41	2480 ± 131.5	6 ± 0.125	228 ± 25.41	1380 ± 164.3	6 ± 0.204
<i>VG</i>	$AU(\overline{PRC})$	76 ± 46.47	433 ± 278.4	6 ± 0.551	40 ± 13.22	206 ± 79.26	5 ± 0.717
<i>hP</i>	<i>hR</i>	1151 ± 14.87	8427 ± 199.8	7 ± 0.101	2467 ± 32.30	19301 ± 267.2	8 ± 0.026
<i>hR</i>	<i>hP</i>	1143 ± 27.55	8277 ± 207.5	7 ± 0.037	2423 ± 27.10	18799 ± 330.2	8 ± 0.067

Quanto as otimizações usando apenas funções voltadas a um só objetivo, apesar de suas ordens não mostrarem alterações tão notáveis como nos exemplos supracitados, essas ainda assim mostraram diferenças significativas. Na otimização de  $AU(\overline{PRC})$  e *hF* houveram diferenças consideráveis quanto aos critérios de *regras* e *testes ativos*, onde o uso de  $AU(\overline{PRC})$  como objetivo principal se mostrou melhor nos dois conjuntos, principalmente no REPBASE em que foram geradas menos de 50% das regras e testes ativos em relação a otimização inversa. Já na execução de *hP* e *hR*, essa não mostrou mudanças significativas quanto sua ordem, no entanto, a utilização de *hR* como objetivo de primeira ordem mostrou resultados ligeiramente melhores porém sem significância estatística. A respeito das otimizações das funções de interpretabilidade *VG* e *PC*, essas apresentaram diferenças consideráveis em todos os critérios de interpretabilidade avaliados. O uso de *VG* como objetivo de maior importância resultou em valores substancialmente inferiores, gerando entre três a seis vezes mais regras, dez vezes mais testes ativos e praticamente o dobro de testes por regra considerando os dois conjuntos de dados.

Como melhores resultados, apontam-se as otimizações adotando *PC* como objetivo prioritário, aonde foram geradas as listas de regras mais interpretáveis para os dois conjuntos de dados. Na ordem, os melhores modelos foram obtidos pelas otimizações de *PC* e *hF*, que ligeiramente superaram a combinação das funções de interpretabilidade *PC* e *VG*. Em seguida, tem-se as execuções de *PC* e  $AU(\overline{PRC})$ , que curiosamente geraram modelos mais simples que os da otimização *VG* e *PC*. Novamente, os piores resultados foram oriundos das execuções com os melhores desempenhos preditivos, aonde foram gerados modelos com mais regras, testes por regra e mais testes ativos. Considerando os dois conjuntos de dados, dos modelos mais complexos para os mais simples, citam-se: *hP* e *hR*, *hR* e *hP*, *hF* e  $AU(\overline{PRC})$ , *hF* e *VG*, e por fim, *hF* e *PC*. Os desvios padrão nessas otimizações também foram altos, principalmente quando comparados com os valores gerados pelas melhores soluções nessa análise. A seguir, os tempos médios de treinamento para essas otimizações assim como seus respectivos valores de desvio padrão são comparados e analisados na Tabela 16.

Tabela 16 – Tempos de treinamento do HC-LGA em minutos.

Objetivo 1	Objetivo 2	PGSB	REPBASE
<i>hF</i>	<i>PC</i>	33 ± 3.137	698 ± 68.39
<i>PC</i>	<i>hF</i>	25 ± 4.519	131 ± 22.33
<i>hF</i>	$AU(\overline{PRC})$	82 ± 25.00	1608 ± 123.8
$AU(\overline{PRC})$	<i>hF</i>	96 ± 13.83	1250 ± 429.3
<i>hF</i>	<i>VG</i>	45 ± 6.706	732 ± 146.3
<i>VG</i>	<i>hF</i>	17 ± 5.274	145 ± 36.50
<i>PC</i>	$AU(\overline{PRC})$	22 ± 4.292	126 ± 57.76
$AU(\overline{PRC})$	<i>PC</i>	33 ± 3.405	98 ± 22.09
<i>PC</i>	<i>VG</i>	<b>7 ± 1.476</b>	<b>22 ± 4.093</b>
<i>VG</i>	<i>PC</i>	10 ± 1.828	59 ± 13.23
$AU(\overline{PRC})$	<i>VG</i>	45 ± 4.884	95 ± 11.29
<i>VG</i>	$AU(\overline{PRC})$	20 ± 5.770	149 ± 55.81
<i>hP</i>	<i>hR</i>	79 ± 15.93	1032 ± 87.31
<i>hR</i>	<i>hP</i>	81 ± 20.11	1067 ± 506

Os menores tempos de treinamento foram obtidos pela otimização de **PC** e **VG**, que levou 7 e 22 minutos para gerar modelos de regras que classificam as instâncias de PGSB e REPBASE, respectivamente. Logo em seguida, tem-se justamente a inversão dessa otimização, **VG** e **PC**, que demandou um pouco menos que o dobro do tempo no PGSB e quase o triplo no REPBASE. Os piores resultados foram alcançados, do mais demorado ao mais rápido, pelas execuções de **hF** e  $AU(\overline{PRC})$ , seguida de sua inversão  $AU(\overline{PRC})$  e **hF**, e logo após **hP** e **hR** seguidos também de sua inversão **hR** e **hP**. Assim como ocorreu nas demais análises de tempo de execução/treinamento, os resultados mais elevados também mostraram os maiores valores de desvio padrão, ocorrendo o contrário quanto as execuções com os menores tempos.

## 8.4 Correlações entre Objetivos

A fim de verificar se as funções objetivo possuem algum tipo de correlação, o que justificaria ou não suas otimizações em conjunto, foram realizadas análises de correlação das combinações executadas. Cabe salientar que foi necessário ter certo cuidado quanto a representatividade das amostras, já que foram verificadas mudanças nas distribuições dos dados quando o número de amostras variava de uma execução para outra. Esse é um ponto importante uma vez que cada teste de correlação é adequado a uma distribuição. A seguir, são mostradas as variações observadas para cada função objetivo, obtidas através da execução do teste de Shapiro-Wilk com  $\alpha = 0.05$ . Apenas as distribuições de **hR**, **hP** e **VG** não são exibidas pois não mostraram alterações, apresentando de forma consistente distribuições com padrões monotônicos.

Para a inspeção visual da distribuição dos dados foi utilizado um gráfico Q-Q (no inglês, *Quantil-Quantile*) (WILK; GNANADESIKAN, 1968), um dispositivo que verifica a validade de uma suposição distribucional em um amostra de dados. Sua ideia é calcular o valor teórico

esperado para cada instância com base na distribuição considerada, que aqui é a distribuição normal. Logo, se os dados seguem a distribuição assumida os pontos no gráfico estarão situados em uma linha reta. Nos gráficos a seguir, cada ponto é um indivíduo (regra) e os eixos  $X$  e  $Y$  são os valores teóricos para a distribuição normal e seus valores para as funções objetivo testadas, respectivamente. Os gráficos Q-Q também foram usados nas correlações de objetivos exibidas nessa seção, porém nesses, ambos os eixos correspondem a valores de funções objetivo.

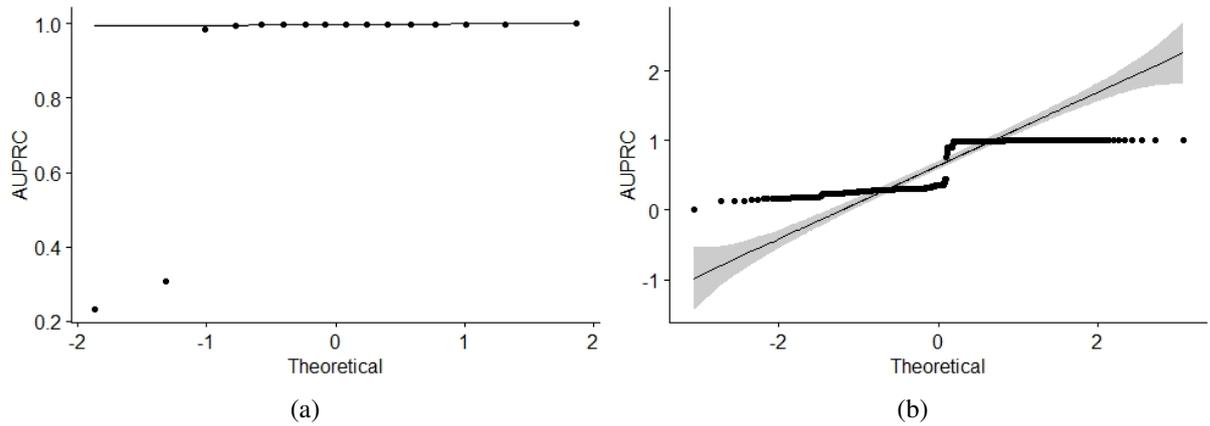


Figura 26 – Distribuição dos dados de  $AU(\overline{PCR})$ .

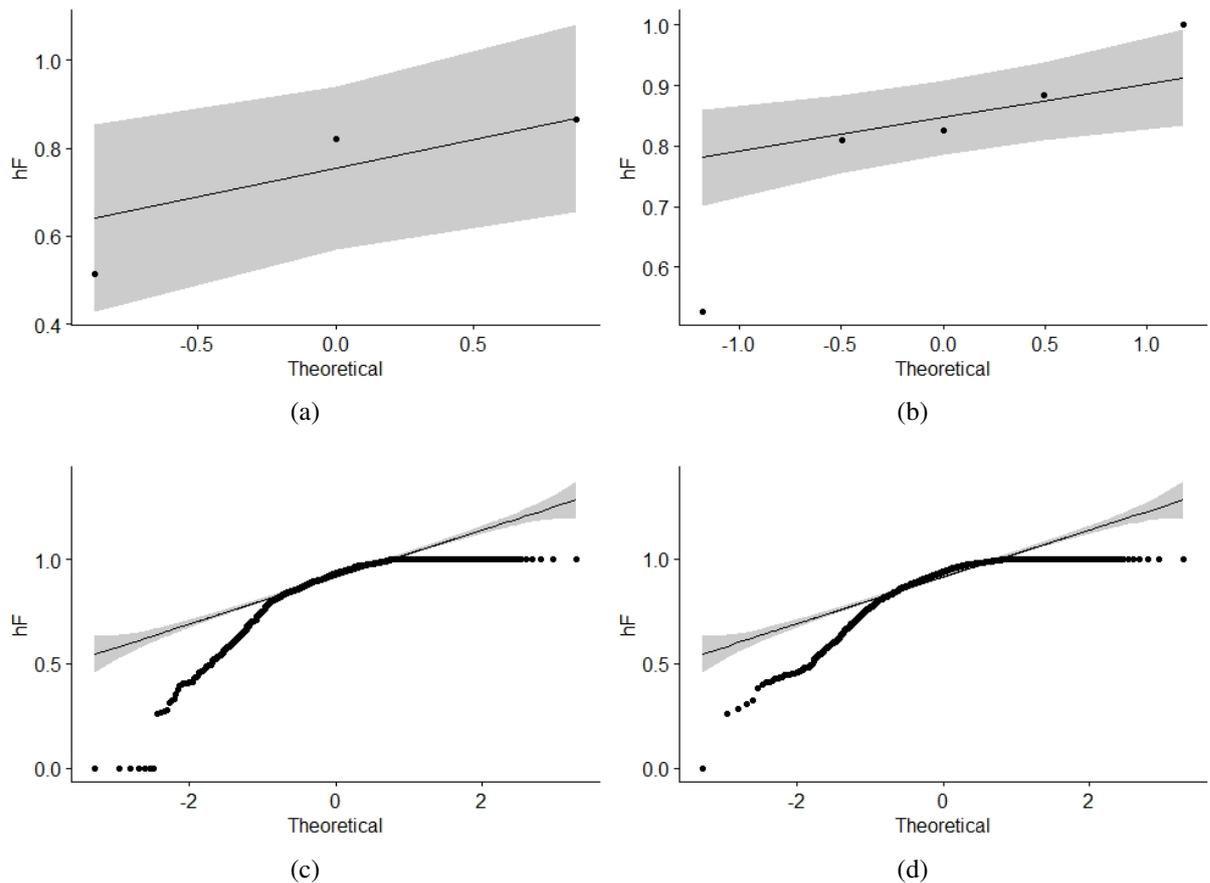


Figura 27 – Distribuição dos dados de  $hF$ .

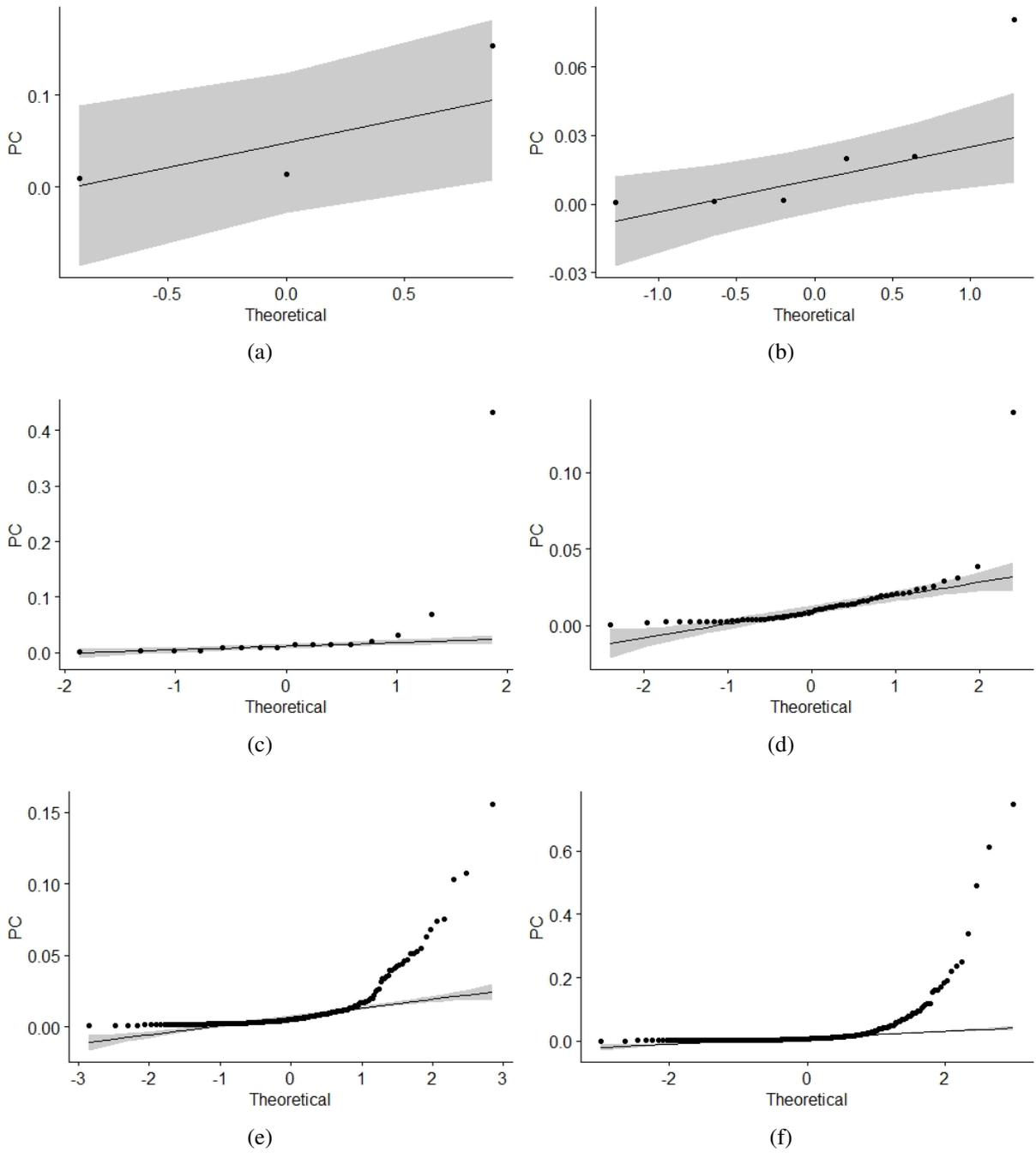


Figura 28 – Distribuição dos dados de PC.

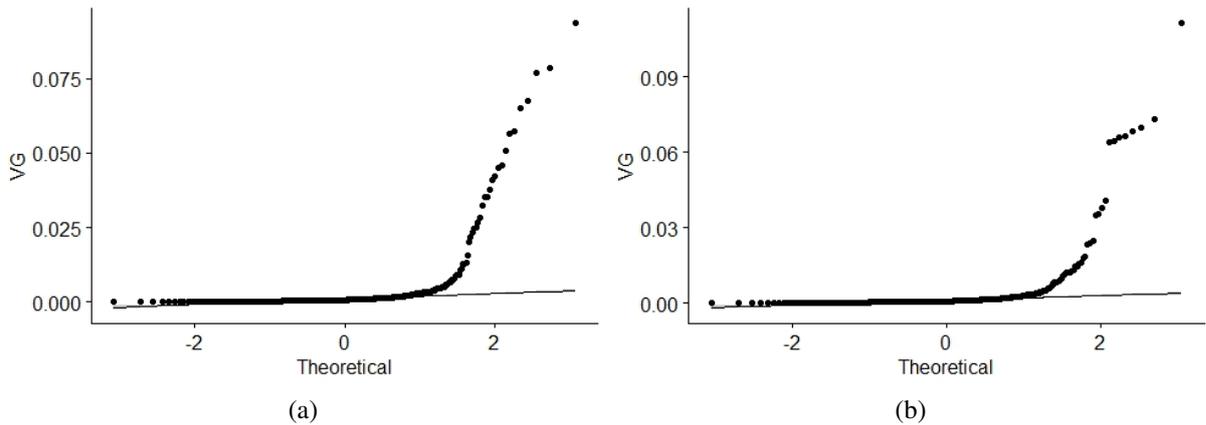
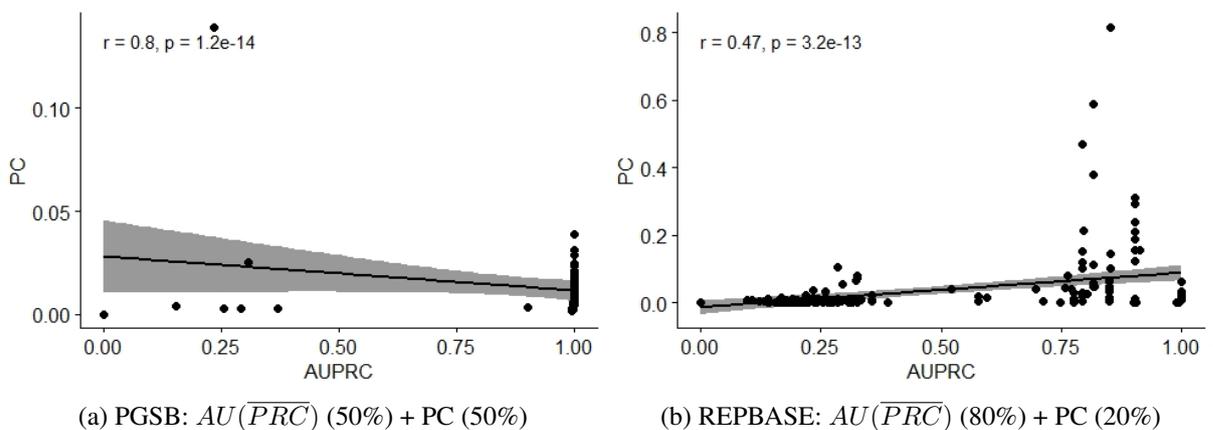


Figura 29 – Distribuição dos dados de VG.

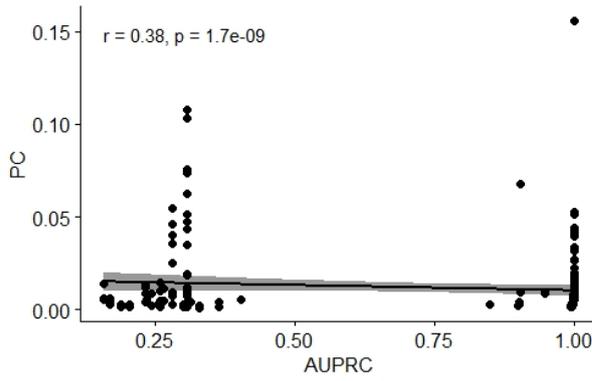
Conforme visto nas distribuições apresentadas pelas Figuras 26, 27, 28 e 29, todas apresentaram mudanças de uma distribuição normal para não normal, formando padrões monotônicos. Acreditamos que isso tenha ocorrido devido a representatividade das amostras que, em alguns casos, não eram numerosas o suficiente para representar o padrão de distribuição de forma significativa. Portanto, para cada valor de objetivo consideramos que suas reais distribuições são demonstradas por amostras que continham um maior número de instâncias.

Nas figuras a seguir, são exibidas as amostras dos valores de objetivos testados em conjunto com pesos nas fórmulas ponderadas do HC-WGA, na qual são avaliadas as correlações entre objetivos. Não foram inclusas as combinações de funções e pesos que não atingiram valores de significância estatística aceitáveis, entretanto, esses podem ser consultados no Apêndice A. Cabe salientar ainda que, mesmo aplicando-se os testes de correlação de Pearson e Spearman de acordo com as distribuições de dados observadas, todos os testes utilizando Pearson foram rejeitados na validação de significância. Desse modo, apenas as correlações de Spearman foram utilizados nos dados oriundos das experimentações com o HC-WGA e HC-LGA.

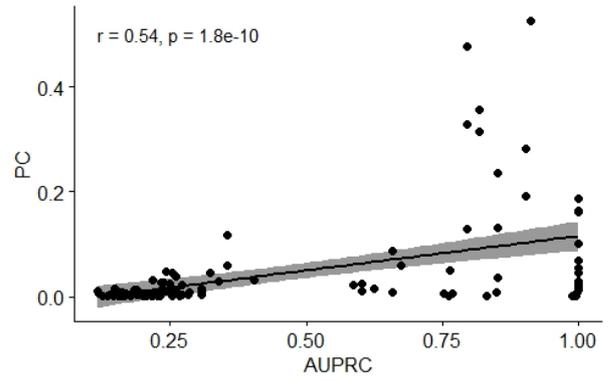


(a) PGSB:  $AU(\overline{PRC})$  (50%) + PC (50%)

(b) REPBASE:  $AU(\overline{PRC})$  (80%) + PC (20%)

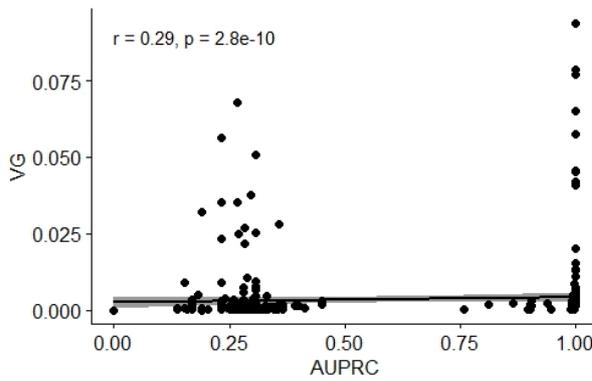


(c) PGSB:  $AU(\overline{PCR})$  (60%) + PC (40%)

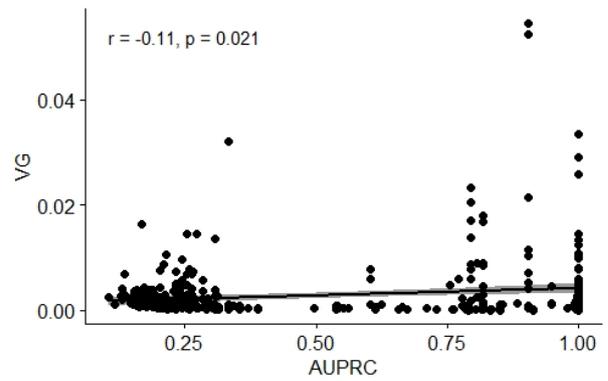


(d) REPBASE:  $AU(\overline{PCR})$  (60%) + PC (40%)

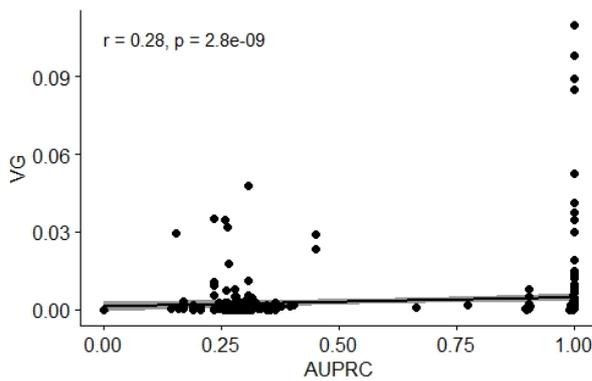
Figura 30 – Correlações dos objetivos  $AU(\overline{PCR})$  e PC utilizando o HC-WGA.



