UNIVERSIDADE FEDERAL DE SÃO CARLOS
Departamento de Computação

Leonardo Utida Alcantara

Árvore de predição semi-supervisionada para predição de localização subcelular de proteínas

São Carlos - SP

2021

Leonardo Utida Alcantara

Árvore de predição semi-supervisionada para predição de localização subcelular de proteínas

Trabalho de graduação apresentado ao programa de graduação em Engenharia de Computação da Universidade Federal de São Carlos, como parte dos requisitos para obtenção do título de bacharel em Engenharia de Computação

Orientador Prof. Dr. Ricardo Cerri

São Carlos - SP

2021

Este trabalho é dedicado à minha família e aos meus amigos.

Resumo

A localização subcelular de proteínas é uma tarefa de classificação de extrema importância, visto que a localização das proteínas dentro de uma célula está diretamente relacionada com as funções dessas proteínas. Como existe uma gama de proteínas que residem em dois ou mais locais ao mesmo tempo ou que se deslocam entre vários locais dentro da célula, normalmente métodos de classificação multirrótulo (CM) são projetados para atacar esse tipo de problema. Essa abordagem já é bem estabelecida na literatura, porém ela apresenta algumas desvantagens, como por exemplo: (i) a necessidade de um grande número de proteínas com localização subcelular anotada para treinar o classificador; (ii) essa abordagem ignora o fato de que as instâncias não rotuladas podem fornecer informações valiosas para a classificação; e (iii) existem diversas áreas de estudo em que instâncias não rotuladas existem em abundância e o processo para rotular uma instância é custoso e consome muito tempo. Aprendizado Semi-Supervisionado (ASS) é uma subárea do aprendizado de máquina tradicional na qual o algoritmo de predição tenta explorar os dados rotulados e não rotulados ao mesmo tempo. Classificação Semi-Supervisionada é uma subcategoria do ASS que usa os dados não rotulados para melhorar a performance de um classificador que já utiliza dados rotulados no processo de treino. O objetivo principal deste projeto foi desenvolver um classificador multirrótulo semi-supervisionado capaz de usar a quantidade abundante de proteínas não rotuladas disponível para melhorar a previsão da localização subcelular de proteínas. O algoritmo de ASS desenvolvido é baseado no *framework predictive clustering trees* e foi construído, testado e analisado em diversos cenários de ASS com a finalidade de validar se o classificador realmente seria capaz de utilizar as informações das instâncias não rotuladas para melhorar o processo de classificação em diversos conjuntos de dados multirrótulo em bases de dados de localização subcelular de proteínas para 3 diferentes taxonomias: Viridiplantae, Virus and Fungi.

Palavras-chave: Aprendizado de máquina, aprendizado de máquina multirrótulo, bioinformática, classificação de dados, aprendizado semi-supervisionado.

Abstract

Protein subcellular localization is a really important classification task, because the location of proteins inside a cell is directly related to these protein's functions. As there are a lot of proteins that reside at the same time in two or more locations in a cell or move between locations, usually supervised multi-label classification methods are designed to attack this problem. This approach is well-established in the literature; however, it presents some disadvantages such as: (i) the need for a large amount of labeled instances to train the classifier; (ii) this approach ignores the fact that unlabeled instances can provide valuable information for the classification; and (iii) there are a lot of areas in which unlabeled data is abundant but manually labelling an instance is too expensive and time-consuming. Semi-Supervised Learning (SSL) is a subfield of traditional machine learning, in which the learner tries to exploit both labeled and unlabeled data at the same time. Semi-Supervised Classification is a in a subcategory of SSL which uses the available unlabeled data to improve the classification prformance of a classification process that already uses labeled data. The main goal of this project was the develop a semi-supervised multi-label classifier able to use the abundant number of unlabeled proteins to improve the prediction of protein subcellular localization. The SSL algorithm developed in this work is based on the predictive clustering tree framework and it was constructed, tested and analysed in many SSL scenarios in order to test whether or not the classifier was able to use the unlabeled instances to help during the classification process in a set of Multi-Label protein subcellular localization datasets, from 3 different taxonomies: Viridiplantae, Virus and Fungi.

Keywords: Machine learning, multi-label machine learning, bioinformatics, data classification, semi-supervised machine learning.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Neste capítulo são apresentadas a contextualização e a motivação deste trabalho, bem como os objetivos e a estrutura do texto.

1.1 Contextualização

Determinar a localização subcelular de proteínas (LSP) é um processo muito importante para a predição e anotação de funções de proteínas, anotação gênica e na identificação de alvos (*targets*) para medicamentos e vacinas. A LSP é importante principalmente em áreas médicas, pois tem papel fundamental no processo de descobrimento e análise de novos medicamentos (REY; GARDY; BRINKMAN, 2005). Proteínas secretadas e da membrana plasmática, por exemplo, são facilmente acessíveis pelas moléculas de medicamentos devido a suas localizações subcelulares (CAMPBELL, 2001). Outra grande importância está na identificação e estudo de certas doenças, já que proteínas em localizações anormais estão associadas a certas doenças, entre elas o Alzheimer e o câncer (CUI et al., 2004). A LSP é importante em diversos estudos, pois pode indicar como, e em quais locais, as proteínas interagem com outras proteínas ou outras moléculas. Além disso, a localização de uma proteína pode auxiliar no entendimento de sua função celular e pode ajudar a entender os complicados caminhos que regulam os processos biológicos em nível molecular (CHOU; SHEN, 2010).

Cada compartimento da célula tem suas características físicas e químicas particulares. Assim, o pH específico de cada um deles auxilia na predição da conformação de proteínas (como o pH ácido do lisossomo e a hidrofobicidade da membrana celular). Além disso, como as organelas têm funções especializadas, a localização de proteínas ajuda a levantar hipóteses sobre a função e possíveis outras macromoléculas com as quais elas interagem. A localização de proteínas também tem importância no comportamento celular, como a presença de elementos em diferentes localizações, antes e após ativação, em vias de sinalização. Por fim, a localização “anormal” pode ser um marcador para doenças e ter papel importante no diagnóstico de doenças, e no desenvolvimento de terapias (GLORY; MURPHY, 2007).

