
**Aprimorando a Predição de Interações Proteicas
no software Genppi: Utilizando os algoritmos
Random Forest e RIS (Reduced Interaction
Sampling) – um novo algoritmo para seleção de
amostras de interações que preserva a
informação topológica da rede.**

Alisson William da Silva



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uberlândia
2025

Alisson William da Silva

**Aprimorando a Predição de Interações Proteicas
no software Genppi: Utilizando os algoritmos
Random Forest e RIS (Reduced Interaction
Sampling) – um novo algoritmo para seleção de
amostras de interações que preserva a
informação topológica da rede.**

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Prof. Doutor Anderson Rodrigues dos Santos.

Uberlândia
2025

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

S586
2025

Silva, Alisson William da, 1975-
Melhoria da quantidade e qualidade de predições do
Genppi via uso do Random Forest em Common Lisp para
identificar similaridade proteica. [recurso eletrônico]
/ Alisson William da Silva. - 2025.

Orientador: Anderson Rodrigues dos Santos.
Dissertação (Mestrado) - Universidade Federal de
Uberlândia, Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
Disponível em: <http://doi.org/10.14393/ufu.di.2025.39>
Inclui bibliografia.
Inclui ilustrações.

1. Computação. I. Santos, Anderson Rodrigues dos, 1971-
, (Orient.). II. Universidade Federal de Uberlândia.
Pós-graduação em Ciência da Computação. III. Título.

CDU: 681.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:

Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074



ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Dissertação, 56/2025, PPGCO				
Data:	24 de janeiro de 2025	Hora de início:	10:40	Hora de encerramento:	12:40
Matrícula do Discente:	12312CCP004				
Nome do Discente:	Alisson William da Silva				
Título do Trabalho:	Melhoria da quantidade e qualidade de predições do Genppi via uso do Random Forest em Common Lisp para identificar similaridade proteica				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Inteligência Artificial				
Projeto de Pesquisa de vinculação:	EDITAL DIRPE Nº 2/2023 - PIBIC Estudo da oxidação anaeróbica de metano via relação sinérgica entre organismos archaea e bactéria utilizando algoritmos de anotação de genomas, construção e análise de rede de interação proteica e análise de dados via bancos de dados relacionais				

Reuniu-se por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta:
Professores Doutores: Bruno Augusto Nassif Travençolo - FACOM/UFU, Sérgio Vale Aguiar Campos - DCC/UFMG e Anderson Rodrigues dos Santos - FACOM/UFU, orientador do candidato.

Os examinadores participaram desde as seguintes localidades: Anderson Rodrigues dos Santos - Itambacuri/MG, Sérgio Vale Aguiar Campos - Lagoa Santa/MG. O outro membro da banca estava em Uberlândia e o aluno Alisson William da Silva participou da cidade de Bom Despacho/MG.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Anderson Rodrigues dos Santos, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir ao candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Sérgio Vale Aguiar Campos, Usuário Externo**, em 24/01/2025, às 16:26, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Bruno Augusto Nassif Travençolo, Professor(a) do Magistério Superior**, em 24/01/2025, às 16:27, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Anderson Rodrigues dos Santos, Professor(a) do Magistério Superior**, em 24/01/2025, às 18:51, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5989076** e o código CRC **EBD84549**.

*Aos que tornaram esta jornada possível, dedico estas páginas com profunda gratidão.
A Deus, pela oportunidade de seguir em frente, mesmo nos momentos mais desafiadores.
Aos meus pais, pilares inabaláveis, que dedicaram suas vidas para que eu pudesse alcançar
meus sonhos. Em especial, ao meu pai, cuja partida em 2024 deixou uma ausência
imensurável, mas cujo legado de amor e apoio continua a guiar meus passos.
Às minhas filhas, que, mesmo distantes geograficamente, permanecem próximas ao meu
coração. Seu amor e inspiração transcendem a distância, e são parte fundamental de quem
sou hoje. À minha companheira, pelo suporte incansável, pelas horas de compreensão e
pelo incentivo que me permitiu seguir adiante, mesmo quando tudo parecia difícil.
E aos meus amigos, pela simples presença em minha vida, lembrando-me que não estamos
sozinhos nas lutas e nas vitórias.
Esta dissertação é um reflexo de todos vocês, que, de alguma forma,
fizeram parte desta trajetória. Meu muito obrigado.*

Agradecimentos

Agradeço especialmente ao meu orientador, Prof. Doutor Anderson Rodrigues dos Santos., pela organização, paciência, disponibilidade e atenção, mas sobretudo, pelos ensinamentos transmitidos, aos quais foram fundamentais para a conclusão deste trabalho.

Gostaria também de expressar minha gratidão à equipe de trabalho que participou do projeto de investigação de proteínas relacionadas à Oxidação Anaeróbica do Metano (AOM). Essa equipe, composta por alunos de iniciação científica, mestrado e de outros cursos, desempenhou um papel essencial no desenvolvimento desta pesquisa. Através de atividades interdisciplinares, discussões técnicas e análises computacionais que mesclaram biologia molecular, bioquímica e bioinformática, cada um de vocês contribuiu de maneira única para este trabalho. Agradeço pelo empenho, colaboração e pela troca de conhecimentos que enriqueceram minha trajetória acadêmica.

Por fim, agradecimentos especiais são direcionados aos professores da Faculdade de Computação da Universidade Federal de Uberlândia, quais foram responsáveis e co-partícipes da minha aprendizagem na área da computação.

“O essencial é invisível aos olhos.”

Antoine de Saint-Exupéry

Resumo

O contexto deste estudo reside na crescente necessidade de métodos computacionais eficientes para prever PPIs, dada a importância dessas interações para a compreensão da biologia celular e o desenvolvimento de novos fármacos. Métodos experimentais são dispendiosos e demorados, tornando as abordagens computacionais uma alternativa promissora. A versão original do *Genppi* já representava um avanço nesse campo, mas sua capacidade era limitada pela dependência de alta similaridade proteica. Este trabalho apresenta um avanço significativo no desenvolvimento do software *Genppi*, uma ferramenta de bioinformática dedicada à predição de redes de interação proteína-proteína (PPI) em genomas bacterianos. A versão 1.5 do *Genppi* incorpora o algoritmo de *Random Forest*, implementado em *Common Lisp*, com o objetivo de superar as limitações da versão anterior e ampliar a precisão e a abrangência na análise de interações proteicas, especialmente em cenários de baixa similaridade sequencial. A pesquisa aborda tanto a implementação do aprendizado de máquina quanto a otimização da geração de redes através do algoritmo “*RIS*” (*Reduced Interaction Sampling*), desenvolvido para lidar com a complexidade computacional inerente à análise de genomas completos.

A metodologia empregada foi multifacetada. Inicialmente, o algoritmo *Random Forest* foi integrado ao *Genppi*, permitindo a classificação de similaridade proteica de maneira eficiente. Em seguida, para otimizar o desempenho do software, o algoritmo “*RIS*” foi implementado para reduzir a complexidade computacional associada à geração de redes de interação a partir de perfis filogenéticos conservados com um grande número de genes. A avaliação da eficácia dessas implementações envolveu a geração de 180 redes de interação em diferentes configurações, variando o modo de operação (Features, *Machine Learning*), o parâmetro *Top_N* (definindo o número de nós mais relevantes) e o número de arquivos analisados. As redes geradas foram submetidas a uma análise rigorosa utilizando diversas métricas topológicas (Grau, *Betweenness Centrality*, *Bridging Centrality*) e testes estatísticos (*Kolmogorov-Smirnov*, Correlação de *Kendall Tau*, Índice de *Fleiss*) para

avaliar a estabilidade, precisão e integridade estrutural das redes.

Os resultados demonstraram que a integração do *Random Forest*, no modo *Machine Learning*, ampliou significativamente a capacidade do *Genppi* de prever interações proteicas. A aplicação do modelo ao genoma da bactéria *Buchnera Aphidicola* revelou uma sobreposição de até 62% com interações documentadas no banco de dados *STRING*, validando a precisão e completude das predições. O modo *Features*, que utiliza características biofísicas pré-definidas, também apresentou resultados consistentes, mantendo uma alta estabilidade estrutural mesmo com a aplicação do algoritmo *RIS*. A análise do algoritmo *RIS* revelou que, embora introduza alguma aleatoriedade, a identificação de nós críticos permanece robusta para valores de *Top_N* maiores ou iguais a 100, indicando que a redução do número de arestas não compromete significativamente a integridade da rede. A Média Global Ponderada (MGP) e outros indicadores confirmaram que o algoritmo preserva a estrutura global das redes, mesmo com a redução da complexidade computacional.

Em conclusão, este trabalho demonstra que a combinação do algoritmo *Random Forest* com o algoritmo *RIS* representa um avanço significativo na predição de redes de interação proteica. A integração do *Random Forest* permitiu ao *Genppi* superar a limitação da dependência de alta similaridade sequencial, enquanto o algoritmo *RIS* otimizou o desempenho do software, permitindo a análise de genomas complexos de forma eficiente. A validação dos resultados com o banco de dados *STRING* e a análise estatística rigorosa confirmam a precisão e a robustez das predições geradas pelo *Genppi* 1.5. A disponibilização pública dos dados e códigos promove a transparência e a reprodutibilidade da pesquisa, incentivando a colaboração e o avanço da bioinformática. Este estudo contribui para o desenvolvimento de ferramentas mais eficazes para a compreensão dos processos moleculares em organismos bacterianos, com potencial para aplicações em biomedicina, biotecnologia e outras áreas científicas. O *Genppi*, aprimorado com essas novas funcionalidades, se consolida como uma solução robusta e escalável para a predição de interações proteína-proteína, baseada em aprendizado de máquina e características biofísicas dos genomas.

Palavras-chave: Genppi. Biologia Computacional. Florestas Randômicas. Redes de Interação Proteica. Bioinformática..

Abstract

The context of this study lies in the growing need for efficient computational methods to predict PPIs, given the importance of these interactions for understanding cellular biology and developing new drugs. Experimental methods are costly and time-consuming, making computational approaches a promising alternative. The original version of *Genppi* already represented an advancement in this field, but its capabilities were limited by its dependence on high protein sequence similarity. This work presents a significant advancement in the development of the *Genppi* software, a bioinformatics tool dedicated to predicting protein-protein interaction (PPI) networks in bacterial genomes. Version 1.5 of *Genppi* incorporates the *Random Forest* algorithm, implemented in *Common Lisp*, with the goal of overcoming the limitations of the previous version and expanding the accuracy and scope of protein interaction analysis, especially in scenarios with low sequence similarity. The research addresses both the implementation of machine learning and the optimization of network generation through the “*RIS*” algorithm (*Reduced Interaction Sampling*), developed to handle the computational complexity inherent in analyzing complete genomes.

The methodology employed was multifaceted. Initially, the *Random Forest* algorithm was integrated into *Genppi*, enabling efficient protein similarity classification. Subsequently, to optimize software performance, the “*RIS*” algorithm was implemented to reduce the computational complexity associated with generating interaction networks from conserved phylogenetic profiles with a large number of genes. Evaluating the effectiveness of these implementations involved generating 180 interaction networks in different configurations, varying the operating mode (*Features*, *Machine Learning*), the *Top_N* parameter (defining the number of most relevant nodes), and the number of files analyzed. The generated networks were subjected to rigorous analysis using various topological metrics (*Degree*, *Betweenness Centrality*, *Bridging Centrality*) and statistical tests (*Kolmogorov-Smirnov*, *Kendall Tau* correlation, *Fleiss’ Kappa*) to assess the stability, accuracy, and structural integrity of the networks.

The results demonstrated that the integration of *Random Forest* in *Machine Learning* mode significantly expanded *Genppi*'s ability to predict protein interactions. Applying the model to the genome of the bacterium *Buchnera Aphidicola* revealed an overlap of up to 62% with interactions documented in the *STRING* database, validating the accuracy and completeness of the predictions. The *Features* mode, which utilizes predefined biophysical characteristics, also presented consistent results, maintaining high structural stability even with the application of the *RIS* algorithm. Analysis of the “*RIS*” algorithm revealed that, although it introduces some randomness, the identification of critical nodes remains robust for *Top_N* values greater than or equal to 100, indicating that reducing the number of edges does not significantly compromise the network's integrity. The *Weighted Global Mean*(MGP) and other indicators confirmed that the algorithm preserves the global structure of the networks, even with reduced computational complexity.

In conclusion, this work demonstrates that the combination of the *Random Forest* algorithm with the *RIS* algorithm represents a significant advancement in the prediction of protein interaction networks. The integration of *Random Forest* allowed *Genppi* to overcome the limitation of dependence on high sequence similarity, while the *RIS* algorithm optimized software performance, enabling efficient analysis of complex genomes. Validation of the results with the *STRING* database and rigorous statistical analysis confirm the accuracy and robustness of the predictions generated by *Genppi* 1.5. Public availability of the data and code promotes transparency and reproducibility of the research, encouraging collaboration and advancement in bioinformatics. This study contributes to the development of more effective tools for understanding molecular processes in bacterial organisms, with potential applications in biomedicine, biotechnology, and other scientific fields. *Genppi*, enhanced with these new features, solidifies its position as a robust and scalable solution for predicting protein-protein interactions based on machine learning and biophysical characteristics of genomes.

Keywords: *Genppi*. Computational Biology. Random Forest. Protein Interaction Networks. Bioinformatics..

Lista de ilustrações

Figura 1 – Dupla-Hélice de DNA: Características e Natureza antiparalela - Adaptado de (STRACHAN; READ, 2016).	36
Figura 2 – Tabela de <i>Códons</i> - Adaptado de (WATSON et al., 2015).	39
Figura 3 – Modelo de árvores de decisão - Adaptado de (STARMER, 2023).	45
Figura 4 – Modelo de “bootstrap” - Adaptado de (STARMER, 2023).	46
Figura 5 – Exemplo de Bag of out - Adaptado de (STARMER, 2023).	47
Figura 6 – Funcionamento dos métodos “Bagging” e <i>Random Forest</i> - Adaptado de (OSHIRO, 2013).	48
Figura 7 – Modelos de Aprendizado de Máquina - Adaptado de (TANG et al., 2023).	58
Figura 8 – Exemplo da saída do arquivo gerado report.txt do diretório gene-neighborhood-conservation-report.	64
Figura 9 – Exemplo da saída do arquivo gerado do diretório pan-genome.	65
Figura 10 – Exemplo da saída do arquivo gerado report.txt do diretório phylogenetic-profiles-report.	66
Figura 11 – Exemplo da saída do arquivo gerado do diretório ppi-files.	67
Figura 12 – Exemplo da saída do arquivo gerado report.txt do diretório ppi-report.	68
Figura 13 – Fluxograma da Metodologia	70
Figura 14 – Fluxograma do processo para geração de testes.	110
Figura 15 – Exemplo de saída Percentual de Presença de nós.	127
Figura 16 – Exemplo de saída dos Cálculos de Similaridade, Correlação e Distribuição.	128
Figura 17 – Comparação do Coeficiente de <i>Jaccard</i> entre arquivos - Exemplo.	130
Figura 18 – Teste de <i>Kolmogorov-Smirnov</i> entre arquivos - Exemplo.	131
Figura 19 – Comparação do Coeficiente de <i>Overlap</i> entre arquivos - Exemplo.	132
Figura 20 – Correlação de <i>Kendall Tau</i> entre arquivos - Exemplo.	133
Figura 21 – Comportamento das Métricas Normalizadas em Função do Top_N no modo <i>Machine Learning</i>	139

Lista de tabelas

Tabela 1	– Genomas empregados e resultados da classificação da Random Forest .	86
Tabela 2	– Avaliação de Desempenho do Modelo de Classificação em Diversos Genomas	87
Tabela 3	– Comparação do Desempenho do Genppi com <i>Features</i> e Machine Learning em Diferentes Conjuntos de Dados	88
Tabela 4	– Resultado da execução da consulta ao banco compareba.dump em diferentes implementações e parâmetros.	90
Tabela 5	– Avaliação das Implementações do <i>Genppi</i>	92
Tabela 6	– Parâmetros variados e combinações testadas	112
Tabela 7	– Presença de <i>Labels</i> em 10 Arquivos (Exemplo Detalhado)	115
Tabela 8	– Comparação de Impacto entre Média Simples e Ponderada	116
Tabela 9	– Resultados para o Modo <i>Features</i> (Dados Filtrados)	134
Tabela 10	– Resultados para Métrica <i>Degree</i> - Modo <i>Machine Learning</i>	136
Tabela 11	– Resultados para Métrica <i>Betweenness Centrality</i> - Modo <i>Machine Learning</i>	137
Tabela 12	– Resultados para Métrica <i>Bridging Centrality</i> - Modo <i>Machine Learning</i>	138

Lista de siglas

AIST Instituto Nacional de Ciência e Tecnologia Industrial Avançada (National Institute of Advanced Industrial Science and Technology)

ANSI Instituto Nacional Americano de Padrões (American National Standards Institute)

ARFF Formato de arquivo de atributo-relação (Attribute-Relation File Format)

BBLN Rede de aprendizagem bilateral (Bilateral-branch learning network)

BLAST Ferramenta de busca básica de alinhamento local (Basic Local Alignment Search Tool)

CDF Função de Distribuição Acumulada

CEGSO Boosting Essential Proteins Prediction by Integrating Protein Complex, Gene Expression, Gene Ontology, Subcellular Localization and Orthology Information

CNN Redes Neurais Convolucionais (Convolutional Neural Networks)

DNA Ácido Desoxirribonucleico (Deoxyribonucleic Acid)

DNN Redes Neurais Profundas (Deep Neural Networks)

EBI Instituto Europeu de Bioinformática (European Bioinformatics Institute)

GENPPI Gerador de interações proteína-proteína (Generator protein-protein interactions)

IA Inteligência Artificial

KNN K-vizinhos mais próximos (K-nearest neighbors)

MGP Média Global de Presença Ponderada

MCR Methyl-coenzyme M reductase

ML Aprendizado de Máquina (Machine Learning)

NCBI Centro Nacional de Informação sobre Biotecnologia (National Center for Biotechnology Information)

ORF Frame de Leitura Aberto (Open Reading Frame)

PCC Coeficiente de correlação de Pearson (Pearson Correlation Coefficient)

PP Proteína-Proteína

PPI Interação Proteína-Proteína (Protein-Protein Interaction)

REPL Loop de leitura-avaliação-impressão (Read-Eval-Print Loop)

RIS Reduced Interaction Sampling

RF Florestas Aleatórias (Random Forests)

ROC Receiver Operating Characteristic

RNA Ácido Ribonucleico (Ribonucleic Acid)

SBCL Steel Bank Common Lisp

STRING Search Tool for the Retrieval of Interacting Genes/Proteins

SVM Máquina de vetores de suporte (Support Vector Machine)

SVM-RFE Máquina de vetores de suporte - Eliminação de recursos recursivos (Support Vector Machine - Recursive Feature Elimination)

Sumário

1	INTRODUÇÃO	25
1.1	Motivação	27
1.1.1	Desafios	29
1.2	Objetivos	30
1.2.1	Objetivo Geral	30
1.2.2	Objetivos Específicos	30
1.3	Hipótese	31
1.4	Contribuições	32
1.4.1	Científicas	32
1.4.2	Práticas	32
1.5	Organização da Dissertação	33
2	FUNDAMENTAÇÃO TEÓRICA	35
2.1	Biotecnologia	35
2.1.1	Genoma, Pangenoma e Proteoma	37
2.1.2	Técnicas da Bioinformática e suas aplicações	37
2.1.3	Estrutura e Sequenciamento do DNA	40
2.2	Redes de Interação Proteína - Proteína	41
2.2.1	Algoritmos de Análises de PPI	42
2.3	Random Forest	43
2.3.1	Árvores de Decisão	43
2.3.2	Definição e características	45
2.3.3	Comparativos a outros métodos de aprendizado de máquina	49
2.4	Common Lisp	50

3	ESTADO DA ARTE	53
3.1	Panorama Tecnológico em Bioinformática	53
3.1.1	Arquivos MultiFasta	53
3.1.2	AAIndex	54
3.1.3	STRING DataBase e BioGRID	54
3.1.4	Algoritmo Needleman-Wunsch e Smith-Waterman	54
3.1.5	Blast	55
3.2	Algoritmos baseados em Aprendizado de Máquina	57
3.3	Trabalhos Relacionados	60
3.4	Genppi	61
4	MATERIAIS E MÉTODOS	69
4.1	Coleta e Preparação de Dados	69
4.1.1	MakeArff	71
4.2	Utilização de Banco de Dados	72
4.3	Funcionalidades Agregadas ao Genppi	73
4.3.1	Features	73
4.3.2	Integração da biblioteca Cl-Random-Forest	74
4.3.3	Reduced Interaction Sampling (RIS)	77
5	RESULTADOS	81
5.1	Experimentos	81
5.1.1	Genomas Utilizados para Treinamento e Testes	82
5.1.2	Desenvolvendo o Modelo de Aprendizado de Máquina	82
5.1.3	Aplicando o Modelo Treinado no Genppi	83
5.1.4	Avaliação da Estabilidade Topológica sob Aleatorização	84
5.2	Resultados	85
5.2.1	Avaliação de Desempenho do Modelo de Classificação	86
5.2.2	Comparação de Resultados: <i>Genppi</i> v1.0 vs. <i>Genppi</i> v1.5	88
5.2.3	Comparação com o STRING	89
5.2.4	Avaliação das Implementações	91
5.3	Discussão	93
5.3.1	Motivo da escolha do Random Forest	93
5.3.2	Validação de novas interações	94
5.3.3	Confiabilidade Computacional e Relevância Biológica nas Predições de Interações Proteicas	95

6	CONCLUSÃO	97
6.1	Limitações	99
6.2	Disponibilidade do Genppi	99
6.3	Trabalhos Futuros	100
	REFERÊNCIAS	101

APÊNDICES 107

APÊNDICE A	– AVALIAÇÃO DO IMPACTO DO ALGORITMO RIS NA GERAÇÃO DE REDES DE INTERA- ÇÃO PROTEÍNA-PROTEÍNA NO GENPPI .	109
A.1	Introdução	109
A.2	Metodologia	110
A.2.1	Base de Dados	111
A.2.2	Cenário de Testes	112
A.2.3	Métricas Utilizadas	113
A.3	Resultados	125
A.3.1	Organização do Material Disponibilizado	125
A.3.2	Média Global de Presença Ponderada	126
A.3.3	Métricas de Similaridade, Correlação e Distribuição	128
A.3.4	Resumos das Médias Globais	134
A.4	Avaliação	140
A.4.1	Modo <i>Features</i>	140
A.4.2	Modo <i>Machine Learning</i>	140
A.5	Conclusão	143

Introdução

O estudo das proteínas é de fundamental importância no desenvolvimento de novos fármacos. Em especial, o papel das redes de interação proteína-proteína (PPI) é necessário para a compreensão da biologia celular. As proteínas são as moléculas responsáveis pela execução de diversos processos celulares, incluindo a regulação gênica, a sinalização e organização celular. As PPIs são essenciais para a interação e comunicação entre essas proteínas. Todavia, a identificação de PPIs é um processo desafiador, pois os métodos experimentais são caros e demorados. Considerando ainda o aumento significativo na disponibilidade de genomas bacterianos sequenciados na última década, os métodos computacionais tornaram-se uma alternativa eficiente para prever redes de interação proteína-proteína (RASTI; VOGIATZIS, 2019).

O *Genppi*, acrônimo de *Generator protein-protein interactions*, é um software autônomo de bioinformática desenvolvido na Universidade Federal de Uberlândia para criar redes de interação de proteínas a partir de genomas. Ele é escrito em *Common Lisp*, compilado com SBCL, pode processar centenas de genomas, dependendo da memória RAM disponível. Por meio da análise de genomas bacterianos busca evidências de eventos evolutivos que indicam interações de proteínas. Estes eventos incluem: vizinhança gênica conservada ¹, a fusão gênica ² e o perfil filogenético conservado ³. Utiliza-se de uma heurística para comparação par-a-par de sequência de aminoácidos de proteínas com base na similaridade de sequências e da estrutura secundária das proteínas. Esta heurística se vale de uma análise linear sobre a quantidade de aminoácidos (representados por letras) presente em cada par de proteínas avaliadas (ANJOS, 2021).

¹ Intervalos recorrentes de genes nos genomas;

² Quando dois genes individuais de um organismo se fundem como uma sequência única formando um novo gene em outro organismo;

³ Genes que ocorrem e estão ausentes nos mesmos genomas de um conjunto de referência.

A similaridade proteica é um indicador inicial de que as proteínas podem compartilhar funções e estruturas comuns, potencialmente permitindo a interação entre elas. No entanto, é importante destacar que, antes de inferir sobre as interações, é necessário identificar quais proteínas são iguais ou similares. Isso envolve a geração do pangenoma central, que representa as proteínas comuns entre diferentes genomas. Em seguida, a análise da vizinhança desses genes pode revelar interações entre pares de proteínas, mesmo na ausência de similaridade proteica. O *Genppi* utilizava um critério de similaridade para determinar se as proteínas eram suficientemente semelhantes para serem consideradas como candidatas a entrarem na lista de investigação de características que definem uma interação. O software estabelecia um limite mínimo de identidade proteica de 90% para considerar as proteínas como similares, embora oferecesse opções para ajustar esse critério. Percentuais inferiores ao padrão possibilitavam prever um número maior de PPIs, porém não garantia a eficiência na montagem de perfis filogenéticos (ANJOS et al., 2021).

A indução de similaridades proteicas por meio de aprendizado de máquina é uma técnica inovadora para a predição de redes de PPI. A técnica ofereceu diversos benefícios, incluindo a capacidade de inferir similaridades proteicas de forma mais rápida entre proteínas que não possuam um percentual de identidade elevado mas que são muito semelhantes perante outras técnicas presentes em ferramentas do estado da arte, possibilitando a identificação de interações entre proteínas que seriam anteriormente ignoradas pelo *Genppi*.

O campo do aprendizado de máquina desempenha um papel importante em diversas áreas, colaborando com avanços significativos na análise de dados, reconhecimento de padrões e automatização de tomada de decisões. Dentre as diversas técnicas utilizadas nesse campo, a *Random Forests* (RF) ou Florestas Aleatórias ofereceu vantagens significativas para a abordagem do problema em específico. Em primeiro lugar, *Random Forests* foi reconhecida por sua capacidade de lidar com dados complexos e alta dimensionalidade, características frequentemente encontradas em dados de bioinformática, como os utilizados na predição de redes de interação proteína-proteína. Sua capacidade de lidar com variáveis de entrada não linearmente correlacionadas e de identificar padrões sutis dos dados tornou-a uma escolha significativa para esse tipo de problema (NORVIG et al., 2012; LIAW; WIENER, 2002).

Além disso, *Random Forests* mostrou-se altamente robusta em relação a *overfitting*. Isso foi crucial ao lidar com dados biológicos que poderiam levar a modelos excessivamente ajustados aos dados de treinamento, prejudicando a generalização para novos dados. *Random Forests* demonstrou a capacidade de fornecer uma medida de importância para cada variável de entrada, permitindo a identificação das características mais relevantes para a predição. Destacou-se ainda por sua eficácia, versatilidade e resultados promissores em diversas aplicações, oferecendo uma abordagem que se mostrou adaptável às exigências

do mundo moderno da inovação e avanço tecnológico (IZBICKI; SANTOS, 2020).

A busca por implementações otimizadas desempenhou um papel fundamental no aprimoramento do *Genppi*. Uma abordagem que se mostrou promissora para integrar eficientemente as funcionalidades do *Random Forests* ao *Genppi* foi a utilização da biblioteca *CL-RandomForest*, escrita em *Common Lisp*. Essa escolha facilitou a integração do algoritmo ao código-base do *Genppi*, também desenvolvido em *Common Lisp* e compilado em SBCL. A flexibilidade e o poder expressivo dessa linguagem contribuíram para adaptar a biblioteca às necessidades específicas do software, otimizando sua eficiência e mantendo a consistência do projeto (NORVIG et al., 2012).

A incorporação da biblioteca *CL-RandomForest* ao *Genppi* permitiu explorar características intrínsecas do *Lisp*, como sua natureza funcional e capacidade de manipulação simbólica. Essa integração foi realizada de forma a minimizar a carga computacional e o consumo de memória do algoritmo, mantendo o foco na escalabilidade e viabilidade técnica do software. Além disso, a compatibilidade do *CL-RandomForest* com a infraestrutura existente do *Genppi* possibilitou a implementação de uma solução robusta, que atendeu aos objetivos propostos sem a necessidade de desenvolver uma nova biblioteca do zero.

O ambiente interativo e a compilação dinâmica oferecidos pelo *Common Lisp* foram elementos fundamentais durante a etapa de adaptação e integração. Esses recursos facilitaram o ajuste do comportamento do algoritmo às particularidades do *Genppi*, permitindo redefinir funções e testar melhorias de forma ágil. Como resultado, foi possível incorporar as funcionalidades do *CL-RandomForest* de maneira eficiente, aprimorando o desempenho do *Genppi* em previsões de redes de interação proteína-proteína, especialmente em cenários de alta complexidade e baixa similaridade.

1.1 Motivação

A oxidação anaeróbica do metano (AOM) é um processo crucial para o controle das emissões de metano, um potente gás de efeito estufa, nos ecossistemas marinhos. Esse fenômeno ocorre em ambientes anaeróbicos, como sedimentos oceânicos profundos, onde consórcios microbianos formados por *archaea* anaeróbicas e bactérias redutoras de sulfato trabalham em associação para oxidar o metano. Apesar de sua importância, os mecanismos moleculares subjacentes a esse processo permanecem pouco compreendidos. Para elucidar essas interações, é necessário investigar as redes de interação proteína-proteína que sustentam essa cooperação metabólica entre os micro-organismos envolvidos.

No contexto da AOM, as *archaea* desempenham um papel central devido à sua capacidade de metabolizar metano em condições extremas. Esses micro-organismos unicelulares pertencem a um dos três domínios da vida, juntamente com bactérias e eucariotos, e

possuem características metabólicas e estruturais únicas. (DEPPENMEIER, 2002). Um exemplo marcante é a via metabólica da metanogênese, mediada pela enzima *methyl-coenzyme M reductase* (MCR), essencial para a sobrevivência das *archaea metanogênicas* em ambientes anaeróbicos. A MCR é composta por três subunidades (*alfa*, *beta* e *gamma*), cuja interação é crucial para a eficiência catalítica e o funcionamento dessa via. Essas interações representam um caso exemplar de redes proteína-proteína fundamentais para processos biológicos essenciais (SHIMA; THAUER, 2005).

Apesar da relevância das interações proteicas nesse contexto, o *Genppi*, projetado para prever redes de interação proteína-proteína em genomas bacterianos, apresentou uma limitação importante ao ser testado com dados de *archaea*. Em um experimento conduzido no laboratório da Universidade Federal de Uberlândia, em janeiro de 2022, foram analisadas as três subunidades da MCR (*alpha-MRG77250.1*, *beta-MRG77246.1* e *gamma-MRG77249.1*) de *archaea Methanosarcinales*, cuja interação é amplamente documentada na literatura. No entanto, o *Genppi* não conseguiu identificar essas interações, evidenciando uma lacuna em sua capacidade preditiva. Essa falha revelou a necessidade de aprimorar o *Genppi*, especialmente em cenários onde as interações entre proteínas são mediadas por mecanismos mais complexos ou ocorrem em organismos filogeneticamente distantes dos modelos tradicionais.

As redes de interação proteína-proteína (PPI) representam uma teia intrincada que sustenta os processos fundamentais da biologia celular. Em um nível molecular, essas interações desempenham papéis cruciais na regulação gênica, na sinalização celular e na organização estrutural. No entanto, a identificação dessas interações é uma tarefa desafiadora, frequentemente exigindo métodos experimentais demorados e dispendiosos. A abordagem computacional oferece uma alternativa promissora, permitindo a previsão de PPIs de forma eficiente.

O *Genppi* é um software de bioinformática projetado para prever redes de interação proteína-proteína em genomas bacterianos. Embora tenha alcançado resultados promissores, sua aplicabilidade era limitada pela necessidade de uma alta similaridade proteica entre as proteínas em questão. A incorporação de um algoritmo de aprendizado de máquina, como a *Random Forest* ao código *Genppi*, representaria um avanço para melhorar a quantidade e qualidade das predições do *Genppi*.

A *Random Forest*, com sua capacidade de discernir padrões complexos nas sequências de aminoácidos, tem o potencial de superar as limitações dos métodos tradicionais. Ao utilizar essa abordagem, esperava-se não apenas uma maior precisão na predição de similaridades proteicas, mas também uma expansão na capacidade de identificar interações entre proteínas menos similares. Além disso, *Random Forests* é altamente paralelizável e escalável, o que significa que pode lidar eficientemente com grandes conjuntos de dados,

acelerando o processo de treinamento e permitindo análises rápidas em tempo real, sendo fundamental para a aplicação em bioinformática, onde a análise de grandes volumes de dados é comum.

A relevância dessa pesquisa vai além do âmbito da bioinformática. Ela contribui significativamente para o entendimento dos intrincados processos celulares que dependem das interações proteína-proteína. Além disso, ao desenvolver ferramentas mais precisas e abrangentes para a predição de redes de PPI, pavimentou-se o caminho para avanços inovadores em áreas como medicina, biotecnologia e desenvolvimento de fármacos.

A motivação surgiu também da necessidade de explorar e abordar desafios específicos relacionados à linguagem *Common Lisp*, que demanda uma curva de aprendizado mais longa. A linguagem *Lisp* é reconhecida como uma linguagem promissora e eficiente para a implementação de algoritmos. Diante desse cenário, buscou-se unir esses dois elementos: a complexidade e eficiência do algoritmo de *Random Forests* e a potencialidade da linguagem *Common Lisp*, ambos num contexto de aperfeiçoamento de um software com aplicabilidade diversas.

Portanto, este estudo não apenas responde a uma necessidade premente na pesquisa biológica, mas também abre portas para novas descobertas e aplicações que podem ter um impacto significativo na nossa compreensão da vida celular e no desenvolvimento de terapias e intervenções biomédicas mais eficazes.

1.1.1 Desafios

O *Genppi* tinha o problema de negligenciar várias proteínas que fossem similares porque a lógica do software exigia um percentual superior a 90% de identidade de proteínas. O alinhamento exato e o heurístico eram muito caros dada às quantidades de alinhamentos que o *Genppi* precisaria fazer. Por exemplo, em um pangenoma composto de 20 genomas de diferentes gêneros/espécies, com uma média de 3000 genes, a quantidade de alinhamentos mesmo que heurísticos como o *Blastp* (página 55) necessitariam serem executados $(3000 \times 20)^2$ vezes, ou 3,6 bilhões de execuções. Isso porque o *Genppi* compara todas as proteínas contra todas para decidir que pares devem ser investigados de acordo com as regras que caracterizam interações.

O desafio consistia em contribuir no campo da biologia computacional por meio de um algoritmo de aprendizado de máquina eficiente, utilizando a mesma linguagem de programação do software original, com acurácia superior a 80%, e que, ao mesmo tempo, obtivesse de forma quase instantânea as proteínas com similaridade proteica maior que 65% em um único tempo de acesso linear, aumentando de forma expressiva o número de redes de interação e reduzindo o número de casos negligenciados para um patamar

aceitável, o que, conseqüentemente, demandaria maior capacidade de processamento.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral desta dissertação é melhorar a capacidade de predição de redes de interação proteína-proteína do software *Genppi* por meio da integração de uma biblioteca escrita em *Common Lisp*, utilizando a mesma linguagem de programação para facilitar o desenvolvimento e a manutenção. Essa biblioteca, que emprega o algoritmo de aprendizado de máquina *Random Forest*, permite inferir similaridades proteicas, mesmo quando essas não atingem o limite de identidade exigido no estado da arte, possibilitando, dessa forma, otimizar o desempenho do *Genppi*.

1.2.2 Objetivos Específicos

- ❑ Realizar uma revisão sistemática da literatura, utilizando artigos indexados no *PubMed* e técnicas de clusterização temática, para identificar e analisar softwares e abordagens computacionais voltados à construção de redes de interação proteína-proteína (PPI) a partir de genomas completos e não anotados. A seleção dos artigos foi baseada na análise dos resumos com o objetivo de mapear ferramentas existentes e avaliar sua aplicabilidade no contexto de genomas bacterianos.
- ❑ Implementar uma nova funcionalidade que altera a heurística baseada no histograma de aminoácidos do *Genppi*, incorporando características físico-químicas por meio de um índice de propensão e valores específicos para cada aminoácido em diferentes regiões de uma proteína, com vistas a alcançar uma análise mais granular das propriedades das proteínas.
- ❑ Incorporar a estrutura básica de um algoritmo de aprendizado de máquina (*Random Forests*) ao código do *Genppi*, considerando a criação das árvores de decisão, a seleção de características, a amostragem dos dados e a agregação dos resultados;
- ❑ Avaliar a eficácia do algoritmo *Random Forest* por meio do uso de métricas de desempenho e realização de testes de controle positivo e negativo, estabelecendo critérios de similaridade para o treinamento e validação do classificador em diferentes conjuntos de dados proteicos.
- ❑ Conduzir uma análise para avaliar o impacto no desempenho e na capacidade de predição do *Genppi*, comparando os resultados com e sem a integração da *Random*

Forest, bem como utilizando a ferramenta *STRING*, amplamente reconhecida como o estado da arte na predição de interações proteína-proteína. Essa análise inclui a avaliação da influência em diferentes parâmetros e configurações do algoritmo.

1.3 Hipótese

A inserção da *Random Forest* no software *Genppi*, aliada à integração de métodos de aprendizado de máquina, proporciona uma melhoria significativa na capacidade de predição de redes de interação proteína-proteína em genomas bacterianos. Esta abordagem permite a identificação de interações entre proteínas com níveis de similaridade mais baixos do que os métodos convencionais exigem, ampliando assim a cobertura e a precisão das redes de PPI previstas.

A hipótese acima é investigada por meio de experimentos e análises comparativas, utilizando conjuntos de dados de referência e métricas de desempenho adequadas. O objetivo é validar a incorporação da *Random Forest* no algoritmo *Genppi*. Portanto, as perguntas associadas às hipóteses deste trabalho, são:

- ❑ Em que medida a inserção da *Random Forest* no software *Genppi* influencia a capacidade de predição de redes de interação proteína-proteína em comparação com o método tradicional?
- ❑ Qual é o impacto da integração de métodos de aprendizado de máquina na precisão e na cobertura das redes de PPI previstas pelo *Genppi*?
- ❑ Quais são as principais diferenças nos resultados obtidos pelo *Genppi* com e sem a inserção da *Random Forest*, em termos de sensibilidade, especificidade e outras métricas de desempenho?
- ❑ Em que medida a melhoria na capacidade de predição de redes de PPI contribui para um melhor entendimento dos processos celulares bacterianos que dependem de interações proteína-proteína?
- ❑ Quais são as possíveis aplicações práticas da melhoria na predição de redes de PPI em pesquisas biomédicas e no desenvolvimento de terapias direcionadas a patógenos bacterianos?
- ❑ Como a inserção da *Random Forest* no *Genppi* pode ser adaptada e aplicada a outras áreas da biologia e da medicina que dependem da análise de interações proteína-proteína, além da microbiologia?

1.4 Contribuições

1.4.1 Científicas

- ❑ Validação da eficácia do algoritmo de *Random Forest*.
- ❑ Ampliação do conhecimento sobre redes de PPI: Ao ampliar a capacidade de predição do *Genppi*, esta dissertação contribui para um melhor entendimento dos complexos processos celulares que dependem das interações entre proteínas. Isso pode levar a *insights* importantes sobre a regulação gênica, a sinalização celular e outros aspectos da biologia celular bacteriana.
- ❑ Aprimoramento da eficiência de métodos computacionais: Além disso, a incorporação da *Random Forest* no *Genppi* pode abrir caminho para o desenvolvimento de abordagens computacionais mais precisas para a predição de PPIs em outros organismos e contextos biológicos. Isso pode beneficiar não apenas a pesquisa em microbiologia, mas também outras áreas da biologia e da medicina que dependem da análise de interações proteína-proteína.
- ❑ Expansão do conhecimento sobre as potencialidades da linguagem *Common Lisp* na área de *machine learning*, proporcionando novas perspectivas para o desenvolvimento de algoritmos e técnicas;

1.4.2 Práticas

- ❑ Disponibilização do código desenvolvido como software de código aberto e livre, permitindo que outros pesquisadores e desenvolvedores acessem e implementem melhorias contínuas. O código está disponível em <<https://genppi.facom.ufu.br/>>.
- ❑ Desenvolvimento de ferramentas avançadas: contribuição para o avanço da bioinformática, oferecendo uma ferramenta mais abrangente para a predição de redes de interação proteína-proteína. A integração de métodos de aprendizado de máquina como a *Random Forest* no software *Genppi* representa uma busca por soluções mais eficazes para os desafios enfrentados na identificação de PPIs.
- ❑ Potencial aplicação em pesquisas e aplicações biomédicas: A melhoria na predição de redes de PPI pode ter importantes implicações para pesquisas biomédicas e aplicações práticas, como o desenvolvimento de terapias direcionadas a patógenos bacterianos. Uma compreensão mais abrangente das interações proteína-proteína bacterianas pode levar ao desenvolvimento de novos tratamentos e intervenções mais eficazes contra doenças bacterianas.