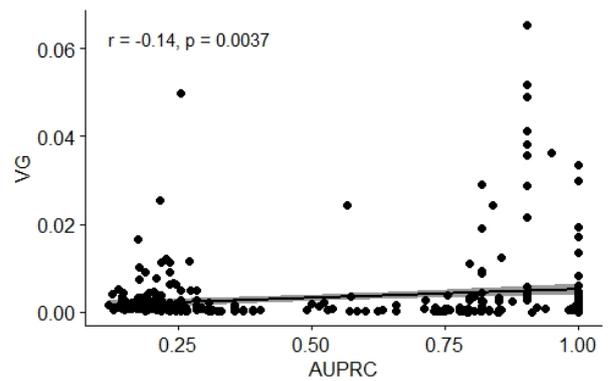
(a) PGSB:  $AU(\overline{PCR})$  (20%) + VG (80%)



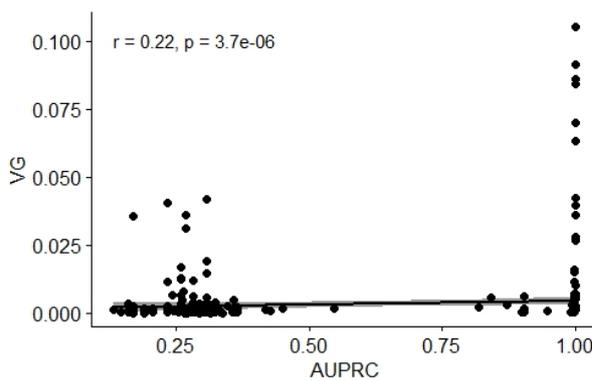
(b) REPBASE:  $AU(\overline{PCR})$  (20%) + VG (80%)



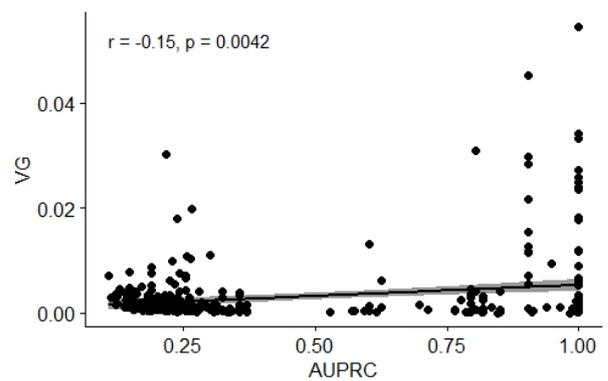
(c) PGSB:  $AU(\overline{PCR})$  (50%) + VG (50%)



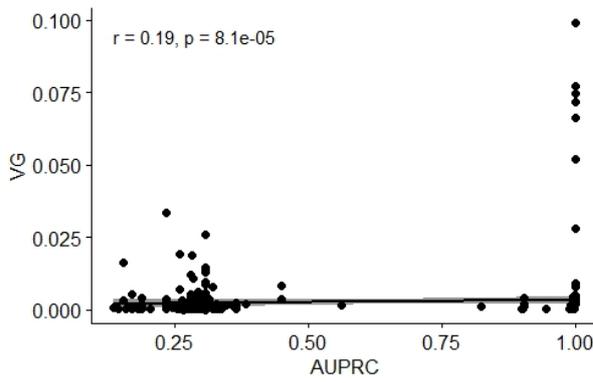
(d) REPBASE:  $AU(\overline{PCR})$  (50%) + VG (50%)



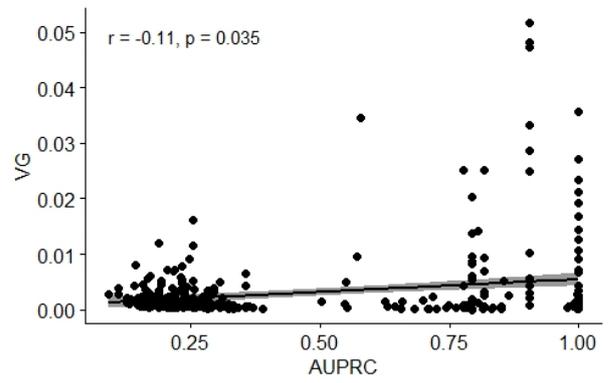
(e) PGSB:  $AU(\overline{PCR})$  (60%) + VG (40%)



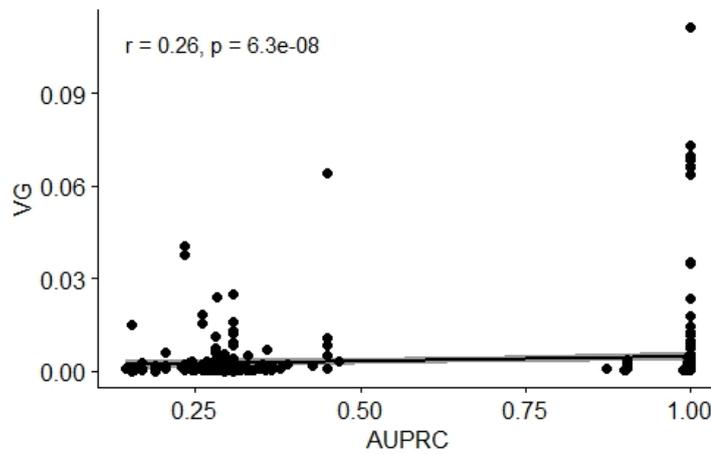
(f) REPBASE:  $AU(\overline{PCR})$  (60%) + VG (40%)



(g) PGSB:  $AU(\overline{PCR})$  (80%) + VG (20%)

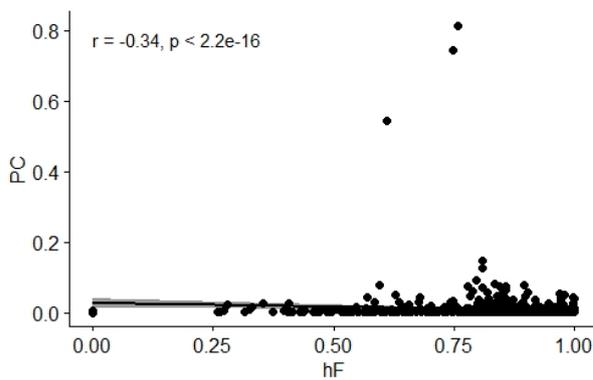


(h) REPBASE:  $AU(\overline{PCR})$  (80%) + VG (20%)

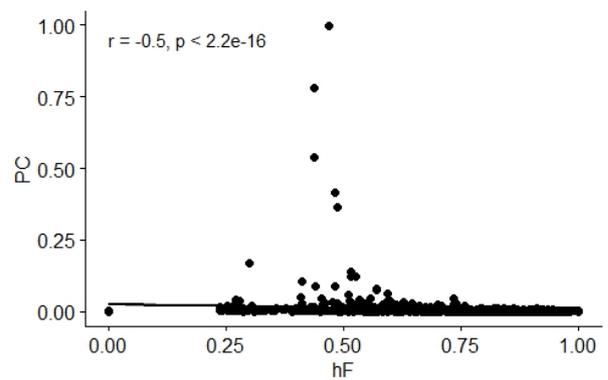


(i) PGSB:  $AU(\overline{PCR})$  (40%) + VG (60%)

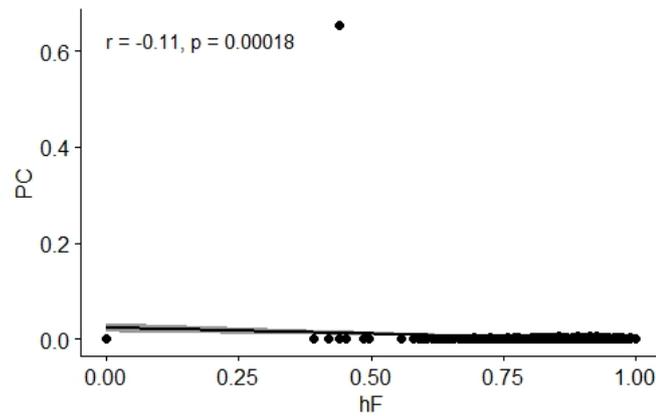
Figura 31 – Correlações dos objetivos  $AU(\overline{PCR})$  e VG utilizando o HC-WGA.



(a) PGSB: hF (80%) + PC (20%)

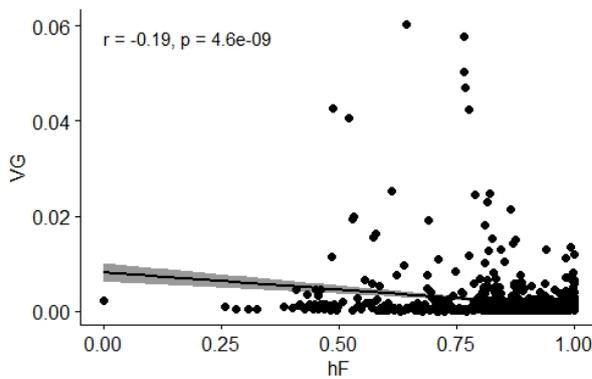


(b) REPBASE: hF (80%) + PC (20%)

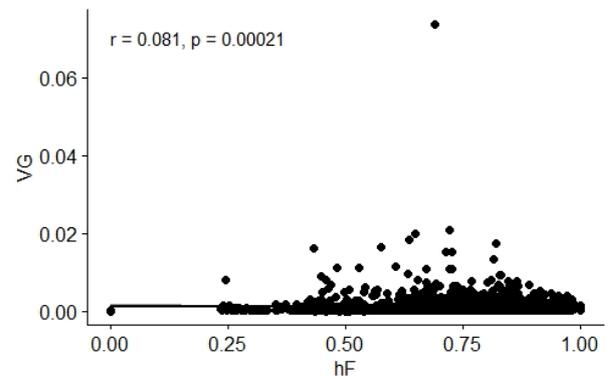


(c) REPBASE: hF (60%) + PC (40%)

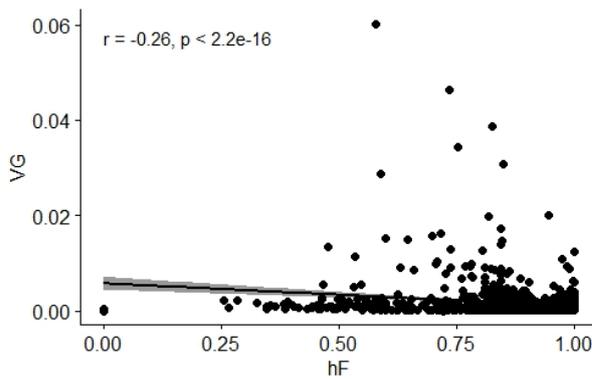
Figura 32 – Correlações dos objetivos hF e PC utilizando o HC-WGA.



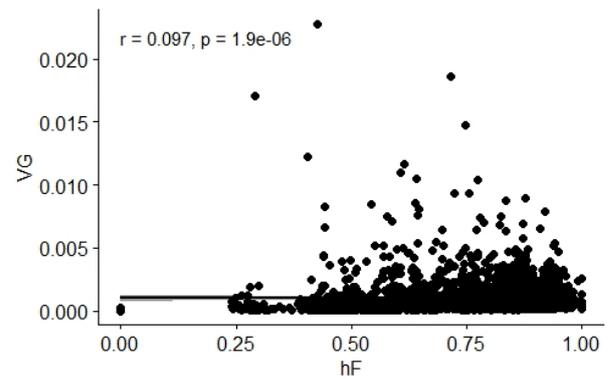
(a) PGSB: hF (20%) + VG (80%)



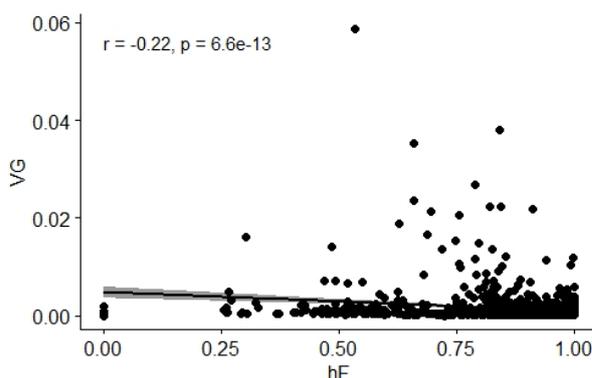
(b) REPBASE: hF (20%) + VG (80%)



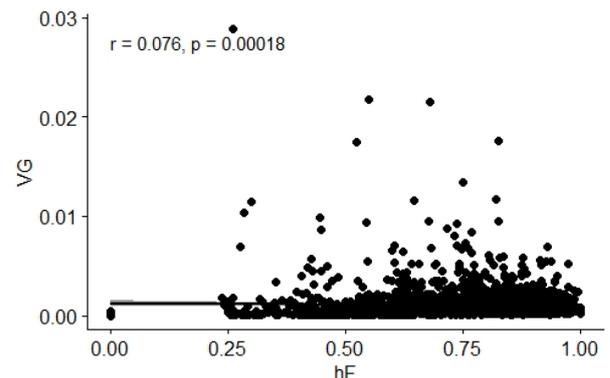
(c) PGSB: hF (40%) + VG (60%)



(d) REPBASE: hF (40%) + VG (60%)



(e) PGSB: hF (50%) + VG (50%)



(f) REPBASE: hF (50%) + VG (50%)

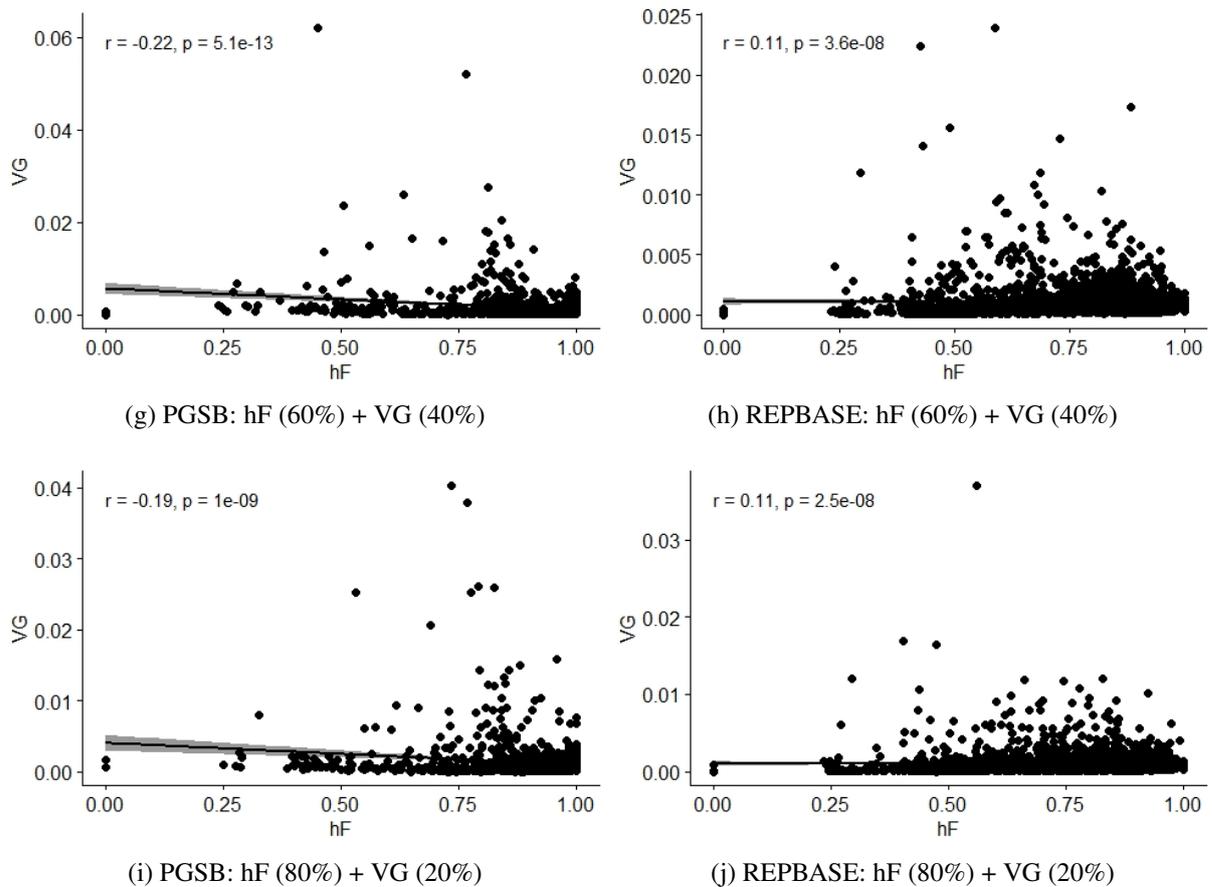


Figura 33 – Correlações dos objetivos hF e VG utilizando o HC-WGA.

Na Tabela 17, é visto um compilado dos coeficientes de correlação entre os objetivos testados no HC-WGA, em que nas colunas de seu cabeçalho são expressos as ponderações utilizadas para cada par de funções objetivos vistos nas linhas e colunas, assim como os conjuntos de dados usados. Nesta, os campos contendo “x” representam que, para dada combinação de funções e pesos, não fora obtido significância estatística suficiente e, portanto, não é possível creditar a correlação. Ademais, uma interpretação para o quão forte são essas correlações, que variam entre  $[-1,1]$ , é a seguinte (RODGERS; NICEWANDER, 1988; MUKAKA, 2012):

- 00 – .19: Muito fraca.
- .20 – .39: Fraca.
- .40 – .59: Moderada.
- .60 – .79: Forte.
- .80 – 1.0: Muito Forte.

Tabela 17 – Correlações dos objetivos testados no HC-WGA.

	20% + 80%	40% + 60%	50% + 50%	60% + 40%	80% + 20%
PGSB					
<b>PC</b>					
<b>AU</b> ( $\overline{PRC}$ )	x	x	<b>0.81</b>	0.38	x
<b>hF</b>	x	x	x	x	<b>-0.34</b>
<b>VG</b>					
<b>AU</b> ( $\overline{PRC}$ )	0.29	0.26	0.28	0.22	0.19
<b>hF</b>	-0.19	-0.26	-0.22	-0.22	-0.19
REPBASE					
<b>PC</b>					
<b>AU</b> ( $\overline{PRC}$ )	x	x	x	<b>0.54</b>	0.47
<b>hF</b>	x	x	x	-0.11	-0.5
<b>VG</b>					
<b>AU</b> ( $\overline{PRC}$ )	-0.11	x	-0.14	<b>-0.15</b>	-0.11
<b>hF</b>	0.08	0.10	0.08	0.11	0.11

Muitos foram os resultados apresentados na Tabela 17 que não atingiram significância estatística suficiente e, portanto, não se pode afirmar que tais pares de objetivos são correlacionados. Foram sete funções que falharam no teste de significância em ambos os conjuntos de dados, sendo que a grande maioria têm relação com **PC**. Apesar de algumas otimizações apresentarem coeficientes de correlação altos, como é o caso de  $AU(\overline{PRC})$  (50%) + **PC** (50%) no PGSB, a mesma foi rejeitada quanto a sua significância no REPBASE. Outros casos semelhantes, em que correlações validadas em um conjunto de dados não são validadas em outro, são:  $AU(\overline{PRC})$  (80%) + **PC** (20%), **hF** (60%) + **PC** (40%) e  $AU(\overline{PRC})$  (40%) + **VG** (60%).

Outro ponto importante observado nessas correlações é a variação no sentido dos coeficientes envolvendo a função **VG** em ambos os conjuntos. Enquanto que no PGSB as correlações entre  $AU(\overline{PRC})$  e **VG** são positivas, o que significa que a medida em que o valor de  $AU(\overline{PRC})$  cresce, **VG** também aumenta, no REPBASE o contrário ocorre. Nesse, as correlações são negativas, assim, quando  $AU(\overline{PRC})$  cresce, **VG** decresce, logo, quando  $AU(\overline{PRC})$  é otimizado **VG** é deteriorado, uma vez que todas as funções utilizadas nesse trabalho são otimizadas de forma a maximizá-las. Esse mesmo padrão de inversões nos coeficientes é visto entre **hF** e **VG** quando comparado aos resultados no PGSB e REPBASE. Acreditamos que tanto esse comportamento quanto as falhas de significância possam ter ocorrido devido ao tamanho e/ou representatividade das amostras, ou ainda, devido a presença de *outliers*<sup>1</sup>. No entanto, mais investigações precisam

<sup>1</sup> Anomalias nos dados; Pontos "fora da curva", em outras palavras, instâncias distantes de outras numa população, fugindo do padrão de distribuição seguido pelos demais.

ser realizadas antes de chegar a uma conclusão. O que se pode utilizar como hipótese é que, de fato, existe algum tipo de correlação entre  $AU(\overline{PRC})$  e **VG**, além de **hF** e **VG**.

Desconsiderando os problemas supracitados, para o PGSB tem-se  $AU(\overline{PRC})$  (50%) + **PC** (50%) como a correlação positiva mais forte enquanto que **hF** (80%) + **PC** (20%) correspondem a correlação negativa mais elevada. Para o REPBASE, a correlação positiva mais forte também foi alcançada por  $AU(\overline{PRC})$  + **PC**, porém com as ponderação 60% + 40%, já o coeficiente de correlação negativa mais alto foi apresentado por  $AU(\overline{PRC})$  (60%) + **VG** (40%). No entanto, a fim de se obter conclusões sólidas acerca das correlações entre as funções objetivo executadas, é preciso desconsiderar aquelas que apresentaram problemas quanto a significância estatística mínima ou que são relacionadas a **VG**, dada as inconsistências previamente mencionadas e que tornaram os resultados inconclusivos. Assim, pode se assumir que  $AU(\overline{PRC})$  e **PC** possuem uma correlação positiva com coeficientes de correlação que variam entre [0.38, 0.81], enquanto que **hF** + **PC** têm uma correlação negativa onde os coeficientes variam entre [-0.11, -0.5]. Consequentemente, esses coeficientes variam em função do tamanho das amostras.

A seguir, são apresentados os gráficos Q-Q das amostras de valores das funções objetivos otimizadas em conjunto no **HC-LGA**, assim como seus respectivos coeficientes de correlação. Não foram inclusas as correlações entre objetivos que não atingiram os valores mínimos de significância estatística. No entanto, essas podem ser vistas no Apêndice A.

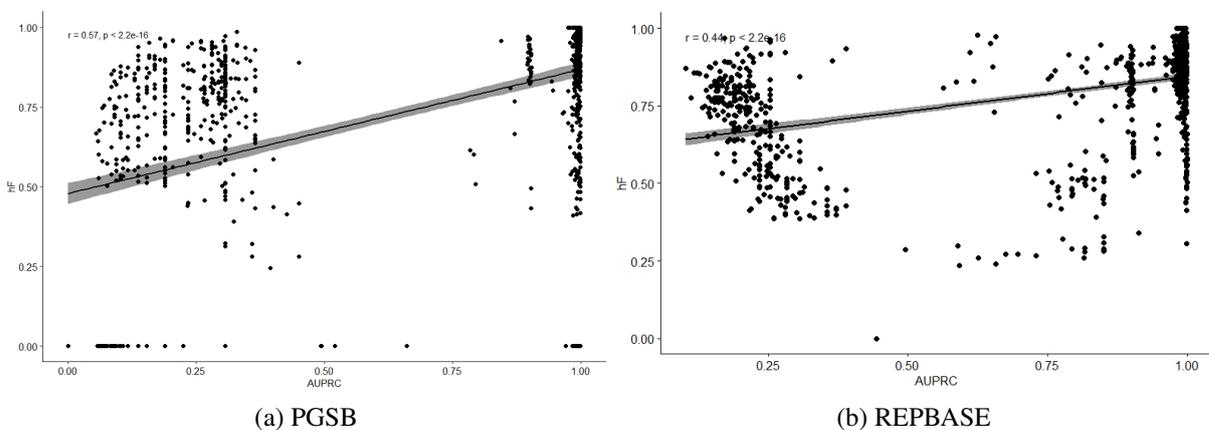


Figura 34 – Correlações dos objetivos  $AU(\overline{PRC})$  e **hF** utilizando o **HC-LGA**.