A localização subcelular de proteínas (LSP) é uma tarefa árdua e que consome muito tempo. Além disso, com a realização de diversos projetos em larga escala de sequenciamento de genomas, cada vez mais sequências de proteínas são descobertas, o que dificulta ainda mais essa tarefa. Desse modo, métodos computacionais de Aprendizado de Máquina (AM) vêm sendo amplamente utilizados para atacar este problema de maneira rápida e precisa. Esses métodos realizam a predição da localização subcelular de proteínas (PLSP) (WAN;

MAK; KUNG, 2012).

1.2 Motivação

Atualmente existem diversos métodos de AM para realizar a PLSP, porém a grande maioria deles é limitado à predição de proteínas simples-rótulo, e desse modo não servem para a predição de proteínas que residem em dois locais simultaneamente ou então proteínas que se deslocam entre múltiplos locais dentro da célula. Esses métodos normalmente excluem esses tipos de proteínas de suas bases de dados, ou se baseiam na suposição de que elas não existem. No entanto essas proteínas são de extrema importância para determinados processos metabólicos, principalmente os que ocorrem em mais de um compartimento da célula (WAN; MAK; KUNG, 2012). Para resolver esse problema são necessárias abordagens que fazem uso de métodos de AM que consigam classificar instâncias únicas (cada proteína) em um conjunto de classes (locais) simultaneamente. Trata-se de uma tarefa desafiadora, e conhecida na literatura de AM como Classificação multirrótulo (CM). A Figura 1 ilustra um problema de CM para proteínas virais. Na figura, cada instância representa o *accession number* de uma proteína, que pode ser associada a duas ou mais classes (locais) simultaneamente.

	Classes				
Proteína	**Membrana celular**	**Citoplasma**	**Retículo endoplasmático**	**Núcleo celular**	**Secretada**
P59632	**X**	**X**			**X**
P52638	**X**		**X**		
Q912Z9		**X**		**X**	**X**

Figura 1 – Exemplificação da Classificação multirrótulo para PLSP.

A CM é usada em diversas áreas do conhecimento, como por exemplo tarefas de categorização de documentos (GONÇALVES; QUARESMA, 2003; LAUSER; HOTHO, 2003; LUO; ZINCIR-HEYWOOD, 2005), diagnósticos médicos (KARALIC; PIRNAT, 1991), classificação de imagens (BOUTELL et al., 2004b; SHEN et al., 2003) e Bioinformática (CLARE; KING, 2001; ELISSEEFF; WESTON, 2001b; ZHANG; ZHOU, 2005; VENS et al., 2008; CERRI et al., 2016), principalmente na área relacionada a classificação de proteínas.

Atualmente, muitos dos classificadores criados para a PLSP são classificadores supervisionados, ou seja, utilizam proteínas rotuladas, para criar suas bases de dados de treinos e então treinar o classificador. Esse tipo de classificação desconsidera as proteínas que não possuem localização rotulada. Normalmente, a maioria dos trabalhos relacionados à PLSP consideram uma proteína rotulada quando ela possui alguma indicação experimental

de que a proteína está associada a uma localização específica. Para facilitar o entendimento, ao longo deste trabalho, proteínas com localização anotada experimentalmente serão chamadas de Proteínas Rotuladas (PRs) e proteínas que não possuem localização anotada experimentalmente serão nomeadas Proteínas não Rotuladas (PNRs).

Embora utilizar apenas instâncias de proteínas rotuladas seja a abordagem mais utilizada, ela apresenta algumas desvantagens:

- PRs são difíceis de obter pois precisam ser anotadas com abordagens experimentais que, como discutido, são processos trabalhosos e demorados;
- essas sequências de proteínas com localização anotadas de forma experimental representam apenas uma pequena parte de todas as sequências presentes nas bases de conhecimento (CAO et al., 2013; XU et al., 2009). PRs representam apenas 6,6 % de todas as instâncias do UniProtKB / Swiss-Prot (versão 2021_03), o maior repositório de informações de proteínas;
- normalmente nem todos as PRs recuperadas das bases de conhecimento podem realmente ser utilizadas pois algumas delas são proteínas homólogas ou muito semelhantes entre si. Caso essas proteínas sejam consideradas no processo de treinamento do classificador, é possível causar problemas de *overfitting*, que é definido pelo uso de modelos ou procedimentos que incluem mais termos do que o necessário ou usam abordagens mais complicadas do que o necessário (HAWKINS, 2004);
- considerar apenas PRs na construção de bases de dados pode levar a conjuntos de dados com número insuficiente de instâncias para realizar uma boa classificação. Por exemplo, este é o caso do conjunto de dados de *benchmark* de vírus usado nos trabalhos de WAN; MAK; KUNG e SHEN; CHOU, que possui apenas 207 proteínas, sendo que apenas 8 delas eram localizadas no “capsídeo viral”.

Métodos supervisionados ignoram o fato de que as PNRs podem fornecer informações valiosas para a classificação. Por exemplo, elas contêm informações sobre a distribuição dos atributos de uma base de dados (CARAGEA et al., 2010; XU et al., 2009). Em situações onde há uma grande quantidade de dados não rotulados e o custo para rotulá-los não é acessível, o Aprendizado Semi-Supervisionado (ASS) pode oferecer estrategias para superar as limitações comentadas acima.

ASS é um paradigma de AM focado em utilizar dados rotulados e não rotulados ao mesmo tempo para o processo de aprendizado de máquina. Nesse trabalho estamos interessados na Classificação Semi-Supervisionada (CSS), uma subcategoria do ASS, que tem o foco de utilizar dados não rotulados para melhorar o processo de classificação que já utilizaria dados rotulados. Esse tipo de classificação é extremamente útil em problemas em

que os dados rotulados são difíceis de se obter e os dados não rotulados são abundantes como é o caso da PLSP.