1.5 Organização da Dissertação

Esta dissertação foi organizada em seis capítulos para apresentar o estudo de maneira clara e compreensível. Cada capítulo aborda os conceitos e métodos empregados, os experimentos realizados e os resultados obtidos.

O Capítulo 1 apresenta as considerações iniciais e o contexto em que está inserido a pesquisa, motivação para seu desenvolvimento e objetivos a atingir. Além disso, é apresentada a hipótese que norteia o estudo e as contribuições esperadas.

O Capítulo 2 aborda os conceitos fundamentais necessários para o entendimento do estudo. São discutidos tópicos sobre biotecnologia, genoma, pangenoma e proteoma, além de princípios de bioinformática e análise de sequências de proteínas. Apresenta ainda uma base teórica sobre *Random Forest* e *Common Lisp*.

O Capítulo 3 apresenta o estado da arte em bioinformática, destacando os principais conceitos, terminologias e técnicas empregadas atualmente. É fornecida uma visão geral das ferramentas e métodos existentes para a análise e predição de redes de interação proteína-proteína, incluindo as principais bases de dados e algoritmos utilizados. Essa apresentação fornece uma base necessária para a compreensão mais clara do contexto em que se insere o *Genppi* e dos trabalhos relacionados.

O Capítulo 4 detalha os procedimentos metodológicos empregados no estudo. Abrange a preparação da base de dados, a incorporação do algoritmo de *Random Forest* em *Common Lisp* e a avaliação do desempenho do modelo.

O Capítulo 5 apresenta os experimentos realizados com diferentes combinações de características extraídas e a avaliação do desempenho do modelo de classificação. Além disso, é apresentada uma análise comparativa dos resultados e a avaliação em diferentes genomas.

O Capítulo 6 oferece uma conclusão, explora a aplicação dos métodos propostos, com ênfase em suas contribuições e limitações dentro do contexto da predição de redes de interação proteína-proteína.

Revisão de Literatura e Fundamentos Teóricos

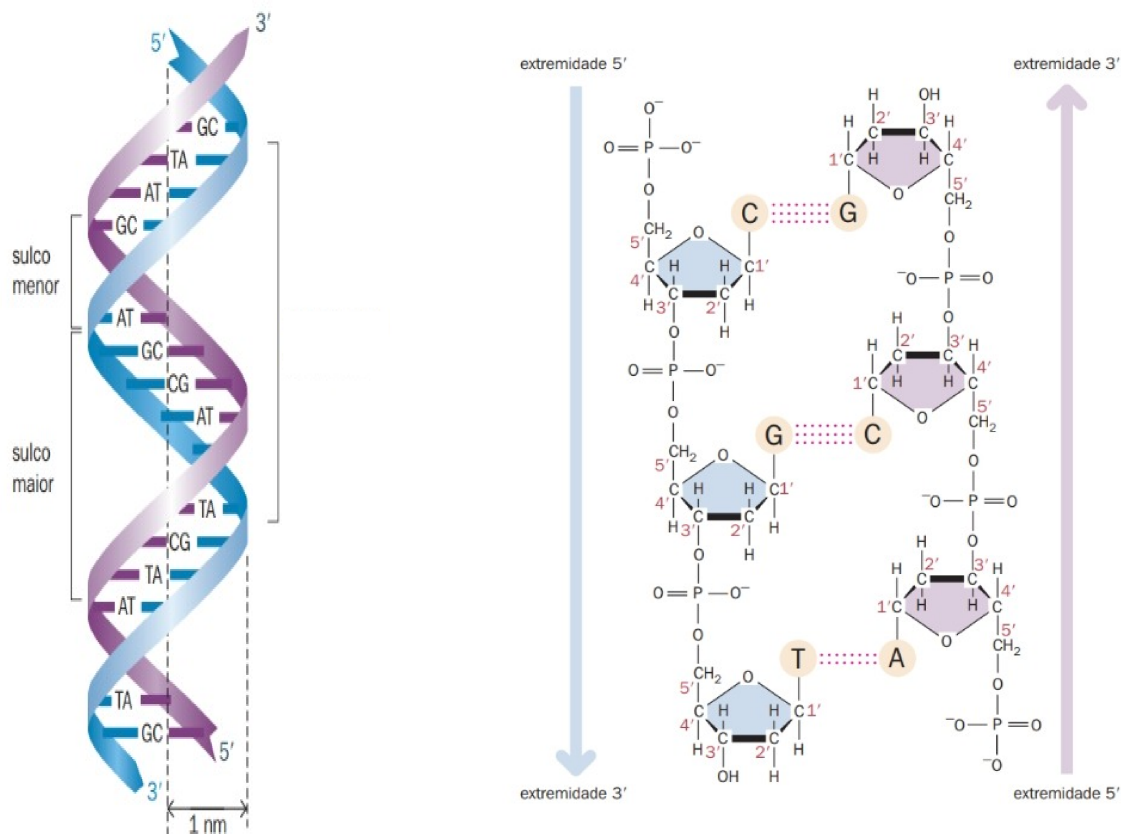
2.1 Biotecnologia

As moléculas são organizadas em estruturas complexas. As células são responsáveis por todas as funções vitais dos seres vivos, incluindo a reprodução, o crescimento e o metabolismo. Existem diferentes tipos de moléculas, mas elas podem ser classificadas em duas categorias principais: moléculas orgânicas e moléculas inorgânicas. (ATKINS, 2003).

Dentre as moléculas orgânicas, o ácido desoxirribonucleico (DNA), da categoria dos ácidos nucleicos, é considerada uma molécula essencial para a vida e também responsável pela síntese do ácido ribonucleico (RNA) - intermediadora capaz de expressar as informações presentes no DNA, obtida por um processo denominado transcrição. O DNA é composto por duas cadeias de nucleotídeos que se entrelaçam formando uma dupla hélice. Cada nucleotídeo é composto por um açúcar de cinco carbonos, um grupo fosfato e uma base nitrogenada. A estrutura do DNA é altamente regular, com cada volta da hélice medindo aproximadamente 10 pares de bases. A estabilidade da dupla hélice é garantida por ligações de hidrogênio entre as bases e pela empilhamento das bases. A direcionalidade das cadeias é determinada pela numeração dos carbonos, com a cadeia superior indo do carbono 5 para o carbono 3 e a cadeia inferior indo do carbono 3 para o carbono 5, conforme é demonstrado na figura 1. Tais características estruturais são fundamentais para a função do DNA (WATSON; BERRY, 2009).

O DNA é uma molécula polimérica que armazena as informações genéticas. Ele é formado por duas fitas, uma de sentido direto e outra de sentido inverso. As duas fitas são complementares, o que significa que as bases nitrogenadas de uma fita se ligam às bases complementares da outra fita. A sequência de bases do DNA determina a sequência

de aminoácidos das proteínas. O DNA é dividido em genes, que são as unidades básicas da informação genética. O DNA de um organismo é responsável por todas as suas características, incluindo sua aparência, comportamento e fisiologia (BROWN, 2000).



(a) Características: As duas fitas de DNA se enrolam uma na outra, produzindo um sulco menor e um sulco maior na dupla hélice.

(b) Natureza antiparalela: As duas fitas antiparalelas correm em direções opostas ligando átomos de carbono 3 dos resíduos de açúcar, a átomos de carbono 5.

Figura 1 – Dupla-Hélice de DNA: Características e Natureza antiparalela - Adaptado de (STRACHAN; READ, 2016).

A transcrição do DNA para o RNA é responsável pela síntese de proteínas. Esse processo ocorre no núcleo das células e é realizado pela RNA-polimerase, uma enzima que se liga ao DNA e copia sua sequência nucleotídica. As bases nitrogenadas do DNA se ligam às bases nitrogenadas complementares do RNA, formando pares de base. Adenina (A) se liga a timina (T), citosina (C) se liga a guanina (G) e, por fim, uracila (U) se liga a adenina. A sequência de bases do RNA é complementar à sequência de bases do DNA. Essa sequência de bases determina a sequência de aminoácidos da proteína que será sintetizada (STRACHAN; READ, 2016).

As proteínas exercem uma diversidade de papéis, capazes de assumir formas e funções infinitamente complexas, variando desde o catalismo das reações químicas, o transporte de nutrientes até a construção de estruturas para defender o corpo contra patógenos. Apesar dessa variedade, todas as proteínas são construídas a partir de um conjunto de apenas vinte blocos chamados aminoácidos. A sequência e o arranjo desses aminoácidos determinam a forma final e, conseqüentemente, a função da proteína (VERLENGIA et al., 2013).

2.1.1 Genoma, Pangenoma e Proteoma

Genoma, pangenoma e proteoma representam conjuntos completos de informação genética e molecular. Enquanto o **genoma** abrange todas as informações genéticas de um organismo individual, incluindo todos os genes e sequências não codificantes de DNA, o **pangenoma** considera a variabilidade genética entre diferentes cepas de uma mesma espécie, incluindo os genes comuns a todas as cepas (*core genome*) e os genes variáveis entre as cepas (*accessory genome*) e revela a diversidade genética e adaptativa de uma população. Ao contrário do genoma, que é estático para um organismo individual e composto por cromossomos lineares de DNA, o pangenoma leva em consideração a variabilidade genética entre diferentes cepas de uma mesma espécie, incluindo genes que conferem vantagens adaptativas específicas a certas cepas, como resistência a antibióticos ou capacidade de colonizar diferentes ambientes (HARISHA, 2013).

Por sua vez, o **proteoma**, relativas às proteínas expressas por um organismo, tecido ou célula em um determinado momento, reflete as interações complexas entre o genoma, o ambiente e os processos celulares, sendo crucial para compreender a fisiologia e a função biológica de um sistema. O estudo do genoma fornece informações importantes sobre a evolução, fisiologia e relação com outros organismos, enquanto a análise do pangenoma ajuda a entender a variabilidade genética e as adaptações evolutivas que ocorreram ao longo do tempo. Já o estudo do proteoma revela informações sobre as vias metabólicas, interações proteína-proteína, modificações pós-traducionais e expressão diferencial de proteínas em diferentes condições fisiológicas ou patológicas, sendo fundamental para a descoberta de biomarcadores e alvos terapêuticos em medicina e biotecnologia (HARISHA, 2013).

2.1.2 Técnicas da Bioinformática e suas aplicações

Na bioinformática, os dados biológicos são analisados e interpretados usando uma variedade de técnicas e ferramentas computacionais. Uma das principais áreas de atuação é a análise de sequências de DNA, RNA e proteínas, onde são aplicados algoritmos e

métodos de análise para extrair informações úteis dos dados genômicos e proteômicos (LESK, 2019).

Uma técnica comum na bioinformática é o alinhamento de sequências, onde as sequências são comparadas para identificar similaridades e diferenças. Isso pode ser feito usando algoritmos para alinhar sequências de DNA, RNA ou proteínas e identificar regiões conservadas ou homólogas. A predição de genes e elementos regulatórios no genoma também é outra técnica muito utilizada, que envolve a identificação de regiões codificadoras de proteínas, sítios de ligação de fatores de transcrição e outras características genômicas importantes. Algoritmos de predição de genes, como o *GeneMark* e o *Augustus*, são empregados para identificar genes em sequências genômicas (GAUTHAM, 2006).

Além disso, a análise de expressão gênica é uma área importante da bioinformática, que envolve a quantificação e comparação dos níveis de expressão de genes em diferentes condições experimentais. Isso pode ser feito usando técnicas de *RNA-Seq*, que permitem a quantificação de transcritos de RNA em larga escala, ou microarranjos de DNA, que permitem a análise simultânea da expressão de milhares de genes. Outras técnicas comuns incluem a modelagem de estruturas de proteínas, a predição de interações proteína-proteína e proteína-ligante, a análise de redes biológicas e a inferência de vias metabólicas. Essas técnicas são aplicadas em uma variedade de contextos, desde a pesquisa básica em biologia molecular até o desenvolvimento de novas terapias e tratamentos médicos (POLANSKI; KIMMEL, 2007).

Uma técnica muito importante para o entendimento de como a informação genética é traduzida e transcrita em proteínas funcionais é conhecida como “degenerescência do código genético” ou “degeneração codificada”. No código genético, uma sequência de três nucleotídeos (também chamada de *triple* ou *códon*) corresponde a um aminoácido específico. No entanto, devido à degenerescência do código genético, mais de um *códon* pode codificar o mesmo aminoácido. Isso ocorre porque há 64 possíveis combinações de três nucleotídeos (4 bases elevadas à 3ª potência), mas apenas 20 aminoácidos comuns utilizados na síntese de proteínas (CRICK, 1968).

Como resultado, alguns aminoácidos são codificados por mais de um *códon*. Por exemplo, o aminoácido leucina é codificado por seis *códons* diferentes: UUA, UUG, CUU, CUC, CUA e CUG. Da mesma forma, a serina é codificada por seis *códons*: UCU, UCC, UCA, UCG, AGU e AGC. Essa redundância no código genético é vantajosa porque minimiza o impacto de mutações pontuais nas sequências de DNA, já que uma mudança em um nucleotídeo pode não afetar necessariamente o aminoácido codificado. A Figura 2 é chamada de “tabela de codificação genética” ou “tabela de códons”. Ela lista todas as possíveis combinações de três nucleotídeos e os aminoácidos correspondentes (CRICK, 1968).

Existem outras características e sinais importantes nas sequências genômicas que influenciam a tradução do DNA em proteínas. Alguns desses sinais incluem: Sequências de início e término, regiões promotoras e terminadoras, sinais de processamento pós-transcricional, sequências de ligação de ribossomos. Destaca-se as sequências de início e término. As proteínas geralmente começam com um *códon* de início (geralmente AUG, que codifica a metionina) e terminam com um *códon* de parada (também chamado de *códon* de terminação ou codão de parada), como UAA, UAG ou UGA. Esses sinais indicam o início e o fim da sequência de aminoácidos que será traduzida em uma proteína funcional. Essas características e sinais são fundamentais para o processo de expressão gênica e é crucial para projetar e manipular sequências genômicas para diversos fins (LESK, 2010).

		segunda posição					
		U	C	A	G		
primeira posição	U	UUU Phe UUC UUA Leu UUG	UCU UCC Ser UCA UCG	UAU Tyr UAC UAA término UAG término	UGU Cys UGC UGA término UGG Trp	U C A G	terceira posição
	C	CUU Leu CUC CUA CUG	CCU CCC Pro CCA CCG	CAU His CAC CAA Gln CAG	CGU CGC Arg CGA CGG	U C A G	
	A	AUU AUC Ile AUA AUG Met	ACU ACC Thr ACA ACG	AAU Asn AAC AAA Lys AAG	AGU Ser AGC AGA Arg AGG	U C A G	
	G	GUU Val GUC GUA GUG	GCU GCC Ala GCA GCG	GAU Asp GAC GAA Glu GAG	GGU GGC Gly GGA GGG	U C A G	

Figura 2 – Tabela de *Códons* - Adaptado de (WATSON et al., 2015).

Os aminoácidos que são moléculas compostas por um grupo amino, um grupo carboxila e uma cadeia lateral são ligados entre si por ligações peptídicas, que são ligações covalentes entre o grupo carboxila de um aminoácido e o grupo amino de outro. A sequência de aminoácidos formam uma proteína e é determinada pela sequência de bases nitrogenadas do DNA. O tamanho médio de uma proteína é de cerca de trezentos e quarenta e um aminoácidos. No entanto, o tamanho das proteínas pode variar muito, desde proteínas

pequenas, como a insulina, que tem apenas 51 aminoácidos, até proteínas gigantes, como a titina, que tem mais de 34.000 aminoácidos (LESK, 2010).

O tamanho de uma proteína é determinado por sua sequência de aminoácidos. Cada aminoácido tem um tamanho diferente, e a sequência de aminoácidos determina a estrutura tridimensional da proteína. A estrutura tridimensional de uma proteína é importante para sua função. Pode-se trabalhar com proteínas de várias maneiras, seja com técnicas de bioinformática para estudar a estrutura e a função das proteínas ou com técnicas de biologia molecular para manipular as proteínas, por exemplo, para modificá-las ou para criar novas proteínas (STRYER, 1981).

No entanto, a representação das proteínas comumente é feita por letras que representam os *códons*. Isso geralmente é feito para simplificar a representação de sequências de aminoácidos. Cada aminoácido é representado por uma letra única, geralmente a primeira letra do seu nome em inglês (por exemplo, A para alanina, C para cisteína, etc.). Essa representação simplificada é útil em análises de sequências, modelagem estrutural e outras aplicações bioinformáticas (GIULIO, 1997).

2.1.3 Estrutura e Sequenciamento do DNA

As unidades fundamentais do genoma, os genes, são sequências específicas de DNA que contêm instruções para a síntese de proteínas. As proteínas, por sua vez, desempenham uma ampla gama de funções essenciais para o funcionamento adequado das células, desde a estruturação até a regulação de processos bioquímicos. O DNA atua como um modelo, fornecendo as informações necessárias para a produção de proteínas através do processo de transcrição e tradução. Para uma compreensão mais profunda do funcionamento celular, é crucial considerar as escalas envolvidas. Cada célula é um microcosmo complexo, com milhões de moléculas operando em perfeita sincronia para realizar as funções vitais do organismo. No âmbito molecular, diversas forças físicas e químicas interagem e se equilibram, influenciando a estrutura e a função das moléculas biológicas. A compreensão abrangente dessas forças, como as pontes de hidrogênio, é fundamental para desvendar os mecanismos moleculares que sustentam a vida (UZMAN, 2003).

Quando se estuda uma doença, a primeira etapa é identificar quais são as proteínas envolvidas no processo. Isso pode ser um desafio, pois o corpo humano é composto por trilhões de proteínas, e cada uma delas pode desempenhar um papel diferente. O sequenciamento, a análise do DNA complementar e a cristalografia de raios X são técnicas que se complementam utilizadas para estudar a estrutura do DNA. As técnicas fornecem informações sobre a sequência de bases do DNA (LODISH, 2008).

O sequenciamento permite analisar a função das proteínas avaliando sua interação

com outras moléculas, como outras proteínas, carboidratos, lipídios e ácidos nucleicos. Possibilita também identificar as proteínas que interagem com uma proteína específica, o que pode fornecer informações importantes sobre a função dessa proteína. O sequenciamento permite determinar a sequência de bases nitrogenadas em uma molécula de DNA. Existem vários métodos de sequenciamento, mas todos eles têm como objetivo determinar a ordem das bases nitrogenadas na fita de DNA. Depois que a sequência de bases nitrogenadas de um gene é determinada, usa-se essa informação para identificar a proteína que o gene codifica. Isso é feito comparando a sequência de bases nitrogenadas do gene com a sequência de aminoácidos da proteína (POLANSKI; KIMMEL, 2007).

A análise do DNA complementar é uma ferramenta que pode ser utilizada para entender como as proteínas funcionam e desenvolver novas terapias. Ela permite identificar os genes que codificam as proteínas. A cristalografia de raios X é uma técnica que utiliza a difração de raios X para determinar a estrutura tridimensional de moléculas (LODISH, 2008).

Enquanto o sequenciamento fornece informações sobre a sequência de bases do DNA em uma única dimensão, a análise do DNA complementar é uma técnica bidimensional, mais simples e rápida, baseada na propriedade das bases nitrogenadas do DNA de se complementarem. A adenina se liga à timina e a citosina se liga à guanina. Essa propriedade permite que a sequência de bases de uma fita de DNA seja determinada a partir da sequência de bases da outra fita. Depois que a estrutura de uma proteína é determinada, essa informação é utilizada para entender como a proteína funciona (WATSON; CRICK, 1953).

2.2 Redes de Interação Proteína - Proteína

As redes de interação proteína-proteína (PPI) representam uma área fundamental da biologia computacional e da bioinformática, proporcionando *insights* essenciais sobre os complexos mecanismos moleculares subjacentes aos processos celulares. Essas redes capturam as interações físicas e funcionais entre proteínas, desempenhando papéis cruciais na regulação gênica, sinalização celular e organização intracelular. No estado da arte, abordagens computacionais avançadas têm sido desenvolvidas para prever e analisar PPIs em diferentes organismos, incluindo bactérias, levando a descobertas significativas na compreensão da biologia molecular (ANJOS, 2021).

A análise de PPIs geralmente se baseia em evidências biológicas encontradas nos genomas, incluindo a conservação da vizinhança gênica, fusão gênica e perfil filogenético conservado. A vizinhança gênica conservada refere-se à observação de que genes que estão próximos uns dos outros em diferentes genomas tendem a estar envolvidos em interações físicas ou funcionais. Por sua vez, a fusão gênica ocorre quando dois genes individuais

se fundem para formar uma sequência única em um organismo distinto, indicando uma possível interação entre as proteínas codificadas por esses genes. Além disso, o perfil filogenético conservado examina a presença e ausência de genes em genomas de referência, identificando padrões evolutivos que sugerem interações entre proteínas (WALHOUT; VIDAL, 2001).

A disposição dos genes em um genoma também desempenha um papel crucial na análise e predição de redes de interação proteína-proteína, principalmente em genomas com informações limitadas ou não anotados. Isso ocorre porque a proximidade genômica entre os genes pode indicar uma relação funcional entre as proteínas codificadas, sugerindo que essas proteínas possuam uma interação direta ou estejam envolvidas em processos biológicos relacionados. O uso dessas informações para inferir interações potenciais baseia-se em padrões observados em genomas bacterianos e outros organismos onde genes organizados em operons ou clusters frequentemente codificam proteínas que participam de vias metabólicas ou funcionais conjuntas.

Essas abordagens baseadas em evidências biológicas permitem a predição e análise de PPIs de maneira eficiente e precisa, contribuindo significativamente para o entendimento dos sistemas biológicos. Com o desenvolvimento de ferramentas computacionais avançadas, como o *Genppi*, torna-se possível explorar essas interações em larga escala, permitindo a identificação de novos alvos terapêuticos, o desenvolvimento de drogas e vacinas e a compreensão mais profunda dos mecanismos moleculares subjacentes a doenças e processos biológicos. Assim, as redes de interação proteína-proteína continuam a desempenhar um papel crucial na vanguarda da pesquisa biomédica e bioinformática, impulsionando avanços significativos na medicina e biotecnologia (GAVIN et al., 2002).

2.2.1 Algoritmos de Análises de PPI

Existem várias abordagens para prever e analisar interações proteína-proteína. (ALOY; RUSSELL, 2002; BARABASI; OLTVAI, 2004). Alguns dos algoritmos e métodos mais comuns incluem:

- ❑ **Docking molecular:** Utiliza algoritmos de simulação para prever a estrutura tridimensional das interações entre proteínas.
- ❑ **Modelagem de interação proteína-proteína:** Usa modelos computacionais para prever interações com base em estruturas conhecidas de proteínas.
- ❑ **Análise de redes de interação proteína-proteína (PIN):** Utiliza métodos de análise de redes complexas para identificar e analisar padrões de interação entre proteínas em uma escala sistêmica.

2.3 Random Forest

2.3.1 Árvores de Decisão

As árvores de decisão são estruturas essenciais das florestas aleatórias. Constituem o núcleo central do algoritmo *Random Forest* e são versáteis no aprendizado de máquina, executando tarefas de classificação, regressão e, até mesmo, tarefa *multioutput* (problemas com múltiplas saídas). Uma das muitas qualidades das árvores de decisão é que elas exigem pouca preparação de dados. Em particular, elas não exigem o escalonamento ou a centralização das características. Os algoritmos de árvore de decisão mais conhecidos são o ID3, o C4.5 e o CART (GÉRON, 2021).

A árvore de decisão é um sistema baseado em regras, fundamentado numa perspectiva *botton-up* e a partir de dados pré-existentes e não, a partir de um especialista. Nas árvores de decisão, o tempo todo se busca alcançar um número mínimo de decisões de modo que a árvore seja a menos profunda possível e pra isso precisa-se tomar as decisões mais relevantes em cada nó, o mais discriminante, no caso de classificação ou que mais reduzem erro quadrático médio, no caso de regressão. A partir do nó raiz, se testa um atributo e uma divisão binária ocorre sucessivamente até que todo o sistema de regras seja coberto até o nó folha. A árvore é uma estrutura hierárquica que representa um conjunto de decisões. Cada nó interno da árvore representa uma decisão com base em um atributo específico, enquanto as folhas representam as diferentes classes ou valores de saída (HO, 1995).

Um sistema com um número x de atributos, tal que $x > y$, uma árvore possível pode chegar a conclusão de que apenas “ y ” atributos sejam suficientes para definir o conjunto de regras. Os atributos não escolhidos não são relevantes para o sistema de regras ou são redundantes aos atributos selecionados. Todavia os “ x ” atributos concorrerão ao nó raiz e será escolhido apenas um atributo que tiver maior ganho de informação (que mais purifica ou discrimina). No caso da classificação, diz-se que a entropia do nó raiz é maior que a entropia dos nós filhos (KUHN; JOHNSON, 2013).

Numa árvore de decisão, os tipos de atributos determinam como a árvore será criada. Atributos podem ser contínuos, categóricos (ordinais ou não ordinais) ou até mesmo binários. Enquanto atributos contínuos apresentam valores numéricos, como altura ou peso, atributos categóricos só podem assumir valores de um conjunto limitado, como tipo de curso ou cor de vestido. Já os atributos binários têm apenas duas opções, como verdadeiro ou falso. O *Random Forest* é capaz de lidar com todos esses tipos de atributos de forma eficiente (MÜLLER; GUIDO, 2016).

Para lidar com todos esses tipos de atributos, em alguns casos é recomendado fazer ajustes dividindo atributos com imensa quantidade de possibilidades em classes menores,

para garantir um melhor desempenho do modelo. Isso pode ser especialmente importante em casos de atributos categóricos não ordinais, onde a quantidade de possibilidades pode ser extremamente grande e difusa, tornando difícil para o modelo aprender de maneira eficiente. Dividir esses atributos em classes menores pode ajudar a simplificar o modelo e melhorar sua precisão. Para atributos contínuos, o ajuste para um número menor de classes (preferencialmente, binária) com delimitadores de maior e menor associado a pontos de corte ou limiares (“threshold”), pode ser vantajoso para evitar “*overfitting*” (HO, 1995).

As vantagens associadas às árvores de decisão, e consequentemente à *Random Forest*, são diversas. Primeiramente, esses modelos de aprendizado são não paramétricos, o que significa que não impõem condições a priori sobre os dados. Além disso, as árvores de decisão têm a capacidade de manipular dados heterogêneos, sejam eles numéricos ou categóricos. A implementação intrínseca da seleção de atributos permite filtrar atributos irrelevantes e/ou ruidosos, contribuindo para a obtenção de predições mais precisas. Além disso, as árvores de decisão são conhecidas por sua robustez em relação a “*outliers*” e pela facilidade de interpretação, permitindo a conversão direta da sequência de decisões em linguagem natural (GÉRON, 2021).

No entanto, é importante destacar que a obtenção da topologia ótima de uma árvore de decisão é um problema computacionalmente intratável. Portanto, o objetivo de utilizar heurísticas eficientes que gerem árvores quase-ótimas, otimizando os recursos computacionais disponíveis é essencial. Em contextos de classificação de padrões, essas heurísticas minimizam a impureza (no caso de classificação) ou o erro quadrático médio (no caso de regressão) nos nós-filhos, garantindo a qualidade das decisões tomadas pela *Random Forest* (KUHN; JOHNSON, 2013).

Schonlau e Zou (2020) destaca no trecho citado adiante que uma desvantagem das árvores de decisão é a propensão a superajuste, o que levou às ideias de considerar múltiplas árvores.

[...] as árvores de decisão são propensas a superajuste, o que significa que o modelo segue as idiossincrasias do conjunto de dados de teste muito de perto e tem um desempenho ruim em um novo conjunto de dados, ou seja, os dados de teste. O superajuste das árvores de decisão levará a uma baixa precisão preditiva geral, também chamada de precisão de generalização. Uma maneira de aumentar a precisão da generalização é considerar apenas um subconjunto das observações e construir muitas árvores individuais. Introduzida pela primeira vez por Ho (1995), essa ideia do método do subespaço aleatório foi posteriormente estendida e formalmente apresentada como a floresta aleatória por Breiman (2001) (SCHONLAU; ZOU, 2020).

2.3.2 Definição e características

Random Forest se baseia na construção de um conjunto de árvores de decisão. Cada árvore é formada por meio da síntese de um vetor aleatório, amostrado independentemente e seguindo a mesma distribuição de probabilidade. O treinamento em um subconjunto aleatório dos dados de treinamento fornece uma classificação ou previsão individual. Esse vetor aleatório guia o processo de síntese da árvore. A saída final da *Random Forest* é obtida por meio de uma combinação ponderada das saídas de cada árvore, resultando em uma decisão conjunta mais robusta e precisa (BREIMAN, 2001).

Definição - Uma floresta aleatória é um classificador que consiste em uma coleção de classificadores estruturados em árvores $h(x, \Theta_k)$, $k = 1, \dots$ onde o Θ_k são vetores aleatórios identicamente distribuídos independentes e cada árvore lança um voto unitário para a classe mais popular da entrada X (BREIMAN, 2001).

As árvores de decisão criadas (Figura 3) são independentes e são construídas em amostras “*bootstrap*” em vez da amostra original. Isso é chamado de agregação de “*bootstrap*” ou simplesmente “*bagging*”, e, dessa forma, reduz o “*overfitting*”. Isso significa que o *Random Forest* é resistente a “*overfitting*” e tende a generalizar bem para novos dados, mesmo em casos em que o conjunto de treinamento contenha ruídos ou “*outliers*”. Isso se deve à diversidade das árvores no modelo, que ajudam a evitar a superespecialização em padrões específicos dos dados de treinamento (SCHONLAU; ZOU, 2020).

Usando amostras aleatórias (bootstrapped) e considerando para cada etapa, apenas um subconjunto de variáveis, igualmente aleatórias, cria-se uma grande variedade de árvores.

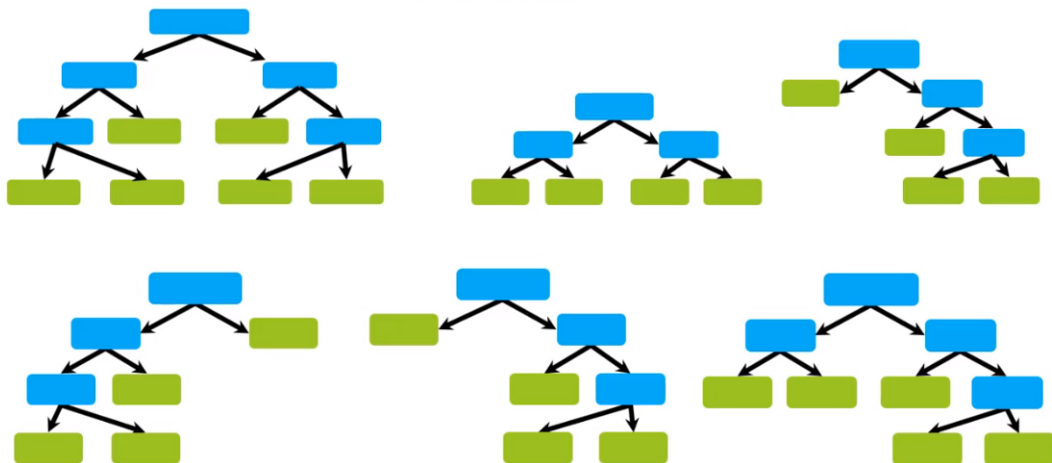


Figura 3 – Modelo de árvores de decisão - Adaptado de (STARMER, 2023).

“*Bootstrap*” é uma técnica estatística que envolve amostrar repetidamente a partir de um conjunto de dados original para criar várias amostras de dados, cada uma com

o mesmo tamanho do conjunto original. Essas amostras de “*bootstrap*” são usadas para estimar a distribuição de um determinado estimador estatístico ou para realizar inferências estatísticas. A tradução literal de “*bagging*” para o português seria “*ensacamento*” ou “*empacotamento*”. O termo “*bagging*” foi criado a partir da junção da palavra “*bootstrap*” com “*aggregating*” (agregação). Ele recebe esse nome porque a técnica de “*bagging*” envolve a criação de múltiplas amostras de “*bootstrap*” e a agregação dos resultados de cada amostra para obter uma estimativa final mais robusta ou uma classificação mais precisa. A ideia é semelhante a colocar várias amostras em uma sacola (*bag*) e, em seguida, combinar os resultados dessas amostras para obter um resultado geral mais confiável (LIAW; WIENER, 2002).

Na Figura 4 temos um exemplo de como é selecionado as amostras dos dados do *dataset* original de um conjunto de dados contendo 04 atributos e 04 instâncias. Verifica-se que no processo de “*bootstrap*” foi selecionada uma mesma amostra para o conjunto “*bootstrapped dataset*”. Por sua vez, na Figura 5 encontra-se a amostra não selecionada também denominada “*Out of Bag*”.

Dataset Original					Bootstrapped Dataset				
Dor no Peito	Boa Circ. Sanguínea	Bloq. Arterial	Peso	Doença Cardíaca	Dor no Peito	Boa Circ. Sanguínea	Bloq. Arterial	Peso	Doença Cardíaca
Não	Não	Não	125	Não	Sim	Sim	Sim	180	Sim
Sim	Sim	Sim	180	Sim	Não	Não	Não	125	Não
Sim	Sim	No	210	Não	Sim	Não	Sim	167	Sim
Sim	Não	Sim	167	Sim	Sim	Não	Sim	167	Sim

Figura 4 – Modelo de “*bootstrap*” - Adaptado de (STARMER, 2023).

No *Random Forest*, em vez de usar todos os recursos disponíveis para criar uma árvore de classificação ou regressão em cada amostra de “*bootstrap*”, a técnica de florestas aleatórias seleciona aleatoriamente um subconjunto dos recursos (chamado de M_try) e usa apenas esses recursos para criar cada árvore. Isso é feito em cada nó da árvore, onde uma amostra aleatória de M_try recursos é considerada para encontrar a melhor divisão. M_try refere-se ao número de preditores (recursos) que são selecionados aleatoriamente em cada nó da árvore de floresta aleatória. O valor de M_try é definido com o objetivo de controlar a aleatoriedade e diversidade das árvores no modelo de floresta aleatória. O valor de M_try é geralmente definido como uma fração do número total de preditores

Dataset Original

Dor no Peito	Boa Circ. Sanguínea	Bloq. Arterial	Peso	Doença Cardíaca
Não	Não	Não	125	Não
Sim	Sim	Sim	180	Sim
Sim	Sim	No	210	Não
Sim	Não	Sim	167	Sim

Aqui está a amostra que não foi utilizada no "Bootstrapped dataset"

↓

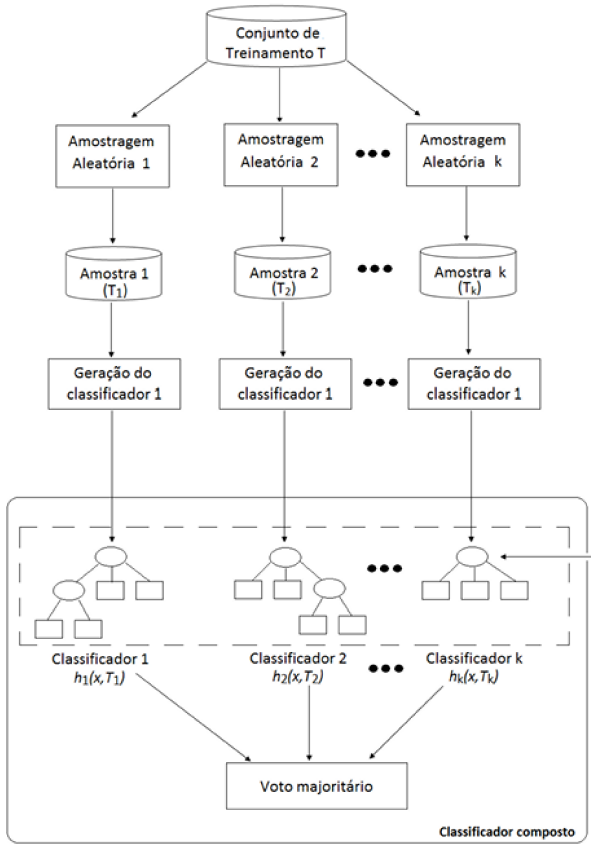
Dor no Peito	Boa Circ. Sanguínea	Bloq. Arterial	Peso	Doença Cardíaca
Sim	Sim	Não	210	No

Figura 5 – Exemplo de Bag of out - Adaptado de (STARMER, 2023).

disponíveis. A escolha do valor de M_try depende do problema e pode variar em diferentes implementações do algoritmo de floresta aleatória (LIAW; WIENER, 2002).

No caso do “*bagging*” (Figura 6), proposto por Breiman em 1996, todas as características (preditores) disponíveis são usadas para construir cada árvore de decisão individual. Cada árvore é treinada em um conjunto de dados de “*bootstrap*”, que é uma amostra aleatória com substituição do conjunto de dados original. Portanto, todas as características são consideradas em cada nó da árvore durante o processo de divisão. O “*bagging*” é uma abordagem de conjunto em que várias árvores são construídas de forma independente e suas previsões são combinadas através de votação ou média para formar a previsão final do modelo. “*bagging*” pode ser pensado como o caso especial de florestas aleatórias obtidas quando $M_try = p$, o número de preditores. Já o *Random Forest* também proposto por Breiman em 2001, é na verdade, uma extensão do “*bagging*” (LIAW; WIENER, 2002).

O *Random Forest*, para promover a diversidade, aborda duas estratégias principais elencadas adiante. Diante disso, ainda que uma árvore qualquer não selecione os melhores dados nem os melhores atributos, é provável que o algoritmo terá um desempenho satisfatório, pois, o mesmo não ocorrerá para o restante das $n-1$ árvores que fazem parte do treinamento. As árvores restantes compensam as possíveis falhas das árvores individuais. Nisso, perde-se na interpretabilidade das árvores de decisão devido ao grande número de árvores construídas, mas ganha-se muito em performance (GÉRON, 2021).



Seja p o número de preditores (atributos) e m , o número de atributos selecionados, na seleção aleatória do algoritmo.

□ se $p = m$, diz-se tratar de “*Bagging*”.

□ se $p > m$, trata-se de *Random Forest*.

Figura 6 – Funcionamento dos métodos “Bagging” e *Random Forest* - Adaptado de (OSHIRO, 2013).

A primeira estratégia apresenta um processo de amostragem com reposição em que cada árvore usa a mesma quantidade de dados mas não com a totalidade dos dados originais. As amostras são selecionadas e pode ocorrer repetições de amostra de forma aleatória. Nesse processo, em média $1/3$ das amostras são “*Out of Bag*”, forçando o algoritmo a processar as amostras do conjunto de dados sob um prisma diferente, em cada uma das árvores de decisão.

A segunda estratégia é minimizar a etapa mais custosa de uma árvore de decisão que é a escolha do melhor atributo, aquele que é mais relevante e que mais discrimina (com maior ganho de informação) a partir do teste em todos os atributos. Para isso, o conjunto de atributos que é testado em cada árvore não é coincidente com o número total de atributos, e sim, uma subamostragem aleatória com atributos candidatos ao nó raiz, tornando a implementação bem mais rápida e com topologias diferenciadas para o mesmo problema, com diversidade de dados do conjunto de treinamento e de escolha de atributos das árvores.

O comitê de máquinas tem bom desempenho global quando cada componente tem bom desempenho individual e quando eles divergem no erro, ou seja, quando tem comportamento descorrelacionado. Aparentemente é contraditório, mas o percentual de erros de uma árvore, ainda que pequeno, deve ser diferente do percentual de erros de outra árvore (amostras diferentes). Para o treinamento do *Random Forest*, uma vez que todas as árvores são construídas, o *Random Forest* utiliza uma técnica de votação (ou média) para combinar as previsões de cada árvore e produzir uma previsão final. No caso da classificação, a classe mais frequente entre as árvores é selecionada como a classe prevista. Na regressão, a média das previsões de todas as árvores é calculada para fornecer o valor final (NORVIG; RUSSELL, 2010).