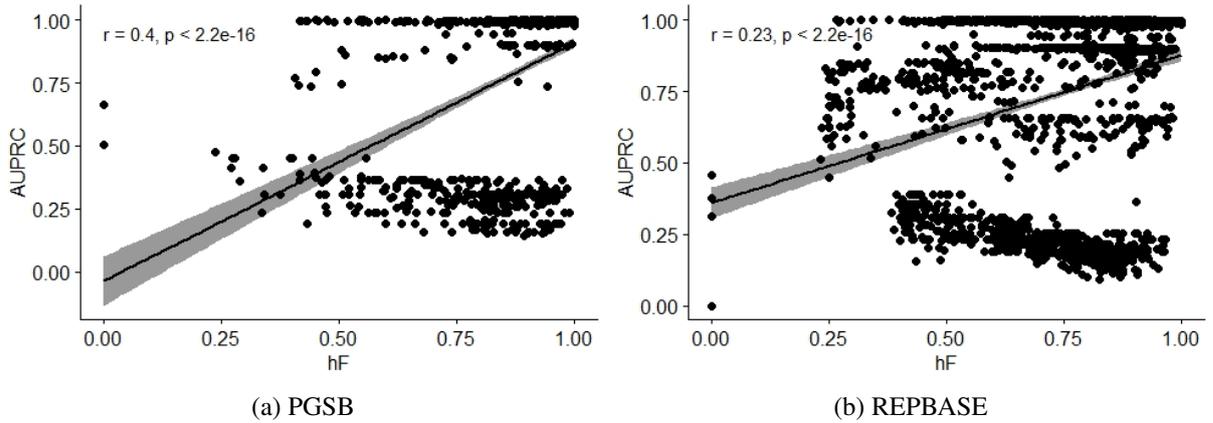


Figura 35 – Correlações dos objetivos  $hF$  e  $AU(\overline{PRC})$  utilizando o  $HC-LGA$ .

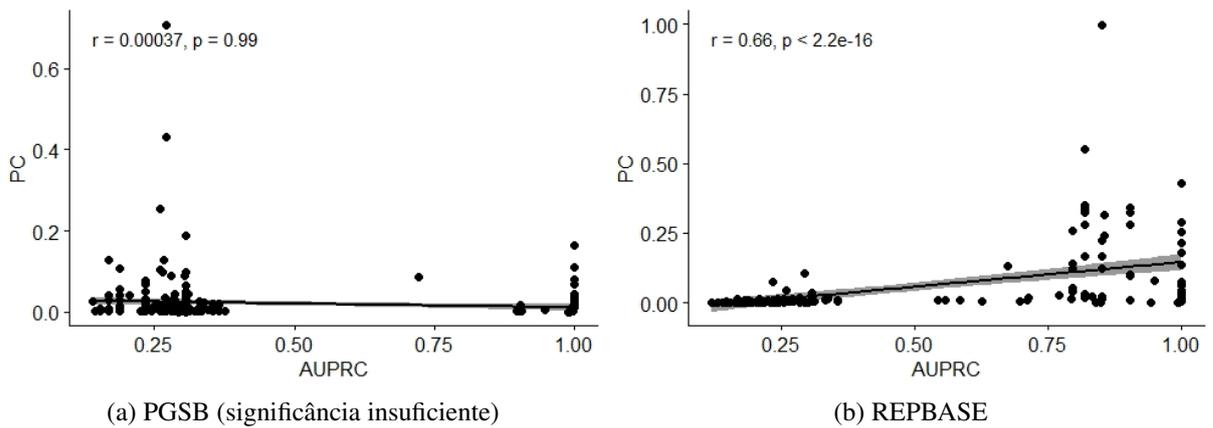


Figura 36 – Correlações dos objetivos  $AU(\overline{PRC})$  e  $PC$  utilizando o  $HC-LGA$ .

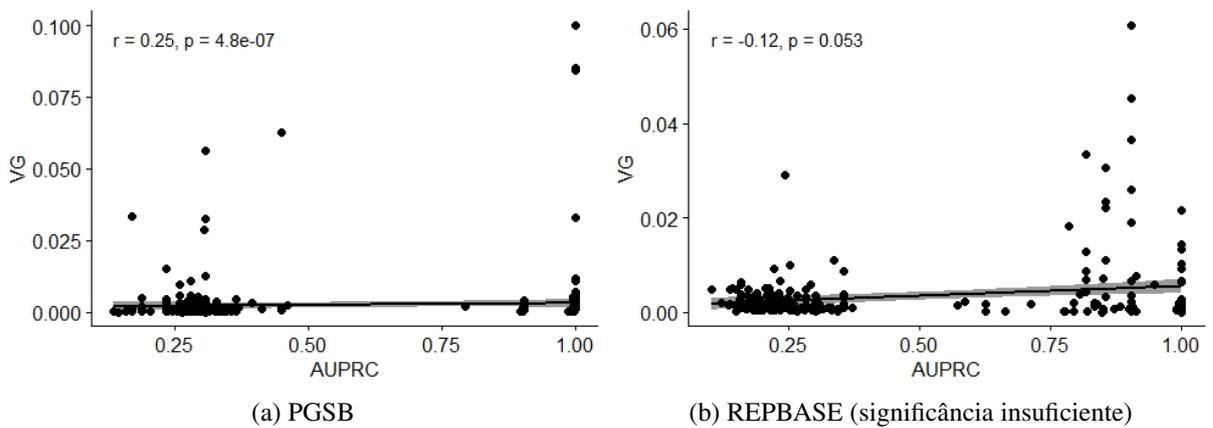


Figura 37 – Correlações dos objetivos  $AU(\overline{PRC})$  e  $VG$  utilizando o  $HC-LGA$ .

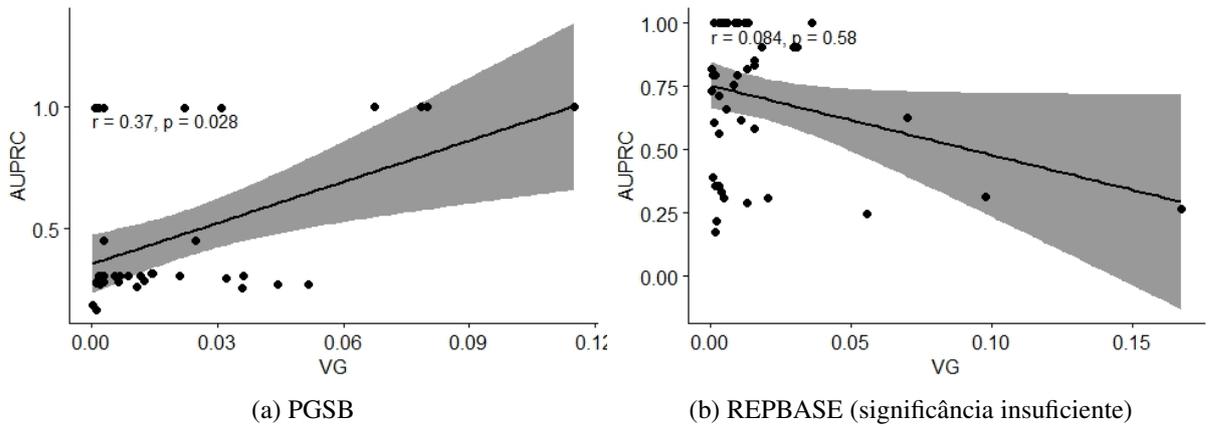


Figura 38 – Correlações dos objetivos VG e  $AU(\overline{PRC})$  utilizando o HC-LGA.

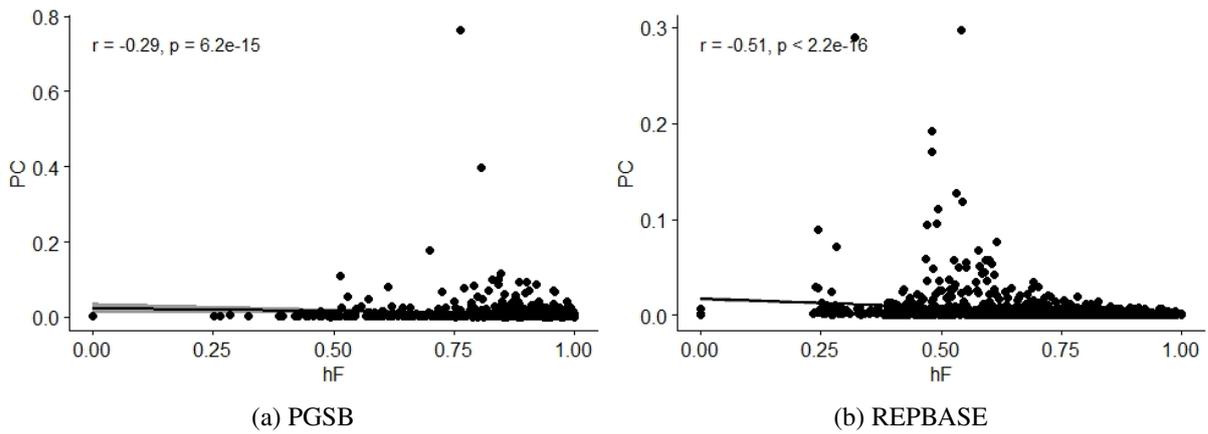


Figura 39 – Correlações dos objetivos hF e PC utilizando o HC-LGA.

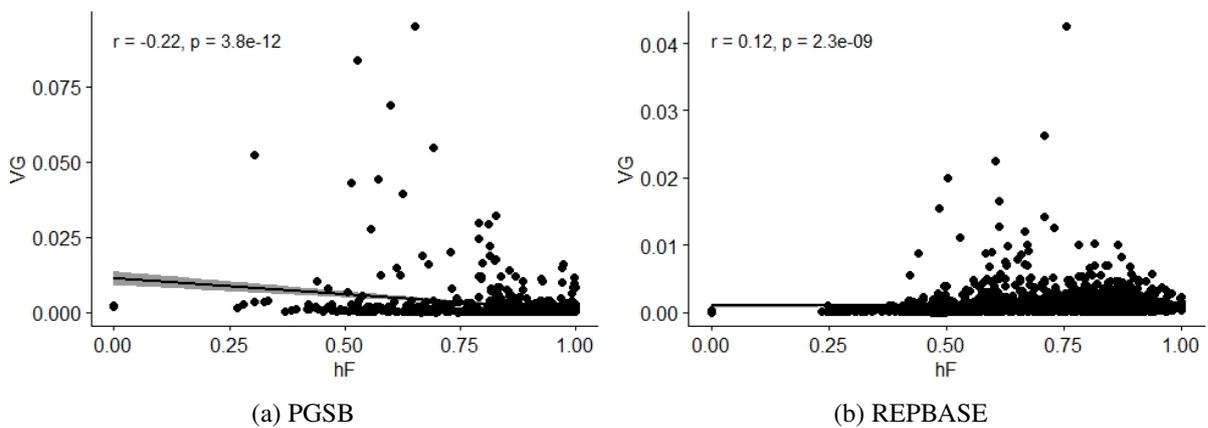


Figura 40 – Correlações dos objetivos hF e VG utilizando o HC-LGA.

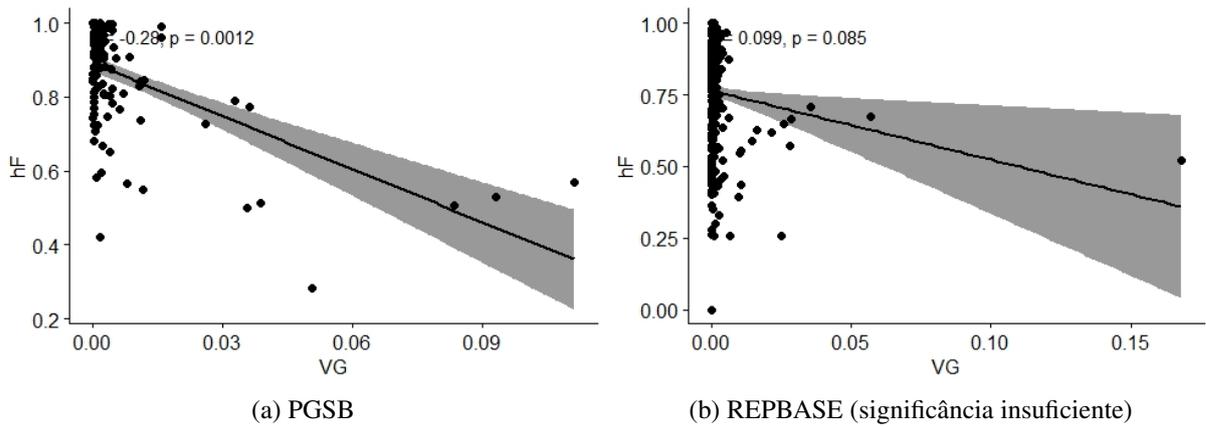


Figura 41 – Correlações dos objetivos VG e hF utilizando o HC-LGA.

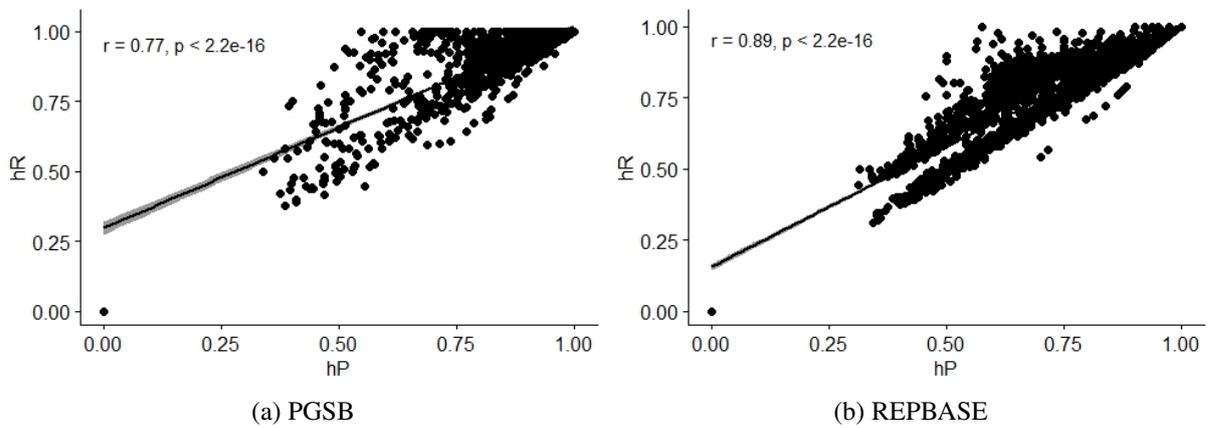


Figura 42 – Correlações dos objetivos hP e hR utilizando o HC-LGA.

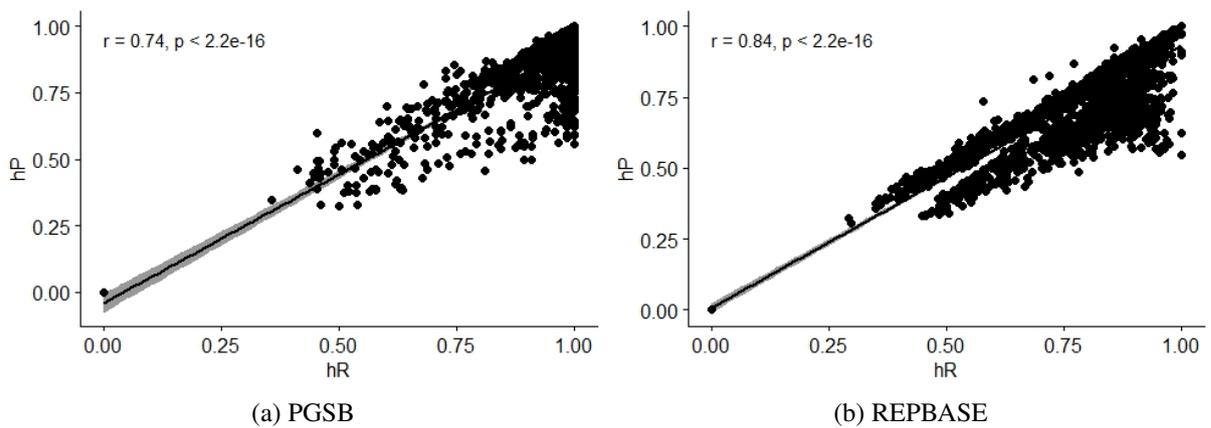


Figura 43 – Correlações dos objetivos hR e hP utilizando o HC-LGA.

Uma compilação das correlações entre as funções objetivo testadas no HC-LGA é exibida na Tabela 18, onde assim como na Tabela 17, os campos “x” representam a não obtenção de significância estatística por parte de determinada otimização, não podendo assim ser validada a correlação entre os objetivos considerados. No entanto, os gráficos Q-Q das amostras que

falharam na prova de significância podem ser consultadas no Apêndice A.

Tabela 18 – Correlações dos objetivos testados no HC-LGA.

	$AU(\overline{PRC})$	<b>hF</b>	<b>hP</b>	<b>hR</b>	<b>PC</b>	<b>VG</b>
PGSB						
<b>AU(<math>\overline{PRC}</math>)</b>	-	0.57	-	-	x	0.25
<b>hF</b>	0.40	-	-	-	<b>-0.29</b>	-0.22
<b>hP</b>	-	-	-	<b>0.77</b>	-	-
<b>hR</b>	-	-	0.74	-	-	-
<b>PC</b>	x	x	-	-	-	x
<b>VG</b>	0.37	-0.28	-	-	x	-
REPBASE						
<b>AU(<math>\overline{PRC}</math>)</b>	-	0.44	-	-	0.66	x
<b>hF</b>	0.23	-	-	-	<b>-0.51</b>	0.12
<b>hP</b>	-	-	-	<b>0.89</b>	-	-
<b>hR</b>	-	-	0.84	-	-	-
<b>PC</b>	x	x	-	-	-	x
<b>VG</b>	x	x	-	-	0.70	-

Quanto aos experimentos com o **HC-LGA**, houveram menos casos de correlações que não atingiram significância estatística, sendo cinco delas no PGSB e seis no REPBASE, a maioria envolvendo as funções **PC** e **VG**. Ainda que em menor quantidade, também ocorreram casos de correlações validadas em apenas um dos conjuntos de dados, como são os casos de  $AU(\overline{PRC})$  e **PC**,  $AU(\overline{PRC})$  e **VG**, **VG** e  $AU(\overline{PRC})$ , **VG** e **hF**, e por fim, **VG** e **PC**. Ademais, foi observado que a utilização de **PC** como objetivo prioritário aplicado aos conjuntos PGSB e REPBASE, resultou em amostras muito pequenas e pouco representativas, o que inviabilizou as análises de correlação com as demais funções. O último ponto importante discutido na análise anterior, que diz respeito as variações nos sentidos das correlações, voltou a ser observado, colaborando para a conclusão de que o uso de **VG** acarreta em anomalias nas correlações.

Tanto no PGSB quanto no REPBASE as correlações mais fortes (e positivas) foram obtidas por **hP** e **hR** e em seguida por **hR** e **hP**. Portanto, assim como é visto com outras correlações como é o caso de  $AU(\overline{PRC})$  e **hF**, que apresentam maior correlação do que **hF** e  $AU(\overline{PRC})$ , a ordem de importância definida para a otimização interfere na correlação das funções objetivo. Desse modo, é importante analisar quais as melhores ordens para que não haja execuções ineficientes. A respeito das correlações negativas mais fortes, essas foram alcançadas por **hF** e **PC** em ambos os conjuntos de dados. Infelizmente, não foi possível verificar se a ordem contrária dessa otimização apresenta mudanças nos coeficientes, já que as execuções usando **PC** como objetivo prioritário falharam nos testes de significância estatística.

Realizando um apanhado geral das correlações apresentadas nas Tabelas 17 e 18, e levando em consideração somente os resultados confiáveis, tem-se que  $AU(\overline{PRC})$  e **hF** possuem

coeficientes de correlação positivos variando entre [0.23, 0.57], ou em termos qualitativos, apresentam uma correlação fraca ou moderada. Já  $AU(\overline{PRC})$  e **PC** possuem coeficientes entre [0.38, 0.81], podendo essa ser uma correlação fraca, moderada ou até mesmo forte dependendo da amostra. Por outro lado, **hF** e **PC** mostram coeficientes de correlação negativos entre [-0.11, -0.55], sendo essa uma correlação muito fraca, fraca ou moderada, enquanto que **hP** e **hR** apresentam uma correlação forte ou muito forte, com coeficientes variando entre [0.74, 0.89].

## 8.5 Comparações entre os Métodos Propostos

Antes de realizar a comparação dos resultados gerados pelos métodos propostos com os métodos da literatura, foi preciso chegar a um consenso acerca de quais os melhores resultados dentre os diversos gerados e discutidos nos experimentos das Seções 8.1, 8.2 e 8.3. Além disso, como uma das visadas contribuições dessa pesquisa é verificar se há ganhos na implementação de Abordagens Multiobjetivo para o problema abordado, é importante comparar as duas soluções multiobjetivo propostas com o também proposto método de otimização simples. Portanto, essa seção aborda justamente a definição dos melhores resultados atingidos por cada método, considerando todos os experimentos realizados com as diversas funções apresentadas e tendo em mente os *trade-offs* entre desempenho preditivo, interpretabilidade e tempo de treinamento, além de ser realizada a comparação direta entre os métodos propostos.

Para cada método proposto foram realizadas três diferentes análises, uma voltada ao desempenho das previsões, outra focada na interpretabilidade dos modelos gerados e uma última que se atendeu aos tempos de treinamento. Como vários critérios foram avaliados, para que se chegasse a uma conclusão justa do melhor resultado atingido e coerente com as preferências dessa pesquisa, definiu-se uma ordem de importância dos critérios considerando as três análises, sendo essa: **Geral** > **Nível 1** > **Classes Folha** > **Nível 2** > **Nível 3** > **Nível 4** > **Regras** > **Testes por Regra** > **Testes Ativos** > **Tempo de Treinamento**. Portanto, assim como ocorre frequentemente em trabalhos de *Data Mining*, deu-se preferência a critérios de desempenho preditivo em vez de interpretabilidade, pois apesar de ser preferível ter um modelo interpretável, essa preocupação vem em segundo plano. Assim, comparou-se os valores de cada experimento em ordem de importância e considerando os resultados do teste estatístico de Wilcoxon. Esse teste foi aplicado a fim de verificar a existência de diferenças significativas entre experimentos e, deste modo, definir os melhores resultados com certo viés estatístico.

A partir dos experimentos com o **HC-GA** vistos na Tabela 8, tem-se as funções **hP**, **hR** e **hF** empatadas como as melhores execuções já que não houveram diferenças significativas entre elas. Quanto as demais funções, essas diferiram com relevância estatística e, portanto, foram consideradas inferiores. As funções hierárquicas voltaram a empatar dentro do intervalo de significância estatística quando levado em conta a análise de interpretabilidade dos modelos, apresentada na Tabela 9. No entanto, a função **hF** foi considerada como a melhor execução

quando considerado os tempos de treinamento apresentados na Tabela 10. Esse é o último critério na ordem de importância previamente apresentada, e onde **hF** se mostrou estatisticamente melhor que os demais, obtendo na média dos dois conjuntos um tempo de 399 minutos, enquanto que o segundo colocado **hR**, levou 634 minutos para gerar induzir os modelos de regras.

Quanto aos resultados do **HC-WGA**, foram seis execuções empatadas estatisticamente considerando a análise de desempenho preditivo apresentada na Tabela 11, sendo essas a fórmula ponderada **hF** (80%) + **PC** (20%) e todas as cinco ponderações de pesos para a combinação **hF** + **VG**. Quando comparados os resultados dessas execuções na análise de interpretabilidade vista na Tabela 12, observou-se uma diferença significativamente menor no número de regras gerado pela fórmula **hF** (20%) + **VG** (80%) em relação as demais, que na média dos dois conjuntos, induziu 1430 regras em oposição as 1711 regras do segundo colocado **hF** (40%) + **VG** (60%), o que resultou em sua definição como a melhor execução para o **HC-WGA**.