A principal vantagem de usar CSS para PSLP é que podemos construir classificadores mais precisos para predizer a LSP de proteínas de taxonomias com poucas instâncias rotuladas. Exemplos de taxonomias que possuem poucas proteínas podem ser vistas na tabela 1 abaixo, que mostra o número de PRs e PNRs e sua relação percentual com o número total de proteínas daquela taxonomia. As informações dessa tabela foram retiradas da base de conhecimento UniProtKB / Swiss-Prot, versão 2021_03 [1].

Tabela 1 – Comparação entre PRs e PNRs = UniprotKB/Swiss-Prot

	Taxonomia	PRs (%)	PNRs (%)	Nº total de proteínas
	Archaea	201 (≈1,02%)	19452 (≈98,98%)	19653
	Bacteria	2845 (≈0,85%)	332221 (≈99,15%)	335066
Eukaryota	**Total**	33534 (≈17,33%)	159987 (≈82,67%)	193521
	Alveolata	115 (≈9,93%)	1043 (≈90,07%)	1158
	Amoebozoa	219 (≈4,97%)	4184 (≈95,03%)	4403
	Euglenozoa	104 (≈11,93%)	768 (≈88,07%)	872
	Fungi	6244 (≈17,66%)	29116 (≈82,34%)	35360
	Metamonada	11 (≈11%)	89 (≈89%)	100
	Viridiplantae	5803 (≈14,18%)	35122 (≈85,82%)	40925
	Viruses	1151 (≈6,77%)	15863 (≈93,23%)	17014
	Total	37731 (≈10,86%)	309704 (≈89,14%)	347435

1.3 Objetivo

Tendo em mente o contexto apresentado, o objetivo deste trabalho é o desenvolvimento de um classificador semi-supervisionado multirrótulo baseado no *framework* de *predictive clustering trees* para realizar a PLSP a partir de dados de proteínas da *Gene Ontology*. Além disso, o classificador criado deve ser capaz de extrair informações dos dados de PNRs para melhorar a classificação em comparação ao classificador utilizando apenas os dados das PRs.

Como objetivos secundários, queremos avaliar se o classificador desenvolvido é capaz de utilizar dados não rotulados para melhorar a classificação utilizando conjuntos de dados multirrótulo de domínios diferentes para conseguir identificar se o algoritmo tem essa capacidade mais geral.

Como objetivos terciário, o classificador desenvolvido deve ser capaz de varias o nível de supervisão, ou seja, controlar o quanto a previsão com base nos dados não rotulados irá afetar o resultado de classificação final.

Para atingir esse objetivo, foi necessário à realização dos seguintes passos:

[1] <https://ftp.uniprot.org/pub/databases/uniprot/previous_releases/release-2021_03/>

- criação de conjuntos de dados com poucas PRs a partir de conjuntos de dados de taxonomias com muitas PRs. Isso foi necessário para simular cenários semi-supervisionados, dado que precisamos saber os rótulos de cada proteína dos experimentos para conseguirmos avaliar se o classificador acertou ou não a localização;
- desenvolvimento da árvore de decisão capaz de utilizar dados rotulados e não rotulados no processo de classificação e que tenha um parâmetro ω capaz de definir o nível de supervisão do classificador;
- experimentos de avaliação do classificador semi-supervisionado utilizando métricas de avaliação multirrótulo para mensurar a performance do classificador em cada um desses cenários criados no primeiro passo.

Por fim, o trabalho busca servir como base teórica para o estudo de métodos semi-supervisionados para PLSP.

1.4 Estrutura do trabalho

Este trabalho está dividido em 5 capítulos. No Capítulo 1, expõe-se o contexto, as motivações para o desenvolvimento deste trabalho e o objetivo para o desenvolvimento da árvore preditiva semi-supervisionada.

No Capítulo 2 expõe-se os trabalhos relacionados, servindo como referencial para o entendimento do que foi implementado. Ele documenta o estudo feito sobre o tema e embasa a organização e a metodologia aplicada na implementação das etapas da solução.

Em seguida, no Capítulo 3, é apresentado o material utilizado para o desenvolvimento prático do trabalho. Também neste capítulo é descrita a metodologia que foi seguida em cada etapa do projeto, passando pela aquisição dos dados, construção das bases de dados e métricas de avaliação utilizadas.

No Capítulo 4 são apresentados detalhes de cada etapa de desenvolvimento. Aqui será melhor detalhado o funcionamento do algoritmo de árvore de decisão semi-supervisionado criado.

No Capítulo 5 serão apresentados e discutidos os resultados obtidos através da avaliação de desempenho do classificador utilizando métricas de avaliação de classificação multirrótulo.

O Capítulo 6 contém considerações finais sobre o trabalho apresentado, sobre a solução produzida e fornece as primeiras percepções sobre possibilidades de evolução do projeto.

2 Revisão Bibliográfica

Este capítulo apresenta o material teórico que foi usado como base para o trabalho. Ele apresenta considerações gerais sobre a predição de localização subcelular de proteínas e tecnologias relacionadas ao desenvolvimento do classificador, como aprendizado de máquina supervisionado, classificação de dados multirrótulo e aprendizado de máquina semi-supervisionado e seus principais conceitos.

2.1 Classificação Multirrótulo

Na literatura de AM, problemas de classificação convencionais são chamados de problemas simples-rótulo. Nesses problemas, um classificador é treinado em um conjunto de instâncias que estão associados com um único rótulo l de um conjunto de rótulos disjuntos L, onde $|L| > 1$. Se $|L| = 2$, então o problema é chamado de problema de classificação binária, e se $|L| > 2$, o problema é chamado de problema de classificação multi-classe. Em uma classificação multirrótulo (CM), as instâncias de treino estão associados a um conjunto de classes $Y \subseteq L$, sendo $|Y| > 1$. (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010). Dado que uma proteína pode estar localizada em múltiplas localizações ou então se movimentar entre duas ou mais localizações, os problema de PLSP é resolvido através de classificação multirrótulo.