2.3.3 Comparativos a outros métodos de aprendizado de máquina

Em relação a outros métodos de IA, a *Random Forest* tem se destacado em várias áreas de aplicação. Em problemas de classificação, regressão e detecção de anomalias, as *Random Forests* frequentemente alcançam resultados competitivos e superam muitos algoritmos tradicionais. Sua capacidade de lidar com dados não lineares, lidar com características categóricas e numéricas, e lidar com problemas de alta dimensionalidade são pontos fortes que a tornam uma escolha viável em diversos cenários (MURPHY, 2012).

Em comparação com outros algoritmos de aprendizado de máquina, como Redes Neurais Artificiais e *Support Vector Machines*, as *Random Forests* também oferecem uma abordagem mais fácil de interpretar e explicar. Enquanto os modelos de redes neurais podem ser considerados como “caixas pretas”, onde as relações entre as variáveis são difíceis de serem compreendidas, as *Random Forests* fornecem uma visão mais clara e intuitiva das decisões tomadas (LOUPPE, 2015).

No entanto, as *Random Forests* também possuem alguns desafios e problemas associados ao seu uso. Um dos principais desafios é o ajuste adequado dos hiperparâmetros do modelo, como o número de árvores e a profundidade máxima de cada árvore. Uma configuração inadequada desses parâmetros pode levar a um desempenho subótimo ou até mesmo a um “*overfitting*”. Além disso, a interpretação dos resultados da *Random Forests* pode ser desafiadora, especialmente quando se trabalha com um grande número de árvores (CRIMINISI; SHOTTON; KONUKOGLU, 2012).

A combinação das decisões individuais de várias árvores em uma *Random Forests* pode dificultar a compreensão do processo de tomada de decisão e a interpretação dos resultados. A complexidade do modelo e a falta de transparência em relação às relações exatas entre as variáveis de entrada e as saídas podem ser consideradas desvantagens em certos contextos (BIAU; SCORNET, 2016).

2.4 Common Lisp

A linguagem de programação *Lisp*, criada por John McCarthy em 1958, é uma das linguagens mais antigas e influentes no campo da inteligência artificial e da computação em geral. Seu nome, derivado de “LIST Processing” (processamento de listas), reflete uma de suas principais características: a representação de dados e programas através de listas encadeadas. *Lisp* foi desenvolvida com o objetivo de ser uma linguagem de programação simbólica e de alto nível, capaz de manipular símbolos e expressões complexas de forma eficiente (NORVIG; RUSSELL, 2010).

Uma das principais razões para a criação do *Lisp* foi a necessidade de uma linguagem que pudesse expressar e manipular símbolos e estruturas de dados de forma flexível. McCarthy estava interessado em desenvolver uma linguagem que permitisse a criação de programas que pudessem aprender e evoluir, aproximando-se do conceito de inteligência artificial. Assim, o *Lisp* foi projetado com recursos poderosos para a manipulação de listas e para a criação de funções recursivas, permitindo a construção de algoritmos complexos (MCCARTHY, 1978).

Lisp foi projetado para fornecer poderosa ferramenta para a manipulação de informações simbólicas, tornando-a adequada para tarefas como processamento de linguagem natural, prova de teoremas e sistemas especialistas. O foco do *Lisp* no processamento simbólico e sua capacidade de modificar código dinamicamente contribuíram para sua relevância contínua no campo da inteligência artificial. [...]

as características únicas do *Lisp*, como suporte à recursão, manipulação simbólica e meta programação, que o tornaram uma ferramenta poderosa para aplicações de IA (NORVIG; RUSSELL, 2010).

Uma das características mais marcantes do *Lisp* é sua notação prefixa, na qual as funções são escritas antes de seus argumentos. Por exemplo, em *Lisp*, a soma de dois números é escrita como $(+ 2 3)$, onde o símbolo “+” é a função de adição e os números 2 e 3 são os argumentos. Essa notação pode parecer estranha para quem está acostumado com outras linguagens, mas ela permite uma sintaxe simples e uniforme, facilitando a manipulação de expressões e a criação de novas funções (GRAHAM, 1993).

Além disso, *Lisp* possui um sistema de macros poderoso, que permite a extensão da própria linguagem. Com as macros, é possível criar novas construções sintáticas, personalizando a linguagem de acordo com as necessidades do programador. Isso dá ao *Lisp* uma flexibilidade e expressividade únicas, tornando-a uma linguagem bastante adaptável a diferentes domínios e paradigmas de programação (KEKKONEN, 2023).

Ao longo dos anos, variantes e dialetos do *Lisp* foram desenvolvidos, cada um com suas próprias características e implementações. Alguns dos dialetos mais conhecidos são o *Common Lisp*, o *Scheme* e o *Clojure*. Cada um deles possui suas particularidades e é

utilizado em diferentes contextos e comunidades. Todavia, há algumas razões pelas quais o *Common Lisp* continua mais popular entre a família de linguagens *Lisp*, tais como a flexibilidade, oferecendo suporte a diversos tipos de programação: procedural, orientada a objetos e funcional, a extensibilidade por meio de macros, a compatibilidade e portabilidade ao padrão *ANSI* e um conjunto abrangente de recursos, como o gerenciamento de memória automático, sistemas de tipo dinâmico e um REPL (*Read-Eval-Print Loop*) (SEIBEL, 2005).

A coisa mais próxima que o *Common Lisp* tem de um lema é a descrição: “a linguagem de programação programável”. Embora enigmática, essa descrição chega à raiz da maior vantagem que o *Common Lisp* ainda tem sobre outras linguagens. Mais do que qualquer outra linguagem, *Common Lisp* segue a filosofia de que o que é bom para o designer da linguagem é bom para os usuários da linguagem. Assim, quando você está programando em *Common Lisp*, você quase nunca deseja que a linguagem suporte algum recurso que torne seu programa mais fácil de escrever, porque, como você verá ao longo deste livro, você mesmo pode adicionar o recurso.

[...] um programa *Common Lisp* tende a fornecer um mapeamento muito mais claro entre suas ideias sobre como o programa funciona e o código que você realmente escreve. Suas ideias não são obscurecidas por códigos padronizados e expressões idiomáticas infinitamente repetidas. Isso torna seu código mais fácil de manter porque você não precisa percorrer resmas de código toda vez que precisar fazer uma alteração. Até mesmo mudanças sistêmicas no comportamento de um programa geralmente podem ser alcançadas com mudanças relativamente pequenas no código real. Isso também significa que você desenvolverá código mais rapidamente; há menos código para escrever e você não perde tempo tentando encontrar uma maneira limpa de se expressar dentro das limitações do idioma (SEIBEL, 2005).

Apesar de ter sido criada há mais de seis décadas, a linguagem *Lisp* ainda mantém sua relevância e promessa na área de inteligência artificial e *machine learning*. Sua sintaxe simples e uniforme, juntamente com recursos avançados como macros e manipulação de listas, tornam *Lisp* uma escolha natural para implementação de algoritmos complexos e experimentações no campo da IA (NORVIG et al., 2012).

No entanto, é importante ressaltar que a popularidade de *Common Lisp* não é como outras linguagens de programação, como *Python* e *Java*. Embora tenha um grupo fiel de seguidores e seja amplamente utilizada em certos domínios, *Lisp* não está entre as dez linguagens mais utilizadas no mercado atualmente. Isso pode ser atribuído, em parte, à sua curva de aprendizado mais íngreme e à falta de ferramentas e bibliotecas atualizadas para suporte a tarefas específicas. (MOGENSEN, 2022).

Ainda assim, a importância de *Lisp* na área de inteligência artificial e *machine learning* é inegável. É uma das linguagens “clássicas” da ciência da computação, baseada em ideias que resistiram ao teste do tempo. Suas características únicas e sua capacidade de representar

e manipular símbolos e estruturas de dados complexas permitem a implementação de algoritmos sofisticados e a resolução de problemas de forma elegante. *Lisp* tem sido amplamente utilizado para o desenvolvimento de sistemas especialistas, sistemas de raciocínio baseado em casos e sistemas de aprendizado simbólico (SEIBEL, 2005).

A capacidade de representar conhecimento de forma simbólica e de executar manipulações sofisticadas de expressões lógicas torna *Lisp* uma escolha natural para a implementação desses sistemas. Além disso, *Lisp* tem sido aplicado no campo do processamento de linguagem natural, onde a manipulação de expressões simbólicas e a construção de gramáticas formais são fundamentais. Através de técnicas como *parsing* e geração de linguagem natural, *Lisp* permite a implementação de sistemas de processamento de linguagem natural avançados, capazes de entender e gerar texto de forma inteligente (TOURETZKY, 1990).

No contexto de *machine learning*, *Lisp* tem sido utilizado na implementação de algoritmos e modelos de aprendizado de máquina, incluindo técnicas como redes neurais, árvores de decisão e algoritmos genéticos. A capacidade de expressar algoritmos complexos e de manipular estruturas de dados de forma flexível torna *Lisp* uma linguagem poderosa para experimentações e pesquisas nessa área (WINSTON; HORN, 1987).

A linguagem *Lisp* tem uma longa história de desenvolvimento de ferramentas e bibliotecas especializadas, muitas das quais foram pioneiras em suas respectivas áreas. Essas ferramentas e bibliotecas, embora possam não ser tão amplamente utilizadas quanto as encontradas em outras linguagens, ainda fornecem recursos valiosos para aqueles que desejam explorar os benefícios e possibilidades do *Lisp* (MOGENSEN, 2022).

O Google tem centenas de programadores *Lisp* trabalhando em passagens aéreas de baixo custo. Três empresas de computadores quânticos financiadas maciçamente usam um compilador quântico escrito em *Common Lisp*. Grammarly, um assistente de escrita baseado em nuvem, usa intensamente o *Lisp*, assim como o sistema de agendamento do Telescópio Espacial James Webb. Eu especulo que essas empresas mantêm seu uso de *Lisp* em silêncio porque é uma verdadeira vantagem competitiva. A maioria tola prefere adicionar mais máquinas do que enfrentar uma curva de aprendizado para uma linguagem mais antiga. No entanto, reuni uma confiança multigeracional que ainda acredita em *Lisp*, e estamos trabalhando em uma versão rica em recursos, à prova de futuro e disponível gratuitamente, especificamente voltada para grandes aplicativos da Web escaláveis (KUNZE; KUNZE, 2022).

Estado da Arte

3.1 Panorama Tecnológico em Bioinformática

A bioinformática é uma área vasta e diversificada que inclui uma ampla gama de aplicações, ferramentas, métodos, algoritmos e tecnologias que abordam os problemas em estudo. As ferramentas mencionadas anteriormente na seção 2.1.2 são mais específicas para análises estruturais e funcionais de proteína. No entanto, existem outras ferramentas, como *Blast* e o *GGSearch*, que são amplamente utilizadas para buscar sequências similares em bancos de dados de sequências genômicas e proteicas, com o objetivo de alcançar similaridade proteica. A seguir, são apresentadas as tecnologias e métodos mais recentes e diretamente aplicáveis ao estudo, bem como algoritmos avançados relevantes.

3.1.1 Arquivos MultiFasta

Os arquivos *multifasta* são arquivos de texto comumente usados na bioinformática que contêm múltiplas sequências de DNA ou proteína em um formato padronizado, o que permite que as ferramentas processem e comparem várias sequências simultaneamente para armazenar várias sequências biológicas em um único arquivo. Cada sequência é representada por um cabeçalho seguido pela sequência de nucleotídeos ou aminoácidos. Eles são utilizados em bancos de dados de sequências biológicas, em alinhamentos de sequências e em análises de similaridade e homologia e geralmente utilizados como dados de entrada em diversas ferramentas bioinformáticas, incluindo o *Blast*, o algoritmo *Needleman-Wunsch*, o *GGSearch* e muitos outros programas de alinhamento e análise de sequências (PEARSON, 2014).

3.1.2 AAIndex

A *AAindex* é um banco de dados que contém informações sobre várias características dos aminoácidos ou propriedades biofísicas e químicas (“Features”) dos aminoácidos, como hidrofobicidade, massa molecular, propriedades de polaridade, entre outros, desenvolvido pelo National Institute of Advanced Industrial Science and Technology (AIST) do Japão e está disponível gratuitamente para uso público. Os índices podem ser utilizados para analisar e comparar sequências de aminoácidos com base em suas propriedades físico-químicas e são frequentemente usadas em análises bioinformáticas para prever ou entender aspectos estruturais e funcionais das proteínas. O *AAindex* pode ser usado como uma referência externa para análise e interpretação de resultados obtidos por outras ferramentas, à luz das propriedades dos aminoácidos (KAWASHIMA; KANEHISA, 2000).

3.1.3 STRING DataBase e BioGRID

STRING DataBase e *BioGRID* são os principais bancos de dados de interação proteína-proteína. O *BioGRID* é uma base de dados abrangente que coleta e cura interações biológicas, incluindo PPIs, de uma variedade de fontes, como literatura científica, experimentos de alto rendimento e bancos de dados relacionados. O *STRING Database* oferece uma ampla coleção de interações conhecidas e previstas entre proteínas, incluindo interações diretas e indiretas. Além disso, fornece informações sobre associações funcionais, redes de interação e análises de enriquecimento funcional (SZKLARCZYK et al., 2021).

3.1.4 Algoritmo Needleman-Wunsch e Smith-Waterman

O algoritmo *Needleman-Wunsch* é um algoritmo clássico de alinhamento global de sequências, comumente usado para comparar duas sequências biológicas, como sequências de proteínas ou DNA. Ele é aplicado para encontrar o melhor alinhamento entre duas sequências, considerando todas as posições de ambos os alinhamentos e é útil para identificar regiões conservadas e variáveis entre as sequências. Já o algoritmo de *Smith-Waterman* é utilizado para alinhamento local. Ambos algoritmos são considerados ótimos, pois garantem encontrar a solução de alinhamento com o maior escore possível, levando em consideração todas as posições das sequências, no caso global, e, no alinhamento local, o objetivo é encontrar regiões de similaridade significativa entre duas sequências, independentemente de sua posição ou comprimento total. No entanto, esses algoritmos podem ser computacionalmente intensivos para sequências muito grandes (GAUTHAM, 2006).

O *GGSearch* é uma ferramenta de bioinformática que implementa o algoritmo de alinhamento global Needleman-Wunsch, além de outras técnicas, para uma variedade de aplicações, incluindo a identificação de genes ou proteínas em genomas ou transcritomas, a análise de homologias entre sequências, a detecção de mutações em sequências, a análise de evolução molecular, dentre outras. Nas pesquisas de sequências de DNA ou proteínas em uma base de dados, ele utiliza correspondência global, o que significa que ele tenta encontrar a melhor correspondência possível para a sequência de pesquisa em toda a base de dados. Desenvolvido pelo *European Bioinformatics Institute* (EBI), está disponível gratuitamente para uso público (PEARSON et al., 1997).

O *SSSearch* implementa o algoritmo de alinhamento local Smith-Waterman ao comparar uma sequência de proteína a um banco de dados de sequência de proteínas ou uma sequência de DNA a um banco de dados de sequência de DNA e é cerca de 10 vezes mais lento que *fasta*, mas é mais sensível para comparação de sequências de proteínas completas. Ambos os pacotes estão disponíveis no pacote *Fasta3* (PEARSON, 1999).

3.1.5 Blast

O *Blast* (*Basic Local Alignment Search Tool*) é uma plataforma desenvolvida pelo National Center for Biotechnology Information (NCBI) e representa um avanço crucial na pesquisa biomédica e genômica. Trata-se de uma ferramenta básica de pesquisa que compara sequências biológicas para encontrar regiões de similaridade, indicando homologias e possíveis relações evolutivas. O programa compara sequências de nucleotídeos ou proteínas com bancos de dados de sequências e calcula a significância estatística das correspondências. O *Blast* pode ser usado para inferir relações funcionais e evolutivas entre sequências, bem como para ajudar a identificar membros de famílias de genes (GAUTHAM, 2006).

O *Blast* utiliza um algoritmo de pesquisa rápida baseado em heurísticas. Possui diferentes variantes, como o *Blastp* (para proteínas), o *Blastn* (para ácidos nucleicos) e outras. Ele é amplamente utilizado para identificar funções de sequências desconhecidas, classificar proteínas e estudar a evolução molecular. O programa oferece uma variedade de opções que podem ser usadas para ajustar o desempenho do programa (CAMACHO et al., 2009) Essas opções incluem:

- ❑ **Mecanismo de alinhamento:** oferece suporte a dois mecanismos de alinhamento: alinhamento local e alinhamento global. O alinhamento local é mais rápido, mas menos preciso do que o alinhamento global.
- ❑ **Métrica de pontuação:** oferece suporte a várias métricas de pontuação, incluindo o alinhamento de pontuação máxima, o alinhamento de pontuação máxima com *gap*

e o alinhamento de pontuação máxima com *gap* e corte.

- ❑ **Métrica de corte:** A métrica de corte é usada para determinar quando o *GGSearch* deve interromper a busca por uma correspondência.

O *Blast* retorna os seguintes resultados para cada correspondência:

- ❑ **Pontuação de correspondência:** Esta pontuação reflete a semelhança entre a sequência de pesquisa e a sequência correspondente.
- ❑ **Porcentagem de identidade:** A porcentagem de identidade é a porcentagem de aminoácidos que correspondem entre as duas sequências.
- ❑ **Cobertura da sequência:** A cobertura da sequência refere-se ao número total de posições na sequência de pesquisa que apresentam correspondências na sequência correspondente.
- ❑ **Posição do início:** A posição do início é a posição na sequência de pesquisa onde a correspondência começa.
- ❑ **Posição do fim:** A posição do fim é a posição na sequência de pesquisa onde a correspondência termina.

Tanto o *GGSearch* quanto o *Blast* são ferramentas da bioinformática muito utilizadas para uma variedade de aplicações. Fáceis de usar e oferecem uma variedade de opções que podem ser usadas para ajustar o desempenho do programa.

3.1.5.1 Blastp

O *Blastp*, uma variante do *Blast*, realiza comparações entre uma sequência de proteína específica e um extenso banco de dados de proteínas em tempo de execução linear $O(n)$, oferecendo uma alternativa ao uso do algoritmo de alinhamento exato de sequências biológicas Needleman–Wunsch (LEE; KIM; UY, 2020). Identifica regiões de similaridade entre as sequências biológicas. Sua utilidade transcende a mera identificação de correspondências, estendendo-se à inferência de relações funcionais e evolutivas entre as sequências analisadas. Além disso, o *Blastp* contribui para a identificação de membros pertencentes a famílias de genes, ampliando seu impacto na compreensão da diversidade genômica.

O diferencial do *Blastp* como executável instalado no sistema operacional está na capacidade de realizar essas análises localmente, sem a necessidade de recorrer ao servidor online do NCBI. Esse recurso é especialmente valioso quando os pesquisadores lidam com conjuntos massivos de dados ou quando utilizam informações que ainda não foram disponibilizadas à comunidade científica.

Ao empregar o *Blastp*, exploramos as similaridades e divergências entre sequências de proteínas, não apenas para identificar homologias, mas também para obter insights valiosos sobre a evolução e a funcionalidade de genes específicos. Assim, o NCBI, por meio do *Blastp* e de sua ampla gama de ferramentas, continua a impulsionar a pesquisa biomédica, fornecendo recursos fundamentais para a comunidade científica enfrentar os desafios mais complexos.

3.2 Algoritmos baseados em Aprendizado de Máquina

Diferentes algoritmos de aprendizado de máquina são usados como classificadores de proteínas para identificação de similaridade proteica ou na identificação de interações proteína-proteína.

Na classificação de proteínas, os algoritmos podem ser usados para atribuir uma classe ou rótulo a uma sequência de aminoácidos. Eles são usados em uma variedade de aplicações, incluindo: previsão de estrutura e função de proteínas, identificação e/ou pesquisa de proteínas. A complexidade dos classificadores de proteínas varia de acordo com o algoritmo usado. Alguns classificadores são relativamente simples, como os baseados em índices de propensão de aminoácidos. Outros classificadores são mais complexos, como os baseados em aprendizado de máquina (BARABASI; OLTVAI, 2004).

Os classificadores baseados em aprendizado de máquina são mais complexos porque são treinados em um conjunto de dados de exemplos rotulados, que consistem em sequências de proteínas com suas classes ou rótulos correspondentes. Estes podem aprender padrões complexos nas sequências de proteínas. Isso os torna mais precisos do que os classificadores baseados em índices de propensão de aminoácidos, especialmente para proteínas com estruturas ou funções atípicas. As pesquisas em andamento de novos algoritmos de classificação tem como finalidade a melhoria da qualidade dos dados de treinamento ou aplicação de classificadores de proteínas a novos problemas. Destaca-se a predominância de pesquisas baseados em redes neurais e aprendizado de máquina profundo (*Deep Learning*) (MIN; LEE; YOON, 2017).

Em relação à predição de redes de interação proteína-proteína, Tang et al. (2023) também enfatiza que os métodos de aprendizado profundo, como redes neurais profundas (DNN) e redes neurais convolucionais (CNN), têm recebido mais atenção do que os modelos tradicionais de aprendizado de máquina devido à complexidade inerente da predição de PPI e ao aumento do volume de dados disponíveis, especialmente de proteínas interagentes e ainda apresenta uma sequência histórica dos modelos de aprendizado de máquina, conforme Figura 7.

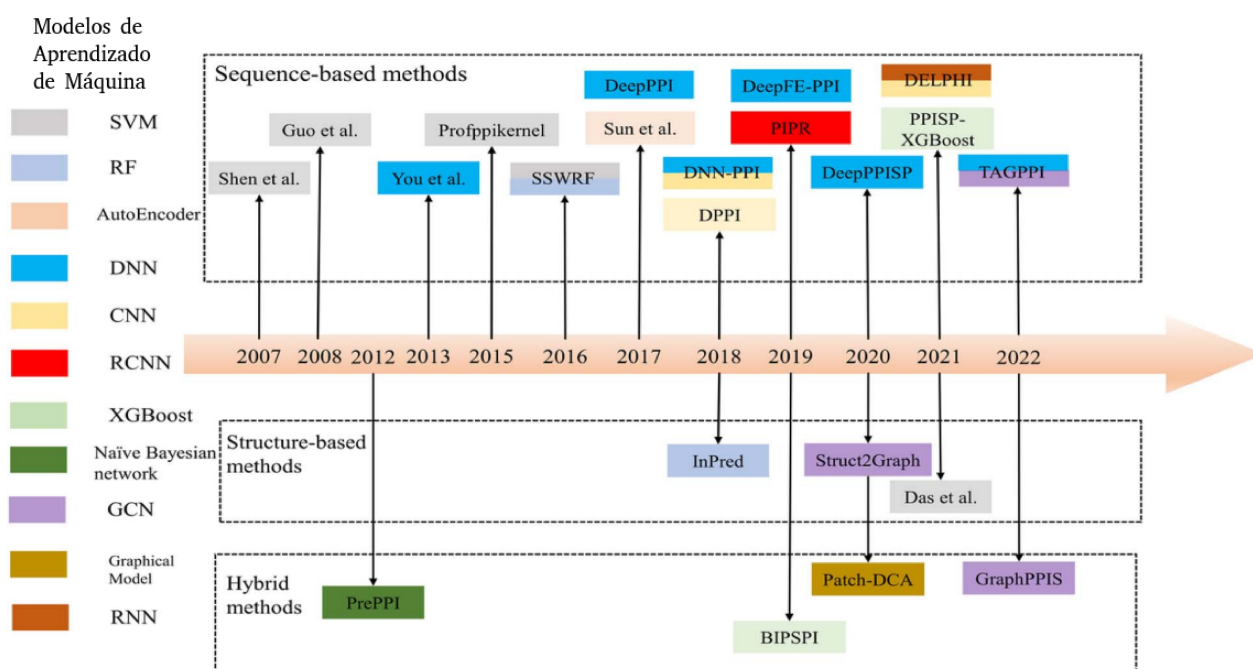


Figura 7 – Modelos de Aprendizado de Máquina - Adaptado de (TANG et al., 2023).

Todavia, na predição para identificação de redes de interação proteína-proteína, em revisão abrangente de ferramentas computacionais disponíveis para identificação de interação proteína-proteína entre humanos e patógenos bacterianos, considerando os métodos tradicionais de aprendizado de máquina (ML), o *Random Forest* demonstrou uma forte capacidade de generalização, alcançando as melhores taxas de acurácia para diferentes proporções de dados (CHEN et al., 2021). O estudo destaca a escassez de estudos abrangentes que tenham se concentrado na avaliação de modelos de aprendizado de máquina como método computacional primário para prever interações entre hospedeiro e patógeno. Foram avaliados modelos de aprendizados de máquina divididos em dois grupos: os baseados em árvores e modelos baseados em Kernel (*Support Vector Machine*, *Logistic Regression* e *Naive Bayes*). Verificou-se o potencial de *Random Forest* como um modelo eficaz e que pode auxiliar na identificação de novos alvos terapêuticos e no desenvolvimento de combate a doenças infecciosas.

Rasti e Vogiatzis (2019) discute a importância da seleção de características (*feature selection*) na análise de redes de interação proteína-proteína. Os autores enfatizam que a escolha das características corretas pode ter um impacto significativo na precisão e na eficácia dos modelos de previsão de interações proteína-proteína. Métodos como SVM-RFE (*Support Vector Machine - Recursive Feature Elimination*) e PCC (*Pearson Correlation Coefficient*) são mencionados como ferramentas eficazes para identificar um subconjunto ótimo de características que podem melhorar o desempenho dos modelos de previsão. O

SVM-RFE é uma abordagem que combina o algoritmo de *Support Vector Machine* com um processo de eliminação recursiva de características. O PCC é uma medida estatística que avalia a correlação linear entre duas variáveis. Essa abordagem ajuda a reduzir a dimensionalidade dos dados e a eliminar características redundantes ou irrelevantes, melhorando assim a capacidade dos modelos de aprendizado de máquina de generalizar e fazer previsões precisas.

Os autores ainda ressaltam a importância de considerar não apenas as características topológicas das redes de interação proteína-proteína, mas também características biológicas e funcionais, como expressão gênica, anotações funcionais e localização celular. A integração desses diferentes tipos de características pode fornecer uma visão mais abrangente e precisa das interações entre proteínas e da essencialidade dos genes. Também cita vantagens e desvantagens do método *Random Forest* em relação a outros métodos de aprendizado de máquina, como redes neurais e *Support Vector Machine*, ao escolher o método de aprendizado de máquina mais apropriado para uma determinada aplicação na análise de redes de PPI.

As vantagens são:

- ❑ Robustez a *overfitting*, especialmente em conjuntos de dados de alta dimensionalidade;
- ❑ Capacidade de lidar efetivamente com variáveis irrelevantes ou redundantes;
- ❑ Eficiência computacional, especialmente em conjuntos de dados grandes, em comparação com redes neurais e *Support Vector Machine*.

As desvantagens são:

- ❑ Dificuldade de interpretação;
- ❑ Sensibilidade a ruído.

Em outra revisão sistemática de estratégias de aprendizado de máquinas para predição função de proteínas, Yan et al. (2023) destaca a importância da integração de informações de várias fontes e o potencial de algoritmos específicos, como redes neurais convolucionais (CNN), redes neurais recorrentes (RNN), K-vizinhos mais próximos (KNN) e *Random Forest*, para aprimorar a precisão e a eficiência desse processo. A integração de duas técnicas distintas como redes neurais e *Random Forest* supera o desempenho de classificadores independentes. Além disso, aplicação de *Random Forest* é discutida em relação à predição de funções moleculares, destacando a superioridade de *embeddings* de proteínas não supervisionadas em comparação com características de sequência e estrutura feitas manualmente.

A *Random Forest*, portanto, demonstrou um potencial significativo na predição de interações proteína-proteína, destacando-se por sua capacidade de lidar com grandes conjuntos de dados, capturar relações não lineares entre variáveis e adaptar-se a conjuntos de dados heterogêneos, o que sugere que pode ser ainda melhor explorada devido a esse potencial. A utilização de outros métodos de aprendizado e a integração de técnicas avançadas de aprendizado de máquina podem ampliar ainda mais a capacidade da *Random Forest* na predição de interações proteína-proteína, permitindo a captura de relações ainda mais complexas e aprimorando a precisão das previsões (DHAKAL et al., 2022).

3.3 Trabalhos Relacionados

A predição de redes de interação proteína-proteína (PPI) tem sido um foco contínuo de pesquisa em bioinformática, especialmente com o surgimento de ferramentas que utilizam técnicas de aprendizado de máquina para ampliar a acurácia e a profundidade dessas previsões. No entanto, desde o lançamento da versão inicial do *Genppi*, há cerca de quatro anos, não se observa um avanço significativo no desenvolvimento de ferramentas que ofereçam suporte para a predição de redes de PPI a partir de genomas completos e não anotados de qualquer organismo, incluindo espécies bacterianas e arqueias menos estudadas. O *Genppi*, na versão 1.0, destacou-se como um pioneiro ao oferecer uma abordagem que permite essa generalização em predições de PPI, e até o momento, nenhum outro trabalho ou software disponível apresenta a mesma flexibilidade ou eficiência em cenários semelhantes.

A ferramenta mais amplamente utilizada para a predição de interações proteicas, o *STRING* (*Search Tool for the Retrieval of Interacting Genes/Proteins*), fornece uma base de dados extensiva e integrada de interações proteicas, reunindo informações de múltiplas fontes, como experimentos bioquímicos e associações genéticas (SZKLARCZYK et al., 2015). Apesar de sua abrangência e da facilidade de uso para análises comparativas entre organismos, o *STRING* é limitado a um conjunto específico de genomas previamente conhecidos e estudados, o que restringe sua aplicabilidade para pesquisadores que trabalham com genomas bacterianos inéditos ou menos conhecidos. Tais limitações tornam difícil a comparação direta entre o *STRING* e o *Genppi*, exceto no sentido de validação da proposta, vez que o *Genppi* foi desenvolvido com a proposta de preencher essa lacuna e possibilitar a predição em qualquer genoma bacteriano ou de arqueas.

Além disso, embora existam outras metodologias para a predição de PPI, muitas delas são projetadas para aplicações específicas e apresentam restrições em relação ao tipo de organismo e à origem dos dados. Por exemplo, o *CEGSO* propõe uma abordagem que integra informações de complexos proteicos e localização subcelular para predição de

proteínas essenciais (ZHANG et al., 2021). No entanto, esta abordagem é teoricamente robusta, mas carece de uma implementação de software que permita sua utilização prática em plataformas como o *Genppi*. Outra proposta recente, o *BBLN: A Bilateral-Branch Learning Network* (KANG et al., 2023), introduz um modelo de aprendizado para prever interações de proteínas desconhecidas, mas limita-se a um conjunto restrito de dados testados. As dificuldades práticas de execução do *BBLN* também limitaram sua avaliação comparativa com o *Genppi*. Embora o artigo *BBLN* fornecesse condições para a reprodução do código e realização de testes independentes, o programa apresentou problemas de funcionalidade ao ser executado. Tentativas de contato por e-mail com os autores do *BBLN* foram realizadas para obter esclarecimentos e orientações sobre a reprodução dos testes, com o intuito de comparar desempenho e resultados entre o *BBLN* e o *Genppi*. No entanto, não houve respostas aos questionamentos enviados.

Durante o desenvolvimento da versão original do *Genppi*, outros algoritmos como o *A Novel Network-Based Algorithm for Predicting Protein-Protein Interactions Using Gene Ontology* (HU et al., 2021), que utilizam uma abordagem baseada na Ontologia Genética e um modelo Bayesiano, foram analisados, mas apresentaram desempenho inferior na predição de interações PPI em relação ao *Genppi* (OLIVEIRA; SANTOS, 2016). A análise topológica de redes PPI, conforme descrita em estudos como *Topological scoring of protein interaction networks* (SARDIU et al., 2019), também reforça a relevância de ferramentas como o *Genppi*, cuja capacidade de facilitar análises topológicas em redes de interação se provou essencial para aprofundar o conhecimento sobre as funções e os processos biológicos.

Com base nesses trabalhos, torna-se evidente que o *Genppi*, mesmo em sua versão 1.0, ocupa uma posição singular no cenário atual de predição de redes de PPI. A ausência de ferramentas que ofereçam um desempenho e flexibilidade equivalentes reforça a relevância de aprimorar a plataforma *Genppi*. A próxima seção detalhará as funcionalidades e a estrutura do *Genppi* na versão original, proporcionando uma visão mais aprofundada de sua contribuição inicial para a predição de redes de interação proteína-proteína.

3.4 *Genppi*

Criado por William Anjos, Gabriel Lanes, Vasco Azevedo e Anderson Santos do Laboratório de Biologia Computacional, Faculdade de Computação (FACOM), Universidade Federal de Uberlândia e do Departamento de Genética, Universidade Federal de Minas Gerais, *Genppi* é um gerador de rede de interação de proteínas *ab initio*. Ele foi desenvolvido para criar redes de interações a partir de genomas e cria redes de interações com base em três características: vizinhanças de genes conservados, perfis filogenéticos e fusões de genes.

O *Genppi* é uma ferramenta de análise de genomas bacterianos que revoluciona a forma como os cientistas estudam a interação entre proteínas. Com sua capacidade de processamento eficiente, o software pode analisar grandes conjuntos de dados genômicos em questão de minutos, fornecendo insights valiosos sobre a conservação de genes, a filogenia e, principalmente, as interações proteína-proteína. Esses resultados são fundamentais para a pesquisa biomédica, permitindo que os cientistas entendam melhor como as proteínas interagem e como essas interações afetam a saúde humana (ANJOS et al., 2021).

Em contraste com as abordagens tradicionais, que muitas vezes dependem de métodos experimentais caros e demorados, o *Genppi* oferece uma solução computacional inovadora e acessível para prever interações proteína-proteína biologicamente relevantes. Sua capacidade de detectar interações entre proteínas em genomas recém-sequenciados, incluindo proteínas não previamente identificadas, representa um avanço significativo na análise de dados genômicos e no desenvolvimento de terapias personalizadas. Com o número de genomas disponíveis crescendo rapidamente, o *Genppi* se torna uma ferramenta essencial para a pesquisa em microbiologia e bioinformática, permitindo que os cientistas façam descobertas mais rápidas e precisas (ANJOS, 2021).

O software inspeciona genomas representados como uma proteína multifasta, procurando uma vizinhança conservada, perfil filogenético e fusão de genes. Ele permite transferir a decisão de quantos e quais genomas usar para a construção de uma rede de interação de proteínas para o usuário final. Foi escrito em *Common Lisp* e compilado com SBCL para criar redes complexas de proteínas previstas a partir de centenas de genomas.

Além disso, são oferecidas versões específicas adaptadas a diferentes capacidades de memória RAM, como 4GB, 8GB, 16GB e 32GB, possibilitando sua adequação às exigências do usuário e ao hardware disponível. Com suporte para sistemas Linux, Windows e Macintosh, o software aproveita as particularidades de cada ambiente operacional. As variantes “Speed” e “Memory” foram especialmente otimizadas para proporcionar velocidade e eficiência de memória, respectivamente, a fim de atender aos diversos requisitos de desempenho. Essa flexibilidade possibilita a utilização do *Genppi* em uma ampla gama de tarefas, desde processamentos leves até operações mais intensivas (ANJOS et al., 2021).

O *Genppi* pressupõe que as proteínas estejam ordenadas de forma correspondente à disposição de seus genes na fita de DNA, refletindo assim a organização genômica subjacente. Essa abordagem permite uma integração eficaz entre as informações genômicas e proteômicas durante o processo de análise e predição de interações proteína-proteína.

O *Genppi* requer um parâmetro obrigatório como dado de entrada, que é um arquivo multi-fasta contendo as sequências de aminoácidos das proteínas a serem analisadas. Esse arquivo deve seguir a convenção padrão, onde cada entrada é precedida por um símbolo de maior que (>) seguido pelo nome da proteína e uma descrição adicional, seguidos pelas

linhas contendo a sequência de aminoácidos correspondente. O software pressupõe que as proteínas estejam ordenadas de forma correspondente à disposição de seus genes na fita de DNA, refletindo assim a organização genômica subjacente. Essa abordagem permite uma integração eficaz entre as informações genômicas e proteômicas durante o processo de análise e predição de interações proteína-proteína.

Além disso, o *Genppi* pode aceitar outros parâmetros opcionais, como configurações para ajustar a sensibilidade da análise, especificações de formato de saída dos resultados e opções para personalização de algoritmos e métodos de predição. Esses parâmetros secundários permitem uma adaptação flexível do software às necessidades específicas de cada estudo, proporcionando uma análise mais precisa e abrangente das interações proteína-proteína (ANJOS, 2021).

O estabelecimento das relações proteína-proteína (PP) é feita com base em nós e vértices, construindo uma rede complexa a partir destes. As ORFs (“Open Reading Frame”), consideradas como vértices, e os relacionamentos, considerados como arestas, formam a estrutura dessa rede. A qualidade e estabilidade das arestas associadas, juntamente com o número de vértices, determinam a informação disponível na rede. O *Genppi* permite a exploração de relações próximas e distantes entre os vértices, definidas pelo número de arestas diretamente conectadas a cada vértice e pela importância de um conjunto de vértices como o mais provável de ser percorrido ao conectar um par arbitrário de pontos em uma rede, respectivamente. O *Genppi* ainda emprega várias medidas de centralidade para análise de dados sob uma perspectiva topológica, como PageRank, Coeficiente de Ponte, Centralidade de Ponte, Densidade e Diâmetro de uma rede (ANJOS et al., 2021).

O software não dispõe de uma interface gráfica, porém, é possível gerar visualizações gráficas a partir dos relatórios produzidos, utilizando ferramentas externas como R, Python ou Matlab. Para isso, são disponibilizados scripts R que podem ser executados para produzir gráficos de barras e caixas a partir dos relatórios de perfil filogenético e conservação da vizinhança gênica. Esses gráficos fornecem uma representação visual das características das proteínas analisadas e das interações identificadas, facilitando a interpretação e análise dos resultados obtidos pelo *Genppi*. Após a execução, são geradas pastas contendo os seguintes relatórios:

- ❑ **Gene neighborhood conservation report:** este diretório contém um arquivo texto denominado report.txt que exibe 5 (cinco) colunas separadas por vírgulas (“,”) e as linhas separadas por ponto e vírgula (“;”). As colunas representam dados respectivamente sobre: nome do genoma, nome do gene, número de genes conservados, número de genes não conservados e número de genomas de cada gene conservado, conforme está representado na Figura 8.

```

#---Gene neighborhood conservation report---

#Minimum identity percentage: 92,55%,
Expansion type: fixed, Window size: 10

#-----

#Columns per row: Genome, Gene,
Number of genes conserved,
Number of genes not conserved,
Number of genomes of each conserved gene;

Ba_Ak, BAKON_002, 7, 3, 4 4 2 2 2 4 2;
Ba_Ak, BAKON_003, 7, 3, 4 2 2 2 4 2 5;
Ba_Ak, BAKON_004, 7, 3, 2 2 2 4 2 4 2;
Ba_Ak, BAKON_005, 2, 8, 2 2;
Ba_Ak, BAKON_006, 2, 8, 2 2;
Ba_Ak, BAKON_007, 4, 6, 2 2 2 2;
Ba_Ak, BAKON_008, 3, 7, 2 4 2;
Ba_Ak, BAKON_009, 2, 8, 2 2;
Ba_Ak, BAKON_013, 1, 9, 2;
Ba_Ak, BAKON_014, 0, 10, 0;
Ba_Ak, BAKON_018, 2, 8, 5 2;
Ba_Ak, BAKON_019, 1, 9, 2;
Ba_Ak, BAKON_020, 4, 6, 2 2 2 2;
Ba_Ak, BAKON_021, 4, 6, 2 3 2 2;
Ba_Ak, BAKON_022, 3, 7, 2 2 2;
Ba_Ak, BAKON_025, 6, 4, 2 2 3 2 3 3;
Ba_Ak, BAKON_026, 4, 6, 2 2 2 2;
Ba_Ak, BAKON_029, 6, 4, 2 2 2 2 2 2;
Ba_Ak, BAKON_032, 8, 2, 2 4 4 2 4 3 4 4;
Ba_Ak, BAKON_033, 5, 5, 2 2 2 2 2;

```

Figura 8 – Exemplo da saída do arquivo gerado report.txt do diretório gene-neighborhood-conservation-report.