Já em relação aos experimentos com o **HC-LGA**, os melhores resultados de desempenho preditivo vistos na Tabela 14 considerando tanto o PGSB quanto o REPBASE, foram obtidos pelas otimizações em conjunto das funções **hP** e **hR**, **hR** e **hP**, **hF** e **PC**, **hF** e **VG**, e por fim, **hF** e  $AU(\overline{PRC})$ , todas essas empatadas segundo o intervalo de significância estatística. Sendo assim, analisou-se o critério seguinte na ordem de importância: os números de regras gerados por essas execuções. Segundo a análise de interpretabilidade exibida na Tabela 15, observou-se que a otimização de **hF** e **PC** gerou 1543 regras na média entre os conjuntos, obtendo superioridade estatística em relação ao segundo colocado **hF** e **VG**, que gerou 1698 regras.

Portanto, como configurações finais de cada método tem-se **HC-GA** usando **hF** como função de aptidão, **HC-WGA** com a fórmula ponderada **hF** (20%) + **VG** (80%) e, **HC-LGA** com as funções **hF** e **PC**. A Tabela 19 faz um apanhado geral dos critérios avaliados, de forma que contabiliza uma pontuação final para determinar o melhor método de otimização e, para isso, considera-se apenas a média dos melhores resultados nos dois conjuntos de dados. Cabe salientar que quando há um empate entre métodos, ambos recebem +1 em sua pontuação final, ademais, apenas o melhor resultado pontua. Os significados dos símbolos utilizados são descritos abaixo:

- ▲: Melhor resultado.
- △: Segundo melhor resultado.
- ▷: Empate.
- ▽: Pior resultado.

Tabela 19 – Comparação de desempenho entre os métodos propostos.

	Geral	N1	N2	N3	N4	Folhas	Regras	Testes	Testes p/ Regra	Tempo	Total
HC-GA	▷	▷	▷	▷	▲	▷	▽	▽	▷	△	6
HC-WGA	▽	▷	▽	▽	▽	▽	▲	▲	▷	▽	3
HC-LGA	▷	▲	▷	▷	△	▷	△	△	▷	▲	7

Cabem algumas considerações a respeito dos resultados mostrados na Tabela 19. Tomando como base somente a pontuação final (coluna **Total**), em que todos os critérios têm a mesma relevância e aonde foram comparados os valores médios sem a aplicação de um teste estatístico, tem-se o **HC-LGA** como o melhor método já que apresentou os valores mais altos, seguido do **HC-GA** em segundo e o **HC-WGA** em último lugar. No entanto, aplicando testes estatísticos nos valores médios e seguindo a ordem de importância definida, o **HC-WGA** aparece em primeiro lugar visto que gerou um número significativamente menor de regras em relação ao segundo colocado **HC-LGA**. Esse por sua vez também mostrou vantagem estatística quanto ao número de regras induzidas em comparação com o **HC-GA**. Desse modo, pode-se inferir que as Abordagens Multiobjetivo configuram boas soluções ao problema da indução de regras para a **HC** de **TEs**, mostrando inclusive superioridade estatística ao método de otimização simples.

## 8.6 Comparações com os Métodos da Literatura

Nessa seção são comparados os melhores resultados alcançados pelos métodos propostos juntamente com diversos métodos da literatura aplicados aos conjuntos PGSB e REPBASE, os quais foram organizados de acordo com a taxonomia de [Wicker et al. \(2007\)](#). Nesses comparativos houve a preocupação de incluir um exemplo de cada abordagem e estratégia hierárquica, bem como exemplares não-hierárquicos e tradicionalmente usados no domínio, para que dessa forma, fosse realizada uma análise mais ampla e representativa do estado-da-arte. Como representante das ferramentas utilizadas em problemas de Bioinformática, adotou-se os populares métodos baseados em homologia BLASTn e RepeatMasker, escolhidos por serem mais genéricos e não específicos para determinadas famílias de **TEs** como acontece com outras ferramentas apresentadas no Capítulo 2. Quanto aos demais métodos, esses seguem a Abordagem Global ou uma das três estratégias da Abordagem Local. Em conformidade com a Abordagem Global tem-se o método Clus-HMC, além dos métodos propostos **HC-GA**, **HC-WGA** e **HC-LGA**. Oriundos da estratégia **LCN**, foram incluídos os métodos apresentados em [Santos et al. \(2018\)](#) que utilizam de classificadores tradicionais como base, sendo eles: **C4.5**, **RF**, **NB**, **KNN**, **MLP** e **SVM**. Da estratégia **LCPN**, foram incorporados os métodos nLLCPN e LCPNB, enquanto que da **LCL** foram inclusos os métodos LCL+SWN e LCL+SP, todos propostos em [Nakano et al. \(2017\)](#).

Cabe salientar que dentre os métodos da literatura supracitados, somente o Clus-HMC foi executado e suas predições obtidas, para que assim fossem submetidas as medidas de avaliação hierárquica assim como fora feito com os métodos propostos. Em relação aos demais

métodos, esses foram apenas referenciados bem como seus resultados já reportados em trabalhos da literatura. Como as saídas ou predições do Clus-HMC são valores reais entre  $[0, 1]$  que representam as probabilidades de associação das classes, tal como ocorre com os métodos propostos, foi aplicado o mesmo limiar de 0.5 para obter as classificações finais.

Na Tabela 20, são apresentados os resultados da medida-F hierárquica por nível e considerando toda a hierarquia de classes, assim como os desvios padrão gerados por BLASTn e RepeatMasker reportados originalmente em Nakano (2018). Cabe ressaltar que os valores da coluna “Não classificado” correspondem a porcentagem de instâncias no qual os métodos não foram capazes de classificar. Já na Tabela 21, são exibidas apenas os resultados gerados pelos métodos hierárquico considerando as três medidas hierárquicas aplicadas a toda a hierarquia de classes, juntamente com seus valores de desvio padrão.

Tabela 20 – Resultados dos métodos de Homologia considerando a Medida-F Hierárquica.

Método	Nível 1	Nível 2	Nível 3	Nível 4	Geral	Não Classificado
PGSB						
BLASTn	0.90 ± 0.01	0.88 ± 0.01	<b>0.73 ± 0.02</b>	0.34 ± 0.05	0.80 ± 0.01	<b>0.55 ± 0.01</b>
RepeatMasker	<b>0.91 ± 0.07</b>	<b>0.91 ± 0.08</b>	0.72 ± 0.01	<b>0.47 ± 0.06</b>	<b>0.81 ± 0.01</b>	0.58 ± 0.08
REPBASE						
BLASTn	0.68 ± 0.08	0.68 ± 0.08	0.56 ± 0.06	0.21 ± 0.15	0.54 ± 0.06	0.91 ± 0.02
RepeatMasker	<b>0.68 ± 0.01</b>	<b>0.78 ± 0.01</b>	<b>0.70 ± 0.02</b>	<b>0.62 ± 0.08</b>	<b>0.68 ± 0.02</b>	<b>0.60 ± 0.01</b>

Tabela 21 – Comparações entre métodos hierárquicos considerando medidas hierárquicas.

	hP	hR	hF	hP	hR	hF
	PGSB			REPBASE		
HC-GA	0.85 ± 0.008	<b>0.89 ± 0.006</b>	<b>0.87 ± 0.007</b>	0.78 ± 0.007	0.79 ± 0.007	0.78 ± 0.007
HC-WGA	0.86 ± 0.006	0.88 ± 0.007	<b>0.87 ± 0.006</b>	0.77 ± 0.005	0.77 ± 0.007	0.77 ± 0.005
HC-LGA	0.86 ± 0.008	0.88 ± 0.009	<b>0.87 ± 0.008</b>	0.78 ± 0.004	0.79 ± 0.006	0.78 ± 0.005
Clus-HMC	0.87 ± 0.004	0.83 ± 0.005	0.85 ± 0.004	0.76 ± 0.016	0.67 ± 0.008	0.71 ± 0.006
LCN-C4.5	0.84 ± 0.020	0.83 ± 0.027	0.83 ± 0.023	0.76 ± 0.043	0.56 ± 0.031	0.65 ± 0.035
LCN-RF	0.87 ± 0.028	0.86 ± 0.028	<b>0.87 ± 0.028</b>	0.67 ± 0.036	0.62 ± 0.045	0.64 ± 0.039
LCN-NB	0.59 ± 0.134	0.51 ± 0.135	0.55 ± 0.136	0.52 ± 0.067	0.49 ± 0.073	0.51 ± 0.070
LCN-KNN	<b>0.90 ± 0.027</b>	0.83 ± 0.028	0.86 ± 0.025	0.69 ± 0.048	0.58 ± 0.065	0.63 ± 0.055
LCN-MLP	0.84 ± 0.014	0.86 ± 0.018	0.85 ± 0.015	0.69 ± 0.042	0.62 ± 0.044	0.66 ± 0.040
LCN-SVM	<b>0.90 ± 0.014</b>	0.67 ± 0.016	0.76 ± 0.014	0.78 ± 0.044	0.54 ± 0.033	0.64 ± 0.036
nLLCPN	0.83 ± 0.031	0.85 ± 0.025	0.84 ± 0.027	<b>0.83 ± 0.010</b>	0.84 ± 0.007	<b>0.84 ± 0.005</b>
LCPNB	0.84 ± 0.017	0.86 ± 0.015	0.85 ± 0.013	<b>0.83 ± 0.011</b>	0.85 ± 0.012	<b>0.84 ± 0.010</b>
LCL+SWV	0.79 ± 0.008	0.86 ± 0.015	0.82 ± 0.011	0.78 ± 0.006	0.84 ± 0.004	0.81 ± 0.005
LCL+SP	0.80 ± 0.009	0.88 ± 0.018	0.84 ± 0.012	0.80 ± 0.005	<b>0.87 ± 0.005</b>	0.83 ± 0.005

Conforme é visto nos resultados da Tabela 20, é alta a quantidade de instâncias não classificadas pelo BLASTn em ambos os conjuntos de dados. Essa quantidade aumenta consideravelmente do PGSB para o REPBASE, subindo de 55% para mais de 90% do total de instâncias não classificadas, o que pode ser um indício de que a ferramenta tem problemas em lidar com maiores quantidades de dados. Além disso, é expressiva a deterioração no desempenho preditivo

de um conjunto para o outro mesmo classificando uma taxa menor de instâncias, passando de 0.81 no PGSB para 0.54 no REPBASE. Por sua vez, os resultados do RepeatMasker foram ligeiramente superiores aos do BLASTn no PGSB, mostrando inclusive uma menor taxa de instâncias não classificadas. As exceções foram para os Níveis 3 e 4, sendo a diferença consideravelmente maior nesse último. Em relação ao REPBASE, o RepeatMasker alcançou resultados significativamente superiores em todos os critérios, com exceção para o Nível 1 ao qual o desempenho foi o mesmo do BLASTn. Outro ponto interessante fora que o RepeatMasker manteve certa consistência de um conjunto de dados para o outro, apresentando uma porcentagem de instâncias não classificadas bem semelhante, diferente do que ocorreu com o BLASTn.

Ainda que considerado somente as baixas taxas de instâncias classificadas pelas ferramentas de homologia, resultados ruins foram obtidos para a medida-F hierárquica, sendo muito inferiores quando comparados aos resultados dos métodos hierárquicos apresentados na Tabela 21. Nesses, percebe-se que o único método a mostrar valores mais baixos que os dessas ferramentas foi o LCN-NB (fundamentado no Naive Bayes), no entanto, pelo fato de métodos baseados em ML não sofrem da desvantagem de não classificar instâncias como ocorre com os baseados em homologia, pode se considerar que o LCN-NB ainda é superior.

Exclusivamente sobre os resultados expressos na Tabela 21, ficaram empatados com os valores mais altos no PGSB, os métodos globais HC-GA, HC-WGA, HC-LGA e os locais LCN-RF e LCN-KNN, esse último considerado tecnicamente equiparável aos supracitados. Com exceção de LCN-NB e LCN-SVM, os demais métodos mostraram resultados semelhantes no PGSB. Com relação ao REPBASE, o mesmo comportamento não se repetiu, visto que os melhores resultados gerados por nLLCPN e LCPNB além de LCL+SP, considerados estatisticamente equiparáveis, mostraram valores bem distantes dos demais. O método LCL-SWV alcançou o segundo melhor resultado, já os métodos propostos HC-GA, HC-WGA e HC-LGA obtiveram somente os terceiros melhores valores, não mantendo o bom desempenho alcançado no PGSB.

Em relação as comparações de desempenho entre abordagens hierárquicas, os métodos exemplos de LCPN e LCL foram os únicos a mostrar uma estabilidade em seus resultados nos dois conjuntos, apresentando valores bastante parecidos entre seus semelhantes. Apesar de não apresentarem tal comportamento, os métodos globais, principalmente os propostos, se mostraram equiparáveis entre si em cada conjunto. Além disso, quanto os valores de desvio padrão, os maiores foram obtidos pelos métodos LCN enquanto que os menores foram dos métodos globais.

Na Tabela 22 é apresentado um comparativo entre o desempenho dos métodos globais propostos nessa pesquisa com classificadores clássicos que operam como métodos *flat*. Esses experimentos foram realizados com o intuito de verificar o quão hábil são os métodos hierárquicos em predizer as classes folhas da hierarquia e, para isso, se comparou suas predições com as de classificadores específicos para tal tarefa. Conseqüentemente, não foram utilizadas medidas hierárquicas para avaliar os resultados e sim medidas tradicionais como Precisão (P), Revocação (R) e Medida-F (F). Com exceção dos métodos multiobjetivo HC-WGA e HC-LGA, os demais

resultados foram previamente publicados em [Pereira e Cerri \(2017\)](#), sendo referenciados nesse comparativo. No entanto, cabe ressaltar que os resultados reportados para o [HC-GA](#) foram aprimorados durante essa pesquisa de mestrado, saltando de 0.50 para 0.83 no PGSB e de 0.23 para 0.69 no REPBASE, considerando a Medida-F. Adicionalmente, julgamos que tais experimentos foram pertinentes uma vez que as classes folhas de uma hierarquia detêm o conhecimento mais específico a respeito das classes.

Tabela 22 – Resultados usando medidas não-hierárquicas. Adaptado de [Pereira e Cerri \(2017\)](#).

	C4.5	NB	KNN	SVM	MLP	HC-GA	HC-WGA	HC-LGA
PGSB								
P	0.66 ± 0.011	0.45 ± 0.041	<b>0.80 ± 0.009</b>	0.68 ± 0.022	0.41 ± 0.141	<b>0.80 ± 0.010</b>	<b>0.80 ± 0.006</b>	<b>0.80 ± 0.007</b>
R	0.66 ± 0.011	0.20 ± 0.009	0.80 ± 0.009	0.69 ± 0.012	0.30 ± 0.110	<b>0.87 ± 0.009</b>	0.86 ± 0.010	0.86 ± 0.011
F	0.66 ± 0.011	0.28 ± 0.016	0.80 ± 0.009	0.68 ± 0.017	0.32 ± 0.072	<b>0.83 ± 0.009</b>	<b>0.83 ± 0.007</b>	<b>0.83 ± 0.007</b>
REPBASE								
P	0.46 ± 0.008	0.38 ± 0.111	<b>0.69 ± 0.006</b>	0.56 ± 0.013	0.24 ± 0.086	0.68 ± 0.008	0.68 ± 0.006	0.68 ± 0.008
R	0.46 ± 0.007	0.11 ± 0.003	0.68 ± 0.007	0.49 ± 0.004	0.22 ± 0.056	<b>0.70 ± 0.007</b>	0.68 ± 0.013	<b>0.70 ± 0.011</b>
F	0.46 ± 0.007	0.16 ± 0.014	<b>0.69 ± 0.006</b>	0.52 ± 0.008	0.22 ± 0.051	<b>0.69 ± 0.007</b>	0.68 ± 0.008	<b>0.69 ± 0.009</b>

Em [Pereira e Cerri \(2017\)](#), os classificadores tradicionais mostraram resultados superiores aos apresentados pelo [HC-GA](#), em especial o [KNN](#), que gerou os melhores resultados em todos os critérios avaliados e para ambos os conjuntos de dados. Nos presentes experimentos com o [HC-GA](#) aprimorado, o método hierárquico alcançou resultados superiores ou competitivos com o [KNN](#), até então o melhor método a realizar uma classificação *flat* nesses conjuntos. O [HC-GA](#) juntamente com os métodos multiobjetivo [HC-WGA](#) e [HC-LGA](#) apresentaram os melhores resultados para o PGSB, empatando com o [KNN](#) quanto a Precisão mas lhe superando quanto a Revocação e a Medida-F. No REPBASE os resultados foram mais semelhantes. Os três métodos propostos mostraram valores ligeiramente inferiores ao [KNN](#) na Precisão, em contrapartida, [HC-GA](#) e [HC-LGA](#) mostraram resultados superiores na Revocação enquanto que o [HC-WGA](#) empatou. Por fim, considerando os valores de Medida-F houve um empate entre [KNN](#), [HC-GA](#) e [HC-LGA](#), configurando-as como soluções equiparáveis.

Visando verificar o quão interpretáveis são as regras geradas pelos métodos propostos, essas foram comparadas com as regras induzidas pelo método global considerado estado-da-arte para [HC](#), denominado Clus-HMC. Na Tabela 23, são avaliados e comparados os números de regras geradas, quantos testes estão ativos nessas regras e a média de testes por regra.

Tabela 23 – Análise de interpretabilidade das regras geradas por métodos globais.

Método	Regras	Testes Ativos	Testes por Regra	Regras	Testes Ativos	Testes por Regra
PGSB						
HC-GA	1180	8749	7	2507	19889	8
HC-WGA	<b>1037</b>	<b>7438</b>	7	<b>2397</b>	<b>18640</b>	8
HC-LGA	1151	8427	7	2467	19301	8
Clus-HMC	1375	18103	13	2433	29941	11
REPBASE						

Conforme essas comparações, os métodos propostos apresentaram resultados relativamente semelhantes entre si, obtendo inclusive o mesmo número de testes por regra em cada conjunto de dados. Em relação ao Clus-HMC, esse apresentou modelos menos interpretáveis em comparação com o [HC-GA](#), [HC-WGA](#) e [HC-LGA](#), que lhe superaram com significância estatística em todos os critérios avaliados e nos dois conjuntos. Em suma, o [HC-WGA](#) obteve os melhores resultados tanto no PGSB quanto no REPBASE, mostrando inclusive superioridade estatística em relação aos demais. Curiosamente, esse método gerou os piores resultados quanto ao desempenho preditivo, reforçando ainda mais a assertiva de que quando um modelo apresenta bom desempenho esse acaba sacrificando sua interpretabilidade e vice-versa.

# Capítulo 9

## CONSIDERAÇÕES FINAIS

---

---

Nesta Dissertação de Mestrado foram propostos e investigados três métodos hierárquicos globais que induzem listas de regras de **HC**, denominados **Hierarchical Classification with a Genetic Algorithm (HC-GA)**, **Hierarchical Classification with a Weighted Genetic Algorithm (HC-WGA)** e **Hierarchical Classification with a Lexicographic Genetic Algorithm (HC-LGA)**. Esses métodos baseiam-se em Algoritmos Genéticos para gerar e evoluir regras que classificam instâncias em múltiplos níveis de uma hierarquia de classes estruturada em Árvore. Enquanto que o **HC-GA** constitui um método de busca e otimização simples, os métodos **HC-WGA** e **HC-LGA** implementam as Abordagens Multiobjetivo de Soma Ponderada e Lexicográfica, respectivamente. Essas soluções multiobjetivo foram propostas com a intenção de endereçar e maximizar os objetivos de desempenho preditivo e interpretabilidade nos modelos gerados, qualidades fortemente desejadas em soluções de **ML**, *Data Mining* e também na Bioinformática.

Os métodos propostos foram aplicados a **HC** de instâncias de **Transposable Element (TE)**s oriundas dos dois conjuntos de dados públicos PGSB e REPBASE, estruturados segundo a taxonomia hierárquica de **TEs** desenvolvida por [Wicker et al. \(2007\)](#). Diversos experimentos foram realizados com esses métodos, os quais podem ser sumarizados em: (i) validação do desempenho preditivo geral, nas classes folha e nos níveis hierárquicos da hierarquia de **TEs**; (ii) análise de interpretabilidade das listas de regras induzidas segundo os critérios de número de regras, número de testes ativos e a média de testes por regra; e (iii) os tempos de treinamento requeridos por cada configuração de função ou funções objetivo (multiobjetivo). Adicionalmente, foram realizadas comparações entre os métodos propostos conforme todos os critérios avaliados anteriormente de forma isolada. Por fim, as últimas comparações foram feitas com uma série de métodos hierárquicos representantes de cada abordagem e estratégia de **HC** da literatura, juntamente com duas populares ferramentas de Bioinformática baseadas em homologia.

Para a avaliação apropriada das predições dos métodos de **HC**, usou-se as medidas **hierarchical Precision (hP)**, **hierarchical Recall (hR)** e **hierarchical F-measure (hF)**. Além disso, visando garantir a significância estatística dos resultados, foram aplicados testes estatístico de Wilcoxon nas comparações entre funções e entre métodos. Como essa pesquisa também focou na investigação de abordagens multiobjetivo, realizou-se testes de correlação de Pearson e Spearman

entre funções objetivo, utilizando como ferramenta gráfica de inspeção os gráficos Q-Q. No restante deste capítulo são apresentadas as contribuições da pesquisa e os trabalhos futuros.

## 9.1 Contribuições da Pesquisa

Ao decorrer deste trabalho foram gerados materiais aos quais consideramos como contribuições válidas para a literatura e o avanço no estado-da-arte dos domínios abordados. Dentre essas, destacamos como principais contribuições as propostas e investigações dos três métodos hierárquicos globais **HC-GA**, **HC-WGA** e **HC-LGA**. Esses métodos já se encontram disponíveis e abertos ao público<sup>1,2</sup> para utilização e/ou colaboração. Ademais, não se tem conhecimento nem foram encontrados trabalhos que proponham métodos globais hierárquicos baseados em Algoritmos Genéticos Multiobjetivo, ou ainda, que apliquem abordagens multiobjetivo a problemas hierárquicos, tampouco endereçando a classificação de **TEs**.

Uma contribuição para o nicho de **HC** foram os comparativos realizados com uma gama de métodos oriundos de diferentes abordagens e estratégias hierárquicas. Apesar desses comparativos fazerem parte de um escopo fechado que é a **HC** de **TEs**, as análises e observações feitas podem ser úteis ao desenvolvimento de trabalhos que também visam aplicar abordagens hierárquicas a dados com hierarquias semelhantes, ou ainda, que se proponham a investigar a implicação dessas abordagens a aspectos de qualidade dos modelos, como o desempenho preditivo, interpretabilidade ou tempo de treinamento.

Também houveram contribuições ao domínio da Bioinformática e as pesquisas voltadas ao estudo de **TEs**. Através das comparações dos métodos hierárquicos com duas ferramentas de homologia frequentemente adotadas em problemas de Bioinformática, pode se observar as desvantagens dessas em relação aos métodos baseados em **ML**. Dentre essas estão a não indução de modelos de inferência a partir dos dados, a incapacidade de classificar instâncias em alguns casos assim como as altas taxas de instâncias não classificadas, além do desempenho preditivo ruim. Adicionalmente, os métodos de homologia ignoram as relações hierárquicas entre classes, e como já fora mostrado na literatura, a utilização de tais informação é benéfica (**SILLA; FREITAS, 2010**). Além disso, esse trabalho apresentou métodos que classificam **TEs** utilizando a popular e, relativamente recente, taxonomia hierárquica de classes de **TEs** proposta por **Wicker et al. (2007)**, que até então não havia sido utilizada por métodos de **HC** globais.

Outra contribuição que julgamos relevante foram os comparativos entre o método de otimização simples **HC-GA** e os métodos multiobjetivo **HC-WGA** e **HC-LGA**. Essas comparações tiveram como propósito a investigação das vantagens e desvantagens da utilização de abordagens multiobjetivo em problemas hierárquicos, usando como “estudo de caso” o problema da **HC** de **TEs**. Como visto nas análises realizadas, os métodos multiobjetivo superaram o método de

<sup>1</sup> <https://github.com/geantrindade/HC-GA>

<sup>2</sup> <https://github.com/geantrindade/HC-LGA>

otimização simples com significância estatística considerando os critérios avaliados, sendo esses o desempenho preditivo analisado usando medidas hierárquicas, a interpretabilidade avaliada conforme a complexidade do modelo, e o tempo de treinamento. Acreditamos que tais comparativos sirvam para evidenciar que a investigação de abordagens multiobjetivo possui potencial de aplicação não só em problemas hierárquicos, mas também em problemas que exigem modelos com bom desempenho preditivo e interpretabilidade.