A CM pode ser investigada por meio do uso de duas abordagens: dependente e independente de algoritmo. A abordagem independente de algoritmo utiliza algoritmos tradicionais de classificação simples-rótulo para tratar problemas multirrótulo. Para isso, é necessário realizar uma transformação do problema multirrótulo original em vários problemas simples-rótulo. Já na abordagem dependente de algoritmo, são criados algoritmos específicos para tratar o problema multirrótulo diretamente, sem a necessidade de transformações. Esses algoritmos podem ser baseados em algoritmos de classificação convencionais, como por exemplo Máquinas de Vetores de Suporte (VAPNIK, 1995) e Árvores de Decisão (QUINLAN, 1993).

2.2 Métodos de Localização Subcelular de Proteínas (LSP)

2.2.1 Métodos moleculares

Métodos de Biologia Molecular são usados para obtenção de informação funcional de proteínas por meio de testes e análise em laboratório. Esses métodos são os mais confiáveis, porém como o número de proteínas recentemente descobertas vêm aumentando

exponencialmente esse tipo de métodos se torna inviável pois consome muito tempo e dinheiro além de ser muito trabalhoso. Desse modo os métodos de PLSP são usados para contornar esse problema, criando classificadores a partir de informações relacionadas à proteína ou sua sequência genética (WAN; MAK; KUNG, 2013).

2.2.2 Métodos baseados em predição

Métodos convencionais que realizam a LSP por meio de predição são aqueles que fazem uso do aprendizado de máquina para criar ferramentas capazes de definir a LSP. Esses métodos podem ser divididos em duas grandes categorias: métodos baseados em sequências e métodos baseados em anotações. O primeiro deles se baseia no fato de que existem padrões similares nas sequências genômicas que podem indicar funções e elementos estruturais similares entre esses genomas (JUNCKER et al., 2009). Já o segundo método faz uso das correlações entre anotações de uma proteína (normalmente relacionadas à função das mesmas) e sua localização subcelular (WAN; MAK; KUNG, 2012). Nesse trabalho estamos interessados em criar um classificador que realize a PLSP a partir de dados de anotações da *Gene Ontology*, ou seja, estamos interessados nos métodos baseados ema anotações.

2.2.2.1 Métodos baseados em anotações

Métodos baseados em anotações, também conhecidos por métodos baseados em conhecimento, são métodos que usam as informações contidas em bancos de dados e na literatura científica para realizar a PLSP. Podem se baseados em literatura cientifica e em bancos de dados.

Quando baseado em literatura científica, esse método realiza a PLSP através do conjunto de informações obtidos do processamento de linguagem natural na literatura científica associado com as informações guardadas em bancos de dados.

Quando baseado apenas em banco de dados, esse método faz a correlação entre as anotações de proteínas, normalmente relacionadas às funções das mesmas, e sua localização dentro da célula. Nos últimos tempos, houve um profundo avanço e enriquecimento de informação nos conjuntos de dados de anotações de proteínas, o que faz esse tipo de método ser cada vez mais atrativo e eficiente. Esse método extrai as informações das proteínas de bancos de dados de anotações para criar um conjunto de dados que é utilizado no treinamento de um classificador utilizando Aprendizado de Máquina. Hoje em dia existem diversos bancos de dados que podem ser usados, como por exemplo o *Swiss-Prot keywords*, *PubMed abstracts* e *Gene Ontology Annotation database* (GOA).

Nesse trabalho estamos interessados em utilizar os dados da GOA como atributos

para o conjunto de dados de treinos e dessa forma utilizaremos a abordagem baseada em anotações de banco de dados.

2.3 Aprendizado Semi-Supervisionado

ASS é um paradigma de AM relacionado ao estudo de como os computadores podem aprender a partir de dados rotulados e não rotulados (ZHU; GOLDBERG, 2009). O ASS pode ser visto como um meio termo entre o aprendizado supervisionado, no qual o processo de aprendizagem é feito apenas com exemplos rotulados, e o aprendizado não supervisionado, no qual o processo de aprendizagem é feito apenas a partir de exemplos não rotulados (SADARANGANI; JIVANI, 2016). Esse paradigma possui diversas subcategorias (TRIGUERO; GARCIA; HERRERA, 2015; ZHU, 2008), mas podemos citar as principais como:

- **classificação semi-supervisionada (CSS)**: no qual o objetivo é construir um classificador capaz de utilizar ambos dados rotulados e não rotulados, utilizando os dados não rotulados para minimizar os erros do processo de classificação que já utilizaria dados rotulados;
- **regressão semi-supervisionada**: no qual o objetivo é construir um regressor capaz de utilizar ambos dados rotulados e não rotulado, utilizando os dados não rotulados para minimizar os erros do processo de regressão que já utilizaria dados rotulados;
- **agrupamento semi-supervisionado**: no qual o objetivo é construir um algoritmo de agrupamento de dados capaz de utilizar dados rotulados ou então informações de restrições entre pares de dados para obter melhores resultados (QIN et al., 2019).

Nesse trabalho estamos interessados na CSS, pois esse tipo de classificador é extremamente útil em problemas em que os dados rotulados são difíceis de se obter e os dados não rotulados são abundantes como é o caso da PLSP.

Um dos métodos mais comuns de ASS e que foi considerado para a PLSP no desenvolvimento deste trabalho é o *Self-Learning* (SL) Também chamado de *Self-Training*. No SL, o classificador é primeiramente treinado usando apenas os dados rotulados disponíveis (em geral, uma pequena quantidade de dados rotulados), depois disso o classificador é usado para classificar todos os dados não rotulados no conjunto de treino. A partir disso, os exemplos não rotulados que foram rotulados com a maior confiabilidade pelo classificador são adicionados ao conjunto de treino. O processo é então repetido até que um critério de parada seja atendido. O problema desse método é que o algoritmo assume que as previsões de maior confiança estão corretas e isso pode não ser verdade (SADARANGANI; JIVANI, 2016);

A abordagem com SL foi descartada, pois ela apresenta um problema específico em que erros são reforçados quando o classificador erra uma predição, dado que ela passa a ser utilizada no conjunto de dados de treino, como uma verdade. Este erro é então reforçado nas próximas iterações, o que pode levar a uma diminuição no desempenho preditivo (LEVATIć et al., 2017).