- **Pan genome:** este diretório contém dois tipos de arquivos, apresentando o mesmo conteúdo, porém com forma diferente de exibição. O primeiro é chamado de pan-genome-format-1.txt. Nele é exibido blocos que especifica determinada proteína e seu genoma, seguido das proteínas similares e seu respectivo genoma (Figura 9(a)). O segundo é chamado de pan-genome-format-2.txt, que exibe em cada linha 3 (três) colunas separadas por vírgula (“,”), onde a primeira representa o genoma, a segunda representa o gene e a última, as listas de similaridade (Figura 9(b)).

```

-----Pan genome-----
Minimum identity percentage: 92,55%
-----

Protein: BAKON_002 | Genome: Ba_Ak

Similar proteins:

Protein: BUMP002_CDS00585 | Genome: Ba_G002;
Protein: BUSg_002 | Genome: Ba_Sg;
Protein: BUAMB_002 | Genome: Ba_Ua;
-----

Protein: BAKON_003 | Genome: Ba_Ak

Similar proteins:

Protein: bbp_003 | Genome: Ba_Bp;
Protein: BUMP002_CDS00584 | Genome: Ba_G002;
Protein: BUSg_003 | Genome: Ba_Sg;
Protein: BUAMB_003 | Genome: Ba_Ua;
-----

Protein: BAKON_004 | Genome: Ba_Ak

Similar proteins:

Protein: BUMP002_CDS00583 | Genome: Ba_G002;
Protein: BUSg_004 | Genome: Ba_Sg;
Protein: BUAMB_004 | Genome: Ba_Ua;
-----

Protein: BAKON_005 | Genome: Ba_Ak

Similar proteins:

Protein: BUSg_005 | Genome: Ba_Sg;
-----

Protein: BAKON_006 | Genome: Ba_Ak

Similar proteins:

Protein: BUAMB_006 | Genome: Ba_Ua;
-----

```

a) pan-genome-format-1.txt

```

#Columns per row: Genome, Gene, Similar list

Ba_Ak, BAKON_002, BUMP002_CDS00585 BUSg_002 BUAMB_002;
Ba_Ak, BAKON_003, bbp_003 BUMP002_CDS00584 BUSg_003 BUAMB_003;
Ba_Ak, BAKON_004, BUMP002_CDS00583 BUSg_004 BUAMB_004;
Ba_Ak, BAKON_005, BUSg_005;
Ba_Ak, BAKON_006, BUAMB_006;
Ba_Ak, BAKON_007, BUMP002_CDS00580;
Ba_Ak, BAKON_008, BUMP002_CDS00579 BUSg_008 BUAMB_008;
Ba_Ak, BAKON_009, BUMP002_CDS00578;
Ba_Ak, BAKON_013, bbp_013 BUMP002_CDS00574 BUSg_013 BUSg19 BUAMB_013;
Ba_Ak, BAKON_014, BUMP002_CDS00573;
Ba_Ak, BAKON_018, bbp_020 BUMP002_CDS00570 BUSg_018 BUAMB_018;
Ba_Ak, BAKON_019, bbp_021 BUMP002_CDS00569 BUSg_019 BUAMB_019;
Ba_Ak, BAKON_020, BUMP002_CDS00567;
Ba_Ak, BAKON_021, BUMP002_CDS00566 BUSg_022 BUAMB_021;
Ba_Ak, BAKON_022, BUMP002_CDS00565;
Ba_Ak, BAKON_025, BUMP002_CDS00562 BUSg_026;
Ba_Ak, BAKON_026, BUSg_027;
Ba_Ak, BAKON_029, BUMP002_CDS00558;
Ba_Ak, BAKON_032, bbp_033 BUMP002_CDS00555 BUSg_033 BUAMB_031;
Ba_Ak, BAKON_033, BUSg_034;
Ba_Ak, BAKON_034, BUMP002_CDS00553 BUSg_035 BUAMB_033;
Ba_Ak, BAKON_035, BUMP002_CDS00552 BUSg_036 BUAMB_034;
Ba_Ak, BAKON_036, BUAMB_035;
Ba_Ak, BAKON_037, BUMP002_CDS00550 BUSg_038 BUAMB_036;
Ba_Ak, BAKON_038, BUMP002_CDS00549 BUAMB_037;
Ba_Ak, BAKON_039, BUMP002_CDS00548 BUSg_040 BUAMB_038;
Ba_Ak, BAKON_040, BUMP002_CDS00547 BUSg_041 BUAMB_039;
Ba_Ak, BAKON_048, BUMP002_CDS00543;
Ba_Ak, BAKON_049, BUMP002_CDS00542 BUSg_046;
Ba_Ak, BAKON_050, BUMP002_CDS00541;
Ba_Ak, BAKON_052, BUAMB_048;
Ba_Ak, BAKON_053, BUAMB_049;
Ba_Ak, BAKON_055, BUMP002_CDS00536 BUSg_052 BUAMB_051;
Ba_Ak, BAKON_057, bbp_054 BUMP002_CDS00534 BUSg_054 BUAMB_053;
Ba_Ak, BAKON_059, bbp_056 BUMP002_CDS00532 BUSg_056 BUAMB_055;
Ba_Ak, BAKON_063, BUMP002_CDS00528 BUAMB_059;
Ba_Ak, BAKON_064, BUMP002_CDS00527;
Ba_Ak, BAKON_065, bbp_061 BUMP002_CDS00526 BUSg_062 BUAMB_061;
Ba_Ak, BAKON_072, BUMP002_CDS00522 BUSg_066;
Ba_Ak, BAKON_074, BUAMB_070;
Ba_Ak, BAKON_080, BUAMB_076;
Ba_Ak, BAKON_081, BUMP002_CDS00513 BUAMB_077;
Ba_Ak, BAKON_082, BUMP002_CDS00512 BUSg_076 BUAMB_078;
Ba_Ak, BAKON_083, BUMP002_CDS00511 BUSg_077 BUAMB_079;
Ba_Ak, BAKON_084, bbp_070 BUMP002_CDS00510 BUSg_078 BUAMB_080

```

b) pan-genome-format-2.txt

Figura 9 – Exemplo da saída do arquivo gerado do diretório pan-genome.

- ❑ **Phylogenetic profiles report:** este diretório contém um arquivo texto denominado report.txt que exhibe para todos os genomas, a identificação do genoma, o total de perfis e a quantidade de interações entre proteína-proteína, exibindo ao final os perfis com o total de genes, total de redes de interações proteína-proteína (PPI), o peso que teve e o número de genomas, conforme está representado na Figura 10.

-----Report of complete ppi prediction by phylogenetic profile-----
 Minimum identity percentage: 92,55%

Genome: Ba_Ak
 Total profiles: 13
 Total ppi: 5810

Profile 1	=> Total genes: 1	Total ppi: 0	Weight: 0.000	Number of genomes: 4;
Profile 2	=> Total genes: 1	Total ppi: 0	Weight: 0.000	Number of genomes: 3;
Profile 3	=> Total genes: 1	Total ppi: 0	Weight: 0.000	Number of genomes: 3;
Profile 4	=> Total genes: 2	Total ppi: 1	Weight: 0.055	Number of genomes: 4;
Profile 5	=> Total genes: 3	Total ppi: 3	Weight: 0.082	Number of genomes: 4;
Profile 6	=> Total genes: 8	Total ppi: 28	Weight: 0.109	Number of genomes: 3;
Profile 7	=> Total genes: 17	Total ppi: 136	Weight: 0.136	Number of genomes: 2;
Profile 8	=> Total genes: 23	Total ppi: 253	Weight: 0.164	Number of genomes: 3;
Profile 9	=> Total genes: 29	Total ppi: 406	Weight: 0.191	Number of genomes: 3;
Profile 10	=> Total genes: 31	Total ppi: 465	Weight: 0.218	Number of genomes: 2;
Profile 11	=> Total genes: 46	Total ppi: 1035	Weight: 0.245	Number of genomes: 5;
Profile 12	=> Total genes: 58	Total ppi: 1653	Weight: 0.273	Number of genomes: 2;
Profile 13	=> Total genes: 61	Total ppi: 1830	Weight: 0.300	Number of genomes: 4;

Genome: Ba_Bp
 Total profiles: 15
 Total ppi: 1043

Profile 1	=> Total genes: 1	Total ppi: 0	Weight: 0.000	Number of genomes: 3;
Profile 2	=> Total genes: 2	Total ppi: 1	Weight: 0.055	Number of genomes: 3;
Profile 3	=> Total genes: 3	Total ppi: 3	Weight: 0.082	Number of genomes: 3;
Profile 4	=> Total genes: 4	Total ppi: 6	Weight: 0.109	Number of genomes: 4;
Profile 5	=> Total genes: 4	Total ppi: 6	Weight: 0.109	Number of genomes: 3;
Profile 6	=> Total genes: 4	Total ppi: 6	Weight: 0.109	Number of genomes: 3;
Profile 7	=> Total genes: 5	Total ppi: 10	Weight: 0.136	Number of genomes: 4;
Profile 8	=> Total genes: 6	Total ppi: 15	Weight: 0.164	Number of genomes: 2;
Profile 9	=> Total genes: 6	Total ppi: 15	Weight: 0.164	Number of genomes: 3;
Profile 10	=> Total genes: 6	Total ppi: 15	Weight: 0.164	Number of genomes: 2;
Profile 11	=> Total genes: 7	Total ppi: 21	Weight: 0.191	Number of genomes: 2;
Profile 12	=> Total genes: 8	Total ppi: 28	Weight: 0.218	Number of genomes: 4;
Profile 13	=> Total genes: 14	Total ppi: 91	Weight: 0.245	Number of genomes: 2;
Profile 14	=> Total genes: 22	Total ppi: 231	Weight: 0.273	Number of genomes: 5;

Figura 10 – Exemplo da saída do arquivo gerado report.txt do diretório phylogenetic-profiles-report.

□ **Ppi files:** este diretório contém vários arquivos, qual recebe o nome do genoma seguido da extensão *.dot* ou *.sif*. Os arquivos *.sif* são usados para gerar grafos no Cytoscape e R (Figura 11(a)). Eles exibem as interações entre nós (nó origem e nó destino indicadas por seus nomes) e uma relação “*pp*” (perfil filogenético) ou “*cn*” (vizinhança conservada).

Os arquivos *.dot* são usados para gerar grafos no *Gephi*⁴. Ele também exibe as interações entre as proteínas e ainda um peso que indica a magnitude da relação entre os dois nós (Figura 11(b)).

⁴ Gephi é um software utilizado para trabalhar com grafos e redes. Ele permite criar visualizações interativas e explorar dados complexos de forma intuitiva.

Na pasta de ferramentas do *Genppi*, temos um arquivo denominado *countdots*, feito em linguagem C, destinado a informar o número de nós e arestas de um determinado arquivo de extensão *.dot*.

Tal arquivo é muito útil para sumarizar de maneira rápida o números total de proteínas que possui interações, bem como o número total de interações entre as proteínas.

Para sua execução, basta informar o nome do arquivo, como por exemplo “*countdots name.dot*”.

O resultado é exibido em 3 colunas, contendo respectivamente: o nome do arquivo, o número de nós e o número de arestas.

"BAKON_001"	cn	"BAKON_002"	graph Ba_Ak {
"BAKON_001"	cn	"BAKON_003"	"BAKON_001" -- "BAKON_002" [WEIGHT = 0.14];
"BAKON_001"	cn	"BAKON_004"	"BAKON_001" -- "BAKON_003" [WEIGHT = 0.14];
"BAKON_001"	cn	"BAKON_005"	"BAKON_001" -- "BAKON_004" [WEIGHT = 0.14];
"BAKON_002"	cn	"BAKON_003"	"BAKON_001" -- "BAKON_005" [WEIGHT = 0.14];
"BAKON_002"	cn	"BAKON_004"	"BAKON_002" -- "BAKON_003" [WEIGHT = 0.56];
"BAKON_002"	cn	"BAKON_005"	"BAKON_002" -- "BAKON_004" [WEIGHT = 0.56];
"BAKON_002"	cn	"BAKON_006"	"BAKON_002" -- "BAKON_005" [WEIGHT = 0.28];
"BAKON_002"	cn	"BAKON_007"	"BAKON_002" -- "BAKON_006" [WEIGHT = 0.28];
"BAKON_002"	cn	"BAKON_008"	"BAKON_002" -- "BAKON_007" [WEIGHT = 0.28];
"BAKON_002"	cn	"BAKON_009"	"BAKON_002" -- "BAKON_008" [WEIGHT = 0.56];
"BAKON_002"	cn	"BAKON_010"	"BAKON_002" -- "BAKON_009" [WEIGHT = 0.28];
"BAKON_002"	cn	"BAKON_011"	"BAKON_002" -- "BAKON_010" [WEIGHT = 0.14];
"BAKON_002"	cn	"BAKON_012"	"BAKON_002" -- "BAKON_011" [WEIGHT = 0.14];
"BAKON_003"	cn	"BAKON_004"	"BAKON_002" -- "BAKON_012" [WEIGHT = 0.14];
"BAKON_003"	cn	"BAKON_005"	"BAKON_002" -- "BAKON_591" [WEIGHT = 0.3];
"BAKON_003"	cn	"BAKON_006"	"BAKON_002" -- "BAKON_587" [WEIGHT = 0.3];
"BAKON_003"	cn	"BAKON_007"	"BAKON_002" -- "BAKON_586" [WEIGHT = 0.3];
"BAKON_003"	cn	"BAKON_008"	"BAKON_002" -- "BAKON_570" [WEIGHT = 0.3];
"BAKON_003"	cn	"BAKON_009"	"BAKON_002" -- "BAKON_564" [WEIGHT = 0.3];
"BAKON_003"	cn	"BAKON_010"	"BAKON_002" -- "BAKON_554" [WEIGHT = 0.3];
"BAKON_003"	cn	"BAKON_011"	"BAKON_002" -- "BAKON_553" [WEIGHT = 0.3];
"BAKON_003"	cn	"BAKON_012"	"BAKON_002" -- "BAKON_535" [WEIGHT = 0.3];
"BAKON_003"	cn	"BAKON_013"	"BAKON_002" -- "BAKON_531" [WEIGHT = 0.3];
"BAKON_004"	cn	"BAKON_005"	"BAKON_002" -- "BAKON_529" [WEIGHT = 0.3];
"BAKON_004"	cn	"BAKON_006"	"BAKON_002" -- "BAKON_528" [WEIGHT = 0.3];
"BAKON_004"	cn	"BAKON_007"	"BAKON_002" -- "BAKON_526" [WEIGHT = 0.3];
"BAKON_004"	cn	"BAKON_008"	"BAKON_002" -- "BAKON_524" [WEIGHT = 0.3];
"BAKON_004"	cn	"BAKON_009"	"BAKON_002" -- "BAKON_522" [WEIGHT = 0.3];
"BAKON_004"	cn	"BAKON_010"	"BAKON_002" -- "BAKON_518" [WEIGHT = 0.3];
"BAKON_004"	cn	"BAKON_011"	"BAKON_002" -- "BAKON_513" [WEIGHT = 0.3];
"BAKON_004"	cn	"BAKON_012"	"BAKON_002" -- "BAKON_512" [WEIGHT = 0.3];
"BAKON_004"	cn	"BAKON_013"	"BAKON_002" -- "BAKON_511" [WEIGHT = 0.3];
"BAKON_004"	cn	"BAKON_014"	"BAKON_002" -- "BAKON_510" [WEIGHT = 0.3];
"BAKON_005"	cn	"BAKON_006"	"BAKON_002" -- "BAKON_505" [WEIGHT = 0.3];
"BAKON_005"	cn	"BAKON_007"	"BAKON_002" -- "BAKON_504" [WEIGHT = 0.3];

a) extensão *.sif*

b) extensão *.dot*

Figura 11 – Exemplo da saída do arquivo gerado do diretório ppi-files.

- ❑ **Ppi report:** este diretório contém um arquivo texto denominado report.txt que exibe a quantidade de interações previstas para cada genoma, separados pelo tipo da relação: se por perfil filogenético ou por vizinhança conservada. Todavia, o valor não é sumarizado de forma unitária. O valor exibido não é agregado de maneira única; em outras palavras, se houver interação de A para B e de B para A (o mesmo evento), ele será contabilizado duas vezes ou mais, dependendo do número de ocorrências. No entanto, idealmente, deveria ser considerado apenas uma vez. Por esse motivo, o programa mencionado anteriormente, denominado “countdots”, oferece uma visualização mais precisa e permite compreender o resultado real.

-Amount of predicted interactions-

```
-----
Genome name:      Ba_Ak
Number of ppi by cn: 2330
Number of ppi by pp: 5810
-----
Genome name:      Ba_Bp
Number of ppi by cn: 749
Number of ppi by pp: 1043
-----
Genome name:      Ba_G002
Number of ppi by cn: 3020
Number of ppi by pp: 6394
-----
Genome name:      Ba_Sg
Number of ppi by cn: 2253
Number of ppi by pp: 3489
-----
Genome name:      Ba_Ua
Number of ppi by cn: 2244
Number of ppi by pp: 4042
-----
```

Figura 12 – Exemplo da saída do arquivo gerado report.txt do diretório ppi-report.

Materiais e Métodos

Nesta seção, são descritos os materiais e métodos utilizados, com foco nas etapas de pré-processamento dos dados, bem como nos procedimentos de integração e validação do algoritmo no *Genppi*. O propósito central é incorporar a biblioteca Cl-Random-Forest no *Genppi* e avaliar sua eficácia na predição de redes de interação proteína-proteína (PPI) em genomas bacterianos. Para facilitar a compreensão da metodologia empregada, a Figura 13 ilustra em um fluxograma as etapas-chave da pesquisa. As fases descritas incluem desde a obtenção e preparação dos dados, passando pela incorporação do *Random Forest* no *Genppi*, até a avaliação do desempenho do modelo. Cada uma dessas etapas é abordada em detalhes nas seções seguintes deste e do próximo capítulo.

4.1 Coleta e Preparação de Dados

A coleta e preparação dos dados representam etapas cruciais para a confiabilidade das análises subsequentes no *Genppi*. Inicialmente, dados genômicos públicos são obtidos de fontes renomadas, como o *STRING*, uma base de dados que integra interações proteicas conhecidas e previstas, abrangendo diversas fontes de evidência. O *STRING* é amplamente reconhecido como um padrão ouro para validação de novas ferramentas de predição, facilitando análises comparativas entre diferentes espécies e oferecendo uma interface intuitiva para visualização de redes de interação.

A plataforma do *STRING* oferece uma interface intuitiva para a visualização de redes de interação, facilitando a identificação de módulos funcionais e a interpretação de contextos biológicos complexos. A constante atualização e a inclusão de novas funcionalidades fazem do *STRING* uma ferramenta valiosa para pesquisadores de bioinformática e biologia de sistemas, contribuindo significativamente para o avanço do conhecimento sobre interações proteicas e suas implicações na saúde e na doença (SZKLARCZYK et al., 2015).

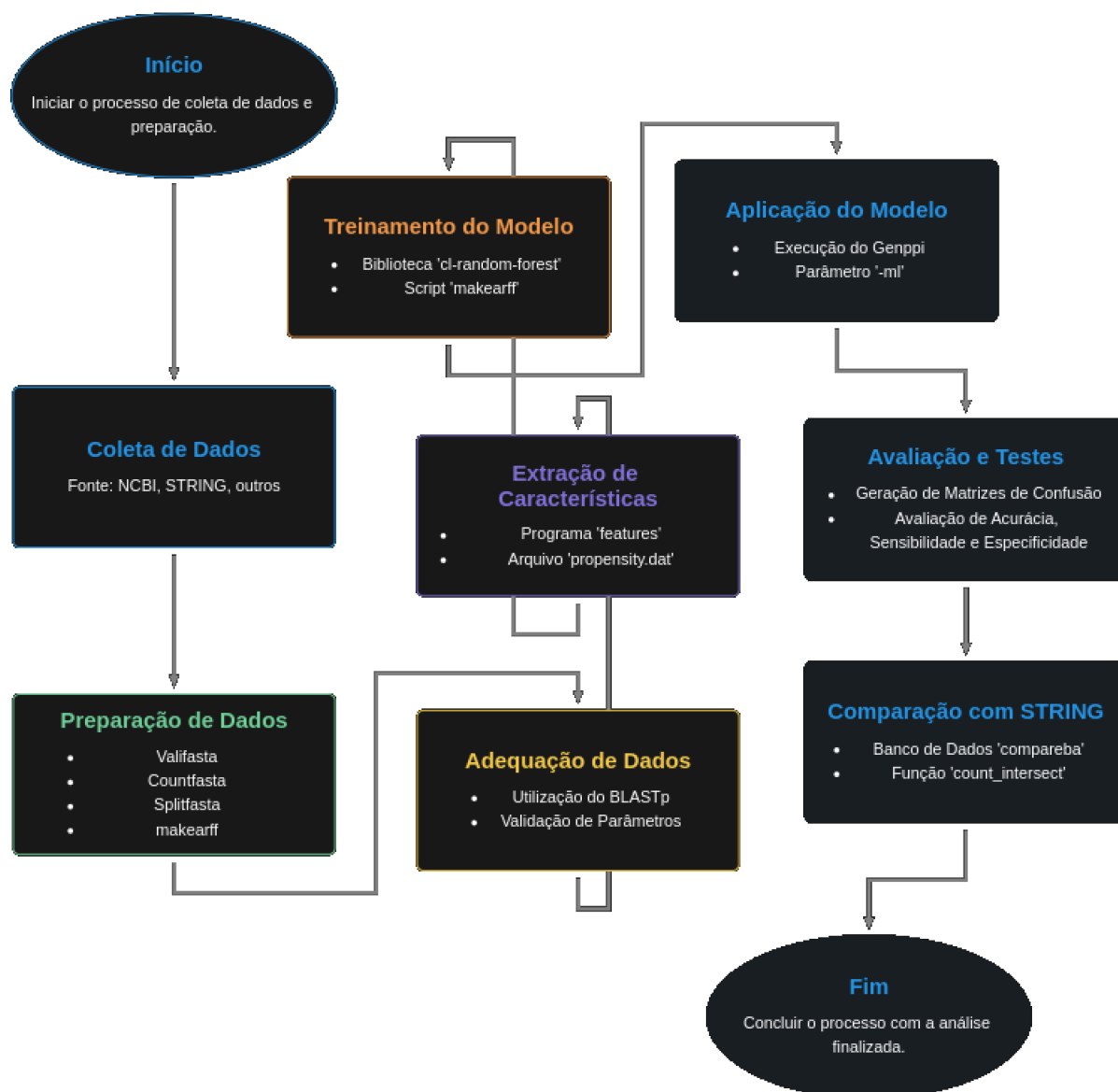


Figura 13 – Fluxograma da Metodologia

A preparação dos dados envolve vários procedimentos para garantir a consistência e organização necessárias para a análise. Um dos programas utilizados é o *Valifasta*, que assegura a remoção de caracteres não alfabéticos e padroniza os nomes das proteínas em arquivos multifasta, garantindo a consistência dos dados. Além disso, o *Countfasta* é empregado para quantificar o tamanho das proteínas, enquanto o *Splitfasta* divide as sequências proteicas em arquivos menores para análises específicas. Esses processos são fundamentais para a precisão e confiabilidade das análises subsequentes.

Outro elemento essencial na preparação dos dados é o uso da ferramenta *Blastp*, que permite a criação de bancos de dados com *makeblastdb*. Este passo é crucial para

a comparação e validação dos dados, assegurando que as sequências proteicas estejam adequadamente estruturadas para as análises. A seguir, os dados são processados utilizando o script *makearff*, que valida os parâmetros e estrutura os dados para o algoritmo *Random Forest*, essencial para a predição de interações proteína-proteína (PPI).

Nesta etapa, os dados preparados são adequados para uso com a biblioteca *CL Random Forest*, uma ferramenta desenvolvida para integrar o algoritmo *Random Forest* ao *Genppi*. Esta adequação permite uma análise mais detalhada das características das proteínas, aproveitando as propriedades específicas das sequências proteicas. A eficácia deste processo é garantida pela capacidade do *Random Forest* de lidar com dados complexos e de alta dimensionalidade, promovendo uma melhor generalização dos resultados.

A integração da *CL Random Forest* ao *Genppi* exige um pré-processamento meticuloso dos dados, assegurando que todas as características relevantes das proteínas sejam consideradas. Este processamento inclui a criação de arquivos ARFF compatíveis com o software Weka, amplamente utilizado em aprendizado de máquina. A geração desses arquivos facilita a aplicação eficiente de algoritmos nas análises proteicas, melhorando a precisão das predições e a robustez dos resultados.

Para assegurar a integridade dos dados e a eficácia dos modelos, é fundamental uma constante atualização e refinamento dos algoritmos utilizados. Este processo contínuo de melhoria permite que o *Genppi* mantenha-se atualizado com as mais recentes descobertas e técnicas de análise, proporcionando resultados cada vez mais precisos e confiáveis.

4.1.1 MakeArff

O script *makearff* desempenha um papel crucial na preparação e estruturação dos dados, facilitando análises de similaridade entre proteínas e a construção de conjuntos de dados para aplicação em técnicas de aprendizado de máquina. Desenvolvido como um script *BASH*, ele automatiza e simplifica a preparação de dados, integrando características relevantes para análises mais aprofundadas.

O processo inicia-se com a validação de parâmetros, assegurando a correta entrega de proteínas e características desejadas. Utilizando a ferramenta *blastp* para analisar a similaridade entre proteínas, o script cria eficientemente um banco de dados com *makeblastdb*. Após obter os resultados, realiza a concatenação de proteínas, a filtragem de interações e a geração de pares únicos, fundamentais para a criação de conjuntos de dados estruturados.

A relevância do *makearff* reside na habilidade de automatizar a preparação de dados, tornando as análises mais eficientes. Ao facilitar a identificação de interações proteicas significativas e a construção de conjuntos de dados estruturados, o script se torna ins-

trumental para a aplicação de algoritmos de aprendizado de máquina. Isto permite a detecção de padrões complexos nas interações proteicas, proporcionando uma abordagem mais eficaz na exploração das redes proteicas intrincadas subjacentes ao fenômeno biológico em estudo.

O *makearff* foi incorporado ao projeto *Genppi* e está disponível no repositório GitHub do projeto, sob a pasta ‘src/tools’. Sua aplicação é fundamental para a compreensão das complexas interações entre proteínas, um aspecto crucial na nossa pesquisa.

4.2 Utilização de Banco de Dados

Para realizar uma análise robusta das interações proteína-proteína (PPI), utilizamos dados provenientes do *STRING*, reconhecido como um padrão-ouro na análise de interações proteicas. Este banco de dados fornece informações detalhadas sobre interações conhecidas e previstas, integrando múltiplas fontes de evidência. Para facilitar a gestão e análise desses dados, empregamos o sistema gerenciador de banco de dados relacional PostgreSQL.

O banco de dados utilizado em nossa análise, disponível no repositório do *Genppi* no GitHub sob o nome “compareba.dump”, foi projetado especificamente para estudar o genoma *Buchnera Aphidicola*, identificado como *Ba_ak*. Este banco de dados possui duas tabelas principais: *protein* e *related*. A Tabela *protein* armazena os identificadores das proteínas e sua origem genômica, obtidos tanto da rede de interação do *STRING* quanto do *Genppi*. Já a Tabela *related* registra as interações entre proteínas do mesmo genoma, permitindo uma comparação eficaz dessas interações.

Para garantir a precisão das interações proteicas, desenvolvemos a função “*count_intersect_distinct*”. Esta função retorna uma tabela com o nome do genoma e o número de interações únicas, considerando as interações com e sem repetições. No contexto do *STRING*, onde a interação entre A e B é considerada diferente da interação entre B e A, a contagem de interações sem repetições é igual à metade do total. Em contrapartida, no *Genppi*, as interações duplicadas são eliminadas usando o programa *Gephi*, frequentemente utilizado para visualizar as redes de interação geradas pelo *Genppi* (BASTIAN; HEYMANN; JACOMY, 2009).

Por meio do script “*makearff*”, o processo inicia-se com a validação dos parâmetros, assegurando a correta entrega das proteínas e características desejadas, além de verificar a presença dos programas de bioinformática necessários. Utilizando a ferramenta *Blastp*, o script cria eficientemente no banco de dados com *makeblastdb*. Posteriormente, realiza a concatenação de proteínas, filtragem de interações e geração de pares únicos, fundamentais para conjuntos de dados estruturados.

A concatenação é vital porque o programa de aprendizado de máquina analisa sequências de proteínas em pares para determinar similaridades, conforme definido pelo algoritmo heurístico de alinhamento de sequências do *Blastp*. Todo este processo assegura que os dados estejam preparados e organizados para análises avançadas, proporcionando resultados confiáveis e precisos na predição de interações proteína-proteína.

4.3 Funcionalidades Agregadas ao Genppi

O desenvolvimento da versão 1.5 do *Genppi* trouxe diversas inovações tecnológicas, dentre as quais destacam-se o *Features.lisp* e a incorporação do algoritmo *Random Forest*. Ambas as adições visam aprimorar a capacidade de predição de interações proteína-proteína (PPI) e aumentar a eficiência do software no processamento de dados genômicos e proteômicos.

4.3.1 Features

O *Features.lisp* é um programa desenvolvido *in house* que utiliza o arquivo *propensity.dat* para mapear características específicas de proteínas. Este arquivo contém valores específicos para cada aminoácido, possibilitando a criação de um índice de propensão para cada característica de cada aminoácido presente em todas as proteínas de um arquivo multifasta. As características físico-químicas representadas no *propensity.dat* incluem acidez/basidade, polaridade, massa, entre outras, num total de dez características disponíveis na base de dados do site AAindex.

Cada linha do arquivo *propensity.dat* representa uma característica, enquanto cada coluna representa um aminoácido, com valores numéricos indicando a propensão de cada aminoácido para a referida característica. A seleção desses índices baseou-se em sua eficácia na classificação de proteínas conforme seus locais subcelulares mais comuns em organismos procarióticos.

O programa *Features* lê o arquivo FASTA contendo diversas sequências de aminoácidos e o arquivo de propensão, que inclui informações sobre as propriedades físico-químicas dos aminoácidos. Para cada sequência no arquivo FASTA, o *Features* calcula um histograma que conta a ocorrência de cada aminoácido na sequência e a soma ponderada das propriedades físico-químicas para a sequência completa, bem como para as regiões inicial, média e final da sequência.

Em termos de análise, o *Features* calcula o valor de cada característica ao longo de toda a proteína e segmenta a proteína em início, meio e fim, resultando em quatro valores para cada característica. Isto resulta em um total de quarenta valores por proteína. Além disso,

a contagem simples da quantidade de cada aminoácido adiciona mais vinte características numéricas a cada proteína, totalizando sessenta valores que definem o índice de propensão de uma proteína, permitindo uma análise granular das suas propriedades.

4.3.2 Integração da biblioteca Cl-Random-Forest

A integração do algoritmo *Random Forest* ao *Genppi* foi motivada pela necessidade de melhorar a predição de redes de interação proteína-proteína (PPI) em genomas bacterianos, superando as limitações da versão 1.0. O algoritmo *Random Forest* foi escolhido por sua capacidade de lidar com conjuntos de dados complexos e de alta dimensionalidade, comuns nas informações genômicas e proteômicas analisadas pelo *Genppi*.

O *Random Forest* constrói uma coleção de árvores de decisão, cada uma treinada com um subconjunto aleatório dos dados e variáveis disponíveis. Essa abordagem promove a diversidade entre as árvores, impedindo que características dominantes exerçam uma influência excessiva no modelo, o que resulta em melhor generalização. Cada árvore estabelece regras de decisão e forma uma estrutura semelhante a um fluxograma, contribuindo coletivamente para a tomada de decisões.

Para implementar essa técnica, optou-se pela biblioteca *Cl-Random-Forest*, que oferece suporte para classificação multiclasse e regressão univariada. Esta biblioteca foi escolhida pela sua robustez e eficiência, superando outras implementações populares como *scikit-learn* (Python/Cython) e *ranger* (R/C++). Além disso, a *Cl-Random-Forest* inclui um refinamento global, conforme descrito por (REN et al., 2015) apud (IMAI, 2017), tornando-a ainda mais rápida e precisa."

O código utilizado para treinar o modelo de *Random Forest* no *Genppi* 1.5 está disponível no repositório GitHub, sendo dependente das bibliotecas *Cl-Random-Forest* e *cl-store*, que são instaladas via Quicklisp. Diferentes conjuntos de treinamento e teste foram empregados para gerar o modelo final de predição de similaridades entre pares de proteínas.

Essas inovações tecnológicas visam aprimorar significativamente a eficácia do *Genppi* na predição de interações proteína-proteína, proporcionando uma ferramenta mais robusta e precisa para análises genômicas e proteômicas.

4.3.2.1 Pacote Cl-Random-Forest

Para a integração do pacote *Cl-Random-Forest* ao *Genppi*, o conhecimento das diversas funções que o algoritmo possui é necessário para o entendimento e posterior à implementação das árvores aleatórias. Adiante segue um conjunto dessas funções para a criação e operação de um modelo de *Random Forest* em *Common Lisp* do principal

arquivo do pacote. O texto seguinte descreve, de forma resumida as principais funções e suas funcionalidades, apresentando também pontos de destaque em listas para melhor compreensão.”

- **Estrutura Geral do Código:** O código inclui funções para a construção de árvores de decisão, manipulação de dados e execução do algoritmo de *Random Forest*. As funções mais relevantes podem ser agrupadas em categorias, como leitura de dados, construção de árvores, treino e avaliação do modelo.

- **Leitura e Manipulação de Dados:** O código contém funções dedicadas à importação e formatação dos dados que serão usados para treinar o modelo. Estas funções permitem que os dados sejam lidos de arquivos ou estruturas internas e preparados de acordo com as necessidades do algoritmo:

- ❑ **Funções de leitura de arquivo:** Responsáveis por carregar conjuntos de dados e convertê-los para listas ou matrizes utilizáveis.
- ❑ **Funções de formatação:** Realizam transformações nos dados, como a normalização ou extração de atributos.

- **Construção das Árvores de Decisão:** O cerne do código consiste em funções que constroem árvores de decisão, as quais compõem a floresta. Essas funções realizam as seguintes tarefas:

- ❑ **Divisão de nós:** Implementam a lógica para dividir os nós com base em critérios como entropia ou ganho de informação.
- ❑ **Criação recursiva de subárvores:** Constrói a árvore de forma recursiva até atingir uma condição de parada, como uma profundidade máxima ou um número mínimo de exemplos em um nó.

- **Treino e Predição:** Após a construção das árvores, o código inclui funções para o treino do modelo de *Random Forest* e a realização de predições.

- ❑ **Função de treino:** Coordena a criação de múltiplas árvores com subconjuntos aleatórios de dados para formar a floresta.
- ❑ **Função de predição:** Usa as árvores da floresta para realizar votações e determinar a classe ou valor final com base nas respostas das árvores individuais.

- **Avaliação de Desempenho:** Por fim, o código inclui métodos para avaliar a eficiência do modelo.

- ❑ **Cálculo de métricas:** Funções que medem a acurácia, precisão e outras métricas de avaliação.
- ❑ **Validação cruzada:** Implementa mecanismos de teste que dividem os dados em partes de treino e teste para fornecer estimativas mais robustas do desempenho.

As funções do código trabalham de forma integrada para possibilitar a construção, treino e avaliação de modelos de *Random Forest*. O código foi desenvolvido para ser modular, facilitando a reutilização e a extensão dos módulos existentes.

4.3.2.2 Random-forest.lisp

Para a utilização do pacote Cl-Random-Forest, um conjunto de procedimentos fazem necessários a fim de preparar o conjunto de dados de treinamento aos parâmetros exigidos pelo pacote, como por exemplo, é permitido a criação de árvore de decisão, de florestas aleatórias, previsões e testes, além de refinamento global caso se justifique. Ainda se pode realizar poda, paralelização, regressão, etc.

Random-forest.lisp é um arquivo incorporado para o treinamento do modelo utilizado no Genppi que visa atender aos parâmetros exigidos pelo pacote Cl-Random-Forest. Este arquivo criado possui diversas funções auxiliares para um modelo de floresta aleatória ("*Random Forest*") em *Common Lisp*. Adiante está um resumo descritivo das funções principais, organizadas por funcionalidade:

Pacotes e Configurações Iniciais: O código inicia com a definição de um pacote (`defpackage`) chamado `:random-forest`, que importa símbolos do pacote `:cl` e exporta várias funções principais. Ele também carrega pacotes adicionais (`Cl-Random-Forest` e `cl-store`), que são bibliotecas externas necessárias para a implementação do modelo de floresta aleatória e serialização de dados.

- Funções Utilitárias:

- ❑ **file-or-directory:** Verifica se o caminho fornecido é um diretório ou um arquivo regular. Retorna `:directory` se for um diretório, `:regular-file` se for um arquivo, ou `nil` caso contrário.
- ❑ **string-empty-p:** Confere se uma string está vazia ao verificar se seu comprimento é zero.
- ❑ **split-sequence:** Divide uma sequência de acordo com um delimitador especificado. Caso a opção `remove-empty-subseqs` seja verdadeira, as subsequências vazias resultantes são removidas.

- Funções para Manipulação de Dados:

- ❑ **read-arff-file**: usada para ler arquivos de dados no formato ARFF (Attribute-Relation File Format). Esta função permite a leitura de dados estruturados para treinamento e teste do modelo.

- Funções de Treinamento e Previsão:

- ❑ **train-predict**: Treina o modelo de floresta aleatória com um conjunto de dados de treinamento e, em seguida, realiza previsões com um conjunto de teste. Esta função encapsula o fluxo principal de treino e previsão, sendo a função principal para análise de dados.
- ❑ **predict**: Realiza previsões usando um modelo de floresta aleatória previamente treinado. Dado um novo conjunto de dados, a função prevê a classe ou valor para cada entrada.

- Serialização e Desserialização:

- ❑ **serialize-forest**: Serializa o modelo de floresta aleatória para que ele possa ser salvo em disco, permitindo a reutilização do modelo posteriormente sem precisar treiná-lo novamente.
- ❑ **deserialize-forest**: Lê um modelo de floresta aleatória previamente salvo e o reconstrói na memória para que possa ser utilizado novamente.

O conjunto de funções forma uma estrutura para construir, treinar, salvar e carregar modelos de floresta aleatória. As funções utilitárias ajudam nas operações de manipulação de strings e arquivos, enquanto as funções principais se concentram em manipulação de dados e execução do modelo.

4.3.3 Reduced Interaction Sampling (RIS)

Ao implementar técnicas de aprendizado de máquina, observamos um aumento significativo no número de interações identificadas em comparação com abordagens que não utilizam aprendizado de máquina. Esse crescimento trouxe desafios adicionais para a análise topológica da rede resultante. Para mitigar esse problema, foi desenvolvido o algoritmo denominado “*RIS*” (*Reduced Interaction Sampling*), incorporado ao código-fonte do *Genppi* 1.5 (Algoritmo 1).

A depender da versão do código fonte do *Genppi*, o *RIS* foi implementado com o nome de “*Split*”.

Para compreender a necessidade do algoritmo *RIS*, considere a geração de redes de interação. Suponha que um perfil filogenético composto por cinco genes conservados seja

encontrado em diversos genomas. Nesse caso, cada genoma onde esse perfil for identificado terá arestas criadas entre todos os cinco homólogos desses genes, resultando em seis arestas.

Ao analisar genomas de organismos evolutivamente próximos, a quantidade de genes em perfis filogenéticos conservados tende a aumentar significativamente. Por exemplo, é comum encontrar perfis filogenéticos conservados com mais de quinhentos genes em organismos com mais de dois mil genes. Isso poderia gerar até 124.251 arestas, com cada gene possuindo 499 conexões. Para perfis com setecentos genes conservados, o número de arestas pode alcançar 243.951, com cada gene conectado a 699 outros. Poucos perfis grandes podem resultar em redes com centenas de milhares de arestas ou até mais de um milhão, proporcional ao nível de similaridade entre os genomas.

Para controlar esse crescimento rápido no número de arestas e no grau médio dos vértices, desenvolvemos um processo que realiza uma amostragem significativa das interações em perfis filogenéticos grandes. A ideia é criar o máximo de interações possíveis sem incluir todas as arestas, demonstrando que um gene A interage com um gene B sem todas as conexões entre A e B. Denominamos essa abordagem de *RIS*, pois ela divide o grande conjunto de genes em várias porções menores. O *RIS* gera perfis menores equivalentes a 90% do tamanho do perfil original, com cada perfil derivado contendo uma quantidade menor de genes selecionados aleatoriamente.

Inicialmente, o *RIS* desordena a lista de proteínas similares, processo conhecido como *shuffle*. Em seguida, essa lista é subdividida em sub-listas correspondentes a 90% do número de elementos da lista original, com cada sub-lista recebendo aleatoriamente metade menos um elemento da lista desordenada. O programa é então executado separadamente para cada sub-lista, reduzindo o número de combinações possíveis e, conseqüentemente, o número total de arestas na rede. Essa divisão e execução em sub-grupos facilita a análise topológica da rede resultante.