Por fim, consideramos como contribuições as revisões da literatura apresentadas nesse documento. Acreditamos que, principalmente, os levantamentos bibliográficos acerca dos TEs e Classificação Hierárquica nos Capítulos 2 e 3, podem servir de base para pesquisas que venham explorar esses temas futuramente. Na revisão sobre TEs, além de diversas referências acerca de seu conceito e histórico, foram apresentadas uma variedade ferramentas que podem ser utilizadas em sua classificação ou que são exclusivamente voltadas a esse propósito. Do mesmo modo, no capítulo de HC esse tipo distinto de classificação é conceituado em detalhes, onde são exemplificados as estruturas hierarquias utilizadas, as restrições quanto a classificação e como são exploradas as hierarquias. Além disso, um gama de métodos de HC oriundos de diferentes abordagens e estratégias hierárquicas são apresentados. Todo esse referencial teórico representa um material de consulta que compila informações potencialmente úteis a pesquisas futuras.

Ademais, com o desenvolvimento dos métodos aqui propostos e os experimentos realizados, foram geradas até então as seguintes publicações:

Periódico:

- PEREIRA, G. T., CERRI, R.: **Hierarchical and Non-Hierarchical Classification of Transposable Elements with a Genetic Algorithm**. In: Journal of Information and Data Management (JIDM), 2018.

Conferências:

- PEREIRA, G. T., SANTOS, B. Z., CERRI, R.: **A Genetic Algorithm for Transposable Elements Hierarchical Classification Rule Induction**. In: IEEE International Congress on Evolutionary Computation, 2018, Rio de Janeiro. Anais do IEEE International Congress on Evolutionary Computation (CEC 2018).
- PEREIRA, G. T., CERRI, R.: **Classificação Hierárquica e Não Hierárquica de Elementos Transponíveis**. In: 5th Symposium on Knowledge Discovery, Mining and Learning (KDMiLe), 2017, Uberlândia. p. 69-76.
- SANTOS, B. Z., PEREIRA, G. T., NAKANO, F. K., CERRI, R.: **Strategies for Selection of Positive and Negative Instances in the Hierarchical Classification of Transposable Elements**. In: Brazilian Conference on Intelligent Systems (BRACIS), 2018, São Paulo. Proceedings of the 7th Brazilian Conference on Intelligent Systems.

## 9.2 Trabalhos Futuros

Nesta dissertação foram gerados diversos materiais que podem ser estendidos em trabalhos futuros. O mais natural a ser realizado em seguida, é a investigação mais aprofundada dos métodos multiobjetivo **HC-WGA** e **HC-LGA**. Dada as limitações de tempo da pesquisa de mestrado, a quantidade de experimentações com esses métodos teve que ser reduzida. Deste modo, o mesmo desempenho obtido em um conjunto de dados menor e menos desbalanceado como o PGSB, não foi alcançado num conjunto mais complexo como o REPBASE. A hipótese levantada é que a utilização de populações iniciais maiores assim como um número mais elevado de gerações e tentativas de convergência das regras, podem levar a melhores resultados.

Outro trabalho futuro em potencial, é a alteração do tipo de representação aplicada aos indivíduos nos **GAs** propostos. No momento, tanto o **HC-GA** quanto o **HC-WGA** e **HC-LGA**, utilizam a representação de Michigan, onde cada indivíduo corresponde a uma única regra de **HC** e uma população constitui uma lista de regras. Essa abordagem foi utilizada com o intuito de induzir regras que lidem melhor com as particularidades do problema hierárquico, já que a mesma tende a facilitar a exploração do espaço de busca. No entanto, uma outra alternativa seria a utilização da representação de Pittsburgh em que um único indivíduo representa um conjunto inteiro de regras, possuindo assim um viés mais exploratório. A investigação dessa abordagem é interessante pois diferente de Michigan que busca pelas melhores regras isoladas, a abordagem de Pittsburgh visa encontrar o melhor conjunto de regras.

Ainda em relação aos métodos propostos, pretende-se focar na implementação e teste de novas funções de aptidão que enderecem outros aspectos importantes a serem observados em modelos de classificação, além do aprimoramento e investigação das que já existem. A função **Variance Gain (VG)**, por exemplo, apenas de ter gerado previsões com um bom equilíbrio entre desempenho preditivo e interpretabilidade, apresenta uma falha ou vulnerabilidade caso uma regra que cubra todas as instâncias seja induzida, fazendo com que o valor final de **VG** seja zero, já que, segundo a Equação 6.5,  $\frac{|S_{-r}|}{|S|} \cdot var(S_{-r})$  será zero, e os valores de  $var(S)$  e  $\frac{|S_r|}{|S|} \cdot var(S_r)$  serão os mesmos. Dessa forma, mesmo a regra em questão sendo considerada ótima, já que cobre todas as instâncias remanescente que pertencem ao mesmo (ou similar) conjunto de classes, seu *fitness* seria zero. Além disso, as otimizações em conjunto de **hierarchical Precision (hP)** e **hierarchical Recall (hR)** com o **HC-LGA** necessitam mais investigação, uma vez que suas correlações foram positivas o que indica um comportamento contrário ao senso comum que diz que essas duas medidas são, de certa forma, contraditórias, existindo inclusive um problema de *trade-off* relacionado (JR; FREITAS, 2009; SUN; LIM, 2001).

Um outro trabalho futuro que pode vir a gerar resultados interessantes, sendo esse mais voltado a Otimização Multiobjetivo, é justamente a implementação da única dentre as três abordagens a ficar fora dessa pesquisa, a Abordagem de Pareto. Diferente das demais, trata-se de uma abordagem conceitualmente mais complexa além de ser consideravelmente mais custosa na implementação. Logo, pretende-se adaptar algoritmos já consolidados como o NSGA-II, PAES e

SPEA2 de forma a operarem conforme métodos hierárquicos (DEB et al., 2016). Ademais, visa-se realizar mais experimentos com os métodos HC-WGA e HC-LGA, aumentando o número de objetivos e também particionando objetivos mais gerais em objetivos menores, como por exemplo fragmentar o “desempenho preditivo” na otimização de hP e hR, ou a “interpretabilidade” em *número de testes ativos* numa regra e a *quantidade de testes simples* presentes.

Por fim, visa-se aplicar os métodos propostos em mais conjuntos de dados hierárquicos com mais atributos, classes e hierarquias maiores, ou ainda, em problemas que pertençam a outros domínios como a predição de funções de proteínas e a classificação de documentos.

## REFERÊNCIAS

---

- ABADEH, M. S.; HABIBI, J. Computer intrusion detection using an iterative fuzzy rule learning approach. In: IEEE. *Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International*. [S.l.], 2007. p. 1–6. Citado na página [40](#).
- ABRUSÁN, G.; GRUNDMANN, N.; DEMESTER, L.; MAKALOWSKI, W. Teclase—a tool for automated classification of unknown eukaryotic transposable elements. *Bioinformatics*, Oxford University Press, v. 25, n. 10, p. 1329–1330, 2009. Citado 3 vezes nas páginas [18](#), [26](#) e [27](#).
- AHA, D. W.; KIBLER, D.; ALBERT, M. K. Instance-based learning algorithms. *Machine Learning*, v. 6, n. 1, p. 37–66, 1991. Citado na página [36](#).
- ALBERTS, B.; JOHNSON, A.; LEWIS, J.; RAFF, M.; ROBERTS, K.; WALTER, P. *Biologia molecular da célula*. [S.l.]: Artmed Editora, 2009. Citado na página [23](#).
- ALTSCHUL, S. F.; GISH, W.; MILLER, W.; MYERS, E. W.; LIPMAN, D. J. Basic local alignment search tool. *Journal of molecular biology*, Elsevier, v. 215, n. 3, p. 403–410, 1990. Citado 3 vezes nas páginas [18](#), [26](#) e [72](#).
- ALVES, R. T.; DELGADO, M.; FREITAS, A. A. Knowledge discovery with artificial immune systems for hierarchical multi-label classification of protein functions. In: IEEE. *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*. [S.l.], 2010. p. 1–8. Citado na página [40](#).
- ALVES, R. T.; DELGADO, M. R.; FREITAS, A. A. Multi-label hierarchical classification of protein functions with artificial immune systems. In: SPRINGER. *Brazilian Symposium on Bioinformatics*. [S.l.], 2008. p. 1–12. Citado 2 vezes nas páginas [39](#) e [40](#).
- BAKER, J. E. Reducing bias and inefficiency in the selection algorithm. In: *Proceedings of the second international conference on genetic algorithms*. [S.l.: s.n.], 1987. p. 14–21. Citado na página [47](#).
- BALUJA, S.; CARUANA, R. Removing the genetics from the standard genetic algorithm. In: *Machine Learning: Proceedings of the Twelfth International Conference*. [S.l.: s.n.], 1995. p. 38–46. Citado na página [44](#).
- BANDARU, S.; NG, A. H.; DEB, K. Data mining methods for knowledge discovery in multi-objective optimization: Part a-survey. *Expert Systems with Applications*, Elsevier, v. 70, p. 139–159, 2017. Citado na página [54](#).
- BARROS, R. C.; CERRI, R.; FREITAS, A. A.; CARVALHO, A. C. Probabilistic clustering for hierarchical multi-label classification of protein functions. In: BLOCKEEL, H.; KERSTING, K.; NIJSSEN, S.; ZELEZNY, F. (Ed.). *Machine Learning and Knowledge Discovery in Databases*. [S.l.]: Springer Berlin Heidelberg, 2013, (Lecture Notes in Computer Science, v. 8189). p. 385–400. Citado na página [28](#).

- BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, v. 13, n. Feb, p. 281–305, 2012. Citado na página 73.
- BEYGELZIMER, A.; LANGFORD, J.; ZADROZNY, B. Machine learning techniques—reductions between prediction quality metrics. *Performance Modeling and Engineering*, Springer, p. 3–28, 2008. Citado na página 33.
- BIÉMONT, C. A brief history of the status of transposable elements: from junk DNA to major players in evolution. *Genetics*, Genetics Soc America, v. 186, n. 4, p. 1085–1093, 2010. Citado 2 vezes nas páginas 22 e 23.
- BIÉMONT, C.; VIEIRA, C. Genetics: junk DNA as an evolutionary force. *Nature*, Nature Publishing Group, v. 443, n. 7111, p. 521–524, 2006. Citado 2 vezes nas páginas 23 e 24.
- BLOCKEEL, H.; BRUYNOOGHE, M.; DŽEROSKI, S.; RAMON, J.; STRUYF, J. Hierarchical multi-classification. In: *Workshop Notes of the KDD'02 Workshop on Multi-Relational Data Mining*. [S.l.: s.n.], 2002. p. 21–35. Citado 4 vezes nas páginas 19, 34, 38 e 39.
- BÖHNE, A.; BRUNET, F.; GALIANA-ARNOUX, D.; SCHULTHEIS, C.; VOLFF, J.-N. Transposable elements as drivers of genomic and biological diversity in vertebrates. *Chromosome Research*, Springer, v. 16, n. 1, p. 203–215, 2008. Citado na página 22.
- BORGES, H. B.; NIEVOLA, J. C. Multi-label hierarchical classification using a competitive neural network for protein function prediction. In: *IEEE. Neural Networks (IJCNN), The 2012 International Joint Conference on*. [S.l.], 2012. p. 1–8. Citado 2 vezes nas páginas 39 e 40.
- BOUTELL, M. R.; LUO, J.; SHEN, X.; BROWN, C. M. Learning multi-label scene classification. *Pattern Recognition*, v. 37, n. 9, p. 1757–1771, 2004. Citado na página 36.
- CALSA, T. J.; CARRARO, D. M.; BENATTI, M. R.; BARBOSA, A. C.; KITAJIMA, J. P.; CARRER, H. Structural features and transcript-editing analysis of sugarcane (*saccharum officinarum* l.) chloroplast genome. *Current genetics*, Springer, v. 46, n. 6, p. 366–373, 2004. Citado na página 23.
- CASTRO, L. N. D. *Fundamentals of natural computing: basic concepts, algorithms, and applications*. [S.l.]: CRC Press, 2006. Citado 4 vezes nas páginas 13, 43, 50 e 51.
- CECI, M.; MALERBA, D. Classifying web documents in a hierarchy of categories: a comprehensive study. *Journal of Intelligent Information Systems*, Springer, v. 28, n. 1, p. 37–78, 2007. Citado na página 33.
- CERRI, R. *Redes neurais e algoritmos genéticos para problemas de classificação hierárquica multirrótulo*. Tese (Doutorado) — Universidade de São Paulo, 2014. Páginas 1–203. Citado 11 vezes nas páginas 13, 18, 30, 32, 39, 41, 45, 49, 51, 58 e 60.
- CERRI, R.; BARROS, R. C.; CARVALHO, A. C. D. Hierarchical multi-label classification using local neural networks. *Journal of Computer and System Sciences*, Elsevier, v. 80, n. 1, p. 39–56, 2014. Citado 2 vezes nas páginas 37 e 38.
- CERRI, R.; BARROS, R. C.; CARVALHO, A. C. de; JIN, Y. Reduction strategies for hierarchical multi-label classification in protein function prediction. *BMC bioinformatics*, BioMed Central, v. 17, n. 1, p. 373, 2016. Citado 6 vezes nas páginas 29, 31, 33, 37, 38 e 40.

- CERRI, R.; BARROS, R. C.; CARVALHO, A. C. P. L. F. de. A genetic algorithm for hierarchical multi-label classification. In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2012. (SAC '12), p. 250–255. ISBN 978-1-4503-0857-1. Disponível em: <<http://doi.acm.org/10.1145/2245276.2245325>>. Citado 3 vezes nas páginas 39, 41 e 74.
- CERRI, R.; CARVALHO, A. C. P. d. L. et al. Neural networks and genetic algorithms for hierarchical multi-label classification problems. In: UNIVERSIDADE DE SÃO PAULO-USP. *Joint Conference on Robotics and Intelligent Systems; Best DSc Theses/MSc Dissertations Contest in Artificial and Computational Intelligence, 9th*. [S.l.], 2014. Citado na página 17.
- CERRI, R.; CARVALHO, A. C. P. de. Hierarchical multilabel classification using top-down label combination and artificial neural networks. In: IEEE. *Neural Networks (SBRN), 2010 Eleventh Brazilian Symposium on*. [S.l.], 2010. p. 253–258. Citado 2 vezes nas páginas 35 e 38.
- CERRI, R.; CARVALHO, A. C. P. F. de. New top-down methods using svms for hierarchical multilabel classification problems. In: IEEE. *Neural Networks (IJCNN), The 2010 International Joint Conference on*. [S.l.], 2010. p. 1–8. Citado 2 vezes nas páginas 36 e 38.
- CERRI, R.; CARVALHO, A. de; FREITAS, A. Adapting non-hierarchical multilabel classification methods for hierarchical multilabel classification. *Intelligent Data Analysis*, IOS Press, v. 15, n. 6, p. 861–887, November 2011. Citado na página 36.
- CESA-BIANCHI, N.; RE, M.; VALENTINI, G. Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference. *Machine Learning*, Springer, v. 88, n. 1, p. 209–241, 2012. Citado 2 vezes nas páginas 34 e 38.
- CESA-BIANCHI, N.; VALENTINI, G. Hierarchical cost-sensitive algorithms for genome-wide gene function prediction. In: *Machine Learning in Systems Biology*. [S.l.: s.n.], 2009. p. 14–29. Citado na página 34.
- CHÉNAIS, B. Transposable elements and human cancer: a causal relationship? *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, Elsevier, v. 1835, n. 1, p. 28–35, 2013. Citado na página 23.
- CLAESEN, M.; MOOR, B. D. Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*, 2015. Citado na página 73.
- CLARE, A.; KING, R. D. Predicting gene function in *saccharomyces cerevisiae*. *Bioinformatics*, Oxford Univ Press, v. 19, n. suppl 2, p. ii42–ii49, 2003. Citado 3 vezes nas páginas 19, 39 e 40.
- COELLO, C. A. C.; LAMONT, G. B.; VELDHUIZEN, D. A. V. et al. *Evolutionary algorithms for solving multi-objective problems*. [S.l.]: Springer, 2007. v. 5. Citado na página 55.
- COELLO, C. C. Evolutionary multi-objective optimization: a historical view of the field. *IEEE computational intelligence magazine*, IEEE, v. 1, n. 1, p. 28–36, 2006. Citado 3 vezes nas páginas 52, 55 e 56.
- COHEN, W. W. Fast effective rule induction. In: *Twelfth International Conference on Machine Learning*. [S.l.: s.n.], 1995. p. 115–123. Citado na página 36.

COSTA, E. P.; LORENA, A. C.; CARVALHO, A. C.; FREITAS, A. A.; HOLDEN, N. Comparing several approaches for hierarchical classification of proteins with decision trees. In: *II Brazilian Symposium on Bioinformatics*. Berlin, Heidelberg: Springer-Verlag, 2007. (Lecture Notes in Bioinformatics, v. 4643), p. 126–137. Citado na página 28.

DAI, J.; XIE, W.; BRADY, T. L.; GAO, J.; VOYTAS, D. F. Phosphorylation regulates integration of the yeast ty5 retrotransposon into heterochromatin. *Molecular cell*, Elsevier, v. 27, n. 2, p. 289–299, 2007. Citado na página 23.

DAVIS, J.; GOADRICH, M. The relationship between precision-recall and roc curves. In: *Proceedings of the 23rd international conference on Machine learning*. [S.l.]: ACM, 2006. p. 233–240. Citado na página 62.

DEB, K. *Multi-objective optimization using evolutionary algorithms*. [S.l.]: John Wiley & Sons, 2001. v. 16. Citado 3 vezes nas páginas 52, 54 e 56.

DEB, K.; SINDHYA, K.; HAKANEN, J. Multi-objective optimization. In: *Decision Sciences: Theory and Practice*. [S.l.]: CRC Press, 2016. p. 145–184. Citado 5 vezes nas páginas 13, 52, 53, 56 e 112.

DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, v. 7, n. Jan, p. 1–30, 2006. Citado na página 74.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. *Positive feedback as a search strategy*. [S.l.], 1991. Citado na página 40.

DUTTA, D. Classification rules generation for iris data using lexicographic pareto based multi objective genetic algorithm. In: *Conference proceedings of 3rd international conference on intelligent systems and networks (IISN 2009), At Klawad, Haryana, India*. [S.l.: s.n.], 2009. p. 173–177. Citado na página 55.

EDGEWORTH, F. Y. *Mathematical psychics: An essay on the application of mathematics to the moral sciences*. [S.l.]: Kegan Paul, 1881. v. 10. Citado na página 54.

EIBEN, A. E.; SMITH, J. E. et al. *Introduction to evolutionary computing*. [S.l.]: Springer, 2003. v. 53. Citado 2 vezes nas páginas 12 e 44.

EISNER, R.; POULIN, B.; SZAFRON, D.; LU, P.; GREINER, R. Improving protein function prediction using the hierarchical structure of the gene ontology. In: IEEE. *Computational Intelligence in Bioinformatics and Computational Biology, 2005. CIBCB'05. Proceedings of the 2005 IEEE Symposium on*. [S.l.], 2005. p. 1–10. Citado na página 33.

FAGNI, T.; SEBASTIANI, F. On the selection of negative examples for hierarchical text categorization. In: *Proceedings of the 3rd Language & Technology Conference (LTC'07)*. [S.l.: s.n.], 2007. p. 24–28. Citado na página 33.

FESCHOTTE, C. Transposable elements and the evolution of regulatory networks. *Nature Reviews Genetics*, Nature Publishing Group, v. 9, n. 5, p. 397–405, 2008. Citado 2 vezes nas páginas 18 e 22.

FESCHOTTE, C.; KESWANI, U.; RANGANATHAN, N.; GUIBOTSY, M. L.; LEVINE, D. Exploring repetitive DNA landscapes using REPCLASS, a tool that automates the classification of transposable elements in eukaryotic genomes. *Genome biology and evolution*, Oxford University Press, v. 1, p. 205–220, 2009. Citado na página 26.

- FESCHOTTE, C.; PRITHAM, E. J. DNA transposons and the evolution of eukaryotic genomes. *Annual review of genetics*, NIH Public Access, v. 41, p. 331, 2007. Citado na página 22.
- FINNEGAN, D.; FAWCETT, D. Transposable elements in drosophila melanogaster. *Oxford surveys on eukaryotic genes*, v. 3, p. 1, 1986. Citado 2 vezes nas páginas 18 e 24.
- FINNEGAN, D. J. Eukaryotic transposable elements and genome evolution. *Trends in genetics*, Elsevier, v. 5, p. 103–107, 1989. Citado na página 24.
- FREITAS, A. A. A critical review of multi-objective optimization in data mining: A position paper. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 6, n. 2, p. 77–86, dez. 2004. ISSN 1931-0145. Citado 2 vezes nas páginas 54 e 55.
- FREITAS, A. A.; CARVALHO, A. C. de. A tutorial on hierarchical classification with applications in bioinformatics. In: *In: D. Taniar (Ed.) Research and Trends in Data Mining Technologies and Applications*, Idea Group, 2007. [S.l.: s.n.], 2007. Citado 7 vezes nas páginas 17, 29, 30, 31, 33, 36 e 38.
- FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. Bayesian network classifiers. *Machine learning*, Springer, v. 29, n. 2-3, p. 131–163, 1997. Citado na página 36.
- GHAZI, D.; INKPEN, D.; SZPAKOWICZ, S. Hierarchical versus flat classification of emotions in text. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*. [S.l.], 2010. p. 140–146. Citado na página 32.
- GOLBERG, D. E. Genetic algorithms in search, optimization, and machine learning. *Addison wesley*, v. 1989, p. 102, 1989. Citado 4 vezes nas páginas 47, 49, 50 e 56.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0201157675. Citado 3 vezes nas páginas 45, 50 e 56.
- GOLDBERG, D. E.; HOLLAND, J. H. Genetic Algorithms and Machine Learning. *Machine Learning*, v. 3, p. 95–99, 1988. Citado 2 vezes nas páginas 47 e 50.
- GOLDBERG, D. E.; SASTRY, K. A practical schema theorem for genetic algorithm design and tuning. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. [S.l.: s.n.], 2001. p. 328–335. Citado 2 vezes nas páginas 44 e 56.
- HOEDE, C.; ARNOUX, S.; MOISSET, M.; CHAUMIER, T.; INIZAN, O.; JAMILLOUX, V.; QUESNEVILLE, H. Pastec: an automatic transposable element classification tool. *PLoS One*, Public Library of Science, v. 9, n. 5, p. e91929, 2014. Citado 3 vezes nas páginas 18, 26 e 27.
- HOLLAND, J. H. Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems. *Machine learning: An artificial intelligence approach*, Morgan Kaufmann, v. 2, 1986. Citado na página 44.
- HOLLAND, J. H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. [S.l.]: MIT press, 1992. Citado 3 vezes nas páginas 43, 45 e 47.

- JANICKI, M.; ROOKE, R.; YANG, G. Bioinformatics and genomic analysis of transposable elements in eukaryotic genomes. *Chromosome research*, Springer, v. 19, n. 6, p. 787–808, 2011. Citado na página [22](#).
- JR, C. N. S.; FREITAS, A. A. A global-model naive bayes approach to the hierarchical prediction of protein functions. In: IEEE. *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*. [S.l.], 2009. p. 992–997. Citado 3 vezes nas páginas [39](#), [40](#) e [111](#).
- JUNGJIT, S.; FREITAS, A. A lexicographic multi-objective genetic algorithm for multi-label correlation based feature selection. In: ACM. *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*. [S.l.], 2015. p. 989–996. Citado na página [55](#).
- JURKA, J.; KAPITONOV, V. V.; PAVLICEK, A.; KLONOWSKI, P.; KOHANY, O.; WALICHIEWICZ, J. Repbase update, a database of eukaryotic repetitive elements. *Cytogenetic and genome research*, Karger Publishers, v. 110, n. 1-4, p. 462–467, 2005. Citado na página [22](#).
- KAZAZIAN, H. H. Mobile elements and disease. *Current opinion in genetics & development*, Elsevier, v. 8, n. 3, p. 343–350, 1998. Citado na página [22](#).
- KAZAZIAN, H. H. Mobile elements: drivers of genome evolution. *science*, American Association for the Advancement of Science, v. 303, n. 5664, p. 1626–1632, 2004. Citado 2 vezes nas páginas [18](#) e [22](#).
- KIRITCHENKO, S.; MATWIN, S.; FAMILI, F. Functional annotation of genes using hierarchical text categorization. 2005. Citado 2 vezes nas páginas [61](#) e [74](#).
- LABROU, Y.; FININ, T. Yahoo! as an ontology: using Yahoo! categories to describe documents. In: ACM. *Proceedings of the eighth international conference on Information and knowledge management*. [S.l.], 1999. p. 180–187. Citado na página [39](#).
- LAGEMAAT, L. N. van de; LANDRY, J.-R.; MAGER, D. L.; MEDSTRAND, P. Transposable elements in mammals promote regulatory variation and diversification of genes with specialized functions. *Trends in Genetics*, Elsevier, v. 19, n. 10, p. 530–536, 2003. Citado 2 vezes nas páginas [22](#) e [23](#).
- LEVIN, H. L.; MORAN, J. V. Dynamic interactions between transposable elements and their hosts. *Nature Reviews Genetics*, Nature Publishing Group, v. 12, n. 9, p. 615–627, 2011. Citado na página [23](#).
- LEVY, A.; SELA, N.; AST, G. Transposone and microtransposone: transposed elements influence on the transcriptome of seven vertebrates and invertebrates. *Nucleic acids research*, Oxford Univ Press, v. 36, n. suppl 1, p. D47–D52, 2008. Citado 2 vezes nas páginas [18](#) e [22](#).
- LI, W.; ZHANG, P.; FELLERS, J. P.; FRIEBE, B.; GILL, B. S. Sequence composition, organization, and evolution of the core triticeae genome. *The Plant Journal*, Wiley Online Library, v. 40, n. 4, p. 500–511, 2004. Citado na página [24](#).
- LINDSLEY, D. L.; ZIMM, G. G. *The genome of Drosophila melanogaster*. [S.l.]: Academic Press, 2012. Citado na página [22](#).