2.4 Classificação Semi-Supervisionada

A CSS pode ser extremamente útil em situações em que: (i) anotar instâncias em classes é um processo difícil, pois pode exigir testes de análise, especialistas humanos, dispositivos especiais, experimentos lentos, etc., e (ii) há uma grande quantidade de dados não rotulados disponíveis. Dessa forma a CSS pode fazer uso destes dados não rotulados em abundância para complementar o processo de aprendizagem, sem a necessidade de esforço humano e pode potencialmente melhorar a precisão do algoritmo de previsão. Diversos domínios se encaixam nos problemas descritos acima, como por exemplo: reconhecimento de voz, bioinformática, processamento de texto e categorização de imagens (SADARANGANI; JIVANI, 2016; PISE; KULKARNI, 2008).

A CSS também pode ser dividida em duas categorias de aprendizagem: aprendizagem transdutiva (AT) e aprendizagem indutiva (AI). Na AI o objetivo do classificador criado é ser capaz de generalizar os dados de treino de forma que o mesmo seja capaz de prever as classes de dados não vistos de forma precisa. Já na AT, o objetivo do classificador é e ajustar bem ao conjuntos de treino e prever corretamente as instâncias da base de dados de treino (TRIGUERO; GARCIA; HERRERA, 2015; PRAKASH; NITHYA, 2014). Para um processo de classificação supervisionado tradicional, normalmente AT não é um objetivo a ser alcançado, dado que isso pode gerar problemas de *overfitting*, mas na CSS, também temos interesse de saber se o classificador está conseguindo classificar as instâncias de treino corretamente, dado que nesse paradigma não sabemos as classes de todas as instâncias de treino.

3 Materiais e Métodos

Nessa seção serão apresentados e descritos os materiais e metodologias utilizados no desenvolvimento do presente projeto, abordando o processo de criação de bases de dados utilizadas, assim como os principais conceitos dos algoritmos utilizados. Por fim, são elencadas as métricas utilizadas para avaliar a performance do classificador criado, tendo em vista que é necessário avaliar a performance multirrótulo nos casos de AI e AT.

3.1 Bases de conhecimento

Uniprot Knowledgebase (UniProtKB) (The UniProt Consortium, 2017) é uma base de conhecimento, verificada por especialistas, que age como um ponto central de acesso para informações de proteínas com referências de multiplas fontes. Essa base consiste em 2 grandes seções: UniProtKB/Swiss-Prot e UniProtKB/TrEMBL. A primeira contém anotações de alta qualidade, em geral vindas de anotações confiáveis da literatura, publicadas em artigos e trabalhos científicos ou de anotações manuais, vindas de testes em laboratório. A segunda base consiste em proteínas analisadas computacionalmente, em geral com anotações feitas por algoritmos e por isso são menos confiáveis. Desse modo as proteínas utilizadas para criar as bases de dados deste trabalho vieram apenas da base de conhecimento UniprotKB/Swiss-Prot.

3.1.1 Gene Ontology

Como comentado previamente, com profundo avanço e enriquecimento de informação nos nas bases de conhecimento de proteínas, os métodos baseados em anotações vem ganhando cada vez mais espaço na literatura de PLSP. Entre os bancos de dados mais usados para essa abordagem, a *Gene Ontology* (GO) é um dos mais atrativos e comuns (WAN; MAK, 2015) e foi a abordagem escolhida para a construção das bases de dados desse trabalho. Nessa abordagem as anotações de termos da GO em uma proteína são convertidas em atributos para a construção das bases de dados dos classificadores.

Do forma geral, uma ontologia é uma representação de diversos conceitos sobre um determinado domínio e os relacionamentos entre esses conceitos. Desse modo, a GO é definida como um conjunto de vocabulários padronizados e controlados que descrevem genes, os atributos do produto genético e a correlação entre eles, em qualquer tipo de organismo. A GO é dividida em três ontologias:

- **componentes celulares**: descrevem componentes das células e organismos vivos,

como proteínas, ácidos nucleicos, membranas e organelas. A grande maioria desses componentes reside dentro da célula, porém alguns residem em locais extracelulares;

- **funções moleculares**: descrevem atividades que ocorrem em nível molecular, como atividades de catalização e ligação. Seus termos representam as atividades desempenhadas pelas moléculas;
- **processos biológicos**: são sequências de eventos realizados por um ou mais conjuntos de funções moleculares.

Existem vários conjuntos de dados que utilizam anotações da *Gene Ontology*, como por exemplo o Euk-mPLoc (CHOU; SHEN, 2007), o Hum-PLoc (CHOU; SHEN, 2006a), o Gneg-PLoc (CHOU; SHEN, 2006b) e o Euk-OET-PLoc (CHOU; SHEN, 2006c). Para a construção de conjuntos de dados com anotações da GO, duas etapas devem ser executadas: extração de termos da GO e construção de vetores de atributos para cada proteína utilizando os termos recuperados. A extração de termos da GO pode ser realizada de três maneiras, descritas a seguir.

- escaneamento de um conjunto de sequências de proteínas contra conjuntos de dados de assinaturas de proteínas. Nesses conjuntos é feita uma busca por termos relevantes da GO. Esse tipo de método pode ser encontrado nos trabalhos de MEI; FEI; ZHOU e de BLUM; BRIESEMEISTER; KOHLBACHER. Para a realização desse escaneamento, pode-se utilizar ferramentas específicas, como o InterproScan (ZDOBNOV; APWEILER, 2001);
- usando *accession numbers*[1] das proteínas para busca por termos da GO em bancos de dados de proteínas. Esse método associa diretamente os *accession numbers* das proteínas com os termos da GO. Desse modo, são criados vetores de atributos indicando cada termo da GO que cada proteínas contém. Esse método apresenta um desempenho melhor que os que utilizam o InterProScan, porém não é aplicável para proteínas que não estão anotadas nos conjunto de dados (WAN; MAK, 2015);
- utilização de ferramentas para buscar por homólogos das proteínas estudadas, como por exemplo BLAST (ALTSCHUL et al., 1997), OrthoMCL (LI; STOECKERT; ROOS, 2003; CHEN et al., 2006) e HMMER (FINN; CLEMENTS; EDDY, 2011; MISTRY et al., 2013). Os *accession numbers* desses homólogos são então utilizados para uma busca em bancos de dados de anotações da GO. Esse método é utilizado quando a proteína estudada não está anotada com termos da GO. Ele é aplicável

[1] <https://www.ncbi.nlm.nih.gov/Sequin/acc.html>

para qualquer tipo de sequência de proteínas, e com ele é possível obter um número maior de termos da GO (WAN; MAK, 2015).