Uma questão relevante ao adotar essa estratégia é o destino das proteínas que não formam arestas. Embora algumas interações possam não ser representadas, testes realizados demonstraram que, após a aplicação do *RIS*, a análise da rede de interações praticamente não se altera em termos de importância topológica dos nós. Ou seja, o *RIS* não apenas reduz o número de arestas na rede, facilitando sua análise, mas também mantém a qualidade dos dados, preservando a diferenciação dos nós em medidas de centralidade. Isso ocorre porque a diminuição do grau dos nós é homogênea devido à aleatoriedade, preservando o ranking dos nós mais conectados. Dessa forma, o *RIS* oferece uma solução eficaz para gerenciar o aumento das interações resultantes do aprendizado de máquina, mantendo a integridade e qualidade dos dados para análise subsequente da rede PPI.

Algoritmo 1 Pseudocódigo do Algoritmo RIS

Função replace-item(target, item, replacements)
 pos $\leftarrow$ posição de item em target
if pos **then**
 target $\leftarrow$ concatenação de (subsequência de target até pos), replacements, (subsequência de target a partir de pos+1)
end if
Fim Função

Função substitute-node(target, number, node)
 header, replacements $\leftarrow$ vazios
if target não vazio, node não vazio e number > 0 **then**
 if number < comprimento de target **then**
 header $\leftarrow$ primeiro elemento do número-th elemento de target
 for cada subnó em node **do**
 Adiciona (header, subnó, comprimento de subnó) à lista replacements
 end for
 else
 replace-item(target, número-th elemento de target, replacements)
 end if
end if
Fim Função

Função shuffle-list(target)
 copy $\leftarrow$ cópia de target
for i de comprimento de copy até 2 **do**
 Troca (random(i)th elemento de copy) com ((i-1)th elemento de copy)
end for
Retorna copy
Fim Função

Função split-list(protein-list, dividend, redundants)
 protein-list-shuffled $\leftarrow$ shuffle-list(protein-list)
 n $\leftarrow$ comprimento de protein-list-shuffled
 sublist-size, sublists, pivotal-sublist, non-pivotal-sublists $\leftarrow$ vazios
if dividend < n e redundants < n **then**
 sublist-size $\leftarrow$ arredonda(n / dividend)
 sublists $\leftarrow$ subdivisão de protein-list-shuffled em sublists de tamanho sublist-size
 dividend $\leftarrow$ comprimento de sublists
end if
for cada índice pivotal de 0 até dividend-1 **do**
 pivotal-sublist $\leftarrow$ pivotal-th elemento de sublists
 non-pivotal-sublists $\leftarrow$ sublists sem pivotal-sublist
 for cada redundante **do**
 if existem elementos para escolher de non-pivotal-sublists **then**
 índice do sublist de origem $\leftarrow$ índice aleatório de non-pivotal-sublists
 sublist de origem $\leftarrow$ índice do sublist de origem
 índice do elemento $\leftarrow$ índice aleatório do sublist de origem
 elemento $\leftarrow$ índice do elemento
 if o elemento não está na pivotal-sublist **then**
 adiciona elemento à pivotal-sublist
 remove elemento do sublist de origem
 end if
 end if
 end for
 atualiza o pivotal-th elemento de sublists para pivotal-sublist
end for
Retorna sublists
Fim Função

A implementação do algoritmo *RIS* demandou uma avaliação rigorosa de seu impacto na preservação das propriedades topológicas das redes de interação proteína-proteína. Para isso, foi conduzida uma investigação independente do *Genppi* em ambos os modos operacionais (*Features* e *Machine Learning*), gerando redes distintas devido à aleatorização inerente ao algoritmo. A análise sistemática demonstrou que, embora a redução combinatória introduza variabilidade local, a hierarquia global dos nós críticos (como *hubs* e proteínas com alta *betweenness*) mantém-se estável em cenários de amostragem média (em torno 17,89 % do genoma, com um $Top_N = 100$) ou superior. No modo *Features*, observou-se consistência quase absoluta, com a Média Global de presença ponderada (MGP) atingindo 99,76% a 100% em praticamente todos os cenários testados, indicando imunidade à aleatoriedade em abordagens baseadas em características bioquímicas. Já no modo *Machine Learning*, a MGP variou de 13,36% ($Top_N = 10$, representando 1,79% do genoma) a 74,18% ($Top_N = 200$, representando 35,78% do genoma), revelando que a redundância estrutural compensa parcialmente a fragmentação de perfis filogenéticos. Testes estatísticos, como o da distância de *Kolmogorov-Smirnov* para múltiplas amostras, confirmaram a preservação das distribuições de grau mesmo com redução drástica de arestas.

Estes resultados, detalhados no Apêndice A, fundamentam a adoção do *RIS* como estratégia viável para equilibrar eficiência computacional e integridade topológica, desde que utilizados limiares empíricos adequados.

Experimentos e Resultados

5.1 Experimentos

Os experimentos conduzidos para avaliar a eficácia do *Random Forest* e sua integração ao *Genppi* foram estruturados em três etapas principais. Na primeira etapa, foi avaliada a eficácia do *Random Forest* na classificação de pares de proteínas com diferentes níveis de similaridade. Para isso, foram realizados testes de controle positivo e negativo, definindo critérios para identificar pares de proteínas com alta ou baixa similaridade com base em informações biológicas conhecidas. Essa avaliação buscou validar a capacidade do modelo em distinguir pares similares, proporcionando dados consistentes que poderiam ser utilizados pelo *Genppi* para uma análise mais ampla.

Na segunda etapa, o foco foi avaliar o impacto do algoritmo *RIS* na topologia estrutural das redes geradas pelo *Genppi*. Para isso, foram realizadas rodadas independentes do software em cada modo operacional (*Features* e *Machine Learning*), analisando-se a estabilidade de métricas topológicas críticas (como grau, *betweenness* e coeficiente de clustering) em múltiplos cenários de amostragem (*Top_N* entre 10 e 200 nós). Utilizando técnicas estatísticas multivariadas e medidas de concordância hierárquica (ex.: Correlação de *Kendall Tau*), demonstrou-se que o *RIS* preserva a hierarquia global dos nós mesmo com redução significativa de arestas, embora limiares muito restritos ($Top_N < 50$, que representa 8,95% do genoma) amplifiquem inconsistências locais. Os detalhes metodológicos e resultados brutos estão consolidados no Apêndice A.

Na terceira etapa, foi realizada uma análise focada no desempenho do *Genppi*, agora aprimorado pela integração do *Random Forest*. Essa análise incluiu estudos comparativos com outras ferramentas descritas na literatura, como o *STRING*, para avaliação de redes de interação proteína-proteína. Utilizando métricas de desempenho e validação cruzada, foi possível verificar a capacidade do *Genppi* de avaliar interações com maior abrangência

e precisão. Ferramentas como análise de matriz de confusão foram empregadas para validar os resultados obtidos, demonstrando que a integração do *Random Forest* ao *Genppi* ampliou significativamente a eficiência e a escalabilidade do software na análise de redes de interação proteína-proteína.

5.1.1 Genomas Utilizados para Treinamento e Testes

Genomas de várias espécies bacterianas patogênicas disponibilizados pelo National Center for Biotechnology Information (NCBI) foram selecionadas devido à sua ampla base de dados. As espécies escolhidas incluem *Aeromonas hydrophila*, *Bacillus anthracis*, *Bacteroides fragilis*, *Borrelia burgdorferi*, *Clostridium botulinum*, *Clostridium tetani*, *Corynebacterium Jeikeium*, *Enterobacter huaxiensis*, *Enterococcus faecalis*, *Escherichia coli*, *Francisella tularensis*, *Helicobacter pylori*, *Moraxella Catarrhalis*, *Morganella morganii*, *Mycobacterium Avium*, *Mycobacterium tuberculosis*, *Providencia rettgeri*, *Pseudomonas aeruginosa*, *Salmonella enterica*, *Staphylococcus aureus*, *Streptococcus pyogenes*, *Vibrio cholerae* e *Yersinia pestis*. A escolha dessas espécies foi baseada na sua relevância como patógenos e na vasta literatura científica existente sobre elas.

O uso de dados de organismos modelos, como os encontrados no NCBI, oferece uma base sólida para investigações futuras, permitindo a confirmação de hipóteses e conclusões derivadas do nosso estudo. Ademais, a disponibilidade desses genomas possibilita a criação de conjuntos de dados representativos, bem como a realização de análises comparativas, o que contribui para a robustez e validade dos resultados de nossos experimentos.

5.1.2 Desenvolvendo o Modelo de Aprendizado de Máquina

Para a criação do arquivo de treinamento, selecionamos proteínas de nove genomas de importância médica: *Bacillus anthracis*, *Clostridium tetani*, *Enterococcus faecalis*, *Escherichia coli*, *Francisella tularensis*, *Mycobacterium tuberculosis*, *Salmonella enterica*, *Staphylococcus aureus* e *Vibrio cholerae*. Utilizamos o script *makearff* para realizar uma série de manipulações em arquivos de texto contendo informações genômicas, aplicando filtros e convertendo-os para o formato ARFF, indispensável para o treinamento do algoritmo. Este script possibilitou a comparação dos nove genomas, gerando um arquivo com proteínas identificadas por suas características e classificadas como positivas ou negativas.

Inicialmente, o conjunto de dados apresentava um desequilíbrio significativo entre proteínas positivas (8.858) e negativas (318.309), o que dificultava o treinamento dos modelos de classificação. A aplicação inicial do algoritmo *Random Forest* não obteve resultados satisfatórios devido a essa discrepância. Para solucionar esse problema, realizamos uma

série de testes para determinar a proporção ideal entre proteínas positivas e negativas. Diversas estratégias, como seleção aleatória e ordenação por início, meio ou fim dos dados, foram testadas. Após análises e experimentações, identificamos a proporção de 2,6:1 (positivas:negativas) como a mais adequada.

A execução do programa *Random Forest* foi feita através de chamadas específicas na linha de comando do sistema operacional. O comando utilizado foi: `./randomforest similar_test.arff xxxx similar_training.arff`, onde o primeiro parâmetro é o nome do programa, o segundo parâmetro (*similar_test.arff*) é o arquivo de teste com as proteínas do genoma a ser avaliado, e o terceiro parâmetro (xxxx) indica a quantidade de proteínas negativas presentes no arquivo de teste em formato de dígitos. É importante ressaltar que o arquivo ARFF é cuidadosamente estruturado, de forma que todas as proteínas negativas precedem as positivas. Essa ordenação é essencial, pois rótulos verdadeiros para cada instância de dado são utilizados no treinamento. O programa *Random Forest*, implementado em Lisp, utiliza essas informações para calcular a precisão do modelo, contabilizando acertos e erros. O último parâmetro representa o arquivo de treinamento, que, assim como o arquivo de teste, segue o padrão de instâncias negativas precedendo as positivas. Essa conformidade permite que os arquivos de treinamento e teste sejam intercambiáveis, proporcionando flexibilidade na avaliação e validação do modelo de aprendizado de máquina.

5.1.3 Aplicando o Modelo Treinado no Genppi

Inicialmente, o *Genppi* utilizava a contagem simples das ocorrências dos vinte e seis aminoácidos (incluindo seis não padronizados) como método de análise. Na versão aprimorada, que incorpora sessenta características geradas pelo programa *Features*, o modo padrão de execução, denominado modo *Features*, considera duas proteínas como similares se apresentarem pelo menos vinte e cinco dessas características com uma diferença máxima de uma ou duas unidades. Este valor foi determinado empiricamente, e sua alteração pode afetar a precisão das inferências de similaridades. Se este número fosse significativamente aumentado, por exemplo, próximo das sessenta características usadas pelo *Genppi* 1.5, a versão 1.5 no modo *Features* operaria de maneira semelhante ao *Genppi* 1.0, que utilizava vinte e seis possibilidades de aminoácidos.

As mesmas sessenta características usadas pelo algoritmo *Features* também são aplicadas no programa *Cl-Random-Forest*, que realiza aprendizado de máquina para criar redes mais complexas de proteínas. O modo de aprendizado de máquina do *Genppi* é ativado pelo parâmetro '-ml', sendo este opcional. Para utilizar o modo de aprendizado de máquina, é necessário baixar o modelo de *Random Forest* previamente treinado para reconhecer

pares de proteínas similares, com mais de 65% de identidade e tamanho de alinhamentos conforme determinado pelo *Blastp*.

Na versão 1.5 do *Genppi*, alguns parâmetros foram padronizados para simplificar a experiência do usuário. O parâmetro *-ppcomplete*, utilizado para predições de PPIs pelo método de perfil filogenético conservado sem aplicação de filtros, agora é ativado por padrão. Isso implica que o algoritmo realizará predições sem a necessidade de reduzir o número de interações preditas.

Adicionalmente, o tipo de expansão da vizinhança de um gene recorrente, antes controlado pelo parâmetro *-expt*, agora está fixo (*fixed*). O tamanho da janela principal (*-w1*) e a quantidade necessária de genes conservados dentro da janela 1 (*-cw1*), previamente definidos como dez e quatro, respectivamente, também são agora valores padrão. Dessa forma, na ausência de parâmetros extras, o *Genppi* 1.5 executará com essas configurações padrão, simplificando o processo e permitindo que os usuários se concentrem mais na interpretação dos resultados obtidos.

Para realizar testes no genoma da *Buchnera Aphidicola*, uma vez clonado o repositório de exemplo, o programa pode ser executado com o comando (no sistema operacional linux): `./genppi -dir test1`. O primeiro parâmetro representa a versão do programa *Genppi* selecionada para a máquina, com diversas compilações disponíveis para diferentes limites de uso de memória. O segundo parâmetro (*'-dir'*) especifica o nome do diretório onde os genomas estão armazenados como arquivos multifasta (um arquivo por genoma), e o último parâmetro (*'test1'*) é o nome do diretório em questão.

Para executar o programa utilizando aprendizado de máquina, utiliza-se o comando: `./genppi -dir test1 -ml`. Os parâmetros são os mesmos, exceto pelo último (*'-ml'*), que instrui o programa a empregar o aprendizado de máquina para identificar proteínas similares.

5.1.4 Avaliação da Estabilidade Topológica sob Aleatorização

A segunda etapa dos experimentos concentrou-se em investigar se a aleatorização introduzida pelo algoritmo *RIS* compromete a integridade topológica das redes de interação. Para isso, executou-se o *Genppi* em 50 rodadas independentes para cada modo operacional (*Features* e *Machine Learning*), gerando redes distintas a cada execução devido ao processo estocástico de fragmentação de perfis filogenéticos. Cada rodada produziu métricas topológicas completas para todos os 559 nós da rede de *Buchnera Aphidicola*, permitindo análises comparativas em vários cenários de *Top_N*, variando entre 10 (1,79% do genoma) a 200 (35,78%), abrangendo desde núcleos mínimos até subconjuntos amplos. Ainda houve a avaliação em diferentes conjuntos de arquivos (10, 20, 30, 40, 50).

A avaliação combinou medidas de estabilidade global e concordância hierárquica. A Média Global de presença ponderada (MGP) quantificou a persistência de nós críticos entre as rodadas, enquanto o Coeficiente de *Jaccard* mediu a sobreposição de conjuntos de nós. Adicionalmente, a Correlação de *Kendall Tau* avaliou a consistência na ordenação hierárquica dos nós por medidas de centralidade. Complementarmente, o Teste de *Kolmogorov-Smirnov* para múltiplas amostras foi aplicado para verificar a homogeneidade distribucional das métricas de grau.

Os resultados revelaram evidências robustas de que a redupadrões distintos entre os modos operacionais. No modo *Features*, observou-se concordância quase perfeita (MGP = 99,76%), com todos os nós presentes em todas as rodadas. Em contraste, no modo *Machine Learning*, a MGP variou significativamente com o *Top_N*, atingindo 74,18% para *Top_N* = 200, o que sugere a existência de um núcleo hierárquico estável mesmo em redes reduzidas. Entretanto, para *Top_N* = 10, a MGP caiu para 13,36%, evidenciando sensibilidade crítica à fragmentação em amostras restritas, com baixa amostragem.

Estes achados, complementados por análises distribucionais detalhadas no Apêndice A, sustentam a conclusão de que o *RIS* é adequado para preservar propriedades topológicas globais quando utilizado com parâmetros conservadores e adequados. A variabilidade residual observada em escalas locais não invalida a utilidade do algoritmo, mas ressalta a importância de critérios empíricos para seleção de limiares em estudos biológicos específicos.

5.2 Resultados

Foi realizada uma análise detalhada da integração do algoritmo *Random Forest* no *Genppi*. Esta avaliação focou na análise do impacto da inserção do *Random Forest* na capacidade de predição de redes de interação proteína-proteína, bem como na avaliação do desempenho e precisão do software com a nova incorporação. A análise crítica dos resultados incluiu a identificação de tendências, padrões e limitações que poderiam influenciar a interpretação dos resultados e a aplicabilidade prática do algoritmo.

Após a avaliação dos resultados, foram propostos refinamentos iterativos no algoritmo *Random Forest*. Estes refinamentos foram testados e avaliados por meio de experimentos adicionais, visando melhorar ainda mais o desempenho e a precisão do algoritmo na predição de redes de interação proteína-proteína em genomas bacterianos. Este processo de avaliação iterativa permitiu o desenvolvimento de uma ferramenta mais robusta e precisa para análises genômicas e proteômicas.

Finalmente, a metodologia empregada também incluiu uma avaliação rigorosa dos modelos de aprendizado de máquina, utilizando métricas como acurácia, sensibilidade e especificidade para validar as predições realizadas. Estes indicadores foram essenciais

para garantir a confiabilidade das redes de interação proteica geradas, permitindo uma comparação direta com os dados do *STRING*. Todos esses procedimentos foram fundamentais para assegurar a eficácia da nova integração e sua capacidade de fornecer resultados precisos e confiáveis.

5.2.1 Avaliação de Desempenho do Modelo de Classificação

Na Tabela 1 são apresentados todos os diferentes genomas ou pares de genomas empregados para a avaliação dos resultados do modelo de classificação elaborado com o algoritmo Cl-Random-Forest. Utilizando o *Blastp*, foram selecionados conjuntos de resultados considerados similares (positivos) ou não similares (negativos), com base em critérios de identidade proteica e tamanho de alinhamentos superiores a 65%, valor empiricamente selecionado como corte para definir similaridade. Esses conjuntos de dados constituíram arquivos ARFF de teste para o classificador implementado com o algoritmo *Random Forest*.

Tabela 1 – Genomas empregados e resultados da classificação da Random Forest

Genomes	Verdadeiro Negativo	Falso Negativo	Verdadeiro Positivo	Falso Positivo	Total Positivo	Total Negativo	Total
Aeromonas Hydrophila	7279	1086	65	23	88	8365	8453
Aeromonas Hydrophila - Bacteroides Fragilis	19379	6103	206	62	268	25482	25750
Aeromonas Hydrophila - Clostridium Botulinum	26387	3631	163	58	221	30018	30239
Aeromonas Hydrophila - Enterobacter Huaxiensis	34136	3631	163	58	944	40248	41192
Bacteroides Fragilis	5988	3417	128	35	163	9405	9568
Bacteroides Fragilis - Clostridium Botulinum	20065	6101	212	58	270	26166	26436
Clostridium Botulinum	6403	1987	79	21	100	8390	8490
Enterobacter Huaxiensis	10736	2160	68	9	77	12896	12973
Enterobacter Huaxiensis - Clostridium Botulinum	24692	6336	213	47	260	31028	31288
Enterobacter Huaxiensis - Bacteroides Fragilis	29796	7908	170	31	201	37704	37905
Morganella Morganii	3834	991	125	3	128	4825	4953
Providencia Rettgeri	4652	1394	51	7	58	6046	6104
Yersinia Pestis	6831	879	150	26	176	7710	7886
Streptococcus Pyogenes	1562	156	17	2	19	1718	1737
Mycobacterium Avium	4631	10027	170	7	177	14658	14835
Mycobacterium Avium - Corynebacterium Jeikeium	9577	13181	465	37	502	22758	23260
Mycobacterium Avium - Moraxella Catarrhalis	7605	10552	208	36	244	18157	18401
Corynebacterium Jeikeium	1201	585	45	8	53	1786	1839
Corynebacterium Jeikeium - Moraxella Catarrhalis	3422	1121	83	37	120	4543	4663
Moraxella Catarrhalis	521	92	35	29	64	613	677
Helicobacter Pylori	724	128	9	12	21	852	873
Helicobacter Pylori - Borrelia Burgdorferi	1522	854	400	69	469	2376	2845
Helicobacter Pylori - Pseudomonas Aeruginosa	15727	6307	134	35	169	22034	22203
Pseudomonas Aeruginosa	12646	5346	113	23	136	17992	18128
Pseudomonas Aeruginosa - Borrelia Burgdorferi	14911	6035	503	80	583	20946	21529
Borrelia Burgdorferi	241	594	389	56	445	835	1280
Média	10556	3869	168	33	229	1452	

Após a aplicação do algoritmo Cl-Random-Forest sobre os dados coletados, foram geradas matrizes de confusão para avaliar a eficácia do modelo de classificação em relação a cada genoma, representadas pela contagem de erros e acertos em relação às classes

“negativo” e “positivo”, essenciais para calcular as medidas de avaliação do algoritmo *Random Forest* treinado.

Foram calculadas medidas de avaliação, incluindo a acurácia, sensibilidade e especificidade para cada genoma avaliado. Os resultados dessas métricas estão na Tabela 2.

Tabela 2 – Avaliação de Desempenho do Modelo de Classificação em Diversos Genomas

Genomes	Sensibilidade	Especificidade	Acurácia
Aeromonas Hydrophila	0,74	0,87	0,87
Aeromonas Hydrophila - Bacteroides Fragilis	0,77	0,76	0,76
Aeromonas Hydrophila - Clostridium Botulinum	0,74	0,88	0,88
Aeromonas Hydrophila - Enterobacter Huaxiensis	0,88	0,85	0,85
Bacteroides Fragilis	0,79	0,64	0,64
Bacteroides Fragilis - Clostridium Botulinum	0,79	0,77	0,77
Clostridium Botulinum	0,79	0,76	0,76
Enterobacter Huaxiensis	0,88	0,83	0,83
Enterobacter Huaxiensis - Clostridium Botulinum	0,82	0,80	0,80
Enterobacter Huaxiensis - Bacteroides Fragilis	0,85	0,79	0,79
Morganella Morganii	0,98	0,79	0,80
Providencia Rettgeri	0,88	0,77	0,77
Yersinia Pestis	0,85	0,89	0,89
Streptococcus Pyogenes	0,89	0,91	0,91
Mycobacterium Avium	0,96	0,32	0,32
Mycobacterium Avium - Corynebacterium Jeikeium	0,93	0,42	0,43
Mycobacterium Avium - Moraxella Catarrhalis	0,85	0,42	0,42
Corynebacterium Jeikeium	0,85	0,67	0,68
Corynebacterium Jeikeium - Moraxella Catarrhalis	0,69	0,75	0,75
Moraxella Catarrhalis	0,55	0,85	0,82
Helicobacter Pylori	0,43	0,85	0,84
Helicobacter Pylori - Borrelia Burgdorferi	0,85	0,64	0,68
Helicobacter Pylori - Pseudomonas Aeruginosa	0,79	0,71	0,71
Pseudomonas Aeruginosa	0,83	0,70	0,70
Pseudomonas Aeruginosa - Borrelia Burgdorferi	0,86	0,71	0,72
Borrelia Burgdorferi	0,87	0,29	0,49
Média	0,81	0,72	0,73

A análise dos dados revela uma média de sensibilidade de 0,81, especificidade de 0,72 e acurácia de 0,73 para os 26 genomas avaliados. Isso indica que o método é bastante sensível, sendo capaz de identificar corretamente uma alta proporção de similaridades verdadeiras. As similaridades falso positivas em maior quantidade não são necessariamente um problema porque o *Genppi* ainda analisará as características de vizinhança gênica e perfil filogenético conservados para definir por uma interação. Dificilmente duas proteínas falso positivas para similaridades possuirão estes tipos de conservação.

Ao observar os dados individuais, notamos que alguns genomas, como o *Morganella Morganii* e o *Mycobacterium Avium*, apresentam sensibilidade excepcionalmente alta (0,98 e 0,96, respectivamente), mas a especificidade e a acurácia são relativamente mais baixas. Genomas como o *Yersinia Pestis* e o *Streptococcus Pyogenes* apresentam alta acurácia (0,89 e 0,91, respectivamente), indicando um equilíbrio entre sensibilidade e especificidade.

Essas variações destacam a importância de considerar o contexto genômico específico ao interpretar os resultados das predições de similaridades proteína-proteína. Essa abordagem sistemática de avaliação permite uma compreensão abrangente do desempenho do modelo de classificação em relação aos diferentes genomas analisados, contribuindo para uma interpretação mais precisa dos resultados obtidos.

5.2.2 Comparação de Resultados: *Genppi* v1.0 vs. *Genppi* v1.5

A análise dos resultados obtidos pelo *Genppi* nas versões 1.0 comparada com a incorporação das funcionalidades “*Features*” e “Machine Learning” no *Genppi* 1.5, revela uma significativa melhoria na predição de redes de interação de proteínas. Observamos uma tendência consistente de aumento no número de nós e arestas, indicando uma maior complexidade e abrangência das redes geradas após as implementações.

Inicialmente, na versão 1.0 do *Genppi*, os genomas de *Buchnera Aphidicola* resultaram em um número relativamente baixo de nós e arestas, conforme evidenciado nos dados da Tabela 3. No entanto, após a introdução das funcionalidades “*Features*” e “Machine Learning”, houve um notável aumento tanto no número de nós quanto de arestas em todas as amostras analisadas. Essa expansão das redes de interação de proteínas sugere uma maior capacidade do software aprimorado em identificar e prever relações entre as proteínas, proporcionando uma visão mais abrangente e detalhada dos processos biológicos subjacentes.

Tabela 3 – Comparação do Desempenho do *Genppi* com *Features* e Machine Learning em Diferentes Conjuntos de Dados

File	Número Proteínas Genoma	Genppi 1		Genppi 1.5			
				Features		Machine Learning	
		<i>Nodes</i>	<i>Edges</i>	<i>Nodes</i>	<i>Edges</i>	<i>Nodes</i>	<i>Edges</i>
Ba_Bp.dot	553	46	91	217	1746	539	30220
Ba_Ua.dot	591	111	314	450	6115	581	40049
Ba_G002.dot	621	120	395	529	9166	617	41969
Ba_Sg.dot	620	88	203	430	5583	614	41483
Ba_Ak.dot	559	119	339	448	7917	559	44005
Summary	2944	484	1342	2074	30527	2910	197726

Os resultados também indicam uma tendência de crescimento na complexidade das redes geradas, refletida no aumento significativo no número total de nós e arestas. Em suma, os resultados comparativos entre as diferentes versões do *Genppi* demonstram os benefícios da utilização de recursos adicionais na predição de redes de interação de proteínas.

5.2.3 Comparação com o STRING

Os dados apresentados a seguir referem-se ao genoma *Ba_ak* da bactéria *Buchnera Aphidicola* e foram obtidos através de uma consulta sql ao banco de dados *compareba.dump*. O arquivo SQL “*count_intersect_distinct.sql*” contém a definição de uma função chamada “*count_intersect*”. Esta função retorna uma tabela com informações sobre a interseção de diferentes conjuntos de dados genômicos. O código da função envolve a identificação de elementos únicos e interseções entre esses elementos, calculando métricas como a proporção de interseções únicas e a proporção de interações entre elementos. Após a definição da função, há uma instrução SELECT que chama a função “*count_intersect*” e retorna os resultados da consulta.

O perfil filogenético conservado infere interações para pares de proteínas que possuem perfis filogenéticos idênticos. É razoável considerar que pares de proteínas com perfis semelhantes também podem ser interagentes. Para incorporar essa noção de similaridade em vez de identidade exata, o parâmetro *-ppdiff tolerated* no *Genppi* estabelece a diferença tolerada entre os perfis filogenéticos. A flexibilidade proporcionada pelo parâmetro *-ppdiff tolerated* permite uma análise mais abrangente das interações proteína-proteína, levando em consideração a similaridade dos perfis filogenéticos (ANJOS, 2021).

A Tabela 4 apresenta os resultados de diferentes implementações do *Genppi*. A nomenclatura adotada reflete os diferentes parâmetros e métodos utilizados. O termo **Features** indica o modo operacional que utiliza exclusivamente as sessenta características descritas na Seção 4.3.1, sem emprego de técnicas de *machine learning*. Neste modo, um par de proteínas é considerado idêntico se compartilhar pelo menos vinte e cinco recursos idênticos. Por outro lado, o termo **ml** refere-se ao modo com *machine learning*, acionado automaticamente quando o critério de identidade do modo *Features* não é atendido.

O sufixo **tol** está vinculado ao parâmetro *-ppdiff tolerated*, que define a diferença máxima aceitável entre perfis filogenéticos para inferir interações. Já o termo **split** está associado ao algoritmo de divisão de perfis filogenéticos (*RIS*), acionado quando um perfil contém mais de N proteínas (e.g., *mlsplit35* utiliza N=35). A implementação padrão (*mlsplit*) adota N=100, equilibrando sensibilidade e complexidade computacional. Esta estrutura de nomenclatura permite identificar rapidamente as combinações de parâmetros, como em *mlsplit35tol2*, que combina N=35 com tolerância intermediária (*tol2*).

Cada linha da tabela representa uma implementação específica, com métricas como *query_nounique*, *query_unique*, *string_intersect_unique*, *unique_intersect_ratio* e *string_proportion*. O objetivo principal é comparar as interações proteicas previstas pelo *Genppi* com as documentadas no banco de dados *STRING*, considerado um padrão ouro para interações proteína-proteína.

Tabela 4 – Resultado da execução da consulta ao banco compareba.dump em diferentes implementações e parâmetros.

Genome	query__nounique	query__unique	string__intersect__unique	unique__intersect__ratio	string__proportion
<i>string</i>	54082	27041	54082	1.0	1.00
<i>Features</i>	7918	7917	2773	2.86	0.10
<i>mlsplit</i>	92137	49077	16681	2.94	0.62
<i>featurestol1</i>	19247	19244	5380	3.58	0.20
<i>mlsplittol1</i>	150789	103743	26924	3.85	1.00
<i>featurestol2</i>	33582	33579	8021	4.19	0.30
<i>mlsplittol2</i>	170849	128579	29846	4.31	1.10
<i>featurestol3</i>	40465	40461	9232	4.38	0.34
<i>mlsplittol3</i>	187547	144468	32278	4.48	1.19
<i>mlsplit25</i>	47854	46634	9509	4.90	0.35
<i>mlsplit10</i>	45430	44216	8844	5.00	0.33
<i>mlsplit35</i>	48196	47103	9335	5.05	0.35
<i>mlsplit35tol1</i>	81182	80055	14573	5.49	0.54
<i>mlsplit35tol3</i>	136538	135617	23840	5.69	0.88
<i>mlsplit35tol2</i>	116101	114998	20034	5.74	0.74

A Tabela contém as seguintes colunas e métricas:

- ❑ *query__nounique*: Número de interações que se repetem na implementação.
- ❑ *query__unique*: Número de interações únicas na implementação.
- ❑ *string__intersect__unique*: Número de interações únicas que coincidem com as documentadas no *STRING*. Essa métrica é crucial para validar a precisão das predições do *Genppi* em relação ao padrão ouro.
- ❑ *unique__intersect__ratio*: Razão entre o número de interações únicas na implementação e o número de interações que coincidem com o *STRING*. Um valor maior que 1 indica que a implementação está prevendo mais interações do que as documentadas no *STRING*. Valores próximos a 1 sugerem alta precisão, mas podem indicar baixa sensibilidade, enquanto valores mais altos aumentam a sensibilidade, mas também o risco de falsos positivos. O valor ideal depende do contexto: um equilíbrio entre sensibilidade e precisão é crucial.
- ❑ *string__proportion*: Proporção de interações no *Genppi* que também estão presentes no *STRING*. Valores superiores a 100% podem indicar superpredição ou duplicação de interações, sugerindo a necessidade de validação experimental.

O `unique_intersect_ratio` é uma métrica crucial para entender a eficácia das diferentes implementações do *Genppi*. Ele representa o número de interações previstas pelo *Genppi* dividido pelo número de interações que coincidem com o *STRING*. Um valor maior que 1 indica que a implementação está prevendo mais interações do que as documentadas no *STRING*, enquanto um valor próximo a 1 sugere alta precisão. Implementações como *mlsplit* (`unique_intersect_ratio` = 2,94) apresentam um equilíbrio razoável entre sensibilidade e precisão, com 62% das interações coincidindo com o *STRING*. No entanto, valores elevados, como em *mlsplittol3* (`unique_intersect_ratio` = 4,48) e *mlsplit35tol3* (`unique_intersect_ratio` = 5,69), podem aumentar o risco de falsos positivos.

A medida que o valor de “*tol*” aumenta (de “*tol1*” para “*tol3*”), o `unique_intersect_ratio` também aumenta. Isso sugere que permitir uma maior diferença entre os perfis filogenéticos pode resultar em uma maior concordância com o *STRING*. Da mesma forma, aumentar o número de proteínas em um perfil filogenético (de “*mlsplit10*” para “*mlsplit35*”) também parece aumentar o `unique_intersect_ratio`. No entanto, é importante notar que `unique_intersect_ratio` maior não indica necessariamente uma melhor predição de interações proteína-proteína. Embora uma maior concordância com o *STRING* possa sugerir uma maior precisão, também é possível que algumas interações válidas sejam perdidas. Portanto, é crucial considerar o equilíbrio entre a sensibilidade e a especificidade ao avaliar a eficácia de diferentes implementações do *Genppi*.

As interações previstas pelo *Genppi* que não estão presentes no banco de dados *STRING* não devem ser descartadas como falsos positivos. Essas interações podem representar descobertas potenciais, especialmente quando consideramos as características de vizinhança gênica e perfil filogenético conservados, que são corretamente implementados e validados entre as versões do *Genppi*. A busca por uma rede de densidade intermediária é essencial para garantir que as predições do *Genppi* sejam abrangentes e precisas.

O *STRING*, embora amplamente utilizado, não é exaustivo e pode não cobrir todas as interações reais, especialmente em organismos menos estudados. Portanto, a comparação com o *STRING* deve ser feita com cautela, reconhecendo suas limitações e utilizando outras fontes de validação quando possível.

5.2.4 Avaliação das Implementações

A eficácia das predições de interações proteína-proteína pode ser compreendida pelas métricas que fornecem uma visão abrangente da qualidade e precisão das predições. Essas métricas incluem:

- *Precision* (Precisão): Mede a proporção de interações previstas pelo *Genppi* que estão documentadas no *STRING*. Valores altos indicam que a maioria das interações

previstas são válidas, enquanto valores baixos sugerem um aumento no número de falsos positivos.

- ❑ *Recall* (Sensibilidade): Refere-se à proporção de interações documentadas no *STRING* que foram previstas pelo *Genppi*. Essa métrica é crucial para avaliar a capacidade do *Genppi* de capturar interações conhecidas. Valores altos de *Recall* indicam uma implementação sensível e abrangente, enquanto valores baixos sugerem que muitas interações conhecidas não foram previstas.
- ❑ *F1-Score*: Combina *Precision* e *Recall* em uma única medida, equilibrando sensibilidade e precisão. Valores altos de *F1-Score* indicam um bom equilíbrio entre a capacidade de prever interações conhecidas e a confiabilidade das predições.
- ❑ Densidade da Rede: Mede a proporção de interações observadas em relação ao número máximo possível de interações. Redes muito densas podem dificultar a análise topológica, enquanto redes muito esparsas podem perder informações valiosas.
- ❑ Grau Médio: Indica o número médio de interações por proteína na rede. Valores altos sugerem que cada proteína tem muitas interações, o que pode ser útil para estudos exploratórios, mas também aumenta o risco de falsos positivos.

Os resultados obtidos para as diferentes implementações do *Genppi* são apresentados na Tabela 5.

Tabela 5 – Avaliação das Implementações do *Genppi*.

Genome	Precision	Recall	F1-Score	Densidade	Grau Médio
<i>features</i>	0,35	0,10	0,16	0,018	28,3
<i>mlsplit</i>	0,34	0,62	0,44	0,315	175,6
<i>featurestoll</i>	0,28	0,20	0,23	0,035	19,3
<i>mlsplittoll</i>	0,18	1,00	0,30	0,665	371,4
<i>featurestoll2</i>	0,24	0,30	0,27	0,052	28,7
<i>mlsplittoll2</i>	0,17	1,10	0,29	0,806	449,8
<i>featurestoll3</i>	0,23	0,34	0,27	0,059	33,1
<i>mlsplittoll3</i>	0,17	1,19	0,29	0,864	483,0
<i>mlsplitt25</i>	0,20	0,35	0,25	0,061	33,4
<i>mlsplitt10</i>	0,20	0,33	0,25	0,057	31,6
<i>mlsplit35</i>	0,19	0,35	0,25	0,060	33,6
<i>mlsplit35toll</i>	0,18	0,54	0,27	0,094	52,1
<i>mlsplit35toll3</i>	0,18	0,88	0,30	0,153	85,5
<i>mlsplit35toll2</i>	0,17	0,74	0,28	0,129	71,9

Implementações como *mlsplit* se destacam pela combinação de uma boa precisão (0,34) e um *Recall* razoável (0,62), resultando em um *F1-Score* de 0,44. Essa implementação

demonstra um equilíbrio entre a capacidade de prever interações conhecidas e a confiabilidade das predições. Por outro lado, implementações como *mlsplittol3* e *mlsplittol2*, apesar de apresentarem altos valores de Recall, geram redes excessivamente densas, o que pode aumentar o risco de falsos positivos.

Além disso, a densidade da rede é uma consideração importante. A implementação *mlsplit* apresenta uma densidade moderada (0,315), o que sugere um equilíbrio adequado para análises topológicas. Em contrapartida, implementações com densidade muito alta, como *mlsplittol3* (0,864), podem dificultar a análise e visualização das interações.

As métricas de Precision, Recall e F1-Score são particularmente úteis para avaliar a qualidade das predições em relação ao *STRING*. Um número elevado de interações não documentadas no *STRING* pode indicar superpredição, enquanto um valor baixo de Recall pode sinalizar a necessidade de ajustes nos parâmetros do *Genppi* para melhorar a captura de interações conhecidas.

Dentre as diversas implementações, a mais significativa é a *mlsplit*, que utiliza *machine learning* e algoritmo *RIS* e gerou um total de 49.077 interações únicas, representando uma proporção de 1,81 vezes mais interações do que as 27.041 interações únicas presentes no *STRING*. Deste total, 16.681 interações coincidem com o *STRING*, representando 62% das interações previstas no *STRING*. A razão entre o número de interações únicas da implementação dividido pelo número de interações que intersectam com o *STRING* foi de 2,94, indicando que para cada interação coincidente existe 1,94 interação que não coincide com o *STRING*. Esses resultados quantitativos demonstram a capacidade do programa *Genppi* munido do algoritmo *RIS* de gerar previsões de interações proteicas com uma proporção significativa de sobreposição com as interações documentadas no *STRING*, ao mesmo tempo em que apresenta uma rede que não seja extremamente densa.

5.3 Discussão

5.3.1 Motivo da escolha do Random Forest

Existe uma vasta gama de opções de aprendizado de máquina que podem ser utilizadas na tarefa de identificar proteínas similares. Uma análise comparativa utilizando algoritmos como *XGBoost* e *KNN* certamente agregaria alto valor numa comparação em um contexto geral de aprendizado de máquina. No entanto, um *benchmark* abrangente contra algoritmos alternativos não é o foco principal deste trabalho e não refletiria necessariamente as restrições e os objetivos de otimização específicos do *Genppi*.

O objetivo principal era aprimorar a capacidade do *Genppi* de prever eficientemente PPIs dentro da estrutura de sua arquitetura existente, que depende fortemente da análise

subsequente de perfis filogenéticos e vizinhanças conservadas. Embora aumentar o *recall* de similaridades proteicas potenciais seja desejável, um aumento ingênuo no número de interações previstas exacerba significativamente a carga computacional dessas etapas posteriores. A principal inovação do *Genppi* não reside simplesmente em identificar similaridades, mas em filtrar e priorizar essas similaridades de forma inteligente para gerar uma rede gerenciável e biologicamente significativa.

A implementação do algoritmo “*RIS*” aborda diretamente esse desafio. Ele permite aproveitar a maior sensibilidade do *Random Forest* para identificar interações potenciais, mitigando simultaneamente o custo computacional associado à análise de um grande número de interações. Além disto, o tempo para treinamento do algoritmo é significativamente menor do que outros como Redes Neurais e *KNN*, o que proporcionou um ganho de tempo considerável durante o desenvolvimento do sistema. Como demonstrado nos resultados, o algoritmo *RIS* mantém uma estrutura de rede robusta, mesmo com um conjunto reduzido de interações.