- LOPES, F. R.; CARAZZOLLE, M. F.; PEREIRA, G. A. G.; COLOMBO, C. A.; CARARETO, C. M. A. Transposable elements in coffee (gentianales: Rubiaceae) transcripts and their role in the origin of protein diversity in flowering plants. *Molecular Genetics and Genomics*, Springer, v. 279, n. 4, p. 385–401, 2008. Citado 2 vezes nas páginas 18 e 22.
- LORD, P. W.; STEVENS, R. D.; BRASS, A.; GOBLE, C. A. Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics*, Oxford University Press, v. 19, n. 10, p. 1275–1283, 2003. Citado na página 40.
- LOUREIRO, T.; CAMACHO, R.; VIEIRA, J.; FONSECA, N. A. Boosting the detection of transposable elements using machine learning. In: SPRINGER. *7th International Conference on Practical Applications of Computational Biology & Bioinformatics*. [S.l.], 2013. p. 85–91. Citado 3 vezes nas páginas 17, 19 e 27.
- MCCLINTOCK, B. *The significance of responses of the genome to challenge*. [S.l.]: Singapore: World Scientific Pub. Co, 1993. Citado 2 vezes nas páginas 17 e 22.
- MELSTED, P.; PRITCHARD, J. K. Efficient counting of k-mers in DNA sequences using a bloom filter. *BMC bioinformatics*, BioMed Central, v. 12, n. 1, p. 333, 2011. Citado na página 70.
- MIKI, Y.; NISHISHO, I.; HORII, A.; MIYOSHI, Y.; UTSUNOMIYA, J.; KINZLER, K. W.; VOGELSTEIN, B.; NAKAMURA, Y. Disruption of the apc gene by a retrotransposal insertion of 11 sequence in a colon cancer. *Cancer research*, AACR, v. 52, n. 3, p. 643–645, 1992. Citado na página 23.
- MITCHELL, M. *An introduction to genetic algorithms*. [S.l.]: MIT press, 1998. Citado na página 50.
- MITCHELL, T. *Machine Learning*. McGraw-Hill, 1997. (McGraw-Hill International Editions). ISBN 9780071154673. Disponível em: <<https://books.google.com.br/books?id=EoYBngEACAAJ>>. Citado na página 28.
- MUKAKA, M. M. A guide to appropriate use of correlation coefficient in medical research. *Malawi Medical Journal*, Medical Association of Malawi, v. 24, n. 3, p. 69–71, 2012. Citado na página 94.
- NAKANO, F. K. *Deep Learning para Classificação Hierárquica de Elementos Transponíveis*. Dissertação (Mestrado) — Universidade Federal de São Carlos - Departamento de Computação, São Carlos - SP, 2018. Citado 2 vezes nas páginas 72 e 104.
- NAKANO, F. K.; PINTO, W. J.; PAPPAS, G. L.; CERRI, R. Top-down strategies for hierarchical classification of transposable elements with neural networks. In: IEEE. *Neural Networks (IJCNN), 2017 International Joint Conference on*. [S.l.], 2017. p. 2539–2546. Citado 8 vezes nas páginas 18, 35, 36, 37, 38, 72, 74 e 103.
- NEKRUTENKO, A.; LI, W.-H. Transposable elements are found in a large number of human protein-coding genes. *TRENDS in Genetics*, Elsevier, v. 17, n. 11, p. 619–621, 2001. Citado 2 vezes nas páginas 18 e 22.
- OTERO, F. E.; FREITAS, A. A.; JOHNSON, C. G. A hierarchical classification ant colony algorithm for predicting gene ontology terms. In: SPRINGER. *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. [S.l.], 2009. p. 68–79. Citado 3 vezes nas páginas 19, 39 e 40.

- OTERO, F. E.; FREITAS, A. A.; JOHNSON, C. G. A new sequential covering strategy for inducing classification rules with ant colony algorithms. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 17, n. 1, p. 64–76, 2013. Citado na página 44.
- PAES, B. C.; PLASTINO, A.; FREITAS, A. A. Improving local per level hierarchical classification. *Journal of Information and Data Management*, v. 3, n. 3, p. 394, 2012. Citado na página 37.
- PARETO, V. *Cours d'économie politique*. [S.l.]: Librairie Droz, 1964. v. 1. Citado na página 54.
- PEARSON, K. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, JSTOR, v. 58, p. 240–242, 1895. Citado na página 76.
- PEREIRA, G. T.; CERRI, R. Classificação hierárquica e não hierárquica de elementos transponíveis. *Proceedings of the 5th Symposium on Knowledge Discovery, Mining and Learning (KDMiLe)*, p. 69–76, 2017. Citado 2 vezes nas páginas 16 e 106.
- PEREIRA, G. T.; SANTOS, B. Z.; CERRI, R. A genetic algorithm for transposable elements hierarchical classification rule induction. In: IEEE. *IEEE Congress on Evolutionary Computation (CEC)*. [S.l.], 2018. Citado 2 vezes nas páginas 18 e 72.
- PIMENTA, A. H. d. M. et al. Geração genética multiobjetivo de bases de conhecimento fuzzy com enfoque na distribuição das soluções não dominadas. Universidade Federal de São Carlos, 2014. Citado na página 40.
- PRAY, L.; ZHAUROVA, K. Barbara McClintock and the discovery of jumping genes (transposons). *Nature Education*, v. 1, n. 1, p. 169, 2008. Citado na página 22.
- QUINLAN, J. R. *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0. Citado 2 vezes nas páginas 36 e 39.
- RAZALI, N. M.; GERAGHTY, J. et al. Genetic algorithm performance with different selection strategies in solving tsp. In: INTERNATIONAL ASSOCIATION OF ENGINEERS HONG KONG. *Proceedings of the world congress on engineering*. [S.l.], 2011. v. 2, p. 1134–1139. Citado 2 vezes nas páginas 13 e 48.
- REBOLLO, R.; ROMANISH, M. T.; MAGER, D. L. Transposable elements: an abundant and natural source of regulatory sequences for host genes. *Annual review of genetics*, Annual Reviews, v. 46, p. 21–42, 2012. Citado na página 22.
- ROCCHIO, J. J. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*, Prentice-Hall Inc., p. 313–323, 1971. Citado na página 39.
- RODGERS, J. L.; NICEWANDER, W. A. Thirteen ways to look at the correlation coefficient. *The American Statistician*, Taylor & Francis Group, v. 42, n. 1, p. 59–66, 1988. Citado na página 94.
- RUSSELL, S.; NORVIG, P.; INTELLIGENCE, A. A modern approach. *Artificial Intelligence*. Prentice-Hall, Egnlewood Cliffs, Citeseer, v. 25, p. 27, 1995. Citado na página 28.
- SANTOS, B. Z.; PEREIRA, G. T.; NAKANO, F. K.; CERRI, R. Strategies for selection of positive and negative instances in the hierarchical classification of transposable elements. In: SBC BRACIS. *Brazilian Conference on Intelligent Systems*. [S.l.], 2018. p. 343–354. Citado 3 vezes nas páginas 35, 72 e 103.

- SASTRY, K.; GOLDBERG, D.; KENDALL, G. Genetic Algorithms. In: BURKE, E. K.; KENDALL, G. (Ed.). *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Methodologies*. [S.l.]: Springer, 2005. cap. 4, p. 97–125. Citado na página 50.
- SELA, N.; KIM, E.; AST, G. The role of transposable elements in the evolution of non-mammalian vertebrates and invertebrates. *Genome biology*, BioMed Central, v. 11, n. 6, p. 1, 2010. Citado na página 22.
- SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples). *Biometrika*, JSTOR, v. 52, n. 3/4, p. 591–611, 1965. Citado na página 75.
- SILLA, C.; FREITAS, A. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, Springer Netherlands, v. 22, p. 31–72, 2010. ISSN 1384-5810. Citado 13 vezes nas páginas 13, 17, 18, 29, 31, 32, 33, 34, 35, 36, 37, 38 e 109.
- SILLA, C. N.; KAESTNER, C. A. Hierarchical classification of bird species using their audio recorded songs. In: IEEE. *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. [S.l.], 2013. p. 1895–1900. Citado na página 33.
- SLOTKIN, R. K.; MARTIENSSEN, R. Transposable elements and the epigenetic regulation of the genome. *Nature Reviews Genetics*, Nature Publishing Group, v. 8, n. 4, p. 272–285, 2007. Citado na página 22.
- SMIT, A. F.; HUBLEY, R.; GREEN, P. *RepeatMasker*. 1996. Citado 2 vezes nas páginas 26 e 72.
- SMITH, S. F. *A learning system based on genetic adaptive algorithms*. Tese (Doutorado) — University of Pittsburgh, Pittsburgh, PA, 1980. Citado na página 44.
- SPEARMAN, C. The proof and measurement of association between two things. *The American journal of psychology*, JSTOR, v. 15, n. 1, p. 72–101, 1904. Citado na página 76.
- SPEARS, W. Recombination parameters. In: BÁCK, T.; FOGEL, D. B.; MICHALEWICZ, Z. (Ed.). *The Handbook of Evolutionary Computation*. Philadelphia, PA: IOP Publishing and Oxford University Press, 1997. cap. E1.3, p. E1.3:1–E1.3:13. Citado na página 49.
- SRINIVASAN, S.; RAMAKRISHNAN, S. Evolutionary multi objective optimization for rule mining: a review. *Artificial Intelligence Review*, Springer, v. 36, n. 3, p. 205, 2011. Citado na página 44.
- STEINBISS, S.; WILLHOEFT, U.; GREMME, G.; KURTZ, S. Fine-grained annotation and classification of de novo predicted ltr retrotransposons. *Nucleic acids research*, Oxford University Press, v. 37, n. 21, p. 7002–7013, 2009. Citado 2 vezes nas páginas 18 e 26.
- STOJANOVA, D.; CECI, M.; MALERBA, D.; DZEROSKI, S. Using ppi network autocorrelation in hierarchical multi-label classification trees for gene function prediction. *BMC bioinformatics*, BioMed Central, v. 14, n. 1, p. 285, 2013. Citado 2 vezes nas páginas 39 e 41.
- SUN, A.; LIM, E.-P. Hierarchical text classification and evaluation. In: IEEE. *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*. [S.l.], 2001. p. 521–528. Citado 2 vezes nas páginas 30 e 111.

- SUN, Z.; ZHAO, Y.; CAO, D.; HAO, H. Hierarchical multilabel classification with optimal path prediction. *Neural Processing Letters*, Springer, v. 45, n. 1, p. 263–277, 2017. Citado 2 vezes nas páginas [39](#) e [41](#).
- TAN, P.-N. et al. *Introduction to data mining*. [S.l.]: Pearson Education India, 2006. Citado na página [28](#).
- TSOUMAKAS, G.; KATAKIS, I. Multi Label Classification: An Overview. *International Journal of Data Warehousing and Mining*, v. 3, n. 3, p. 1–13, 2007. Citado na página [36](#).
- VALENTINI, G. True path rule hierarchical ensembles. In: SPRINGER. *International Workshop on Multiple Classifier Systems*. [S.l.], 2009. p. 232–241. Citado 2 vezes nas páginas [34](#) e [38](#).
- VALENTINI, G. True path rule hierarchical ensembles for genome-wide gene function prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, IEEE Computer Society Press, v. 8, n. 3, p. 832–847, 2011. Citado na página [34](#).
- VALENTINI, G.; RE, M. Weighted true path rule: a multilabel hierarchical algorithm for gene function prediction. In: *Proceedings of the 1st workshop on learning from multi-label data (MLD) held in conjunction with ECML/PKDD*. [S.l.: s.n.], 2009. p. 132–145. Citado na página [34](#).
- VENS, C.; STRUYF, J.; SCHIETGAT, L.; DZEROSKI, S.; BLOCKEEL, H. Decision trees for hierarchical multi-label classification. *Machine Learning*, Kluwer Academic Publishers, Hingham, MA, USA, v. 73, n. 2, p. 185–214, 2008. ISSN 0885-6125. Citado 10 vezes nas páginas [17](#), [19](#), [31](#), [34](#), [38](#), [39](#), [41](#), [62](#), [63](#) e [72](#).
- VETTORE, A. L.; SILVA, F. R. da; KEMPER, E. L.; SOUZA, G. M.; SILVA, A. M. da; FERRO, M. I. T.; HENRIQUE-SILVA, F.; GIGLIOTI, É. A.; LEMOS, M. V.; COUTINHO, L. L. et al. Analysis and functional annotation of an expressed sequence tag collection for tropical crop sugarcane. *Genome research*, Cold Spring Harbor Lab, v. 13, n. 12, p. 2725–2735, 2003. Citado na página [23](#).
- VOLFF, J.-N. Turning junk into gold: domestication of transposable elements and the creation of new genes in eukaryotes. *Bioessays*, Wiley Online Library, v. 28, n. 9, p. 913–922, 2006. Citado 2 vezes nas páginas [18](#) e [22](#).
- WANG, K.; ZHOU, S.; HE, Y. Hierarchical classification of real life documents. In: SIAM. *Proceedings of the 2001 SIAM International Conference on Data Mining*. [S.l.], 2001. p. 1–16. Citado na página [39](#).
- WETZEL, A. Evaluation of the effectiveness of genetic algorithms in combinatorial optimization. 1983. Citado 2 vezes nas páginas [47](#) e [48](#).
- WICKER, T.; SABOT, F.; HUA-VAN, A.; BENNETZEN, J. L.; CAPY, P.; CHALHOUB, B.; FLAVELL, A.; LEROY, P.; MORGANTE, M.; PANAUD, O. et al. A unified classification system for eukaryotic transposable elements. *Nature Reviews Genetics*, Nature publishing group, v. 8, n. 12, p. 973–982, 2007. Citado 14 vezes nas páginas [5](#), [7](#), [13](#), [18](#), [19](#), [20](#), [24](#), [25](#), [26](#), [27](#), [71](#), [103](#), [108](#) e [109](#).
- WILCOXON, F. Individual comparisons by ranking methods. *Biometrics bulletin*, JSTOR, v. 1, n. 6, p. 80–83, 1945. Citado na página [75](#).

WILK, M. B.; GNANADESIKAN, R. Probability plotting methods for the analysis for the analysis of data. *Biometrika*, Oxford University Press, v. 55, n. 1, p. 1–17, 1968. Citado na página [87](#).

ZIMEK, A.; BUCHWALD, F.; FRANK, E.; KRAMER, S. A study of hierarchical and flat classification of proteins. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, IEEE, v. 7, n. 3, p. 563–571, 2010. Citado na página [32](#).

# Apêndice A

## CORRELAÇÕES ENTRE OBJETIVOS QUE NÃO ATINGIRAM SIGNIFICÂNCIA ESTATÍSTICA

---

---

Neste apêndice se encontram os gráficos Q-Q das correlações entre objetivos otimizados em conjunto nos experimentos com o [HC-WGA](#) e [HC-LGA](#). As correlações aqui apresentadas não obtiveram significância estatística e, portanto, não foram adicionadas aos corpos de texto principais desse documento, sendo exibidas exclusivamente por razões complementares.

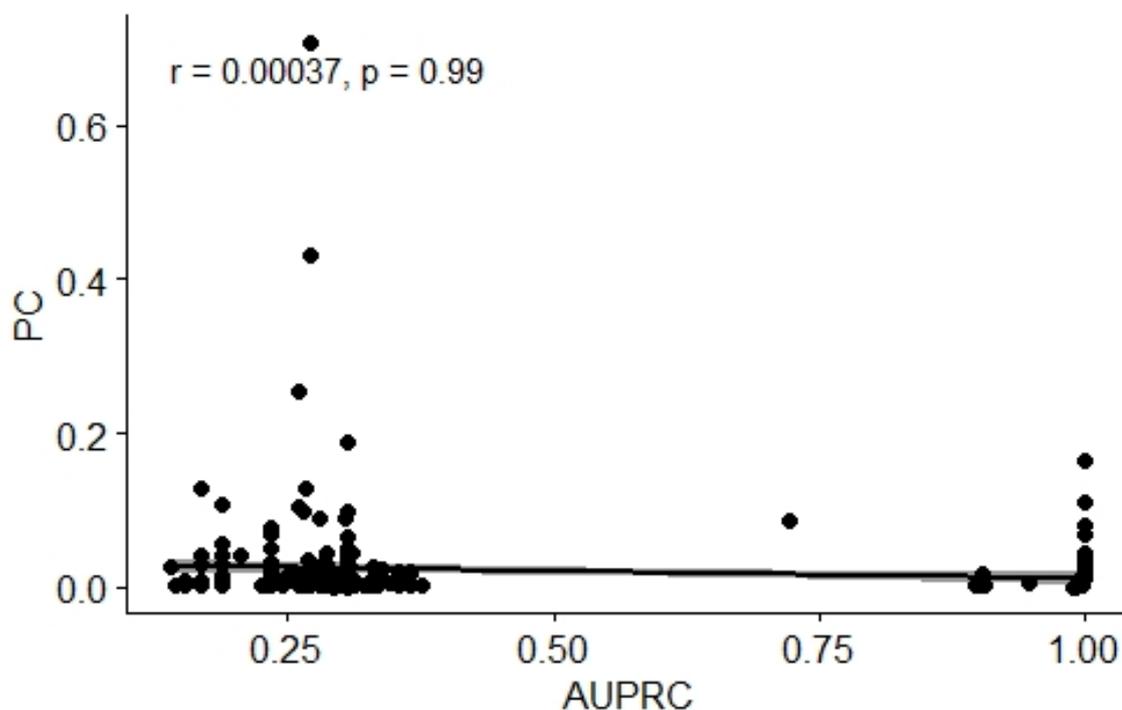


Figura 44 – Correlação de Spearman dos objetivos  $AU(\overline{PCR})$  e  $PC$  no PGSB com [HC-LGA](#).

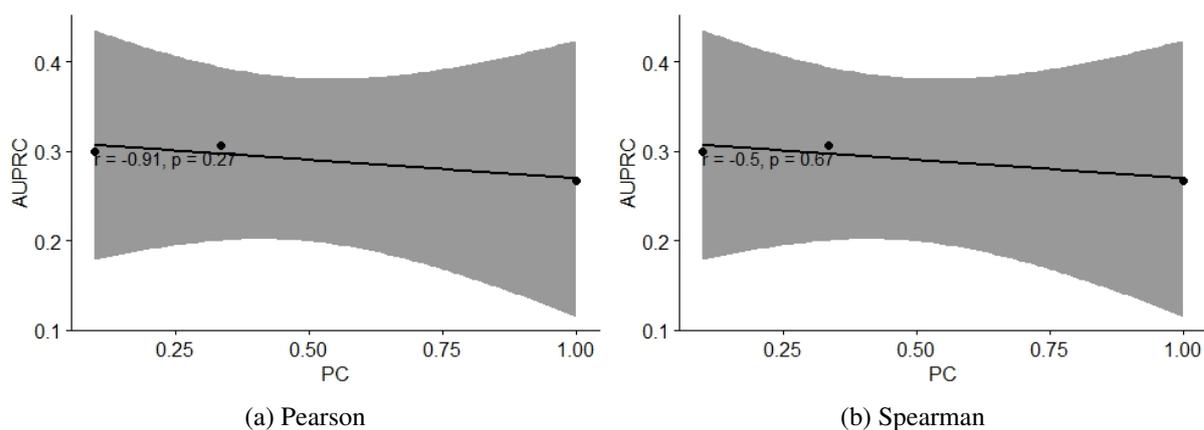


Figura 45 – Correlações dos objetivos PC e  $AU(\overline{PCR})$  no PGSB com HC-LGA.

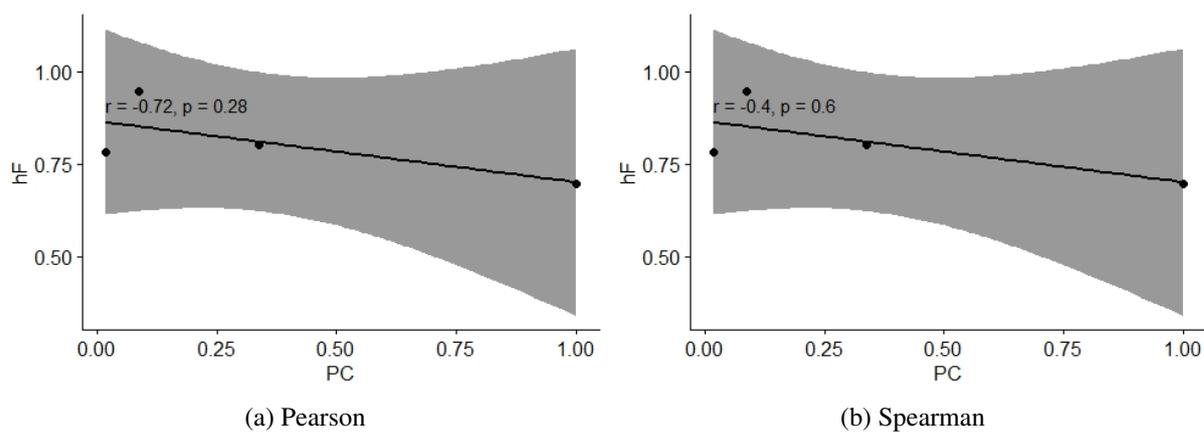


Figura 46 – Correlações dos objetivos PC e hF no PGSB com HC-LGA.

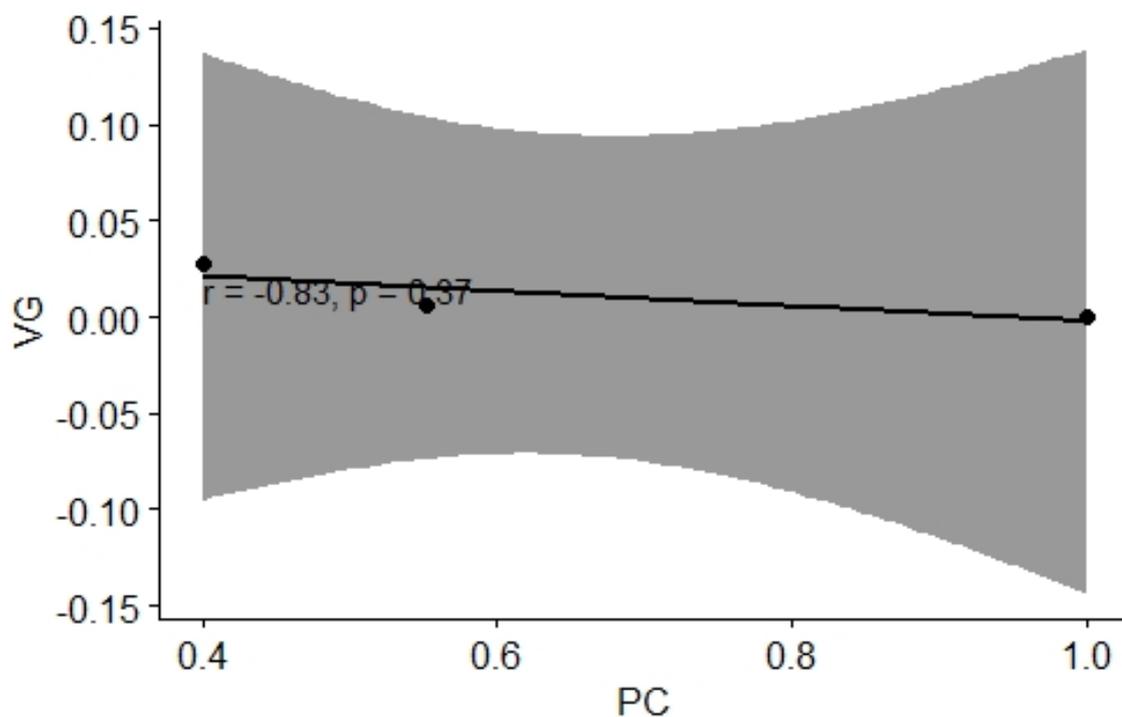


Figura 47 – Correlação de Pearson dos objetivos PC e VG no PGSB com HC-LGA.