Após a extração dos termos da GO, os mesmos são utilizados para a construção de vetores de atributos. Duas estratégias foram ser utilizadas:

- considerar cada termo da GO como uma base canônica de um espaço Euclidiano, onde as coordenadas podem ser 0 ou 1. Assim, a presença ou ausência dos termos da GO é utilizada para construir os vetores de atributos (CHOU; SHEN, 2006a; CHOU; SHEN, 2006b; CHOU; SHEN, 2006c). Essa estratégia ignora o fato que um termo pode ser usado para anotar a mesma proteína diversas vezes (WAN; MAK, 2015);
- considerar a frequência com que cada termo é anotado para cada uma das proteínas estudadas, dado que uma proteínas pode estar anotada diversas vezes em um mesmo termo da GO.

Para maior facilidade na apresentação dos resultados, chamaremos essas estratégias de "binária" (10) e "frequência de termos" (FT), respectivamente.

3.2 Conjunto de dados

Para este trabalho, 6 bases foram criadas utilizando as informações contidas na base de conhecimento UniProtKB/Swiss-Prot (versão 2021_03) para as taxonomias Viridiplantae, Virus e Fungi. Essas bases de dados serão nomeadas P, V e F, respectivamente.

Optamos por utilizar a a extração de termos da GO com *accession numbers*, por ser a abordagem mais comum e simples para criar bases de dados de proteínas para uma determinada taxonomia, dado que conseguimos extrair os *accession numbers* relacionados a uma taxonomia e uma LSP facilmente no *UniProtKB*. O processo de criação dessas bases de dados está descrito abaixo:

1. Primeiramente, para cada organismo, são buscadas todas as proteínas anotadas com as localizações desejadas na base de conhecimento UniProtKB/Swiss-prot;
2. Após isso são extraídos os *acession numbers* do UniProt (ANs) das proteínas;

3. Os ANs são então utilizados como chaves de buscas para buscar termos da GO na principal base de conhecimento de GO, a UniProt-GOA[2] (HUNTLEY et al., 2015) utilizando a ferramenta QuickGO[3];

4. Em seguida são buscados termos da GO na base de dados do Swiss-Prot. Caso algum termo, que não tenha retornado na busca no UniProt-GOA, retorne na busca atual, este é então adicionado à base. Esse método adicional foi incluído para termos uma maior garantia que encontraremos termos da GO para todas as instâncias;

5. Por fim são criados dois conjuntos de dados para cada organismo, cada um utilizando técnicas diferentes para a criação de vetores de atributos. Foram utilizadas as duas técnicas discutidas acima: binária, que utiliza a presença (1) ou ausência (0) dos termos da GO como atributos e FT que utiliza a frequência dos termos.

3.2.1 Conjunto de dados - Mulan

Também utilizamos conjuntos de dados multirrótulo bem definidos na literaura, obtidos do site Mulan[4] (TSOUMAKAS et al., 2011) para testar nosso método em outros tipos de conjuntos de dados, que não só os de proteínas. Os datasets escolhidos foram: ***Scene*** (BOUTELL et al., 2004a), ***Emotions*** (TROCHIDIS et al., 2008) e ***Yeast*** (ELISSEEFF; WESTON, 2001a).

3.2.2 Informações das bases de dados utilizadas

A tabela 2 descreve algumas informações sobre os conjuntos de dados usados em nossos experimentos. ***Card*** é a cardinalidade do conjunto de dados, ou seja, o número médio de rótulos das instâncias daquele conjunto de dados multirrótulo. ***Dens*** representa a densidade do conjunto de dados, que é um nível adimensional da cardinalidade. As equações 3.1 e 3.2 mostram como essas duas medidas são calculadas.

$$Card(D) = \sum_{i=1}^{|D|} \frac{|Y_i|}{|D|} \qquad (3.1) \qquad\qquad Dens(D) = \frac{Card(D)}{|Y|} \qquad (3.2)$$

onde Y_i representa cada class distinta do conjunto de dados D. **MeanIR** representa a média da proporção de desbalanceamento do conjunto de dados D e essa medida representa o quanto uma base de dados multirrótulo está desbalanceada (CHARTE et al., 2015). Essa medida é importante, pois muitos conjuntos de dados multirrótulo apresentam níveis altos de desbalanceamento, fazendo com que o processo de aprendizagem não seja tão

[2] <www.ebi.ac.uk/GOA>
[3] <www.ebi.ac.uk/QuickGO/>
[4] http://mulan.sourceforge.net/datasets-mlc.html

preciso. Podemos notas que alguns conjuntos de dados apresentam níveis bem baixos de desbalanceamento, como é o caso do *Scene* e do *Emotions*, mas para o caso das bases de dados de fungos e do *Yeast*, temos bases de dados mais desbalanceadas. Dessa forma será possível verificar se a performance do algoritmo sofre muita influência do desbalanceamento do conjunto de dados.