Na verdade, experimentos preliminares com algoritmos alternativos (dados não publicados) mostraram que, embora alguns tenham alcançado um *recall* inicial ligeiramente maior, eles resultaram em um aumento drástico no tempo computacional para a análise de perfis filogenéticos e vizinhanças conservadas, anulando, em última análise, quaisquer benefícios. O *Genppi*, com *Random Forest* e o algoritmo *RIS*, alcança um desempenho equilibrado, com uma eficácia de previsão média de aproximadamente 75% para achados de similaridade proteica, o que é suficiente para gerar redes PPI de alta qualidade dentro das restrições do pipeline de análise.

Outro motivo que justifica a favor do *Random-Forest* é a existência da biblioteca implementada em Common-Lisp, mesma linguagem na qual o *Genppi* foi implementado.

5.3.2 Validação de novas interações

A validação biológica de novas interações é um passo importante. No entanto, o número de novas interações é considerável e não existe, até o momento, nenhum método computacional conhecido para validá-las. Tanto o *Genppi* quanto o banco de dados *STRING* sofrem do mesmo problema, com a maioria de suas previsões sendo massivamente preditas apenas por computador. Portanto, as interações proteicas previstas pelo *Genppi* que não estão presentes no banco de dados *STRING* não devem ser descartadas como falsos positivos. Essas interações podem representar descobertas potenciais, especialmente quando consideramos as características de vizinhança gênica e perfil filogenético conservados, que são corretamente implementados e validados entre as versões do *Genppi*. Dessa forma, a identificação de novas interações pode contribuir para a compreensão das dinâmicas

moleculares em organismos pouco estudados.

5.3.3 Confiabilidade Computacional e Relevância Biológica nas Predições de Interações Proteicas

Um aspecto importante a considerar é a densidade da rede gerada. Redes excessivamente densas podem dificultar a identificação de diferenças significativas na topologia da rede, prejudicando análises mais profundas. Além disso, redes densas podem sobrecarregar a análise de métricas como centralidade e modularidade, especialmente em ferramentas como o *Gephi*. Por outro lado, redes muito esparsas podem perder informações valiosas e não capturar interações relevantes. Portanto, a busca por uma rede de densidade intermediária é essencial para garantir que as predições do *Genppi* sejam abrangentes e precisas.

A comparação com o *STRING* deve ser feita com cautela, reconhecendo suas limitações. Embora o *STRING* integre múltiplas fontes de evidências (experimentais, coexpressão, transferência de conhecimento entre espécies, etc.) para atribuir escores de confiança às interações, ele não abrange todas as interações reais e pode conter falsos positivos. Dessa maneira, as interações previstas pelo *Genppi* que não estão registradas no *STRING* devem ser consideradas como candidatas potenciais para validação experimental. Todavia, é crucial que as redes proteicas previstas apresentem uma proporção significativa de sobreposição com as interações proteicas documentadas na base de dados *STRING*. Isso valida a precisão das previsões e fortalece a confiabilidade dos resultados obtidos.

Ao empregar o método de *Features* sem a utilização de *machine learning*, constatou-se uma reduzida ocorrência de arestas duplicadas. Já ao adotar técnicas de *machine learning*, observa-se uma distinção significativa entre os conjuntos de dados “*query_unique*” e “*query_nounique*”. Ao confrontar esses resultados com os dados do banco *STRING*, verificou-se uma interseção aproximada de 60%. Em certos conjuntos de dados, essa interseção atingiu 100% em relação aos dados do *STRING*; entretanto, foram geradas proporções significativamente superiores de interações inexistentes, chegando a quase cinco vezes mais interações ausentes no *STRING*. Além disso, tais redes de interações apresentaram uma alta densidade de arestas, o que dificulta a identificação de diferenças topológicas na estrutura das redes, comprometendo a realização de análises robustas.

Ao gerar um número menor de interações, pode-se perder informação presente no *STRING*, mas há melhores condições para inferir sobre a topologia da rede. Por outro lado, ao gerar um grande volume de interações, é possível obter interseções de até 100% com os dados do *STRING*, porém a densidade da rede aumenta, tornando-a mais desafiadora para análises subsequentes.

Para contornar o problema do número elevado de interações, alguns parâmetros podem

ser ajustados para aumentar a confiança da probabilidade de uma interação. Isso se dá ao fato que uma interação também pode apresentar alta probabilidade de ser verdadeira ou indicar uma probabilidade mais baixa de ser uma interação real. Tanto o *Genppi* como o *STRING* oferecem informações sobre interações conhecidas e previsões de interações entre proteínas. O *STRING*, por exemplo, atribui um valor de confiança ou um escore de confiança para indicar a força ou a probabilidade da interação. Este valor é calculado com base em uma variedade de fontes de dados, incluindo evidências experimentais, coexpressão gênica, similaridade de sequência, entre outros. O valor de confiança atribuído pelo *STRING* e pelo *Genppi* é apresentado como um número entre 0 e 1.

O ajuste cuidadoso dos parâmetros do *Genppi* é fundamental para equilibrar a sensibilidade e a precisão. O parâmetro *-ppdiff tolerated*, por exemplo, permite uma flexibilidade que pode aumentar a sensibilidade da análise, mas também pode introduzir o risco de falsos positivos. Portanto, a análise do *unique_intersect_ratio* em conjunto com a *string_proportion* é fundamental para avaliar a qualidade das predições, considerando o trade-off entre sensibilidade e precisão.

Destaca-se também que quando se utiliza apenas o *Features* sem *machine learning*, o número de arestas que se repetem (*query_nounique* menos *query_unique*) é bastante reduzido.

Além disso, a utilização do *Genppi* como um software de código aberto oferece vantagens significativas para a pesquisa científica, como personalização, controle, colaboração e transparência. Os resultados obtidos demonstram que o *Genppi* é capaz de capturar uma quantidade substancial de interações semelhantes às do *STRING*, validando assim sua eficácia na geração de redes de interação proteína-proteína.

Por fim, a média de acurácia de cerca de 72,63% na seleção de proteínas similares entre genomas, sem a utilização de métodos tradicionais de alinhamento, ressalta o potencial do *Genppi* em melhorar a qualidade e a eficiência das investigações em biologia molecular. Essa comparação quantitativa fornece insights importantes sobre a qualidade e a precisão das interações geradas pelo *Genppi*, reforçando sua relevância como ferramenta na pesquisa de interações proteicas.

A interpretação dos resultados deve ser feita com cautela, considerando sempre as limitações e as características das interações previstas, e buscando um equilíbrio entre a densidade da rede e a precisão das predições. ´

Conclusão

O uso de ferramentas computacionais na biologia tem sido essencial para o progresso no campo, permitindo análises rápidas e precisas, modelagem de sistemas complexos e predições de resultados. Esse avanço tem impulsionado o desenvolvimento de novas terapias e tratamentos, como evidenciado pelo aumento significativo de pesquisas biológicas utilizando recursos computacionais, especialmente nas análises genômicas que visam uma compreensão mais profunda dos sistemas biológicos.

A análise das interações proteína-proteína oferece insights valiosos sobre processos biológicos essenciais que são difíceis de estudar experimentalmente. A construção de redes de interação proteica é uma ferramenta poderosa para inferir processos biológicos complexos, podendo ser realizada por meio de abordagens experimentais ou computacionais. No entanto, métodos experimentais são muitas vezes caros e demorados, o que pode ser um impedimento significativo para pesquisas em grande escala. Essas limitações sublinham a necessidade de métodos alternativos mais eficientes e econômicos. As abordagens computacionais e bioinformáticas se tornam, portanto, cada vez mais relevantes, permitindo a análise de grandes conjuntos de dados e a simulação de processos biológicos de maneira rápida e eficaz.

A predição computacional de redes de interação proteica é uma alternativa eficaz para mitigar os custos e limitações dos métodos experimentais. Essa abordagem utiliza informações biológicas existentes para construir redes de interação, empregando métodos como análise de sequências de aminoácidos, propriedades físico-químicas, estrutura tridimensional, perfil filogenético, vizinhança gênica conservada e expressão gênica. No entanto, a ontologia gênica por si só não é suficiente para predições precisas, e as abordagens atuais, como o *STRING*, enfrentam limitações, especialmente com genomas novos ou pouco estudados.

A versão 1.0 do *Genppi* apresentou uma metodologia inovadora que utiliza características bioquímicas e análises filogenéticas e genômicas comparativas, permitindo a predição

de redes de interação a partir de qualquer genoma, sem depender de dados externos. A capacidade do *Genppi* 1.0 de gerar redes de interação a partir de dados genômicos específicos e inéditos com alta acurácia sugere uma versatilidade significativa, embora sua aplicabilidade fosse limitada pela necessidade de alta similaridade proteica entre as proteínas.

Com a integração do *Random Forest*, o *Genppi* conseguiu superar essas limitações, identificando proteínas com similaridade proteica mais baixa (acima de 65%), o que aumentou significativamente o número de redes de interação preditas. A otimização do processamento necessário para gerar essas redes reduziu o número de casos negligenciados. Assim, o *Genppi* atingiu um novo patamar de desempenho, permitindo predições de redes de interação proteica com maior precisão e eficiência, mesmo para genomas inéditos e sem depender de bancos de dados externos. Os resultados da versão mais recente do *Genppi* demonstram avanços significativos no estudo de interações proteicas.

Os resultados obtidos sugerem que o *Genppi* oferece funcionalidades valiosas para pesquisadores em biologia molecular e bioinformática. Sua capacidade de gerar redes de interação proteica a partir de dados genômicos específicos e inéditos, sem depender de bancos de dados externos e com alta precisão, amplia as possibilidades de investigação científica, especialmente em genomas menos estudados ou em organismos com características únicas.

A pesquisa seguiu uma metodologia estruturada em duas fases principais: uma revisão bibliográfica e um estudo experimental. O objetivo foi incorporar o algoritmo *Random Forest* ao *Genppi* e avaliar sua performance na predição de redes de interação proteína-proteína (PPI) para genomas bacterianos. A primeira fase envolveu uma revisão bibliográfica aprofundada sobre algoritmos de *Random Forest*, implementações em diferentes linguagens de programação e conceitos de bioinformática relacionados à classificação de proteínas.

O estudo experimental incluiu várias etapas, como coleta e preparação de dados, adequação dos dados para uso com a biblioteca *CL-Random Forest*, treinamento e avaliação do modelo, estudo comparativo, avaliação integrada e refinamento do algoritmo. Os resultados dos experimentos mostraram uma melhoria significativa na predição de redes de interação proteína-proteína. Testes de controle positivo e negativo validaram a capacidade do algoritmo de identificar interações com diferentes níveis de similaridade, e estudos comparativos destacaram a superioridade do *Random Forest* em relação a outras ferramentas de predição como o *STRING*, especialmente em termos de eficiência e robustez.

A nova implementação do *Genppi* aumentou substancialmente a capacidade de predição de redes de interação proteína-proteína. A precisão e especificidade das predições foram aprimoradas, resultando em uma ferramenta mais eficaz para análises genômicas e

proteômicas. Os refinamentos iterativos no algoritmo *Random Forest* contribuíram para melhorias adicionais no desempenho e precisão do modelo. A comparação dos resultados com os dados do *STRING* confirmou que a nova implementação do *Genppi* era capaz de identificar interações proteicas com precisão superior às ferramentas tradicionais.

6.1 Limitações

Este estudo aprimorou a capacidade de predição de redes de interação proteína-proteína (PPIs) do software *Genppi*, mas enfrentou desafios importantes. Entre as limitações identificadas, destacam-se os custos computacionais intrínsecos à geração de redes PPI. O cálculo de perfis filogenéticos, a aplicação de técnicas de *machine learning* e a execução do algoritmo *RIS* demandam alto poder de processamento, especialmente com grandes conjuntos de proteínas. O tempo de execução aumenta exponencialmente conforme o número de proteínas e parâmetros como *-ppdiff tolerated*, enquanto redes densas (excedendo 187.000 interações) exigem estratégias de armazenamento e compressão robustas para evitar gargalos.

A validação das predições também enfrentou obstáculos. A dependência do *STRING* — ferramenta amplamente reconhecida, porém de código fechado — limitou a transparência na comparação de resultados. Falsos positivos e negativos em sua base de dados, decorrentes da integração de fontes heterogêneas, podem ter impactado a avaliação da precisão do *Genppi*. Além disso, a validação experimental de interações previstas permanece um desafio crítico: técnicas como *yeast two-hybrid* (Y2H) são custosas e demoradas, enquanto a incerteza sobre a relevância biológica de interações validadas experimentalmente introduz camadas adicionais de complexidade.

Para mitigar essas limitações, propõe-se a otimização algorítmica com paralelização e uso de GPUs, a integração com bases de dados alternativas (como *BioGRID* e *IntAct*), e o desenvolvimento de métricas de confiança para filtrar interações mais plausíveis. A colaboração aberta e o código-fonte disponível do *Genppi* (discutido adiante) são passos estratégicos para ampliar a reprodutibilidade e a transparência metodológica.

6.2 Disponibilidade do Genppi

O *Genppi* é disponibilizado como software autônomo e de código aberto, fortalecendo seu papel na comunidade científica. O executável para Linux e Windows, juntamente com o código-fonte e documentação detalhada, estão acessíveis em <<https://genppi.facom.ufu.br/>> e <<https://github.com/santosardr/genppi>>. Os usuários podem gerar redes PPI a partir de arquivos *multi-fasta*, exportar relatórios para ferramentas como *Gephi* e utilizar

uma plataforma web interativa (<<https://genppi.facom.ufu.br/>>) para visualização de resultados. Essa abertura não apenas facilita a personalização e aprimoramento contínuo, mas também promove a integração com outras ferramentas de bioinformática.

6.3 Trabalhos Futuros

Para trabalhos futuros, sugere-se explorar aplicações práticas do *Genppi* em contextos de pesquisa real que represente problemas complexos, como por exemplo o estudo da oxidação anaeróbica do metano (AOM) nos oceanos. Esse processo ocorre em condições específicas que permitem a oxidação do metano sem a presença de oxigênio. Alguns mecanismos moleculares subjacentes a essa oxidação ainda são pouco compreendidos, sendo necessário investigar as interações proteína-proteína envolvidas. Usar o *Genppi* para explorar essas interações pode ajudar a elucidar as condições específicas e os mecanismos moleculares que permitem essa reação. Isso poderá fornecer informações valiosas sobre as adaptações bioquímicas e os sistemas enzimáticos envolvidos, abrindo portas para o entendimento de processos de mitigação de gases de efeito estufa nos oceanos.

Assim, este estudo não apenas propõe melhorias metodológicas para o *Genppi*, mas também destaca seu potencial para futuras pesquisas aplicadas, incentivando a análise de redes proteicas em cenários de relevância ambiental e ecológica. A continuidade desse trabalho permitirá um avanço significativo na compreensão de interações biológicas complexas, contribuindo para o progresso da biotecnologia e para a conservação de ecossistemas essenciais.

Referências

ALOY, P.; RUSSELL, R. B. Interrogating protein interaction networks through structural biology. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 99, n. 9, p. 5896–5901, 2002. Disponível em: <<https://doi.org/10.1073/pnas.092147999>>.

AN, K. Sulla determinazione empirica di una legge didistribuzione. **Giorn Dell’inst Ital Degli Att**, v. 4, p. 89–91, 1933.

ANJOS, W. F. et al. Genppi: standalone software for creating protein interaction networks from genomes. **BMC bioinformatics**, Springer, v. 22, n. 1, p. 596, 2021. Disponível em: <<https://doi.org/10.1186/s12859-021-04501-0>>.

ANJOS, W. F. d. Genppi: um software autônomo para predição ab initio de redes de interação entre proteínas bacterianas. Universidade Federal de Uberlândia, 2021. Disponível em: <<https://repositorio.ufu.br/bitstream/123456789/32702/9/GenppiSoftwareAutonomo.pdf>>.

ATKINS, P. W. **Atkins’ molecules**. [S.l.]: Cambridge University Press, 2003.

BARABASI, A.-L.; OLTVAI, Z. N. Network biology: understanding the cell’s functional organization. **Nature reviews genetics**, Nature Publishing Group UK London, v. 5, n. 2, p. 101–113, 2004. Disponível em: <<https://doi.org/10.1038/nrg1272>>.

BASTIAN, M.; HEYMANN, S.; JACOMY, M. Gephi: an open source software for exploring and manipulating networks. In: **Proceedings of the international AAAI conference on web and social media**. [s.n.], 2009. v. 3, n. 1, p. 361–362. Disponível em: <<https://doi.org/10.1609/icwsm.v3i1.13937>>.

BIAU, G.; SCORNET, E. A random forest guided tour. **Test**, Springer, v. 25, p. 197–227, 2016. Disponível em: <<https://doi.org/10.1007/s11749-016-0481-7>>.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, p. 5–32, 2001. Disponível em: <<https://doi.org/10.1023/A:1010933404324>>.

BROWN, T. A. **Genética: Um Enfoque Molecular** . [S.l.]: Grupo Gen-Guanabara Koogan, 2000.

CAMACHO, C. et al. Blast+: architecture and applications. **BMC bioinformatics**, Springer, v. 10, p. 1–9, 2009. Disponível em: <<https://doi.org/10.1186/1471-2105-10-421>>.

CHEN, H. et al. Systematic evaluation of machine learning methods for identifying human–pathogen protein–protein interactions. **Briefings in Bioinformatics**, Oxford University Press, v. 22, n. 3, p. bbaa068, 2021. Disponível em: <<https://doi.org/10.1093/bib/bbaa068>>.

CRICK, F. H. The origin of the genetic code. **Journal of molecular biology**, Elsevier, v. 38, n. 3, p. 367–379, 1968. Disponível em: <[https://doi.org/10.1016/0022-2836\(68\)90392-6](https://doi.org/10.1016/0022-2836(68)90392-6)>.

CRIMINISI, A.; SHOTTON, J.; KONUKOGLU, E. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. In: _____. **Foundations and Trends® in Computer Graphics and Vision**. Foundations and trends® in computer graphics and vision: Vol. 7: No 2-3, pp 81-227. [s.n.], 2012. v. 7, n. 2-3, p. 81–227. Disponível em: <<https://doi.org/10.1561/06000000035>>.

DEPPENMEIER, U. The unique biochemistry of methanogenesis. Elsevier, 2002. Disponível em: <[https://doi.org/10.1016/S0079-6603\(02\)71045-3](https://doi.org/10.1016/S0079-6603(02)71045-3)>.

DHAKAL, A. et al. Artificial intelligence in the prediction of protein–ligand interactions: recent advances and future directions. **Briefings in Bioinformatics**, Oxford University Press, v. 23, n. 1, p. bbab476, 2022. Disponível em: <<https://doi.org/10.1093/bib/bbab476>>.

FLEISS, J. L. Measuring nominal scale agreement among many raters. **Psychological bulletin**, American Psychological Association, v. 76, n. 5, p. 378, 1971. Disponível em: <<https://doi.org/10.1037/h0031619>>.

GAUTHAM, N. **Bioinformatics: databases and algorithms**. Alpha Science Int’l Ltd., 2006. 68-90 p. Disponível em: <<https://doi.org/10.1038/415141a>>.

GAVIN, A.-C. et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. **Nature**, Nature Publishing Group UK London, v. 415, n. 6868, p. 141–147, 2002.

GÉRON, A. Mãos à obra: aprendizado de máquina com scikit-learn. **Keras & TensorFlow: Conceitos, ferramentas e técnicas para a construção de sistemas inteligentes**. [S. l.: sn], 2021.

GIULIO, M. D. On the origin of the genetic code. **Journal of theoretical biology**, Elsevier, v. 187, n. 4, p. 573–581, 1997. Disponível em: <<https://doi.org/10.1006/jtbi.1996.0390>>.

GRAHAM, P. **On Lisp**. [S.l.]: Prentice Hall, 1993.

HARISHA, S. **Fundamentals of Bioinformatics**. [S.l.]: IK International Pvt Ltd, 2013.

HO, T. K. Random decision forests. In: IEEE. **Proceedings of 3rd international conference on document analysis and recognition**. 1995. v. 1, p. 278–282. Disponível em: <<https://doi.org/10.1109/ICDAR.1995.598994>>.

HU, L. et al. A novel network-based algorithm for predicting protein-protein interactions using gene ontology. **Frontiers in Microbiology**, Frontiers, v. 12, p. 735329, 2021. Disponível em: <<https://doi.org/10.3389/fmicb.2021.735329>>.

IMAI, S. **CL-Random-Forest**. GitHub, 2017. Disponível em: <<https://github.com/masatoi/cl-random-forest>>.

INMAN, H. F.; JR, E. L. B. The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities. **Communications in Statistics-theory and Methods**, Taylor & Francis, v. 18, n. 10, p. 3851–3874, 1989. Disponível em: <<https://doi.org/10.1080/03610928908830127>>.

IZBICKI, R.; SANTOS, T. M. dos. **Aprendizado de máquina: uma abordagem estatística**. [S.l.]: Rafael Izbicki, 2020.

JACCARD, P. Étude comparative de la distribution florale dans une portion des alpes et des jura. **Bull Soc Vaudoise Sci Nat**, v. 37, p. 547–579, 1901.

KANG, Y. et al. Bbln: A bilateral-branch learning network for unknown protein–protein interaction prediction. **Computers in Biology and Medicine**, Elsevier, v. 167, p. 107588, 2023. Disponível em: <<https://doi.org/10.1016/j.compbiomed.2023.107588>>.

KAWASHIMA, S.; KANEHISA, M. Aaindex: amino acid index database. **Nucleic acids research**, Oxford University Press, v. 28, n. 1, p. 374–374, 2000. Disponível em: <<https://doi.org/10.1093/nar/28.1.374>>.

KEKKONEN, R. Designing domain-specific languages with common lisp macros. 2023.

KENDALL, M. G. A new measure of rank correlation. **Biometrika**, Oxford University Press, v. 30, n. 1-2, p. 81–93, 1938. Disponível em: <<https://doi.org/10.1093/biomet/30.1-2.81>>.

KUHN, M.; JOHNSON, K. **Applied Predictive Modeling**. Springer, 2013. Disponível em: <<https://doi.org/10.1007/978-1-4614-6849-3>>.

KUNZE, F.; KUNZE, L. The history of franz and lisp. **IEEE Annals of the History of Computing**, v. 44, n. 1, p. 103–108, 2022. Disponível em: <<https://doi.org/10.1109/MAHC.2022.3144802>>.

LEE, Y. S.; KIM, Y. S.; UY, R. L. Serial and parallel implementation of needleman-wunsch algorithm. **International Journal of Advances in Intelligent Informatics**, v. 6, n. 1, 2020. Disponível em: <<https://doi.org/10.26555/ijain.v6i1.361>>.

LESK, A. **Introduction to protein science: architecture, function, and genomics**. [S.l.]: Oxford university press, 2010.

_____. **Introduction to bioinformatics**. [S.l.]: Oxford university press, 2019.

LIAW, A.; WIENER, M. Classification and regression by randomforest. **R News**, v. 2, n. 3, p. 18–22, 2002.

LODISH, H. F. **Molecular cell biology**. [S.l.]: Macmillan, 2008.

LOUPPE, G. **Understanding Random Forests: From Theory to Practice**. 2015.

MCCARTHY, J. History of lisp. In: **History of programming languages**. [s.n.], 1978. p. 173–185. Disponível em: <<https://doi.org/10.1145/800025.1198360>>.

MIN, S.; LEE, B.; YOON, S. Deep learning in bioinformatics. **Briefings in bioinformatics**, Oxford University Press, v. 18, n. 5, p. 851–869, 2017.

MOGENSEN, T. Æ. **Programming Language Design and Implementation**. [S.l.]: Springer Nature, 2022.

MÜLLER, A.; GUIDO, S. **Introduction to Machine Learning with Python: A Guide for Data Scientists**. [S.l.]: O'Reilly Media, 2016.

MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. [S.l.]: MIT Press, 2012.

NEWMAN, M. E. J. **Networks: An Introduction**. [S.l.]: Oxford University Press, 2010. 168-235 p.

NORVIG, P.; RUSSELL, S. **Artificial Intelligence: A Modern Approach**. [S.l.]: Pearson Education, 2010.

NORVIG, P. et al. **Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp**. [S.l.]: Morgan Kaufmann, 2012.

OLIVEIRA, G.; SANTOS, A. Using the gene ontology tool to produce de novo protein-protein interaction networks with is_a relationship. **Genet Mol Res**, v. 15, n. 4, p. 10–4238, 2016. Disponível em: <<https://doi.org/10.4238/gmr15049273>>.

OSHIRO, T. M. **Uma abordagem para a construção de uma única árvore a partir de uma Random Forest para classificação de bases de expressão gênica**. Tese (Doutorado) — Universidade de São Paulo, 2013.

PEARSON, W. R. Flexible sequence similarity searching with the fasta3 program package. **Bioinformatics methods and protocols**, Springer, p. 185–219, 1999. Disponível em: <<https://doi.org/10.1385/1-59259-192-2:185>>.

_____. Blast and fasta similarity searching for multiple sequence alignment. **Multiple sequence alignment methods**, Springer, p. 75–101, 2014. Disponível em: <https://doi.org/10.1007/978-1-62703-646-7_5>.

PEARSON, W. R. et al. Comparison of dna sequences with protein sequences. **Genomics**, Elsevier, v. 46, n. 1, p. 24–36, 1997. Disponível em: <<https://doi.org/10.1006/geno.1997.4995>>.

- PEREIRA, G.; GHOSH, P.; SANTOS, A. A bridging centrality plugin for gephi and a case study for mycobacterium tuberculosis h37rv. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, IEEE, v. 18, n. 6, p. 2741–2746, 2021. Disponível em: <<https://doi.org/10.1109/TCBB.2021.3120937>>.
- PN, T.; STEINBACH, M.; KUMAR, V. **Introduction to data mining**. [S.l.]: Addison-Wesley, 2005.
- POLANSKI, A.; KIMMEL, M. **Bioinformatics**. [S.l.]: Springer Science & Business Media, 2007. 155-296 p.
- RASTI, S.; VOGIATZIS, C. A survey of computational methods in protein–protein interaction networks. **Annals of Operations Research**, Springer, v. 276, p. 35–87, 2019. Disponível em: <<https://doi.org/10.1007/s10479-018-2956-2>>.
- REN, S. et al. Global refinement of random forest. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [s.n.], 2015. p. 723–730. Disponível em: <<https://doi.org/10.1109/CVPR.2015.7298672>>.
- SARDIU, M. E. et al. Topological scoring of protein interaction networks. **Nature communications**, Nature Publishing Group UK London, v. 10, n. 1, p. 1118, 2019. Disponível em: <<https://doi.org/10.1038/s41467-019-09123-y>>.
- SCHONLAU, M.; ZOU, R. Y. The random forest algorithm for statistical learning. **The Stata Journal**, SAGE Publications, v. 20, n. 1, p. 3–29, 2020. Disponível em: <<https://doi.org/10.1177/1536867X20909688>>.
- SEIBEL, P. **Practical Common Lisp**. Apress, 2005. Disponível em: <<https://doi.org/10.1007/978-1-4302-0017-8>>.
- SHIMA, S.; THAUER, R. K. Methyl-coenzyme m reductase and the anaerobic oxidation of methane in methanotrophic archaea. **Current opinion in microbiology**, Elsevier, v. 8, n. 6, p. 643–648, 2005.
- STARMER, S. with J. **Random Forest - Construindo e Avaliando**. 2023. <https://www.youtube.com/watch?v=J4Wdy0Wc_xQ&t=0s>. Vídeo online, Acesso em: 13/06/2023.
- STRACHAN, T.; READ, A. **Genética molecular humana**. [S.l.]: Artmed Editora, 2016.
- STRYER, L. Introduction to protein structure and function. In: **Biochemistry**. [S.l.]: Freeman San Francisco, 1981. p. 11–38.
- SZKLARCZYK, D. et al. The string database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. **Nucleic acids research**, Oxford University Press, v. 49, n. D1, p. D605–D612, 2021. Disponível em: <<https://doi.org/10.1093/nar/gkaa1074>>.
- _____. String: functional protein association networks. **Nucleic Acids Research**, Oxford University Press, v. 43, n. D1, p. D447–D452, 2015.

TANG, T. et al. Machine learning on protein–protein interaction prediction: models, challenges and trends. **Briefings in Bioinformatics**, Oxford University Press, v. 24, n. 2, p. bbad076, 2023. Disponível em: <<https://doi.org/10.1093/bib/bbad076>>.

TOURETZKY, D. S. **Common LISP: A Gentle Introduction to Symbolic Computation**. [S.l.]: Benjamin-Cummings Publishing Company, 1990.

UZMAN, A. **Biologia molecular da célula: Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., e Walter, P.** John Wiley & Sons Inc. EUA, 2003. Disponível em: <<https://doi.org/10.1002/bmb.2003.494031049999>>.

VERLENGIA, R. et al. Análises de rna, proteínas e metabólitos: metodologia e procedimentos técnicos. In: **Análises de RNA, Proteínas e Metabólitos: metodologia e procedimentos técnicos**. [S.l.: s.n.], 2013. p. 191–206.

WALHOUT, A. J.; VIDAL, M. Protein interaction maps for model organisms. **Nature reviews Molecular cell biology**, Nature Publishing Group UK London, v. 2, n. 1, p. 55–63, 2001. Disponível em: <<https://doi.org/10.1038/35048107>>.

WATSON, J. D. et al. **Biologia molecular do gene**. [S.l.]: Artmed Editora, 2015.

WATSON, J. D.; BERRY, A. **DNA: The secret of life**. [S.l.]: Knopf, 2009.

WATSON, J. D.; CRICK, F. H. Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. **Nature**, Nature Publishing Group UK London, v. 171, n. 4356, p. 737–738, 1953. Disponível em: <<https://doi.org/10.1038/171737a0>>.

WINSTON, P. H.; HORN, B. K. P. **Lisp**. [S.l.]: Addison-Wesley Professional, 1987.

YAN, T.-C. et al. A systematic review of state-of-the-art strategies for machine learning-based protein function prediction. **Computers in Biology and Medicine**, Elsevier, v. 154, p. 106446, 2023. Disponível em: <<https://doi.org/10.1016/j.compbiomed.2022.106446>>.

ZHANG, W. et al. Cegso: boosting essential proteins prediction by integrating protein complex, gene expression, gene ontology, subcellular localization and orthology information. **Interdisciplinary Sciences: Computational Life Sciences**, Springer, v. 13, p. 349–361, 2021. Disponível em: <<https://doi.org/10.1007/s12539-021-00426-7>>.

Apêndices

Avaliação do Impacto do Algoritmo *RIS* na Geração de Redes de Interação Proteína-Proteína no Genppi

Uma abordagem baseada em métricas estatísticas e comparação de distribuições por meio de testes de verificação das mudanças estruturais promovidas pelo algoritmo RIS no Genppi na geração de redes de interação proteína-proteína.

A.1 Introdução

A geração de redes de interação proteína-proteína (PPI) é uma etapa crítica para estudos de biologia sistêmica, permitindo a análise de vias metabólicas, processos celulares e relações evolutivas. No entanto, a aplicação de técnicas de aprendizado de máquina (*Machine Learning*) para predição de interações introduziu um desafio inesperado: o crescimento exponencial no número de arestas geradas. Esse fenômeno ocorre devido à identificação de perfis filogenéticos conservados com centenas de genes, onde a combinação entre pares de genes resulta em uma explosão combinatória de interações. Por exemplo, um único perfil com 500 genes gera até 124.251 arestas, inviabilizando análises topológicas em redes com múltiplos perfis dessa magnitude.

Para mitigar esse problema, foi desenvolvido o **algoritmo RIS**, integrado ao *Genppi* 1.5, cujo objetivo é reduzir a complexidade computacional sem comprometer a integridade estrutural da rede. A estratégia do *RIS* baseia-se na divisão de perfis filogenéticos grandes em sublistas menores, cada uma contendo uma amostra representativa dos genes originais. Esse processo é realizado em três etapas principais:

- ❑ **Embaralhamento (*Shuffle*):** A lista de genes é aleatorizada para eliminar viés de ordenação.

- ❑ **Subdivisão:** O perfil original é dividido em sublistas de tamanho controlado (90% do número de genes originais), reduzindo o número de combinações por fator quadrático.
- ❑ **Processamento Independente:** Cada sublista é tratada como um perfil distinto, gerando sub-redes menores e gerenciáveis.

A adoção dessa abordagem levantou questões críticas: *Como garantir que a redução de arestas não distorça a topologia da rede?* e *A perda de interações afeta a representatividade biológica?* Para responder a essas perguntas, foram realizados testes rigorosos comparando redes geradas com o *RIS*, utilizando métricas estatísticas e análise comparativa de distribuições.

A.2 Metodologia

Este documento detalha a metodologia empregada para validar o Algoritmo *RIS*, com foco em uma abordagem sistemática de avaliação estrutural.

A figura 14 ilustra o fluxograma completo, desde a geração das redes até a análise comparativa. Todas as implementações se referem ao genoma de referência da *Buchnera Aphidicola*. A escolha por 50 repetições por modo garantiu robustez contra variabilidade estocástica, enquanto a variação do *Top_N* permitiu avaliar como a sensibilidade do algoritmo muda com o tamanho da amostra.

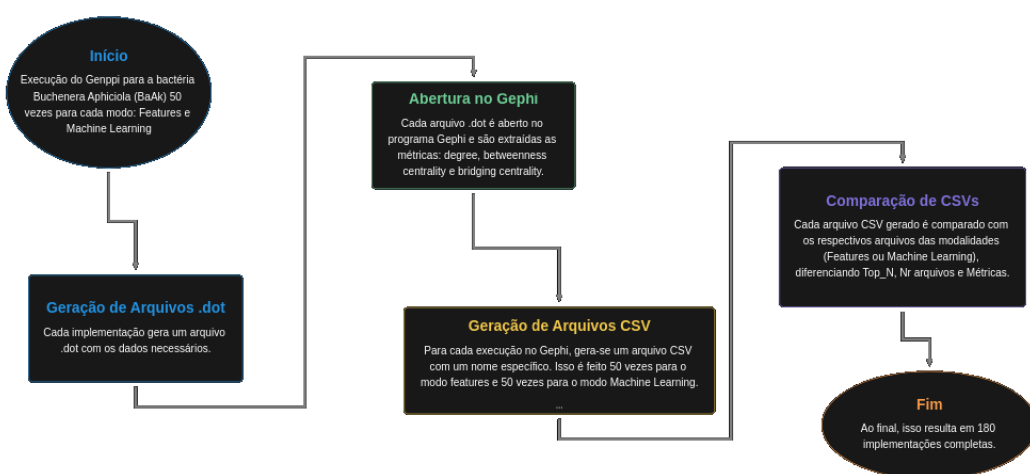


Figura 14 – Fluxograma do processo para geração de testes.

A.2.1 Base de Dados

Foram executadas 50 implementações do *Genppi* 1.5 em cada modo (*Features* e *Machine Learning*), gerando ao final arquivos *.dot* representando as redes.

A.2.1.1 Arquivos *.dot* (*Graph Viz*)

Os arquivos *.dot* contêm a representação gráfica das redes no formato *Graph Viz*.

Código A.1 – Exemplo de arquivo *.dot* gerado

```
1 william@williamubuntu:~/Documentos/Dissertacao/TestesBA/modoML/BAAK$  
  head TesteML08.dot  
2 graph Ba\_Ak {  
3     "BAKON_001" -- "BAKON_002" [WEIGHT = 0.1];  
4     "BAKON_001" -- "BAKON_003" [WEIGHT = 0.2];  
5     "BAKON_001" -- "BAKON_004" [WEIGHT = 0.1];  
6     "BAKON_001" -- "BAKON_561" [WEIGHT = 0.061];  
7     "BAKON_001" -- "BAKON_453" [WEIGHT = 0.061];  
8     "BAKON_001" -- "BAKON_436" [WEIGHT = 0.061];  
9     # ... (continua com outras conexoes)  
10 }
```

Estrutura:

- ❑ Cabeçalho: *graph Ba_Ak* - define o grafo.
- ❑ Arestas: "Nó1" - "Nó2" [*WEIGHT* = valor].
- ❑ Peso numérico: Representa a força da interação (0.0 a 1.0).

Cada arquivo *.dot* foi processado no *Gephi* para extração das métricas, resultando em arquivos *.csv* nomeados pelo padrão:

[Modo]_[NrArquivos].csv.

A.2.1.2 Arquivos *.csv*

Os arquivos *.csv* contêm métricas topológicas extraídas via *Gephi*. Os nomes dos arquivos seguem o padrão [Modo] [NrArquivo].csv, onde:

- ❑ Modo: ML (*Machine Learning*) ou F (*Features*).
- ❑ NrArquivo: Número sequencial do teste (08 no exemplo).

Código A.2 – Exemplo de arquivo CSV gerado

```
1 william@williamubuntu:~/Documentos/Dissertacao/TestesBA/modoML/BAAK$
   head ~/Documentos/DisserTemp/TestesBA/MLCSV/ML08.csv
2
3 Label,Degree,BetweennessCentrality,BridgingCentrality
4 BAKON_001,34,122.005199,4.408293
5 BAKON_002,289,233.843924,0.724917
6 BAKON_003,288,232.844781,0.7261
7 BAKON_004,289,233.841958,0.724242
8 BAKON_561,45,421.805166,8.808386
9 # ... (continua com outros nos)
```

Colunas:

- ❑ *Label*: Identificador único do nó (proteína).
- ❑ *Degree*: Grau do nó (número de conexões).
- ❑ *BetweennessCentrality*: Centralidade de intermediação.
- ❑ *BridgingCentrality*: Centralidade de ponte.

A.2.2 Cenário de Testes

Para compreender o impacto da redução de arestas promovida pelo *RIS*, foram executados testes exaustivos com combinações parametrizadas. A Tabela 6 detalha os valores testados e o cálculo das combinações.

Tabela 6 – Parâmetros variados e combinações testadas

Parâmetro	Valores Testados	Nº Opções
Número de Arquivos	10, 20, 30, 40, 50	5
Modo de Implementação	<i>Features</i> , <i>Machine Learning</i>	2
Métricas Topológicas	<i>Degree</i> , <i>Betweenness</i> , <i>Bridging</i>	3
<i>Top_N</i>	10, 20, 30, 50, 100, 200	6
Total de Combinações		$5 \times 2 \times 3 \times 6 = 180$

Esta matriz experimental permite analisar separadamente:

- ❑ A sensibilidade do algoritmo ao número de arquivos processados;
- ❑ Diferenças entre abordagens bioquímicas (*Features*) e baseadas em *Machine Learning*;
- ❑ Variação nas métricas de centralidade;

□ Efeito do tamanho da amostra (Top_N).

Os 180 cenários foram analisados comparando arquivos *.csv* com os mesmos parâmetros (ou seja, arquivos de *Features* com *Features* e *Machine Learning* com *Machine Learnig*). Foram aplicados testes estatísticos descritos adiante para avaliar divergências nas distribuições das métricas.

A abrangência deste estudo - com 180 implementações analisados fornece evidências estatisticamente significativas sobre a capacidade do *RIS* em preservar propriedades estruturais críticas mesmo com reduções drásticas no número de arestas.

A.2.3 Métricas Utilizadas

A.2.3.1 Métricas Topológicas

Um passo importante ao realizar as diversas implementações do *Genppi* para análises consiste na escolha das métricas a serem selecionadas para investigação. Três métricas topológicas foram escolhidas devido a sua importância: ***Degree***, ***Betweenness Centrality*** e ***Bridging Centrality***.

Cada uma reflete características distintas da organização e dinâmica da rede, conforme detalhado abaixo:

- ***Degree* (Grau)**: Representa o número de conexões diretas de um nó (proteína) com outros nós na rede. No contexto de PPI, um *Degree* elevado indica uma proteína “*hub*”, potencialmente crítica para a integridade estrutural ou funcional da rede. Alterações na distribuição de *Degree* entre redes comparadas podem sugerir mudanças na centralidade de proteínas específicas, afetando a robustez ou sinalização celular.
- ***Betweenness Centrality* (Centralidade de Intermediação)**: Mede a frequência com que um nó atua como ponte nos caminhos mais curtos entre pares de nós. Proteínas com alta *Betweenness Centrality* são “intermediárias” estratégicas, conectando módulos distintos da rede. Variações nessa métrica podem indicar rearranjos na comunicação entre vias biológicas ou na formação de sub-redes especializadas (NEWMAN, 2010).
- ***Bridging Centrality* (Centralidade de Ponte)**: Identifica nós que conectam comunidades diferentes na rede, combinando *Degree* moderado com alta intermediação. Proteínas com elevada *Bridging Centrality* são “pontes” entre grupos funcionais, facilitando a integração de processos celulares. Mudanças nessa métrica sugerem

alterações na modularidade da rede ou na capacidade de coordenação entre sistemas biológicos (PEREIRA; GHOSH; SANTOS, 2021).