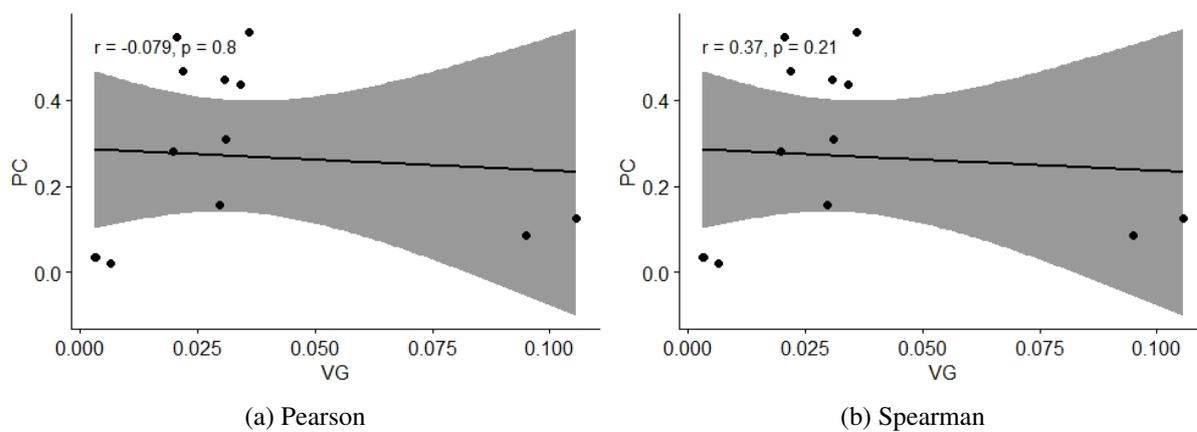


Figura 48 – Correlações dos objetivos VG e PC no PGSB com HC-LGA.

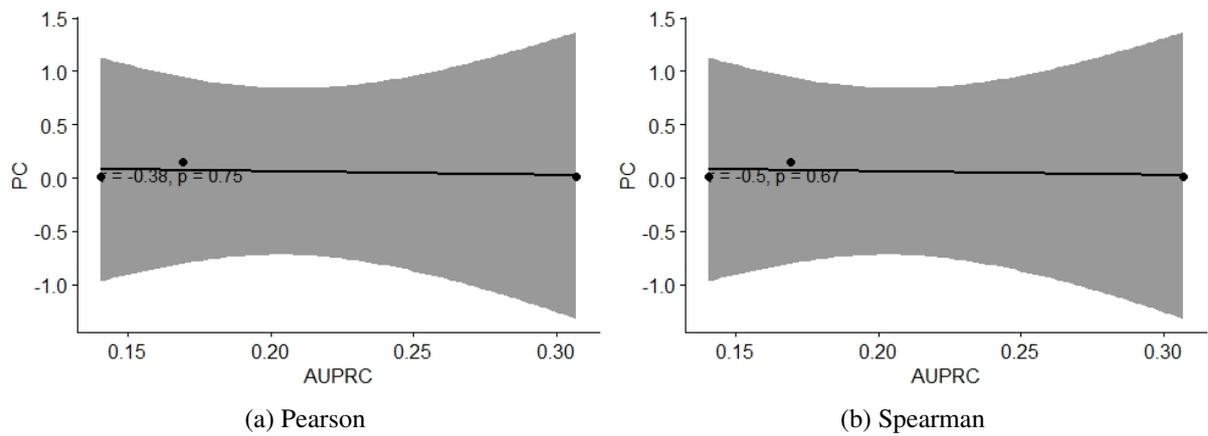


Figura 49 – Correlações de  $AU(\overline{PCR})$  (20%) + PC (80%) no PGSB com HC-WGA.

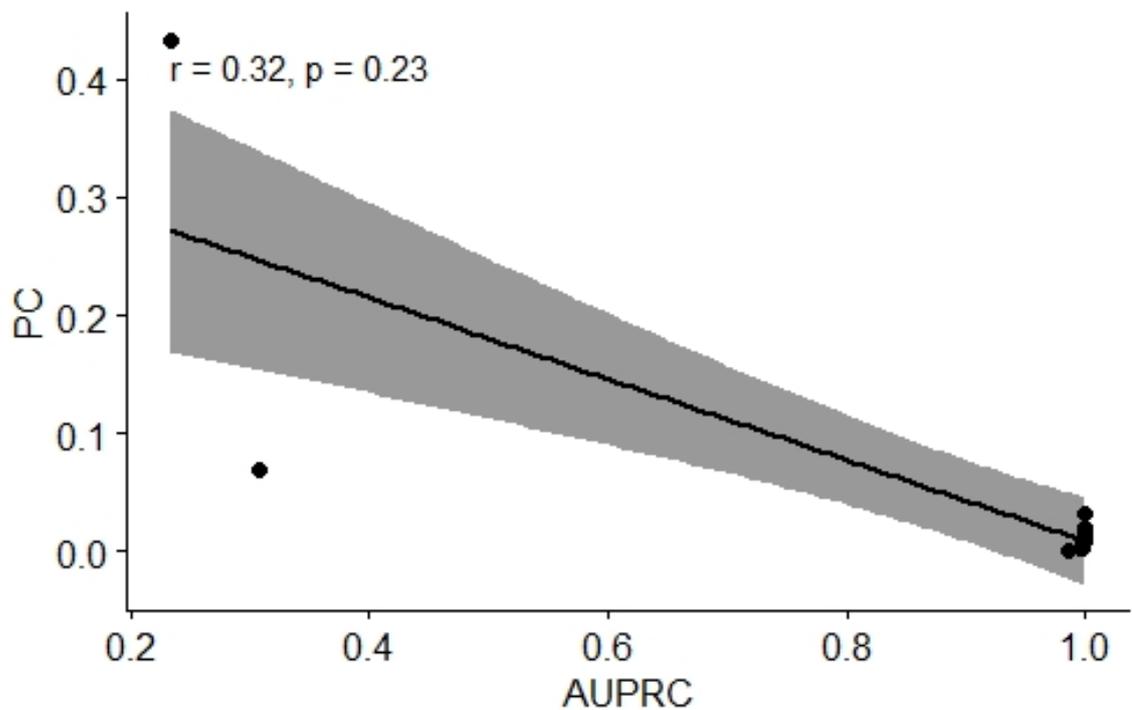


Figura 50 – Correlação de Spearman em  $AU(\overline{PCR})$  (40%) + PC (60%) no PGSB com HC-WGA.

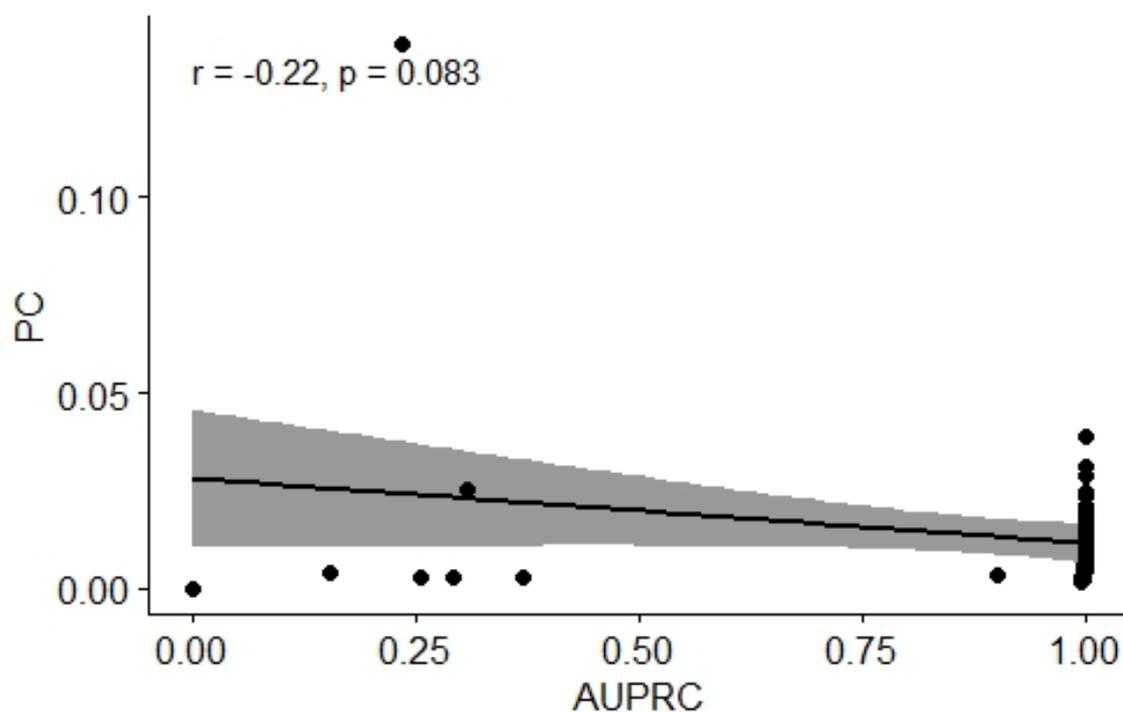


Figura 51 – Correlação de Pearson em  $AU(\overline{PCR})$  (50%) + PC (50%) no PGSB com HC-WGA.

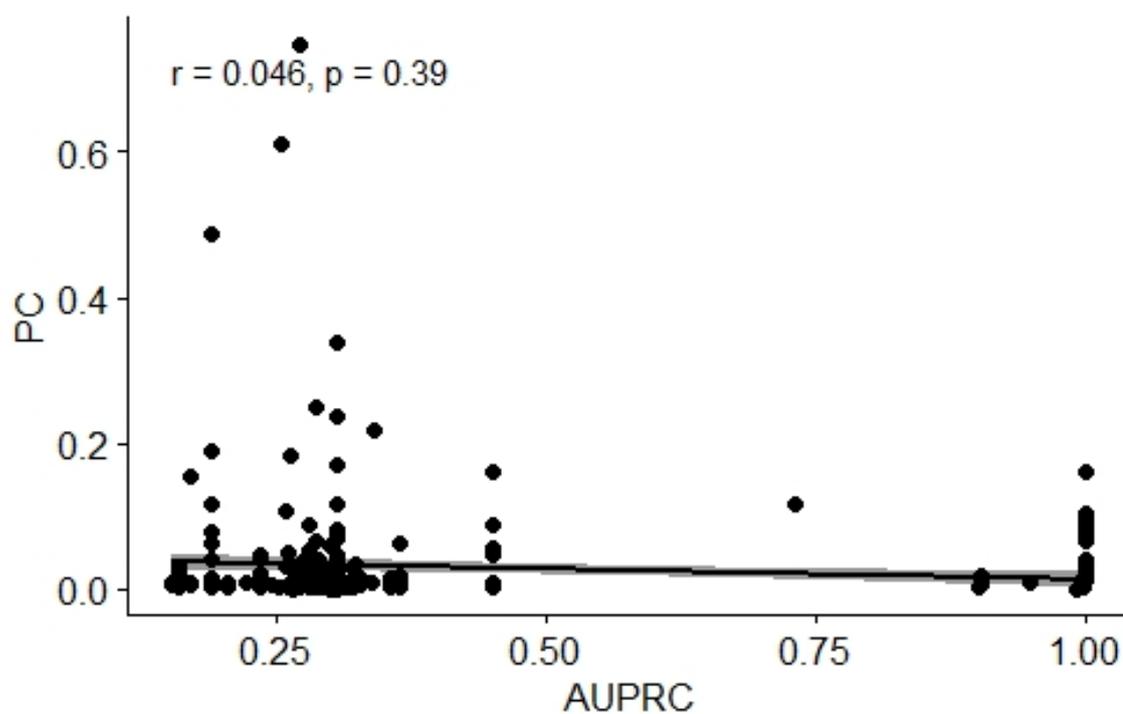


Figura 52 – Correlação de Spearman em  $AU(\overline{PCR})$  (80%) + PC (20%) no PGSB com HC-WGA.

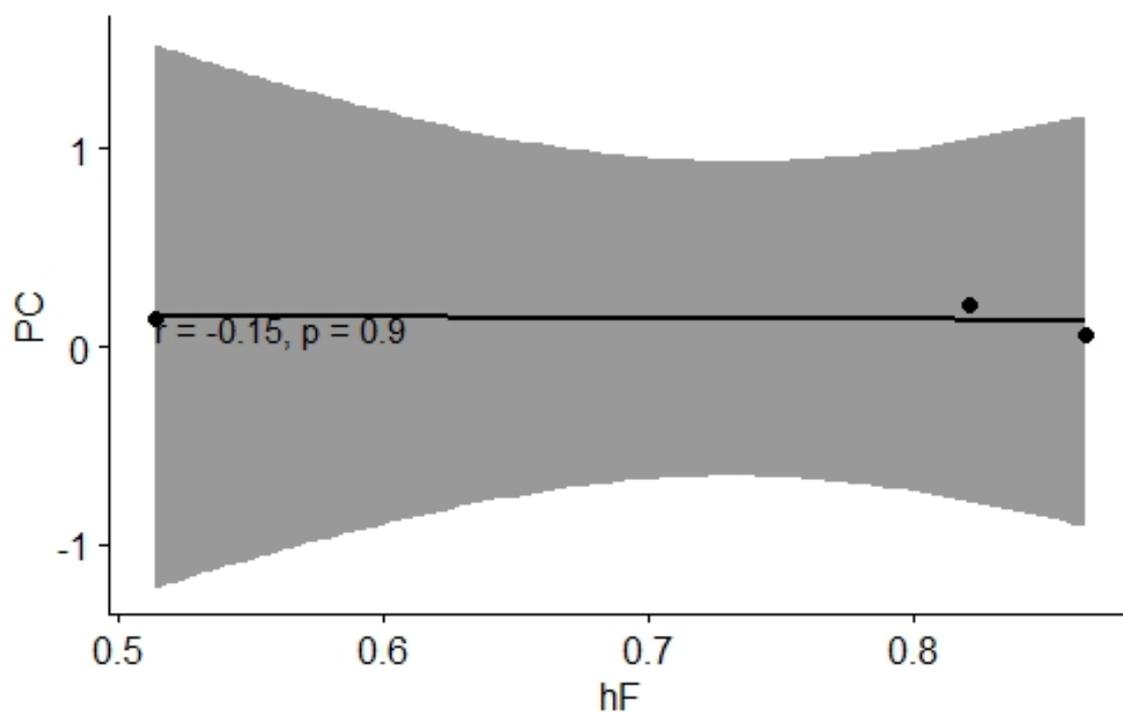


Figura 53 – Correlação de Pearson em hF (20%) + PC (80%) no PGSB com HC-WGA.

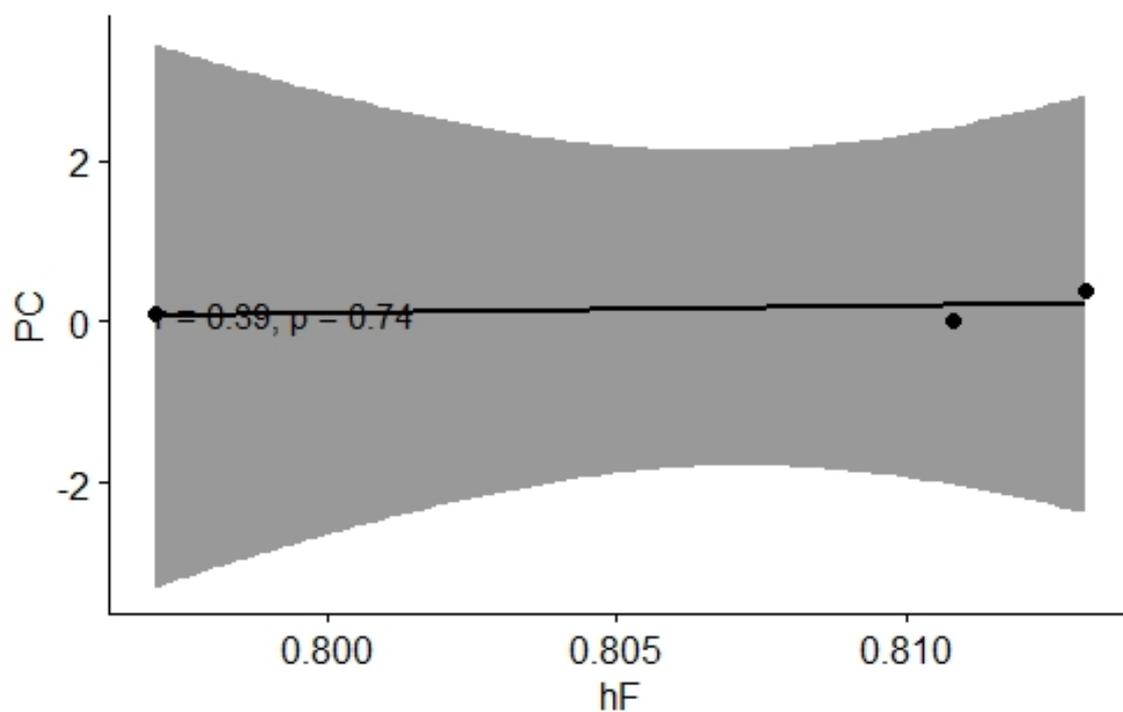


Figura 54 – Correlação de Pearson em hF (40%) + PC (60%) no PGSB com HC-WGA.

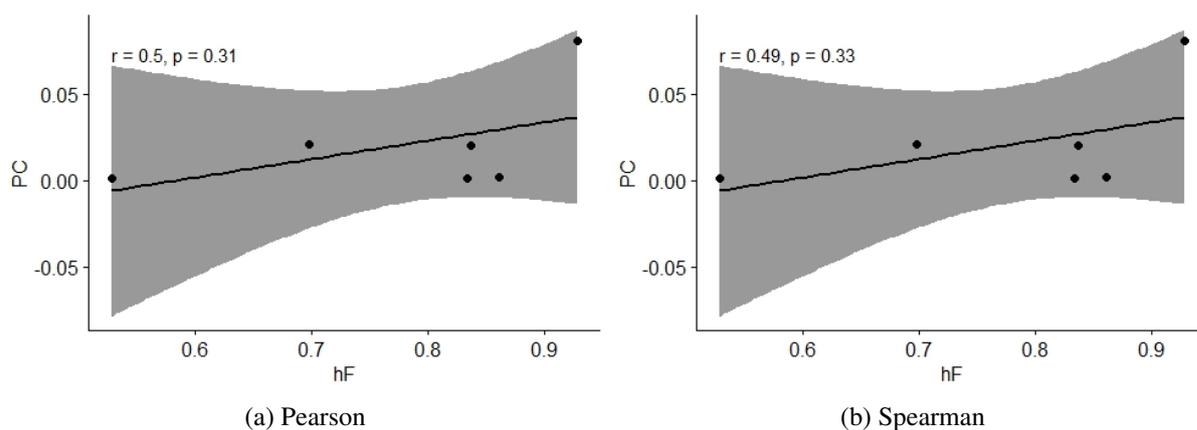


Figura 55 – Correlações de hF (50%) + PC (50%) no PGSB com HC-WGA.

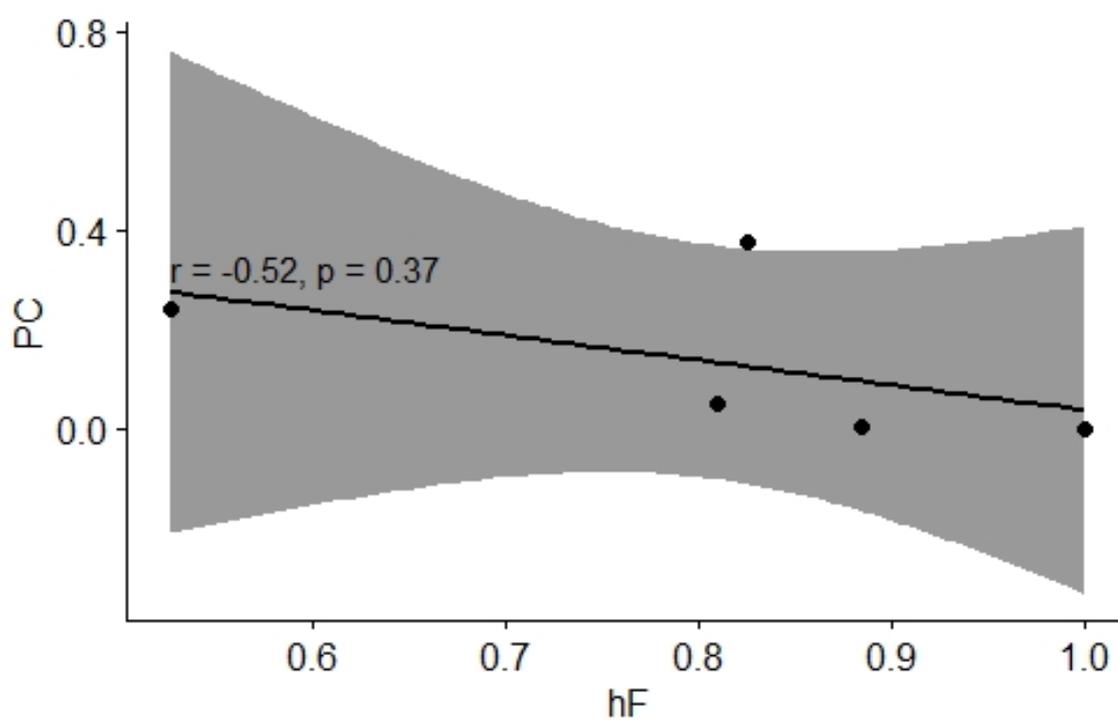


Figura 56 – Correlação de Pearson em hF (60%) + PC (40%) no PGSB com HC-WGA.

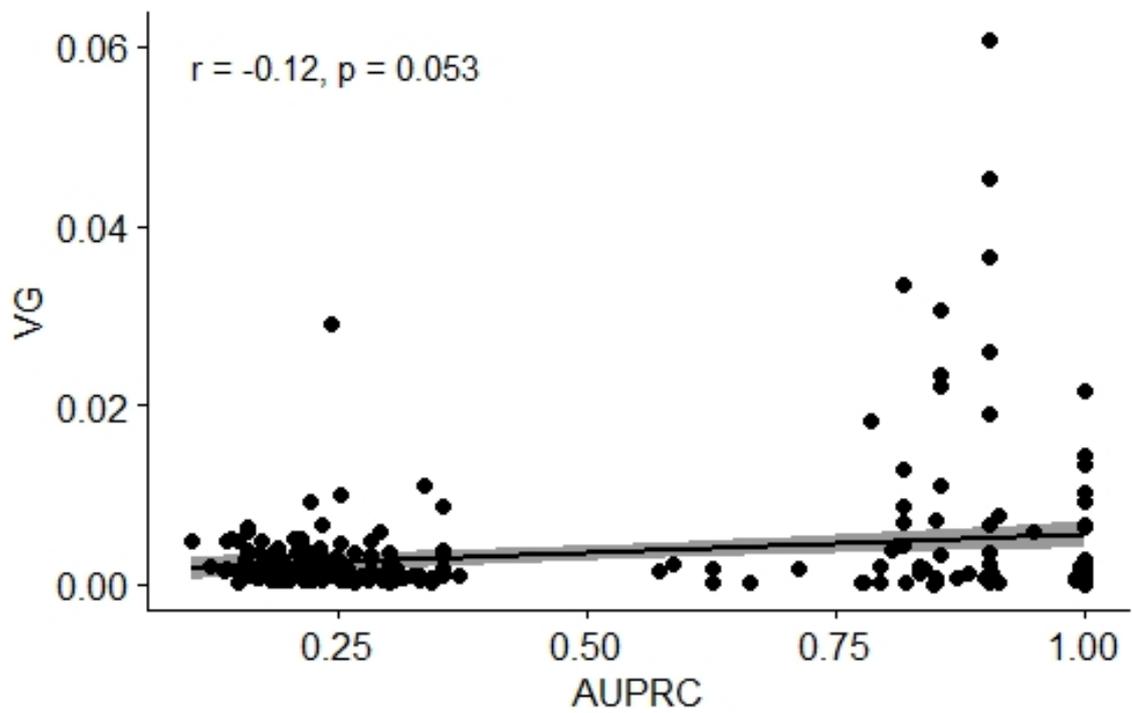


Figura 57 – Correlação de Spearman dos objetivos  $AU(\overline{PCR})$  e VG no REPBASE com HC-LGA.