Tabela 2 – Informações gerais sobre os conjuntos de dados

Dataset	Domínio	Instâncias	Tipo do dado	Classes	Card	Dens	MeanIR
Virus binary	biology	645	Nominal	6	1,356	0,226	4,079
Virus FT	biology	645	Numeric	6	1,356	0,226	4,079
Plants binary	biology	5283	Nominal	12	1,413	0,118	8,639
Plants FT	biology	5283	Numeric	12	1,413	0,118	8,639
Fungus binary	biology	5506	Nominal	10	1,287	0,129	29,168
Fungus FT	biology	5506	Numeric	10	1,287	0,129	29,168
Emotions	music	593	Numeric	6	1,868	0,311	1,478068
Scene	image	2407	Numeric	6	1,074	0,179	1,253784
Yeast	biology	2417	Numeric	14	4,237	0,303	7,196811

As tabelas abaixo mostram informações específicas sobre as bases de dados de proteínas, detalhando as classes internas, assim como informações sobre os múltiplos rótulos de cada instância.

Tabela 3 – Informações sobre os dados da base de dados de Vírus

Localização Subcelular	**Nº de proteínas**
Secretada	36
Membrada do vírion	74
Retículo Endoplasmático do hospedeiro	63
Membrana celular do hospedeiro	86
Citoplasma do hospedeiro	291
Núcleo do hospedeiro	325

Nº de exemplos	**Nº de atributos (termos GO)**	**Nº de classes**	**Nº de labelsets**
645	704	6	29

Nº de proteínas por quantidade de LSPs	
Proteínas de 1 LSP	445
Proteínas de 2 LSPs	174
Proteínas de 3 LSPs	23
Proteínas de 4 LSPs	2
Proteínas de 5 LSPs	1

3.3 Árvores de Decisão

Árvore de Decisão (BREIMAN et al., 1984) é uma abordagem de modelagem preditiva supervisionada amplamente usada em tarefas de classificação. Essa técnica é baseada na ideia de divisão dos dados em grupos homogêneos, na qual usando uma estrutura de dados em formato de árvore, seria possível determinar os valores para novas observações (TAN et al., 2005).

Tabela 4 – Informações sobre os dados da base de dados de Viridiplantae

Localização Subcelular	**Nº de proteínas**
Membrana celular	627
Parede celular	65
Cloroplasto	1122
Citoplasma	1240
Retículo Endoplasmático	369
Extracelular	86
Aparato de Golgi	213
Mitocôndria	426
Núcleo	1869
Peroxissomo	113
Plastídio	1144
Vacúolo	194

Nº de exemplos	**Nº de atributos (termos GO)**	**Nº de classes**	**Nº de labelsets**
5283	5640	12	84

Nº de proteínas por quantidade de LSPs	
Proteínas de 1 LSP	3322
Proteínas de 2 LSPs	1756
Proteínas de 3 LSPs	190
Proteínas de 4 LSPs	12
Proteínas de 5 LSPs	2
Proteínas de 6 LSPs	1

Tabela 5 – Informações sobre os dados da base de dados de Fungi

Localização Subcelular	**Nº de proteínas**
Membrana celular	240
Endosomo	66
Citoplasma	2460
Retículo Endoplasmático	512
Extracelular	14
Aparato de Golgi	237
Mitocôndria	856
Núcleo	2426
Peroxissomo	67
Vacúolo	211

Nº de exemplos	**Nº de atributos (termos GO)**	**Nº de classes**	**Nº de labelsets**
5506	6304	10	66

Nº de proteínas por quantidade de LSPs	
Proteínas de 1 LSP	3980
Proteínas de 2 LSPs	1477
Proteínas de 3 LSPs	42
Proteínas de 4 LSPs	6
Proteínas de 5 LSPs	1

Neste algoritmo pontos de decisão são criados, representados pelos nós da árvore, e para cada um deles o resultado da decisão definirá o caminho a ser percorrido na árvore. Geralmente essa decisão é uma decisão binária, assim cada nó tem sempre dois filhos, mas existem outras abordagens em que cada nó pode ter múltiplas decisões e consequentemente, múltiplos nós filho. Os nós são responsáveis pelas verificações que irão direcionar um ramo ou outro para sequência do fluxo de classificação. Neste projeto, iremos apenas abordar o caso de árvores de decisão binárias.

Para cada nó, portanto, uma pergunta será feita e haverá duas opções de respostas: sim ou não, as quais cada uma leva para uma próxima pergunta diferente, até que um um nó folha seja atingido, que representa o resultado do processo de classificação. Esse processo é ilustrado na Figura 2, que possui uma representação de uma árvore de decisão.

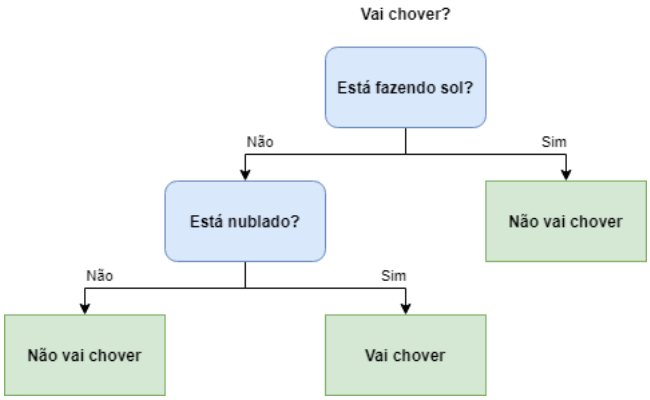

Figura 2 – Ilustração exemplificando uma árvore de decisão simples.

Para que possa ser definido quais valores estarão presentes em cada um dos nós, são usadas algumas estratégias para definir o atributo que melhor divide os espaço amostral, como o cálculo da Entropia ou do Índice Gini de cada atributo.

- **Entropia:** representa a desordem de um conjunto de dados. Quanto maior é o seu valor, maior a desordem temos nos dados. A cada *split* o algoritmo calcula o valor de entropia para todos os atributos do conjunto de dados e a partir deste valor calcula o ganho de informação caso o *split* seja feito com aquele atributos. O atributos selecionado para dividir o conjunto de dados naquele nó é aquele que traz maior ganho de informação;
- **Índice Gini**: similar a abordagem da Entropia, o índice Gini também indica um nível de desordem em um conjunto de dados, mas neste caso ele é inversamente proporcional, ou seja, quanto menor o valor do índice, maior a ordem dos dados. Dessa forma, a cada *split*, o índice Gini é calculado para todos os atributos e aquele com o menor valor é o selecionado para representar a divisão do nó.