A comparação dessas métricas entre redes permite identificar:

- ❑ Estabilidade de *hubs*: Variações em *Degree* podem revelar perda ou surgimento de *hubs*, impactando a resiliência da rede.
- ❑ Reorganização de Fluxos: Alterações em *Betweenness Centrality* indicam modificações em rotas críticas de sinalização ou interações essenciais.
- ❑ Integração Modular: Flutuações em *Bridging Centrality* refletem mudanças na conexão entre comunidades funcionais, possivelmente associadas a adaptações celulares.

A.2.3.2 Média Global Ponderada de Presença dos nós

A média global de presença dos nós é uma métrica estatística que quantifica a representatividade consolidada de nós em um conjunto de redes analisadas, considerando tanto a frequência de aparição de cada nó quanto a diversidade de nós entre os arquivos. Sua formulação busca mitigar distorções causadas pela variação na composição dos nós entre redes distintas, garantindo uma avaliação equilibrada da relevância coletiva dos elementos.

Sua definição formal é descrita a seguir:

- Cálculo da Média Individual: Para um nó i , sua média individual de presença (P_i) corresponde à frequência relativa de aparição nos Top_N nós de cada arquivo:

$$P_i = \left(\frac{C_i}{M} \right) \times 100\% \quad (1)$$

onde:

- ❑ C_i : Número de arquivos em que o nó i está entre os Top_N .
- ❑ M : Número total de arquivos analisados.

- Cálculo da Média Global Ponderada: A média global ($\bar{P}$) é obtida pela soma ponderada das médias individuais, normalizada pela contribuição de cada nó:

$$\bar{P} = \frac{\sum_{i=1}^T (P_i \times w_i)}{\sum_{i=1}^T w_i} \quad (2)$$

com:

- ❑ T : Total de nós únicos identificados em todos os arquivos.
- ❑ w_i : Peso do nó i , definido como $w_i = \frac{C_i}{M}$.

Interpretação:

- Ponderação: A atribuição de pesos (w_i) prioriza nós recorrentes, minimizando o impacto de nós esparsos.
- Estabilidade Estrutural: Valores altos de $\bar{P}$ indicam concordância entre os arquivos sobre nós centrais, enquanto valores baixos sugerem variabilidade contextual ou dinâmica na rede.

A média ponderada foi adotada como uma solução ao problema da média simples ficar distorcida em razão da variedade de nós que surge ao não haver uma concordância completa de nós. Exemplificamos o contexto do problema com o exemplo abaixo:

Suponhamos 10 arquivos (01 a 10) onde cada arquivo possui um $Top_N = 10$ *labels* mais relevantes. A Tabela abaixo mostra a presença dos *labels*:

Tabela 7 – Presença de *Labels* em 10 Arquivos (Exemplo Detalhado)

Label	% Presença	Arquivos	Frequência
BAKON_423	100%	01,02,03,04,05,06,07,08,09,10	10
BAKON_249	100%	01,02,03,04,05,06,07,08,09,10	10
BAKON_124	100%	01,02,03,04,05,06,07,08,09,10	10
BAKON_562	100%	01,02,03,04,05,06,07,08,09,10	10
BAKON_317	100%	01,02,03,04,05,06,07,08,09,10	10
BAKON_570	100%	01,02,03,04,05,06,07,08,09,10	10
BAKON_289	100%	01,02,03,04,05,06,07,08,09,10	10
BAKON_123	100%	01,02,03,04,05,06,07,08,09,10	10
BAKON_134	100%	01,02,03,04,05,06,07,08,09,10	10
BAKON_263	100%	01,02,03,04,05,06,07,08,09,10	10
BAKON_204	90%	01,02,03,04,05,06,07,08,09	9
BAKON_234	10%	10	1
BAKON_235	50%	01,02,03,04,05	5
BAKON_236	10%	06	1
BAKON_237	10%	07	1
BAKON_238	10%	08	1
BAKON_239	10%	09	1
BAKON_240	10%	10	1

a) **Problema da média simples:**

A média simples do percentual de presença individual trata todos os *labels* como igualmente relevantes, mesmo que apareçam em poucos arquivos:

$$\text{Média Simples} = \frac{\sum(\% \text{ de presença})}{\text{Nr total de labels}} = \frac{10 \times 100\% + 90\% + 10\% + 50\% + 5 \times 10\%}{18} \quad (3)$$

$$\text{Média Simples} = \frac{1000\% + 90\% + 10\% + 50\% + 50\%}{18} = \frac{1200\%}{18} \approx \mathbf{66.67\%} \quad (4)$$

Problema: *Labels* raros (como BAKON_236 a 240) distorcem a média, mesmo que 10 *labels* apareçam na maioria quase absoluta dos arquivos.

b) **Solução com a Média Ponderada:**

A média ponderada considera a frequência de cada label (nº de arquivos onde aparece):

$$\text{Média Ponderada} = \frac{\sum(\% \text{ de presença} \times \text{Frequência})}{\sum(\text{Frequência})} \quad (5)$$

$$\begin{aligned} \text{Soma ponderada} &= (10 \times 100\% \times 10) + (90\% \times 9) + (10\% \times 1) + (50\% \times 5) + (5 \times 10\% \times 1) \\ &= (1000\% \times 10) + 810\% + 10\% + 250\% + 50\% \\ &= 10000\% + 1120\% = 11120\% \end{aligned}$$

$$\text{Total de pesos} = (10 \times 10) + 9 + 1 + 5 + 5 = 120 \quad (6)$$

$$\text{Média Ponderada} = \frac{11120\%}{120} \approx \mathbf{92.67\%} \quad (7)$$

c) **Comparação entre Média Simples e Ponderada:**

- ❑ Média Simples (66.67%): Penalizada pelos 6 *labels* raros (10% ou 50%), que representam 33% dos *labels*.
- ❑ Média Ponderada (92.67%): Reflete o domínio dos 10 *labels* frequentes (55.5% do peso total).

Tabela 8 – Comparação de Impacto entre Média Simples e Ponderada

Tipo de Label	% dos <i>labels</i>	% do Peso Total
<i>labels</i> Frequentes (100%)	55.5% (10/18)	83.3% (100/120)
<i>labels</i> Raros	44.5% (8/18)	16.7% (20/120)

A média ponderada é superior porque:

- ❑ Valoriza padrões dominantes: Os 10 *labels* presentes em todos os arquivos contribuem com 83.3% do peso. *labels* presentes na maioria dos arquivos têm maior peso.
- ❑ Reduz sensibilidade a *outliers*: *labels* como BAKON_236 - 240 (10%) têm impacto limitado (1.67% cada). *labels* esparsos não distorcem desproporcionalmente a similaridade.
- ❑ Correspondência intuitiva: Se 83% das ocorrências são *labels* completos, a similaridade deve ser próxima de 90%, não 66%.

A.2.3.3 Coeficiente de *Jaccard* e *Overlap*

O Coeficiente de *Jaccard* é uma métrica estatística que quantifica a similaridade entre dois conjuntos finitos, sendo amplamente utilizado para análise comparativa de redes complexas. Sua formulação é definida por:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \quad (8)$$

onde A e B representam conjuntos de nós de maior centralidade de grau em redes distintas. Este coeficiente varia no intervalo $[0, 1]$, sendo sensível à sobreposição relativa entre os conjuntos, independentemente de suas cardinalidades absolutas. Em análises estruturais, valores elevados indicam compartilhamento significativo de nós críticos entre redes, sugerindo estabilidade topológica (JACCARD, 1901).

O Coeficiente de *Overlap* (*Szymkiewicz-Simpson*) complementa essa análise ao focar na interseção relativa ao menor conjunto, sendo expresso por:

$$O(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)}. \quad (9)$$

Esta métrica é particularmente relevante para identificar a proporção de nós centrais compartilhados, mesmo em redes com tamanhos heterogêneos. Ambos os coeficientes são fundamentais para avaliar a robustez de redes frente a perturbações estruturais (INMAN; JR, 1989).

a) Média Global do Coeficiente de *Jaccard*.

Definição: A média aritmética dos Coeficientes de *Jaccard* calculados para todos os pares de arquivos analisados. Esta métrica agrega informações locais (pares) em uma medida global, permitindo avaliar a estabilidade conjunta do sistema frente a variações estruturais (PN; STEINBACH; KUMAR, 2005).

Formulação:

$$\bar{J} = \frac{1}{C(N, 2)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N J(A_i, A_j), \quad (10)$$

onde N é o número total de arquivos e $C(N, 2)$ representa combinações binárias.

Interpretação:

- $\bar{J} < 0, 3$: Baixa estabilidade global (fragmentação topológica).
- $0, 3 \leq \bar{J} < 0, 7$: Estabilidade moderada (transição crítica).
- $\bar{J} \geq 0, 7$: Alta estabilidade (core estrutural preservado).

Relevância: A média global transcende análises pareadas, oferecendo uma perspectiva sistêmica sobre a consistência dos nós críticos em múltiplas configurações de rede.

b) **Índice de Concordância de *Fleiss* (Adaptado).**

Definição: Adaptação do Índice *Kappa* de *Fleiss* para avaliar concordância multivariada na seleção de nós críticos entre arquivos, corrigindo a concordância esperada por acaso.

Formulação:

$$\kappa_F = \frac{\bar{J} - J_e}{1 - J_e}, \quad (11)$$

onde J_e é o Coeficiente de *Jaccard* esperado sob aleatoriedade, calculado mediante a frequência observada de cada nó nos arquivos (FLEISS, 1971).

Interpretação:

- $\kappa_F < 0, 2$: Concordância insignificante (seleção aleatória).
- $0, 2 \leq \kappa_F < 0, 6$: Concordância parcial (viés estrutural detectável).
- $\kappa_F \geq 0, 6$: Concordância substancial (mecanismo seletivo consistente).

Interrelação com a Média Global: Enquanto $\bar{J}$ quantifica a similaridade média, κ_F discrimina se tal similaridade excede padrões aleatórios, sendo crítico para validar a significância estatística da estabilidade observada. Em cenários com alta heterogeneidade de arquivos, κ_F é preferível por isolar efeitos de viés amostral.

A sinergia entre essas métricas permite uma avaliação estratificada: a média global identifica padrões macroscópicos de estabilidade, enquanto o Índice de *Fleiss* valida sua significância estatística, essencial para distinguir entre consistência estrutural genuína e artefatos de amostragem.

c) **Exemplo Ilustrativo de cálculo dos coeficientes de *Jaccard*, *Overlap*, Média Global de *Overlap* e Índice de *Fleiss*:**

Considere três arquivos (A , B , C) representando redes com nós de maior grau:

$$A = \{1, 2, 3\} \quad (\text{top 3 nós})$$

$$B = \{2, 3, 4\} \quad (\text{top 3 nós})$$

$$C = \{3, 4, 5\} \quad (\text{top 3 nós})$$

Cálculo dos Coeficientes:

□ *Jaccard* entre A e B :

$$J(A, B) = \frac{|\{2, 3\}|}{|\{1, 2, 3, 4\}|} = \frac{2}{4} = 0,5$$

□ *Overlap* entre A e B :

$$O(A, B) = \frac{|\{2, 3\}|}{\min(3, 3)} = \frac{2}{3} \approx 0,67$$

□ *Jaccard* entre A e C :

$$J(A, C) = \frac{|\{3\}|}{|\{1, 2, 3, 4, 5\}|} = \frac{1}{5} = 0,2$$

□ *Jaccard* entre B e C :

$$J(B, C) = \frac{|\{3, 4\}|}{|\{2, 3, 4, 5\}|} = \frac{2}{4} = 0,5$$

Média Global de *Jaccard*:

$$\bar{J} = \frac{0,5 + 0,2 + 0,5}{3} = 0,4 \quad (\text{Estabilidade moderada})$$

Índice de *Fleiss* Adaptado:

□ Frequência dos nós:

$$\text{Nó 1 : } 1/3; \quad \text{Nó 2 : } 2/3; \quad \text{Nó 3 : } 3/3;$$

$$\text{Nó 4 : } 2/3; \quad \text{Nó 5 : } 1/3$$

□ Coeficiente de *Jaccard* esperado (J_e):

$$J_e = \frac{\sum(p_i^2)}{\sum p_i} = \frac{(1/3)^2 + (2/3)^2 + 1^2 + (2/3)^2 + (1/3)^2}{3} \approx 0,54$$

□ Índice κ_F :

$$\kappa_F = \frac{0,4 - 0,54}{1 - 0,54} \approx -0,25 \quad (\text{Concordância insignificante})$$

Interpretação: A média de *Jaccard* ($\bar{J} = 0,4$) indica estabilidade moderada, mas o κ_F negativo sugere que a sobreposição observada é inferior ao esperado por acaso. Isso implica inconsistência na seleção de nós críticos entre os arquivos, possivelmente devido a variações estruturais nas redes.

A.2.3.4 Correlação de *Kendall Tau*

A **Correlação de *Kendall Tau*** (τ) quantifica a concordância ordinal entre dois rankings de nós, sendo crítica para avaliar a estabilidade hierárquica de redes. Foca nos **Top_*N* nós** de maior grau, comparando sua ordenação entre arquivos. É definida por:

$$\tau = \frac{C - D}{\binom{n}{2}}, \quad (12)$$

onde:

□ C : Número de pares concordantes;

□ D : Número de pares discordantes;

□ $\binom{n}{2}$: Combinações possíveis de pares para n nós (KENDALL, 1938).

Pares Concordantes: Mantêm a mesma ordem relativa em ambos rankings:

$$\begin{aligned} \text{Concordante} \iff & (\text{Rank}_i(X) > \text{Rank}_i(Y) \wedge \text{Rank}_j(X) > \text{Rank}_j(Y)) \\ & \vee (\text{Rank}_i(X) < \text{Rank}_i(Y) \wedge \text{Rank}_j(X) < \text{Rank}_j(Y)) \end{aligned}$$

Pares Discordantes: Apresentam ordem relativa invertida entre rankings:

$$\begin{aligned} \text{Discordante} \iff & (\text{Rank}_i(X) > \text{Rank}_i(Y) \wedge \text{Rank}_j(X) < \text{Rank}_j(Y)) \\ & \vee (\text{Rank}_i(X) < \text{Rank}_i(Y) \wedge \text{Rank}_j(X) > \text{Rank}_j(Y)) \end{aligned}$$

a) **Exemplo Ilustrativo:** Considere dois rankings de 4 nós (Top-4) de arquivos distintos:

□ Arquivo 1: $A > B > C > D$;

□ Arquivo 2: $B > A > D > C$.

□ Par (A,B): Discordante ($A > B$ vs $B > A$);

- ❑ Par (A,C): Concordante ($A > C$ vs $A > C$);
- ❑ Par (A,D): Concordante ($A > D$ vs $A > D$);
- ❑ Par (B,C): Concordante ($B > C$ vs $B > C$);
- ❑ Par (B,D): Concordante ($B > D$ vs $B > D$);
- ❑ Par (C,D): Discordante ($C > D$ vs $D > C$).

Cálculo do τ :

$$C = 4, \quad D = 2, \quad \binom{4}{2} = 6$$

$$\tau = \frac{4 - 2}{6} = \frac{2}{6} \approx 0,333 \quad (13)$$

b) Interpretações:

Interpretação Individual:

- ❑ $\tau = 0,333$ indica concordância parcial (33,3% dos pares mantêm ordem);
- ❑ 2 pares discordantes sugerem inconsistências locais.

Interpretação Global:

A média de τ entre todos pares de arquivos avalia estabilidade sistêmica:

- ❑ $\bar{\tau} \approx 0$: Hierarquia inconsistente (Top_N variável entre arquivos);
- ❑ $\bar{\tau} > 0,5$: Ordenação robusta (nós críticos mantêm posições relativas).

Valores Críticos:

- ❑ $\tau \approx 1$: Concordância perfeita;
- ❑ $\tau \approx 0$: Aleatoriedade ordinal (independência entre rankings);
- ❑ $\tau < 0$: Tendência inversa (ranking invertido).

Contexto em Análise de Redes:

- ❑ τ baixo ($< 0,3$): Instabilidade estrutural (ex.: conexões dinâmicas alteram hierarquia);
- ❑ τ moderado ($0,3 \leq \tau < 0,7$): Variações localizadas (ex.: flutuações em nós periféricos);
- ❑ τ alto ($\geq 0,7$): Core hierárquico preservado (nós centrais estáveis).

c) **Variações da Métrica:**

- O **Kendall Ratio Positive** mede a proporção de pares concordantes (ou positivos) em relação ao total de pares possíveis. Em cenários onde a consistência geral é mais importante do que a magnitude da discordância é uma métrica mais indicada de avaliação.

$$\text{Kendall Ratio Positive} = \frac{\text{número de pares concordantes}}{\text{número total de pares}} \quad (14)$$

Esta métrica reflete a proporção de pares que seguem a mesma tendência nos dois rankings comparados. Valores próximos de 1 indicam alta concordância, enquanto valores próximos de 0 indicam pouca ou nenhuma concordância. É particularmente útil para entender a “força geral” da concordância entre os rankings sem considerar diretamente os pares discordantes. Possui como limitação ignorar completamente os pares discordantes, o que pode mascarar problemas em rankings com muitas inconsistências.

- O **Median Kendall** é a mediana dos coeficientes de correlação de *Kendall* calculados para diferentes subconjuntos ou pares de rankings. A mediana é menos sensível a valores extremos (*outliers*) do que a média, tornando-a uma medida robusta para representar a tendência central da correlação ordinal. O *Median Kendall* é especialmente relevante em cenários onde os dados podem conter *outliers* ou quando a distribuição dos coeficientes de *Kendall* não é simétrica.

A.2.3.5 Teste de *Kolmogorov-Smirnov*

O Teste de *Kolmogorov-Smirnov* (KS) avalia diferenças entre **distribuições** empíricas de graus nodais (AN, 1933). Para múltiplos arquivos, utiliza-se uma adaptação multivariada, denominada **Teste de KS para Múltiplas Amostras**:

Definição Formal:

- Estatística KS para k Amostras:

$$D_{\text{global}} = \max_x |F_{\text{global}}(x) - F_i(x)| \quad (15)$$

onde:

- $F_{\text{global}}(x) = \frac{1}{k} \sum_{i=1}^k F_i(x)$ (distribuição acumulada global);
- $F_i(x)$: Função de distribuição empírica do arquivo i .

a) **Construção da Função de Distribuição Acumulada (CDF):**

Para um arquivo com m nós, a CDF $F_i(x)$ é calculada a partir dos graus dos nós:

1. Passo 1: Extrair os graus de todos os nós:

$$\text{Graus} = [34, 289, 288, 289, 45] \quad (\text{Exemplo dos labels citado na seção A.2.1.2}).$$

2. Passo 2: Ordenar os graus em ordem crescente:

$$\text{Graus ordenados} = [34, 45, 288, 289, 289].$$

3. Passo 3: Calcular a CDF empírica:

$$F_1(x) = \begin{cases} 0,2 & \text{se } x \geq 34 \\ 0,4 & \text{se } x \geq 45 \\ 0,6 & \text{se } x \geq 288 \\ 1,0 & \text{se } x \geq 289 \end{cases}$$

b) **Exemplo Ilustrativo:**

Considere 3 arquivos com Top-3 nós:

$$\text{Arquivo1}(\text{Top} - 3) : [289, 289, 288]$$

$$\text{Arquivo2}(\text{Top} - 3) : [300, 295, 290]$$

$$\text{Arquivo3}(\text{Top} - 3) : [285, 284, 280]$$

□ CDFs Individuais:

$$F_1(x) = [0, 33; 0, 66; 1, 0] \quad (\text{valores em } 288, 289, 289)$$

$$F_2(x) = [0, 33; 0, 66; 1, 0] \quad (\text{valores em } 290, 295, 300)$$

$$F_3(x) = [0, 33; 0, 66; 1, 0] \quad (\text{valores em } 280, 284, 285)$$

□ CDF Global:

$$F_{\text{global}}(x) = \frac{1}{3}(F_1 + F_2 + F_3) = [0, 33; 0, 66; 1, 0]$$

□ Cálculo de D_{global} :

$$D_{\text{global}} = \max \begin{pmatrix} |0, 33 - 0, 33|, |0, 66 - 0, 66|, |1, 0 - 1, 0|, \\ |0, 33 - 0, 33|, |0, 66 - 0, 66|, |1, 0 - 1, 0|, \\ |0, 33 - 0, 33|, |0, 66 - 0, 66|, |1, 0 - 1, 0| \end{pmatrix} = 0$$

❑ Caso 1: Se $D_{\text{global}} = 0$ (como no exemplo), todos os arquivos têm CDFs idênticas para os Top_N nós.

❑ Caso 2: Se um arquivo tiver $Top-3 = [100, 90, 80]$, sua CDF será:

$$F_4(x) = [0, 33; 0, 66; 1, 0] \quad (\text{valores em } 80, 90, 100)$$

Neste caso, $D_{\text{global}} = \max(|0, 33 - 0, 33|, \dots, |1, 0 - 1, 0|) = 0$ mesmo com graus diferentes, pois a forma da CDF é preservada.

❑ Caso 3: Considere três arquivos com distribuições de graus:

$$\text{Arquivo1} : F_1(x) = [0.2, 0.5, 0.8, 1.0]$$

$$\text{Arquivo2} : F_2(x) = [0.3, 0.6, 0.9, 1.0]$$

$$\text{Arquivo3} : F_3(x) = [0.1, 0.4, 0.7, 1.0]$$

– Distribuição Global:

$$F_{\text{global}}(x) = \frac{1}{3}(F_1 + F_2 + F_3) = [0.2, 0.5, 0.8, 1.0]$$

– Cálculo de D_{global} :

$$D_1 = \max(|0.2 - 0.2|, |0.5 - 0.5|, |0.8 - 0.8|, |1.0 - 1.0|) = 0$$

$$D_2 = \max(|0.3 - 0.2|, |0.6 - 0.5|, |0.9 - 0.8|, |1.0 - 1.0|) = 0.1$$

$$D_3 = \max(|0.1 - 0.2|, |0.4 - 0.5|, |0.7 - 0.8|, |1.0 - 1.0|) = 0.1$$

$$D_{\text{global}} = \max(0, 0.1, 0.1) = 0.1$$

c) Interpretação:

❑ Valores Críticos:

– $D_{\text{global}} < 0.1$: Consistência distribucional (H_0 não rejeitada);

– $D_{\text{global}} \geq 0.1$: Inconsistência significativa (H_0 rejeitada).

❑ $D_{\text{global}} = 0.1$ indica limiar crítico. Sugere heterogeneidade estrutural entre arquivos.

d) Aplicação em Redes:

❑ Consistência Global: $D_{\text{global}} < 0.05$ implica topologia preservada;

❑ Alerta Estrutural: $D_{\text{global}} \geq 0.15$ sinaliza mudanças drásticas na conectividade;

❑ Exemplo Prático: Se 10 arquivos têm $D_{\text{global}} = 0.18$, conclui-se instabilidade na distribuição de graus.

O teste KS para Top_N nós só detecta diferenças na **forma da distribuição**, não nos valores absolutos. Por isso, se diz que é uma métrica de avaliação da distribuição.

A.3 Resultados

Todos os arquivos gerados durante esta pesquisa estão disponibilizados publicamente por meio do Google Drive no seguinte endereço: <https://drive.google.com/drive/folders/1WP_AtS7K4cxgYS-6dC8V5mp-G4cH4Pv4?usp=sharing>.

A.3.1 Organização do Material Disponibilizado

Os arquivos disponibilizados compreendem:

□ Arquivos *.dot*:

- 50 arquivos gerados das implementações do código *Genppi* aplicado ao genoma da bactéria *Buchnera Aphidicola* no modo **Machine Learning**, armazenados no diretório `modoML`.
- 50 arquivos gerados no modo **Features**, armazenados no diretório `modoF`.

□ Arquivos *.csv*:

- 50 arquivos *.csv* gerados a partir dos arquivos *.dot* no *Gephi*, contendo as métricas de *Degree*, *Bridging Centrality* e *Betweenness Centrality*, armazenados no diretório `MLCSV`.
- 50 arquivos *.csv* correspondentes ao modo **Features**, armazenados no diretório `FCSV`.

□ Código Python:

- O arquivo `genppigephy.py`, responsável por gerar os resultados das implementações.

□ Resultados em PDF:

- `Resultados_Finais_Features.pdf`: Contém 901 páginas referentes a 90 implementações no modo **Features** (numeradas de 01 a 90).
- `Resultados_Finais_ML.pdf`: Contém 901 páginas referentes a 90 implementações no modo **Machine Learning** (numeradas de 91 a 180).

□ Estrutura dos Resultados:

- Cada implementação, em ambos os modos, é composta por:
 - * Página de título.

- * 2 páginas contendo os **Top-7 labels** das proteínas (nós) com as maiores métricas de referência, formatadas para impressão.
- * 1 página exibindo os **Top_N** com percentual de presença individual dos nós, incluindo a média global de presença (média ponderada).
- * 2 páginas contendo cálculos de similaridade, correlação e testes de distribuição entre os arquivos, com as seguintes métricas:
 - Coeficiente de *Jaccard*;
 - Coeficiente de *Overlap*;
 - Teste de *Kolmogorov-Smirnov* (*KS_p*);
 - Correlação de *Kendall Tau*.
- * 4 páginas contendo mapas de calor das métricas:
 - Coeficiente de *Jaccard*.
 - Teste de *Kolmogorov-Smirnov*.
 - Coeficiente de *Overlap*.
 - Correlação de *Kendall Tau*.

□ Arquivos de Dados Gerais:

- `DadosGeraisGenppiModoFeatures.pdf`: Contém o resumo geral das métricas globais para o modo **Features**.
- `DadosGeraisGenppiModoML.pdf`: Contém o resumo geral das métricas globais para o modo **Machine Learning**.
- As colunas dos arquivos incluem: *Jaccard*, *Fleiss*, *Kendall*, *KS_p*, *Top_N*, *NrArq*, *Metrica*, *Nr Impl*.

A.3.2 Média Global de Presença Ponderada

A média global de presença ponderada citada na Seção A.2.3.2 é apresentada nos arquivos `Resultados_Finais`, tanto no modo **Features** quanto no **Machine Learning**, conforme ilustrado na Figura 15.

Cada exibição da implementação inicia com um cabeçalho que é padronizado e exibido em todas as páginas para cada implementação, a exceção do título e ele contém:

□ Número da implementação e dados de configuração:

- *Top_N*: Valor numérico definido para seleção de nós;
- *Nr_Arquivos*: Quantidade de arquivos processados;

- Métrica Topológica: Critério métrico adotado (ex: *Betweenness Centrality*).
- Modo: *Machine Learning* ou *Features*.

Implementação Número 68		
Parâmetros:	Top_N = 20	Modo: Features
	Número de arquivos = 40	Métrica escolhida: Bridgingcentrality
Top Nós com Percentual		
TopN Nós		
% Presença	Label	Arquivos
100.00 %	BAKON_111	00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
100.00 %	BAKON_097	00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
100.00 %	BAKON_354	00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
100.00 %	BAKON_561	00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
100.00 %	BAKON_598	00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
100.00 %	BAKON_392	00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
...	continua	... (amostra formatada para impressão)
Média global de presença dos nós (Ponderada): 99.76%		

Figura 15 – Exemplo de saída Percentual de Presença de nós.

Além do cabeçalho, ainda é exibido:

❑ Dados Individuais de presença dos primeiros *Top_N* nós:

- Percentual individual de presença por nó (formato: 000.00%);
- Label da proteína associada;
- Identificação dos arquivos onde o nó aparece nos *Top_N*.

A formatação da página prioriza a compactação para impressão, exibindo apenas uma amostra representativa quando *Top_N* ou o número de arquivos for muito grande. Para configurações extremas, a visualização é limitada a exibição de um mínimo de 5 nós. Em todos os casos, **a média global ponderada é exibida ao final da visualização** desta página.

A.3.3 Métricas de Similaridade, Correlação e Distribuição

A exibição dos cálculos de similaridade, correlação e dos testes de distribuição se dá conforme apontado na Figura 16.

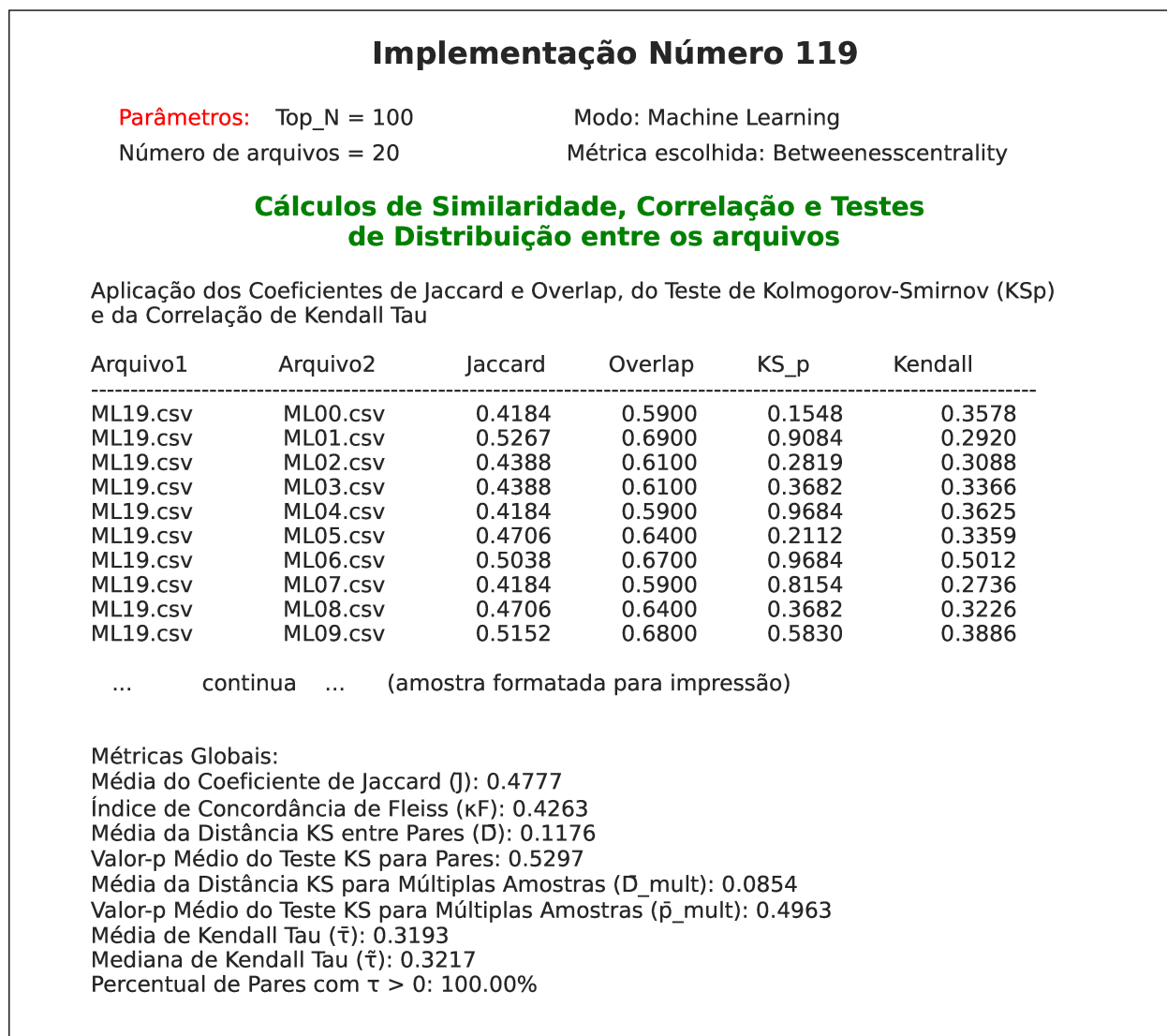


Figura 16 – Exemplo de saída dos Cálculos de Similaridade, Correlação e Distribuição.

O mesmo princípio de compactação para visualização de uma amostra reduzida é aplicado. Isso decorre do fato da possibilidade da quantidade de pares ordenados ser extremamente numeroso.

Na implementação da Figura, temos o total de 190 pares de arquivos. Para n arquivos, o número de pares é dado por:

$$C(n, 2) = \frac{n!}{2!(n-2)!} \quad (\text{ex: } n = 50 \Rightarrow 1.225 \text{ pares}) \quad (16)$$

Calcula-se individualmente as métricas adiante mencionadas para os pares ordenados de arquivos.

- ☐ Coeficiente de *Jaccard*;
- ☐ Coeficiente de *Overlap*;
- ☐ Teste de *Kolmogorov-Smirnov*;
- ☐ Correlação de *Kendall Tau*;

Ao final são exibidas as médias globais do Coeficiente de *Jaccard*, *Kendall Tau* e do Teste KS_p mas também outras medidas de importância global que considera todo o conjunto de arquivos. São elas:

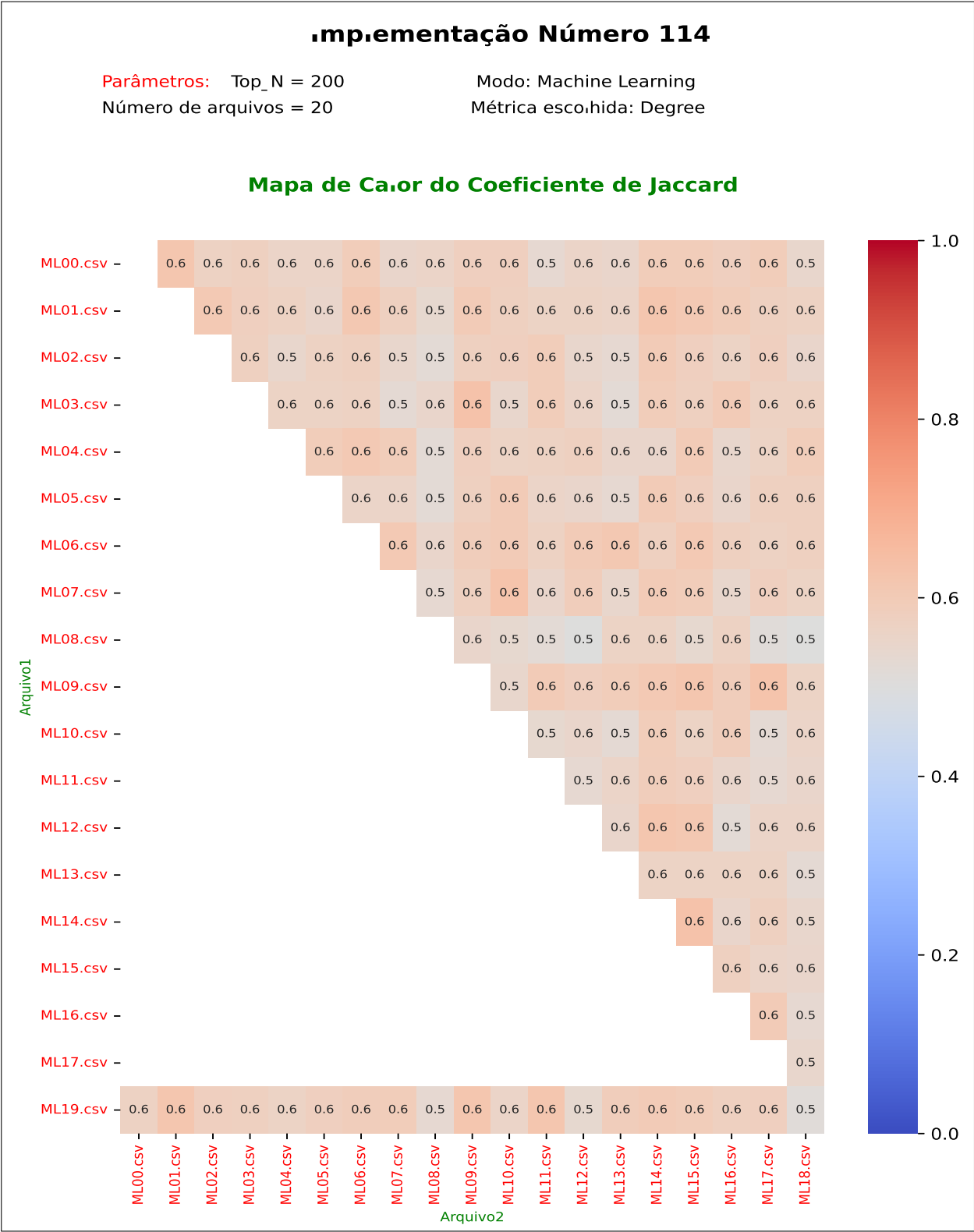
- ☐ Índice de Concordância de *Fleiss*;
- ☐ Teste de KS_p para múltiplas amostras;
- ☐ Mediana de *Kendall Tau*;
- ☐ Proporção de pares de arquivos que tem correlação de *Kendall* positiva ($\tau > 0$).

A.3.3.1 Mapas de Calor

Para cada uma das métricas individuais discutidas na Seção A.3.3, foi implementado um mapa de calor (*heatmap*) visando uma visualização abrangente das diferentes implementações. As Figuras 17, 18, 19 e 20 ilustram exemplos de testes distintos, correspondendo a cada métrica analisada.

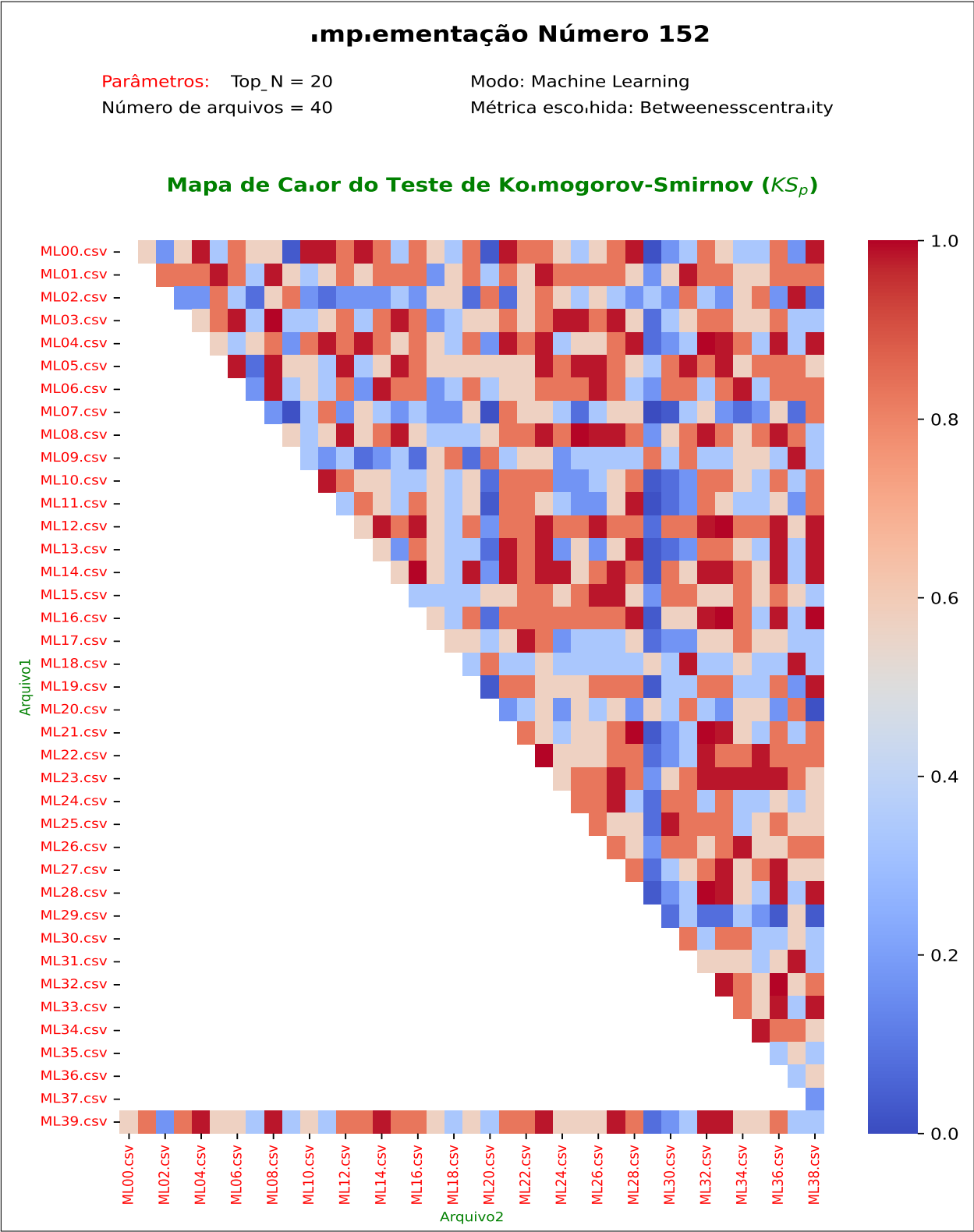
Os intervalos das métricas são normalizados entre 0 e 1, exceto para o coeficiente de *Kendall Tau*, que varia entre -1 e 1 . A escala cromática dos mapas de calor foi padronizada para garantir consistência visual: tons mais intensos na escala vermelha indicam valores próximos ao máximo da métrica, enquanto tons mais frios na escala azulada representam valores próximos ao mínimo.