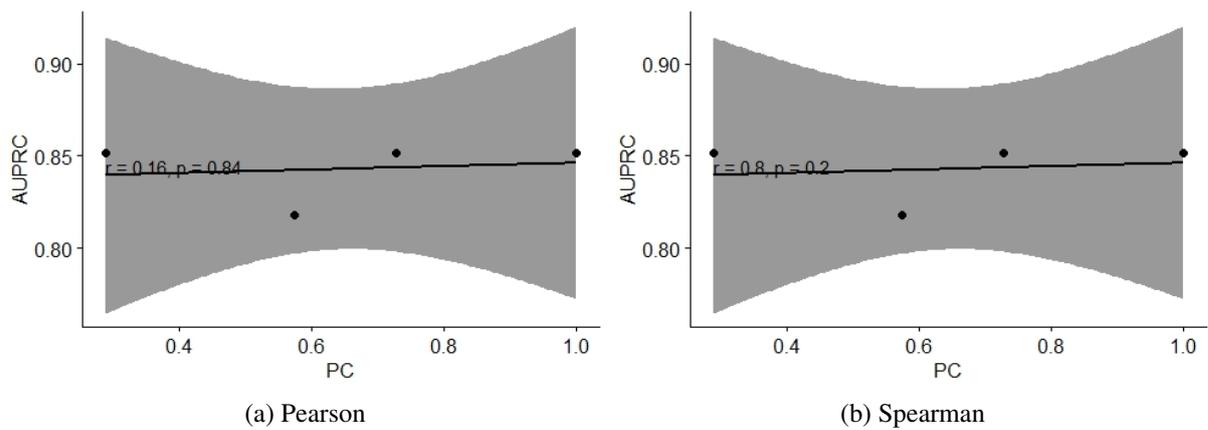


Figura 58 – Correlações dos objetivos PC e  $AU(\overline{PCR})$  no REPBASE com HC-LGA.

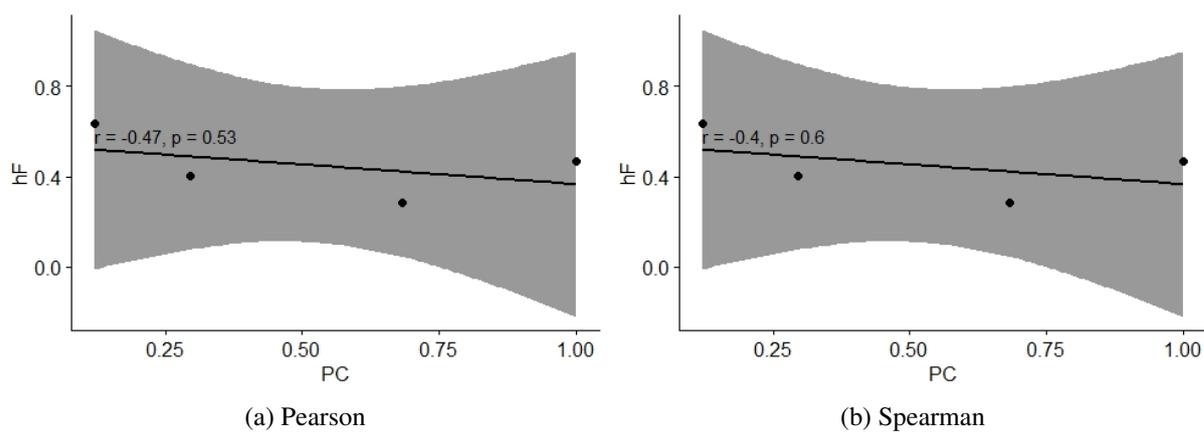


Figura 59 – Correlações dos objetivos PC e hF no REPBASE com HC-LGA.

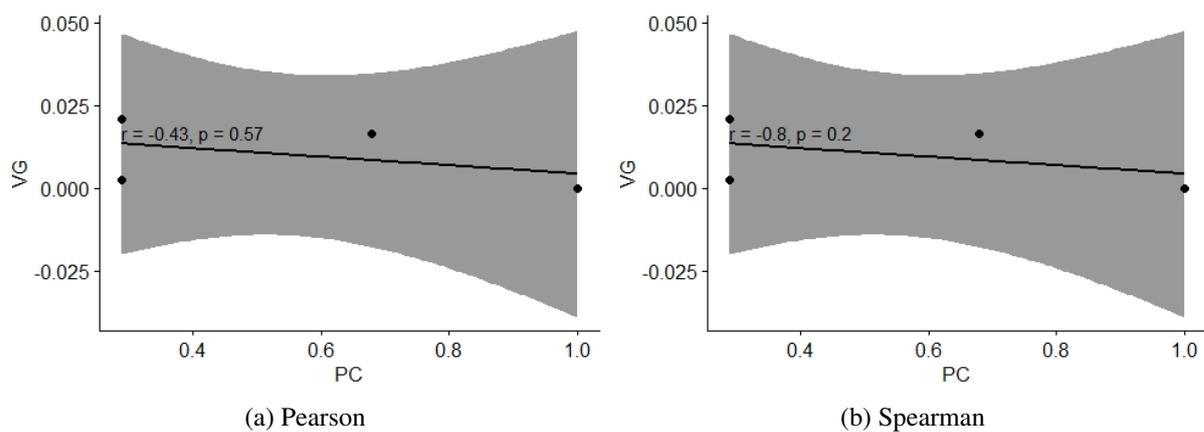


Figura 60 – Correlações dos objetivos PC e VG no REPBASE com HC-LGA.

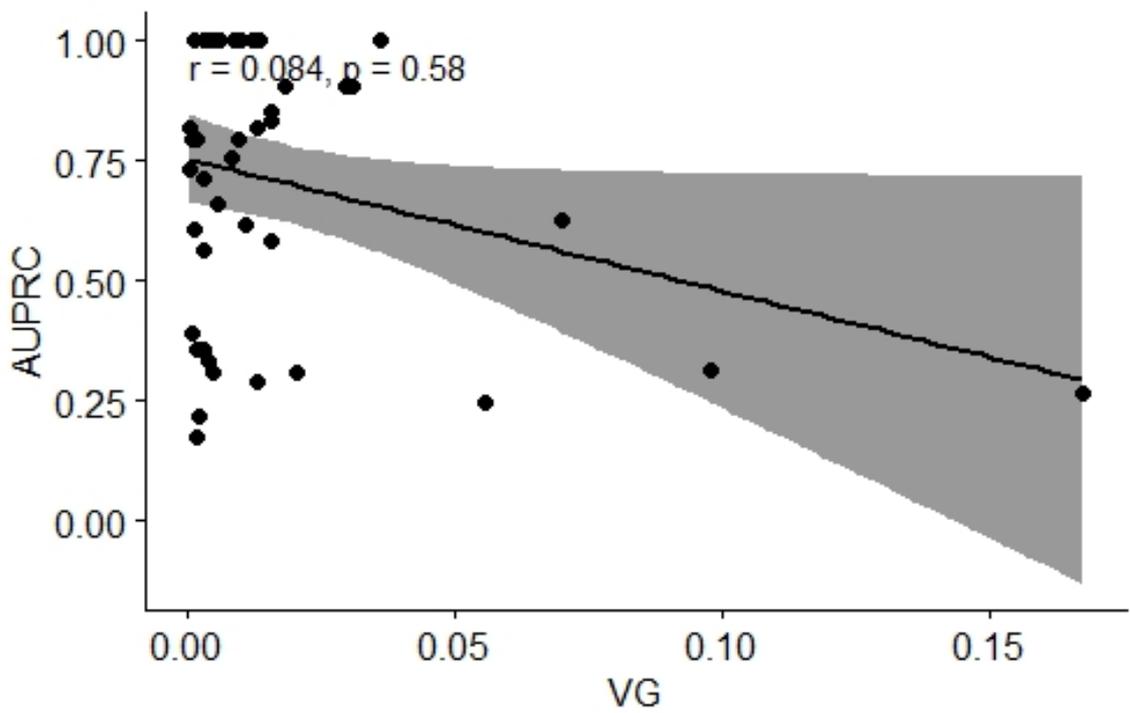


Figura 61 – Correlação de Spearman dos objetivos **VG** e  $AU(\overline{PRC})$  no REPBASE com **HC-LGA**.

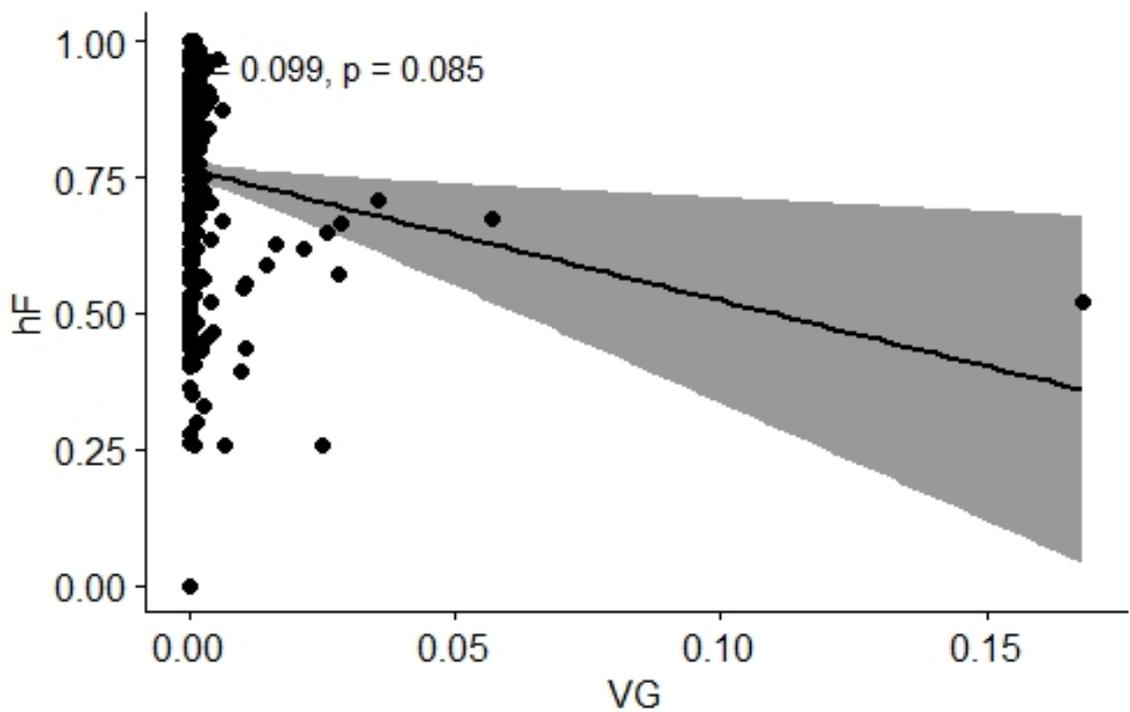


Figura 62 – Correlação de Spearman dos objetivos **VG** e **hF** no REPBASE com **HC-LGA**.

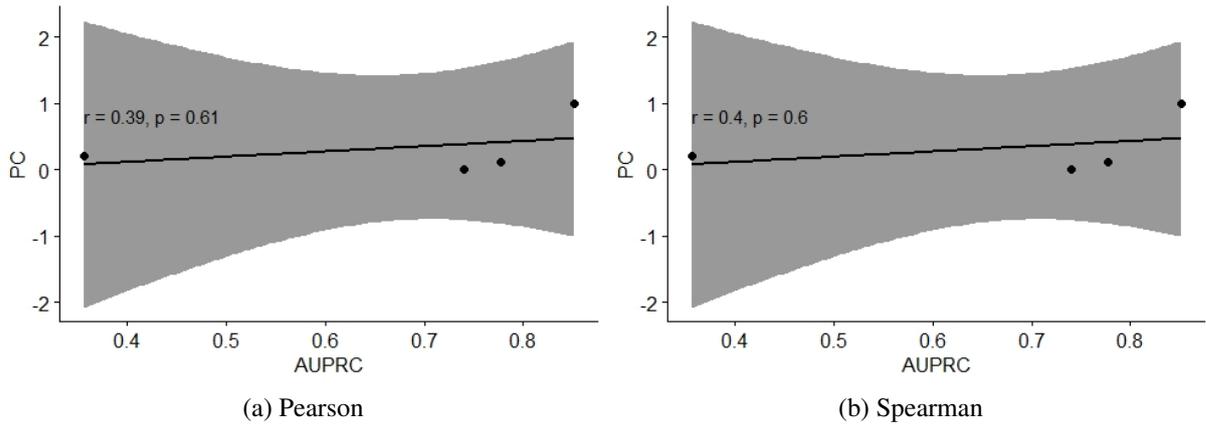


Figura 63 – Correlações de  $AU(\overline{PCR})$  (20%) + PC (80%) no REPBASE com HC-WGA.

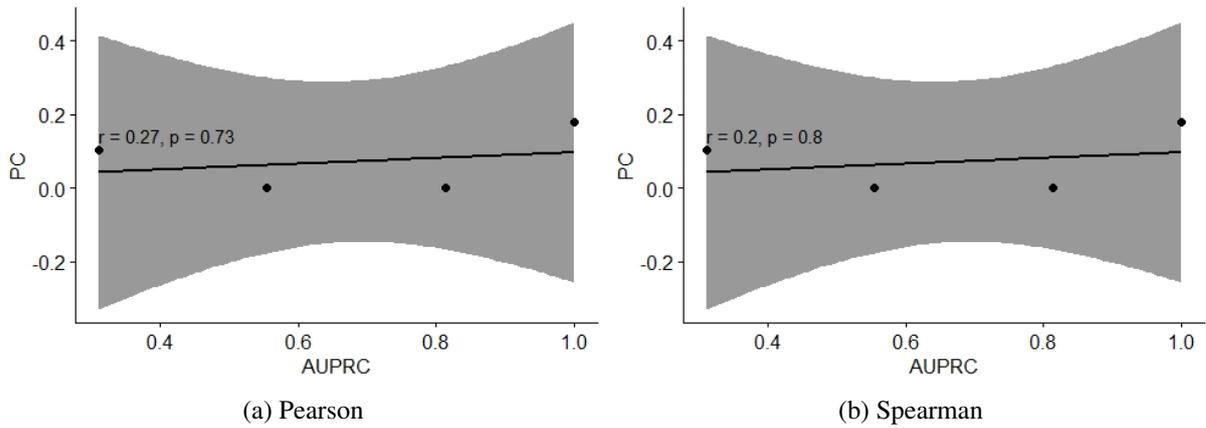


Figura 64 – Correlações de  $AU(\overline{PCR})$  (40%) + PC (60%) no REPBASE com HC-WGA.

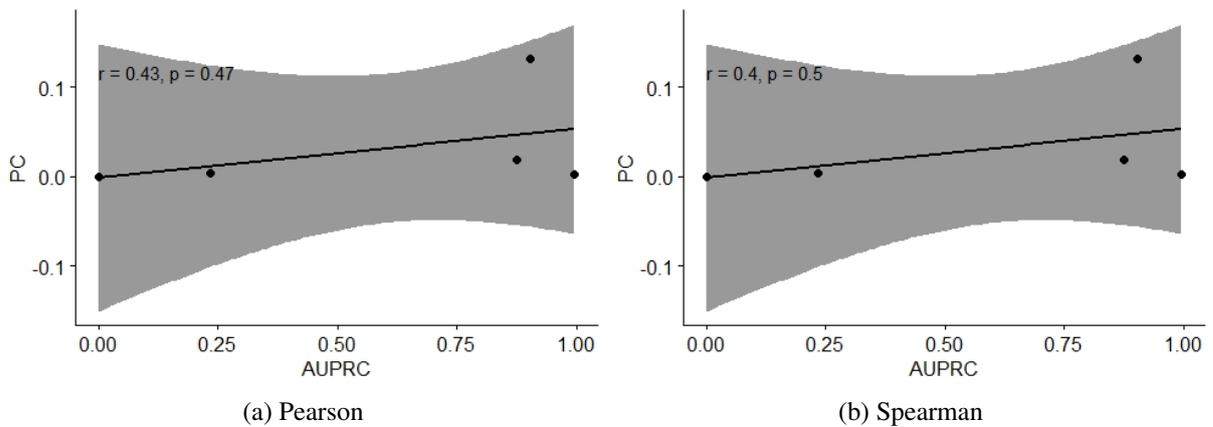


Figura 65 – Correlações de  $AU(\overline{PCR})$  (50%) + PC (50%) no REPBASE com HC-WGA.

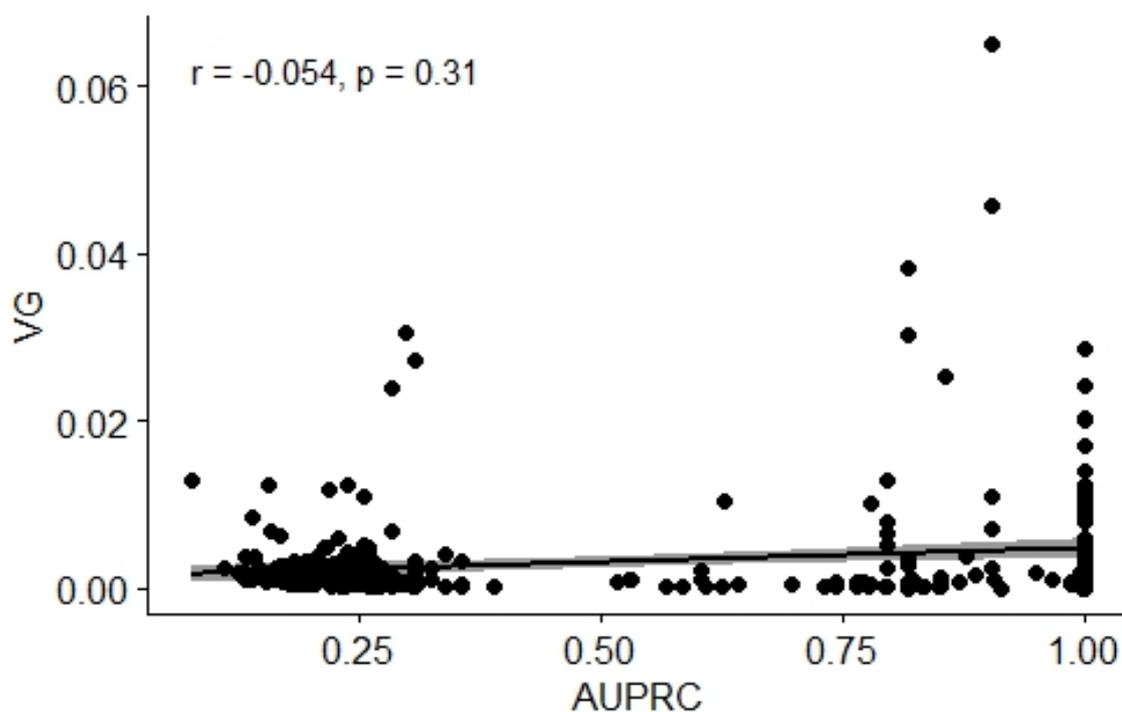


Figura 66 – Correlação de Spearman em  $AU(\overline{PRC})$  (40%) + VG (60%) no REPBASE com HC-WGA.

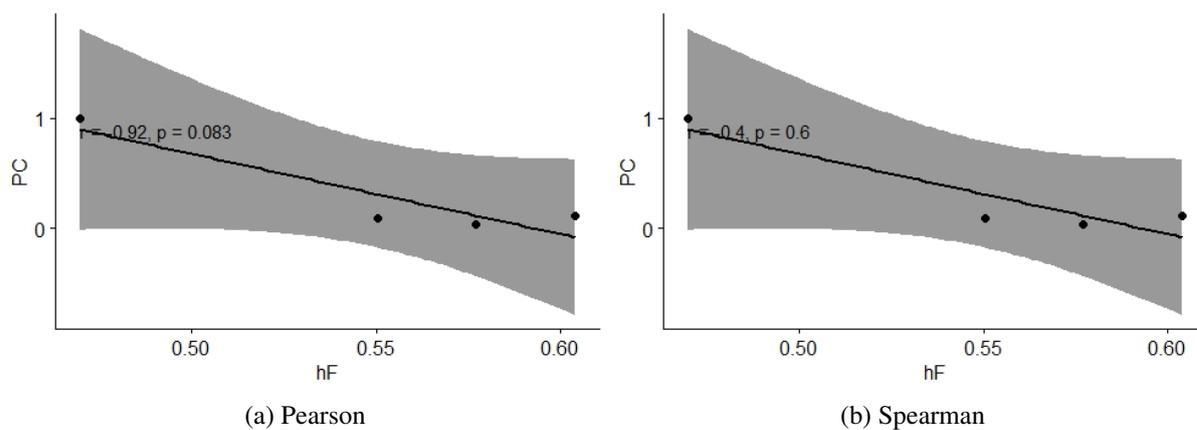


Figura 67 – Correlações de hF (20%) + PC (80%) no REPBASE com HC-WGA.

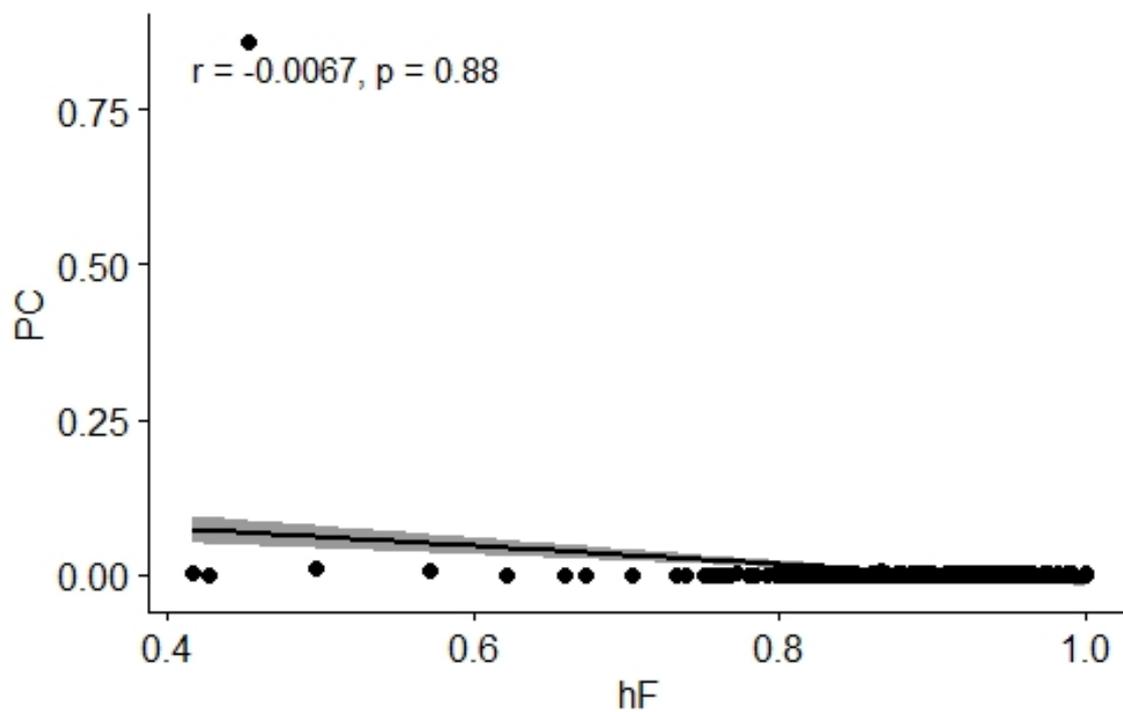


Figura 68 – Correlação de Spearman em **hF** (50%) + **PC** (50%) no REPBASE com **HC-WGA**.