Nos exemplos apresentados, observa-se a variação de parâmetros entre implementações. Para configurações com número reduzido de arquivos (inferior a 20), os valores numéricos podem ser exibidos internamente ao *heatmap* com tamanho de fonte legível. Entretanto, para configurações com mais de 20 arquivos, recomenda-se a redução do tamanho da fonte ou a supressão dos valores numéricos, priorizando a interpretação visual por meio da escala de cores.



- Notas:
- a) Desejável: Próximo a 1.0 (Conjuntos idênticos ou quase idênticos);
 - b) Máximo: 1.0 - (Similaridade Total);
 - c) Mínimo: 0.0 - (Nenhuma interseção entre os Arquivos).

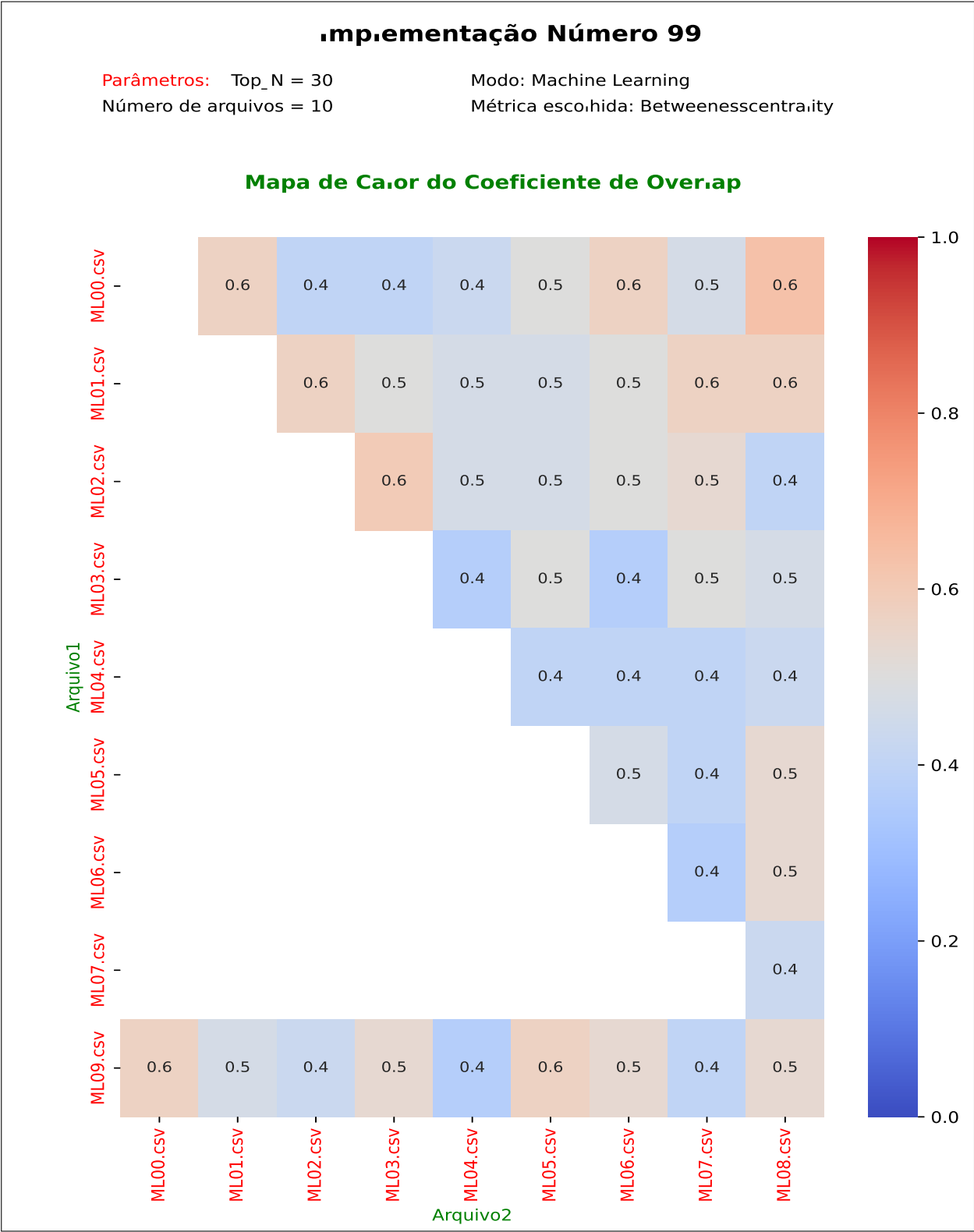
Figura 17 – Comparação do Coeficiente de Jaccard entre arquivos - Exemplo.



Notas:

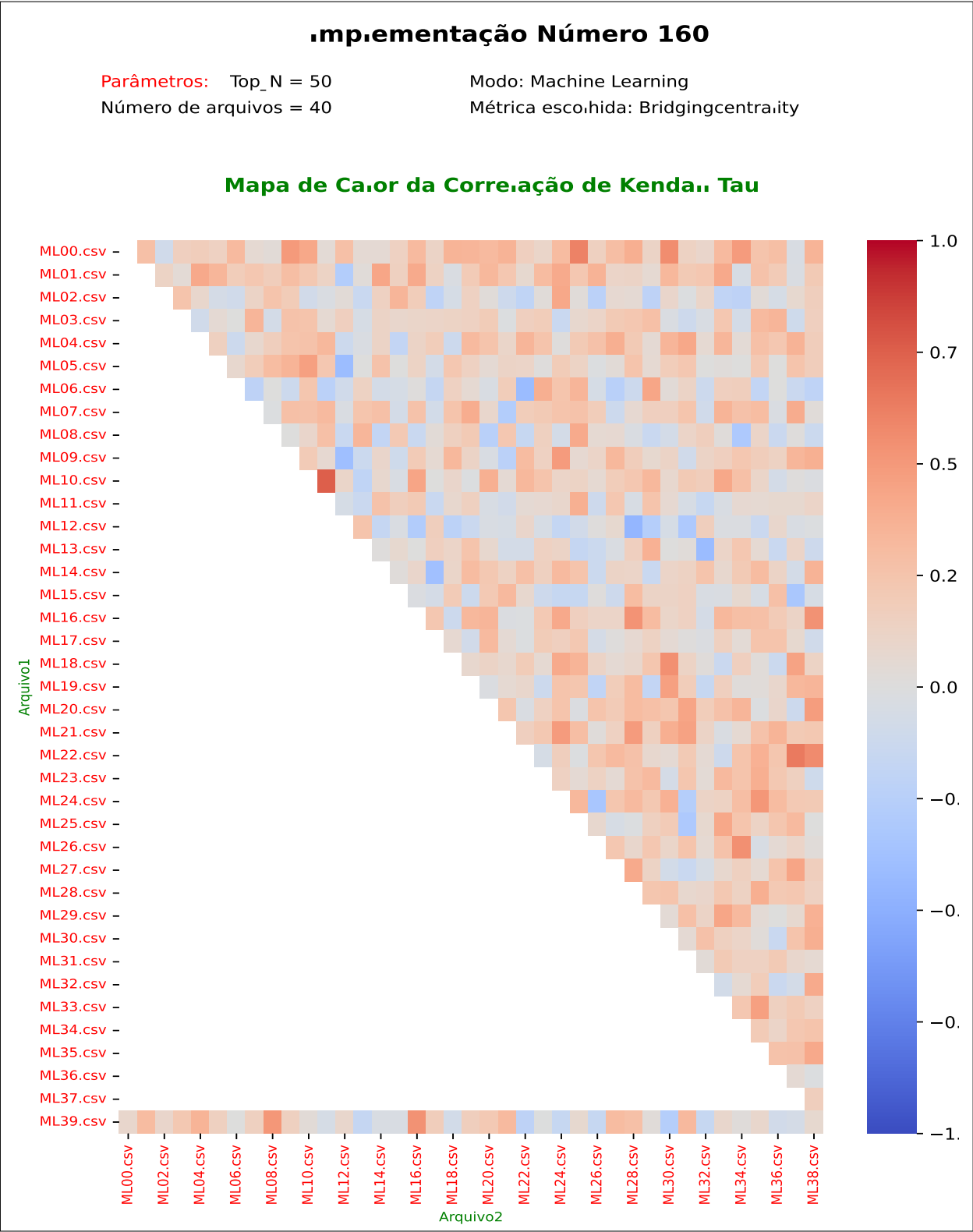
- a) Desejável: Próximo a 0.0;
- b) Máximo: 1.0 - (Distribuições completamente distintas);
- c) Mínimo: 0.0 - (Distribuições idênticas).

Figura 18 – Teste de *Kolmogorov-Smirnov* entre arquivos - Exemplo.



- Notas:
- a) Desejável: Próximo a 1.0 (Conjuntos idênticos ou quase idênticos);
 - b) Máximo: 1.0 - (Similaridade Total);
 - c) Mínimo: 0.0 - (Nenhuma interseção entre os Arquivos).

Figura 19 – Comparação do Coeficiente de *Overlap* entre arquivos - Exemplo.



Notas:

- a) Desejável: Próximo a 1.0 (Alta proporção de pares concordantes);
- b) Máximo: 1.0 - (Pares com correlação positiva);
- c) Mínimo: 0.0 - (Pares com correlação nula).

Figura 20 – Correlação de *Kendall Tau* entre arquivos - Exemplo.

A.3.4 Resumos das Médias Globais

A seguir são apresentados os resultados consolidados das análises de similaridade, correlação e distribuição aplicadas às redes geradas. Os dados fornecem uma visão abrangente do desempenho do *Genppi*. Os dados referentes ao modo *Features* são apresentados na Tabela 9 de forma filtrada. Nela são exibidos apenas os resultados aos quais diferem os dados da convergência completa de 100%. Para todas as demais implementações do modo *Features*, a linha 1 da Tabela os representa.

Tabela 9 – Resultados para o Modo *Features* (Dados Filtrados)

MGP	<i>Jaccard</i>	<i>Fleiss</i>	<i>R-Kendall</i>	<i>KS_p</i>	<i>Top_N</i>	<i>Nr Arq</i>	<i>Met</i>	<i>Implem</i>
100.00%	1.0000	1.0000	1.0000	0.0000	**	**	1	**
099.84%	0.9968	1.0000	1.0000	0.0488	030	040	3	069
099.98%	0.9995	1.0000	1.0000	0.0488	200	040	3	072
099.80%	0.9962	1.0002	1.0000	0.0392	020	050	3	086
099.87%	0.9974	1.0000	1.0000	0.0392	030	050	3	087
099.98%	0.9996	1.0000	1.0000	0.0392	200	050	3	090

Para uma compreensão estruturada dos outros resultados referentes ao modo *Machine Learning*, três Tabelas são disponibilizadas:

- Tabela 10 para a métrica *Degree* (Grau dos Nós);
- Tabela 11 para a métrica *Betweenness Centrality* (Centralidade de Intermediação);
- Tabela 12 para a métrica *Bridging Centrality* (Centralidade de Ponte).

A organização dos dados por métricas topológicas foi escolhida para destacar como cada critério — grau, intermediação e ponteamto — influencia a identificação de nós críticos em redes biológicas, permitindo uma análise comparativa robusta.

A Média Global Ponderada (MGP) é uma das métricas centrais apresentadas, medindo a presença consistente de nós críticos entre as implementações, ponderada pela frequência de aparição em múltiplos arquivos. Essa métrica é complementada pela Média Global de *Jaccard*, que reflete a similaridade média entre conjuntos de nós, isolando variações estruturais e fornecendo uma medida de estabilidade conjunta do sistema.

Além disso, o Índice de *Fleiss* é utilizado para avaliar a concordância entre as implementações, ajustando para o acaso e validando se a similaridade observada excede padrões aleatórios. Uma alta heterogeneidade no Índice de *Fleiss* pode indicar inconsistências ou viés amostral, sendo crítica para isolar variações não significativas.

Os resultados da Razão Positiva de *Kendall Tau* (*R-Kendall*) são apresentados como uma medida de correlação prioritária para este cenário, pois focam na proporção de pares

concordantes, que é mais relevante do que a proporção de pares discordantes na avaliação da estabilidade de rankings de nós. Essa métrica é particularmente útil para validar a robustez do algoritmo *RIS* do *Genppi*, especialmente em cenários com alta variabilidade.

Por fim, o teste de *Kolmogorov-Smirnov* para múltiplas amostras (KS_p) é utilizado para quantificar a divergência entre as distribuições de graus de todos os arquivos, fornecendo uma medida geral que expressa a consistência ou divergência entre as distribuições. Valores baixos de KS_p indicam diferenças estatisticamente significativas, enquanto valores altos sugerem distribuições similares.

Para facilitar a visualização e a comparação entre as métricas, os resultados foram organizados em tabelas separadas, com as implementações ordenadas por sequência ordinal crescente de Top_N (10, 20, ..., 200) e $Nr\ Arq$ (10, 20, ..., 50). Embora a numeração das implementações (coluna *Implem*) não seja totalmente sequencial devido à combinação única de parâmetros, essa ordenação permite uma análise clara do impacto de variações nos parâmetros sobre a consistência e a relevância dos resultados.

Em particular, para um genoma com 559 nós, $TOP_N = 200$ (aproximadamente 36% da rede) atinge valores de MGP de até 75.89%, indicando que as redes biológicas possuem nós críticos altamente redundantes.

A segmentação por métricas topológicas também permite comparar como *Degree*, *Betweenness* e *Bridging* capturam diferentes aspectos da topologia da rede. Enquanto o *Degree* prioriza nós altamente conectados, o *Betweenness* identifica aqueles que atuam como intermediários em caminhos, e o *Bridging* foca em nós que conectam comunidades distintas. Essa abordagem é essencial para entender como variações nos parâmetros do *Genppi* impactam a identificação de nós críticos, especialmente em um genoma como o *Ba_Ak*, onde a complexidade estrutural demanda métricas robustas e complementares.

Os resultados apresentados destacam a importância da escolha de métricas e dos parâmetros para a análise de redes biológicas. A combinação de MGP, *Jaccard*, *Fleiss*, *R-Kendall* e KS_p oferece uma avaliação multidimensional da consistência, similaridade e robustez das redes, sendo fundamental para decisões metodológicas em estudos futuros.

Em seguida, é apresentado na Figura 21, um gráfico que reflete o comportamento das métricas normalizadas em função do parâmetro Top_N para o modo *Machine Learning*. As linhas de tendência destacam padrões importantes, como a convergência da *R-Kendall* para valores máximos, indicando estabilidade nos rankings de nós críticos, e a redução progressiva da distância de *KS*, que sugere maior homogeneidade nas distribuições à medida que Top_N aumenta. Além disso, a saturação da *MGP* e do *Jaccard* em $Top_N \geq 100$ revela um limiar de redundância estrutural nas redes, enquanto oscilações no *Fleiss* sinalizam variações na concordância entre implementações. O gráfico complementa as tabelas oferecendo uma perspectiva dinâmica e visual para a análise parametrizada.

Tabela 10 – Resultados para Métrica *Degree* - Modo *Machine Learning*

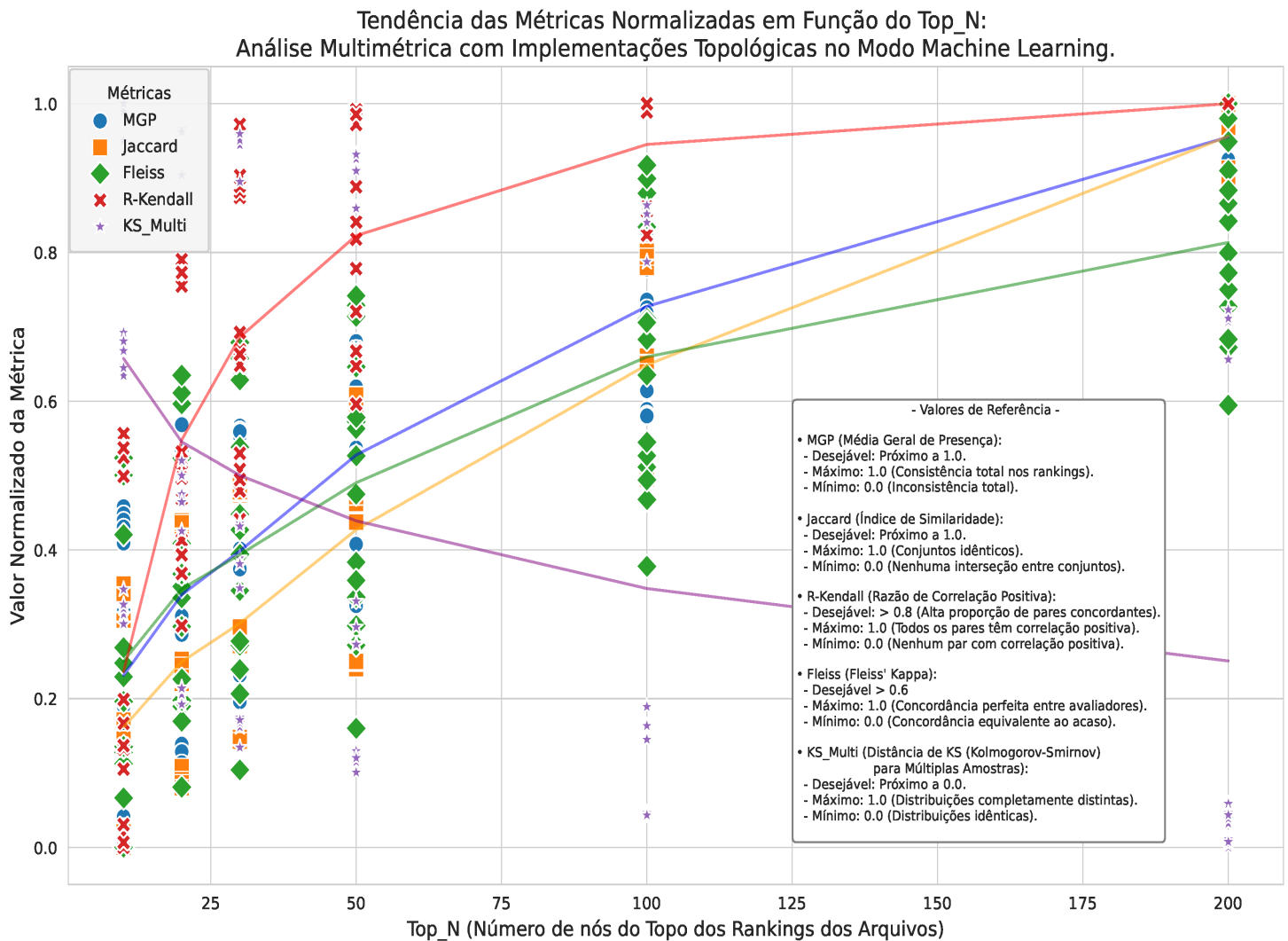
MGP	<i>Jaccard</i>	<i>Fleiss</i>	<i>R-Kendall</i>	<i>KS_p</i>	<i>Top_N</i>	<i>Nr Arq</i>	<i>Met</i>	<i>Implem</i>
033.20%	0.1534	0.0516	0.3111	0.5930	010	010	1	091
039.00%	0.1956	0.0844	0.5778	0.5670	020	010	1	092
040.40%	0.2054	0.1120	0.7556	0.5620	030	010	1	093
052.08%	0.3070	0.1873	0.9111	0.5420	050	010	1	094
061.98%	0.4073	0.2551	1.0000	0.5017	100	010	1	095
075.15%	0.5677	0.3404	1.0000	0.4281	200	010	1	096
027.55%	0.1401	0.0878	0.3632	0.6205	010	020	1	109
035.08%	0.1913	0.1413	0.6105	0.6020	020	020	1	110
038.45%	0.2161	0.1670	0.7368	0.5897	030	020	1	111
048.38%	0.2975	0.2358	0.8737	0.5702	050	020	1	112
059.37%	0.4020	0.3207	0.9895	0.5313	100	020	1	113
073.91%	0.5695	0.4078	1.0000	0.4569	200	020	1	114
025.22%	0.1327	0.1056	0.3103	0.6169	010	030	1	127
032.82%	0.1831	0.1618	0.6069	0.5996	020	030	1	128
037.27%	0.2153	0.1928	0.7195	0.5962	030	030	1	129
046.94%	0.2927	0.2631	0.8552	0.5805	050	030	1	130
058.73%	0.4027	0.3460	0.9908	0.5423	100	030	1	131
073.47%	0.5697	0.4304	1.0000	0.4608	200	030	1	132
023.66%	0.1269	0.1154	0.3141	0.6150	010	040	1	145
031.61%	0.1788	0.1699	0.6000	0.5980	020	040	1	146
036.85%	0.2164	0.2104	0.7205	0.5932	030	040	1	147
046.47%	0.2926	0.2825	0.8256	0.5792	050	040	1	148
058.30%	0.4019	0.3585	0.9923	0.5377	100	040	1	149
073.26%	0.5701	0.4430	1.0000	0.4591	200	040	1	150
023.58%	0.1288	0.1263	0.3380	0.6208	010	050	1	163
031.27%	0.1788	0.1788	0.6278	0.5997	020	050	1	164
036.77%	0.2184	0.2213	0.7322	0.5982	030	050	1	165
046.06%	0.2916	0.2905	0.8237	0.5825	050	050	1	166
057.52%	0.3963	0.3581	0.9910	0.5443	100	050	1	167
072.90%	0.5673	0.4524	1.0000	0.4655	200	050	1	168

Tabela 11 – Resultados para Métrica *Betweenness Centrality* - Modo *Machine Learning*

MGP	<i>Jaccard</i>	<i>Fleiss</i>	<i>R-Kendall</i>	<i>KS_p</i>	<i>Top_N</i>	<i>Nr Arq</i>	<i>Met</i>	<i>Implem</i>
042.00%	0.2233	0.1254	0.6222	0.2290	010	010	2	097
048.90%	0.2784	0.2005	0.8222	0.1685	020	010	2	098
053.07%	0.3173	0.2221	0.9778	0.1360	030	010	2	099
059.44%	0.3804	0.2873	0.9778	0.1172	050	010	2	100
067.38%	0.4695	0.3609	1.0000	0.0828	100	010	2	101
075.89%	0.5781	0.3930	1.0000	0.0608	200	010	2	102
041.40%	0.2441	0.2069	0.6474	0.2362	010	020	2	115
045.60%	0.2753	0.2614	0.8053	0.1809	020	020	2	116
047.93%	0.2951	0.2692	0.9000	0.1587	030	020	2	117
055.89%	0.3680	0.3265	0.9947	0.1334	050	020	2	118
066.30%	0.4777	0.4263	1.0000	0.0854	100	020	2	119
074.79%	0.5811	0.4665	1.0000	0.0670	200	020	2	120
040.87%	0.2482	0.2492	0.6322	0.2406	010	030	2	133
046.60%	0.2923	0.3003	0.8207	0.1759	020	030	2	134
048.78%	0.3106	0.3173	0.9057	0.1513	030	030	2	135
055.31%	0.3699	0.3626	0.9908	0.1281	050	030	2	136
065.85%	0.4792	0.4505	1.0000	0.0867	100	030	2	137
074.54%	0.5837	0.4870	1.0000	0.0640	200	030	2	138
040.29%	0.2481	0.2626	0.6038	0.2436	010	040	2	151
045.76%	0.2890	0.3080	0.8333	0.1800	020	040	2	152
048.47%	0.3118	0.3324	0.9231	0.1546	030	040	2	153
054.60%	0.3666	0.3704	0.9936	0.1253	050	040	2	154
065.68%	0.4806	0.4608	1.0000	0.0841	100	040	2	155
074.22%	0.5823	0.5037	1.0000	0.0631	200	040	2	156
038.94%	0.2401	0.2616	0.6016	0.2550	010	050	2	169
045.63%	0.2908	0.3205	0.8196	0.1806	020	050	2	170
048.31%	0.3128	0.3439	0.9118	0.1567	030	050	2	171
054.08%	0.3639	0.3772	0.9886	0.1281	050	050	2	172
065.17%	0.4769	0.4703	1.0000	0.0850	100	050	2	173
074.18%	0.5834	0.5142	1.0000	0.0650	200	050	2	174

Tabela 12 – Resultados para Métrica *Bridging Centrality* - Modo *Machine Learning*

MGP	Jaccard	Fleiss	R-Kendall	KS_p	Top_N	Nr Arq	Met	Implem
022.00%	0.0753	- 0.0161	0.2889	0.4160	010	010	3	103
029.10%	0.1208	0.0269	0.5333	0.2990	020	010	3	104
031.67%	0.1383	0.0392	0.5556	0.2560	030	010	3	105
038.84%	0.1921	0.0688	0.7778	0.2140	050	010	3	106
055.04%	0.3345	0.1843	0.8889	0.1517	100	010	3	107
072.97%	0.5385	0.2992	1.0000	0.0788	200	010	3	108
015.95%	0.0649	0.0191	0.2053	0.4348	010	020	3	121
023.75%	0.1116	0.0739	0.4421	0.3407	020	020	3	122
027.83%	0.1384	0.0933	0.6263	0.3048	030	020	3	123
035.66%	0.1937	0.1281	0.6789	0.2462	050	020	3	124
051.77%	0.3274	0.2319	0.8895	0.1667	100	020	3	125
071.19%	0.5351	0.3463	1.0000	0.0936	200	020	3	126
015.11%	0.0681	0.0434	0.2184	0.4484	010	030	3	139
022.03%	0.1091	0.0887	0.4966	0.3520	020	030	3	140
026.49%	0.1379	0.1106	0.5862	0.3025	030	030	3	141
033.92%	0.1892	0.1420	0.7195	0.2267	050	030	3	142
050.18%	0.3206	0.2460	0.8851	0.1523	100	030	3	143
070.39%	0.5314	0.3695	1.0000	0.0866	200	030	3	144
014.25%	0.0673	0.0554	0.2295	0.4419	010	040	3	157
021.41%	0.1094	0.1023	0.5179	0.3268	020	040	3	158
026.33%	0.1411	0.1282	0.6090	0.2786	030	040	3	159
033.91%	0.1934	0.1623	0.7385	0.2131	050	040	3	160
049.80%	0.3211	0.2632	0.8603	0.1415	100	040	3	161
069.93%	0.5291	0.3817	1.0000	0.0825	200	040	3	162
013.36%	0.0645	0.0558	0.2106	0.4219	010	050	3	175
020.47%	0.1061	0.1039	0.4980	0.3208	020	050	3	176
025.61%	0.1388	0.1309	0.5984	0.2743	030	050	3	177
033.69%	0.1943	0.1743	0.7355	0.2135	050	050	3	178
049.64%	0.3220	0.2728	0.8596	0.1420	100	050	3	179
069.63%	0.5273	0.3935	1.0000	0.0852	200	050	3	180

**Notas:**

- Topologia da rede: Cada ponto representa uma implementação diferente com métricas como *Degree*, *Betweenness Centrality* ou *Bridging Centrality*. As variações entre pontos de mesma cor refletem diferenças nas métricas topológicas utilizadas;
- As linhas contínuas conectam os valores médios de cada métrica para diferentes valores de Top_N, indicando a tendência geral de comportamento;
- Linhas ascendentes sugerem melhoria na similaridade conforme mais elementos são considerados (MGP, Jaccard, Fleiss e R-Kendall);
- Linha de KS_Multi tem indicador inverso (descendente), ou seja, enquanto as demais métricas apresenta um comportamento onde valores mais altos indicam melhor desempenho, com o ideal sendo próximo de 1, para KS_Multi, valores mais baixos são preferíveis, e o ideal é atingido quando seu valor se aproxima de 0, indicando distribuições mais similares.
- Todas as métricas foram escalonadas para [0, 1] para comparação direta.

Figura 21 – Comportamento das Métricas Normalizadas em Função do Top_N no modo *Machine Learning*.

A.4 Avaliação

A.4.1 Modo *Features*

No modo *Features*, as redes geradas pelo *Genppi* apresentaram uma consistência estrutural quase perfeita. Das 90 implementações analisadas, apenas 5 tiveram resultados diferentes de 100% para a Média Global Ponderada (MGP), com o menor valor registrado sendo 99,87%. Para outras métricas, também apresentou resultado otimizado. Isso indica que a estrutura topológica da rede permaneceu praticamente inalterada em todas as implementações, reforçando a robustez do modo *Features* mesmo com o uso do algoritmo *RIS* para gerar redes estáveis e previsíveis. Portanto, a análise detalhada concentra-se no modo *Machine Learning*, onde foram observadas variações significativas nas métricas.

A.4.2 Modo *Machine Learning*

A.4.2.1 Análise por Número de Arquivos (*Nr Arq*)

A variação do *Nr Arq* impacta as métricas estatísticas de forma distinta, conforme a métrica topológica e o *Top_N*:

- *Fleiss Kappa*: Aumenta consistentemente com o *Nr Arq* para todas as métricas topológicas. Exemplo:

- *Degree* (*Top_N=200*): 0.3404 (*Nr Arq=10*) → 0.4524 (*Nr Arq=50*).

- MGP: Decresce sistematicamente, sugerindo que mais arquivos introduzem ruído em redes já estáveis. Exemplo:

- *Betweenness Centrality* (*Top_N=200*): 75,89% → 74,18%.

- *Jaccard* e *R-Kendall*: Comportamento errático, imprevisível, irregular ou caótico. Exemplo:

- *Bridging Centrality* (*Top_N=30*): R-Kendall varia entre 0.5556, 0.6263, 0.5862, 0.6090 e 0.5984 à medida que aumenta o número de arquivos.

- para a mesma métrica topológica *Degree* (*Top_N=30* tende a crescer) em contrapartida para (*Top_N=10* tende a diminuir.)

- *KS_p*: Sem tendência definida. Exemplo:

- *Degree* (*Top_N=200*): *KS_p* aumenta de 0.4281 para 0.4655.

- *Bridging Centrality* (*Top_N=100*): *KS_p* diminui de 0.1517 para 0.1420.

Pode-se concluir que em relação ao número de arquivos:

- ❑ *Nr Arq* elevado reduz a MGP e aumenta o Fleiss, mas introduz instabilidade em Jaccard e R-Kendall.
- ❑ O *KS_p* não é sensível ao número de arquivos, variando aleatoriamente.

A.4.2.2 Análise por *Top_N*

O parâmetro *Top_N* (quantidade de nós críticos analisados) é o fator mais influente nas métricas de consistência:

- ❑ Para todas as métricas (*Degree*, *Bridging Centrality* e *Betweenness Centrality*), o aumento do *Top_N* elevou a Média Geral de Presença (MGP) (ex: Métrica 2: de 22,33% para 75,89%), corroborando a afirmação de que valores altos de *Top_N* reduzem a sensibilidade à aleatoriedade.
- ❑ *R-Kendall*: Atinge **1.0** para $Top_N \geq 100$, mostrando convergência total nos rankings de nós críticos.
- ❑ *KS_p*: Aumento ligeiro com algumas exceções.
- ❑ *Fleiss*: O índice de *Fleiss* **teórico** varia de **-1** (discordância extrema) a **1** (concordância perfeita) e no gráfico, é importante considerar que tal medida foi normalizada para o intervalo [0,1]. Nos dados de análise apresentados, ele varia de -0.0161 (único caso) qual significa concordância pior que o acaso a 0.5142, indicando que amostras maiores (*Top_N*=200) estabilizam a identificação de nós.

O gráfico revela uma tendência logarítmica: ganhos de consistência são significativos até *Top_N*=150, com retornos marginais além desse valor.

A.4.2.3 Comparação entre Métricas Topológicas

A análise comparativa das métricas topológicas (*Betweenness Centrality*, *Degree* e *Bridging Centrality*) revela padrões consistentes e diferenças significativas em seu desempenho, conforme os critérios avaliados.

Em primeiro lugar, observa-se que a ordem hierárquica de desempenho *Betweenness Centrality* > *Degree* > *Bridging Centrality* é predominante na maioria dos cenários. Essa tendência é clara em métricas como a MGP, onde a *Betweenness Centrality* apresenta valores sistematicamente superiores, seguida pelo *Degree* e, por fim, pela *Bridging Centrality*. Por exemplo, para *Top_N*=200 e *Nr Arq*=10, a MGP atinge 75,89% para *Betweenness*, 75,15% para *Degree* e 72,97% para *Bridging*. Padrão semelhante repete-se em *Fleiss* e

Jaccard, com a *Betweenness* liderando e a *Bridging* ocupando a última posição. Apenas no *R-Kendall* há situações pontuais em que *Degree* e *Betweenness* se equivalem, como em *Top_N=100* e *Nr Arq=10*, onde ambas atingem 1.000, contra 0.8889 da *Bridging*. Essas exceções, porém, não invalidam a tendência geral.

A diferença percentual entre *Betweenness* e *Bridging* é a mais expressiva entre todas as combinações de métricas. Em *Top_N=100* e *Nr Arq=10*, por exemplo, a MGP da *Betweenness* (67,38%) supera a da *Bridging* (55,04%) por mais de 12 pontos percentuais, indicando comportamentos estruturais distintos. Essa disparidade sugere que a *Betweenness Centrality* identifica nós críticos com maior precisão, enquanto a *Bridging* é mais sensível a flutuações na topologia da rede.

Quanto à estabilidade, os dados não corroboram a hipótese de que a *Bridging Centrality* seja a métrica mais estável. Pelo contrário, o *KS_p* — que mede a similaridade entre distribuições — revela que a *Betweenness* apresenta valores consistentemente baixos (ex.: 0.0608 para *Top_N=200*), indicando distribuições distintas e menor aleatoriedade. Já a *Bridging* oscila entre valores intermediários (ex.: 0.0788 para *Top_N=200*), e o *Degree* exibe os maiores *KS_p* (ex.: 0.4281), refletindo maior redundância. Essa variação posicional no *KS_p* (*Betweenness* < *Bridging* < *Degree*) inverte a lógica observada em outras métricas, destacando que a *Betweenness* é a mais estável em termos distributivos.

Por fim, a *Betweenness Centrality* destaca-se como a métrica mais informativa, com a maior média geral de desempenho em critérios como MGP e *Fleiss*. A diferença relativa média entre *Betweenness* e *Bridging* ultrapassa 10% em múltiplos cenários, reforçando um distanciamento informacional significativo. Essa lacuna ressalta a importância de selecionar métricas alinhadas aos objetivos da análise: a *Betweenness* para precisão, o *Degree* para robustez, e a *Bridging* para contextos intermediários.

A.4.2.4 Padrões em Valores Intermediários - Top_N=100

Para *Top_N=100*, que representa 17,89% da rede analisada:

- ❑ MGP: Até 67,38% (Métrica 2), indicando que 20% dos nós capturam padrões estruturais críticos.
- ❑ R-Kendall: 1.0000 em praticamente todas as métricas, mostrando que a aleatoriedade do *RIS* não afeta a hierarquia global.
- ❑ Distância KS para múltiplas amostras: Mantém-se abaixo de 0,15 (exceto para *Degree*), onde valores altos sugerem similaridade distributiva.

Para $Top_N=50$ (8,94% da rede analisada):

- MGP: Atinge até 52,08% (Métrica 1 - Degree), refletindo perda de precisão em redes reduzidas.
- $R\text{-Kendall}$: 0.9111 (Betweenness e Degree), indicando alta concordância mesmo em subamostras menores.
- KS_p : Entre 0.1172 (Betweenness) e 0.5420 (Degree), evidenciando maior aleatoriedade no *Degree*.

A análise de valores intermediários de Top_N revela relações críticas entre granularidade e confiabilidade. Para $Top_N=50$, observa-se que a seleção de nós prioritários captura padrões primários, porém falha em identificar conexões secundárias, como evidenciado pela redução da MGP para 38,84% na métrica *Bridging Centrality*. Em contraste, $Top_N=100$ oferece um equilíbrio mais adequado entre precisão e eficiência, com a MGP atingindo 67,38% para *Betweenness Centrality*, indicando maior capacidade de preservação de estruturas críticas.

Quanto à estabilidade, o KS_p destaca comportamentos distintos: *Betweenness Centrality* apresenta valor de 0.0828 para $Top_N=100$, reforçando sua consistência na identificação de nós, enquanto o *Degree* exibe $KS_p=0.5017$, sugerindo redundância na seleção. Esses padrões orientam escolhas práticas: $Top_N=100$ é recomendado para análises detalhadas com custo computacional moderado, enquanto $Top_N=50$ serve como triagem inicial, porém com limitações na cobertura de padrões complexos.

A.5 Conclusão

A avaliação do impacto do algoritmo *RIS* na geração de redes de interação proteína-proteína demonstrou resultados significativos quanto à sua capacidade de reduzir a complexidade computacional sem comprometer irremediavelmente a integridade estrutural das redes. A análise sistemática, baseada em métricas topológicas e testes estatísticos, revelou padrões distintos entre os modos *Features* e *Machine Learning* (ML), destacando a influência da abordagem metodológica na estabilidade das redes geradas.

No modo *Features*, observou-se consistência estrutural quase absoluta, com a Média Global Ponderada (MGP) atingindo valores próximos a 100% em 85 das 90 implementações. Isso indica que a estratégia baseada em características bioquímicas pré-definidas é robusta mesmo com a aplicação do *RIS*, preservando nós críticos e a topologia da rede. A alta concordância entre arquivos, evidenciada por coeficientes de *Jaccard* ($\geq 0,99$) e Índice

de *Fleiss* adaptado ($\approx 1,0$), reforça que a aleatoriedade introduzida pelo algoritmo não distorceu a representatividade biológica nesse modo.

Em contraste, o modo *Machine Learning* apresentou maior variabilidade, especialmente em métricas como *Bridging Centrality*, onde a MGP oscilou entre 13,36% ($Top_N = 10$) e 74,18% ($Top_N = 200$). Essa flutuação reflete a sensibilidade do ML à fragmentação de perfis filogenéticos, que pode comprometer a conexão entre comunidades funcionais. Entretanto, o aumento progressivo da MGP com o crescimento do Top_N (ex.: 72,90% para $Top_N = 200$) sugere que a redundância estrutural inerente às redes biológicas mitiga parcialmente os efeitos da redução de arestas, preservando um *core* hierárquico estável.

A análise das distribuições de graus, via valores de KS_p (Teste de *Kolmogorov-Smirnov*), revelou heterogeneidade controlada:

- KS_p variou de 0.0608 (diferenças significativas entre as amostras) a 0.6208 (falta de evidência para rejeitar a hipótese nula - distribuição similar), com tendência de queda conforme Top_N aumenta.
- A Correlação de *Kendall Tau* ($\tau = 0,32 - -1,00$) confirmou estabilidade na ordenação de nós críticos para $Top_N \geq 100$, indicando preservação da hierarquia global.

Mapas de calor e médias de *Jaccard* (0,15–0,58) complementaram a análise, mostrando que a similaridade entre redes depende criticamente do Top_N .

Quanto às contribuições metodológicas, o trabalho introduziu uma abordagem para avaliação de redes PPI, integrando métricas clássicas (como *Degree* e *Betweenness*) com adaptações de testes estatísticos (ex.: Índice de *Fleiss* para concordância multivariada). A formulação da Média Global Ponderada destacou-se como uma solução eficaz para contornar distorções causadas por nós esparsos, oferecendo uma medida mais fiel da representatividade estrutural.

Os resultados obtidos têm implicações práticas para estudos de biologia sistêmica. No modo *Features*, a aplicação do *RIS* mostra-se viável para otimização computacional sem perdas significativas. Já no modo *Machine Learning*, recomenda-se o uso de Top_N representativo da rede (≥ 50). Adicionalmente, a disponibilização pública dos arquivos *.dot*, *.csv* e códigos assegura transparência e reprodutibilidade, facilitando futuras investigações.

Em síntese, este trabalho valida a eficácia do algoritmo *RIS* como ferramenta para mitigar a explosão combinatória em redes PPI, equilibrando eficiência computacional e preservação estrutural. As limitações identificadas no modo *Machine Learning* abrem caminho para refinamentos futuros, como a incorporação de pesos biológicos nas subdivisões ou a combinação de múltiplas métricas para análises integradas. Por fim, o trabalho consolida um marco metodológico para avaliação de algoritmos de redução de complexidade em redes biológicas, com aplicações potenciais além do contexto proteico.