3.3.1 Predictive Clustering Trees

Predictive clustering trees (PCTs) (BLOCKEEL; RAEDT; RAMON, 1998) é uma generalização do algoritmo de árvore de decisão explicado acima e pode ser usado tanto para agrupamento e classificação. Em árvores de decisão normais, as folhas contêm as classes e os ramos da raiz até as folhas contêm as condições ou decisões para o processo de classificação. A estrutura de uma PCT vê uma árvore de decisão como uma hierarquia de *clusters*, em que a raiz de um PCT representa um cluster contendo todos os dados, que é

então particionado recursivamente em *clusters* menores usando um critério para selecionar a melhor divisão. As folhas de um PCT correspondem ao nível mais baixo da hierarquia do *cluster* e cada folha é rotulada com o protótipo do *cluster* (previsão) (KOCEV; SLAVKOV; DZEROSKI, 2013).

Os PCTs podem ser construídas utilizando a indução *top-down* de heurísticas de árvores de decisão (BREIMAN et al., 1984). O algoritmo recebe um conjunto de exemplos (E) como entrada e produz a PCT final. Para isso, utiliza uma heurística (h) para selecionar o melhor teste ($t*$) entre os testes (t), também chamado de possíveis divisões, como explicado na seção anterior. Dessa forma o melhor teste representa o atributos que melhor divide o conjunto de dados presente naquele nó em relação as classes. A heurística usada no *framework* PCT é geralmente a redução da variância causada pelo particionamento das instâncias do nó em múltiplas partições de dados (nós filhos) de acordo com o melhor teste ($t*$).

Para encontrar o melhor teste em cada nó, o procedimento *BestTest* procura o melhor teste aceitável calculando a redução da variância para cada teste possível. O procedimento retorna o melhor teste encontrado, o valor heurístico e as novas partições dos dados do nó, que serão utilizados nos nós filhos. Este processo é repetido para cada nó filho, até que nenhum teste aceitável possa ser encontrado, o que significa que nenhum teste é capaz de reduzir significativamente a variância do nó. Nesse caso, o algoritmo cria uma folha, calcula o protótipo das instâncias pertencentes a essa folha para criar o as classes preditas por esse nó. Todo esse processo é denominado indução *top-down* da PCT e o algoritmo abaixo mostra o pseudocódigo exemplificando o *framework* (VENS et al., 2008; KOCEV; SLAVKOV; DZEROSKI, 2013; BLOCKEEL; RAEDT; RAMON, 1998).

Algorithm Algoritmo de indução *top-down* de uma PCT.

procedure InduceTree
Input: A dataset E
Output: A predictive clustering tree
1: $(t*,\ h*\ P*) = BestTest(E)$
2: **if** $t* \neq none$ **then**
3: **for each** $E_i \in P*$ **do**
4: $tree_i = InduceTree(E_i)$
5: **return** $node(t*, \cup_i\{tree_i\})$
6: **else**
7: **return** $leaf(Prototype(E))$

procedure BestTest
Input: A dataset E
Output: The best test($t*$), its heuristic score($h*$) and the partition($P*$) it induces on the dataset(E)
1: $(t*,\ h*\ P*) = (none, 0, \emptyset)$
2: **for each** possible test t **do**
3: $P =$ partition induced by t on E
4: $h = |E|Impurity(E) - \sum_{E_i \in P} |E_i|Impurity(E_i)$
5: **if** $(h > h*) \wedge Acceptable(t,\ P)$ **then**
6: $(t*,\ h*,\ P*) = (t,\ h,\ P)$
7: **return** $(t*,\ h*,\ P*)$

A ideia das PCTs funciona pois, tentando maximizar a redução da variância causada pela divisão em cada nó, garantimos que a homogeneidade do *cluster* também aumente a cada nível que a árvore desce e que o desempenho preditivo melhore de forma geral (BLOCKEEL; RAEDT; RAMON, 1998). A principal diferença entre a estrutura de

PCTs e a estrutura de árvores de decisão normal é que os PCTs tratam as funções de variância e protótipo como parâmetros e esses parâmetros podem ser definidos a qualquer momento, possibilitando a árvore de decisão utilizar múltiplas heurísticas para decidir o melhor teste. Isso também facilita o processo de criação de novas heurísticas, dado que a lógica fica mais extensível com o algoritmo mais padronizável. Por causa disso, os PCTs podem ser usados de várias maneiras, por exemplo: agrupamento (BLOCKEEL; RAEDT; RAMON, 1998; STRUYF; DZEROSKI, 2007), classificação e regressão multi-objetivo (BLOCKEEL; RAEDT; RAMON, 1998; BLOCKEEL; DZEROSKI; GRBOVIĆ, 1999; STRUYF; DZEROSKI, 2006; DEMšAR, 2006), análise de dados de série temporal (DZEROSKI et al., 2007) e até classificação multirrótulo hierárquica (VENS et al., 2008). Para o nosso caso, estamos interessados nas capacidades de classificação multirrótulo das PCTs.

3.3.2 CLUS

Clus é um *framework* Java de código aberto que implementa as PCTs (BLOCKEEL; RAEDT; RAMON, 1998). O Clus generaliza a abordagem da árvore de decisão tradicional criando uma hierarquia de *cluster*, como explicado acima. Dependendo do problema, CLUS pode ser otimizado durante a criação de *clusters* usando diferentes heurísticas para escolher o melhor teste em cada nó e melhorar a precisão do processo de aprendizagem (STRUYF et al., 2018).

CLUS pode realizar tarefas clássicas de classificação ou regressão, mas trabalhos recentes mostraram que pode ser aplicado com sucesso a outras múltiplas tarefas, incluindo *multi-task learning (multi-target classification and regression)*, *structured output learning*, classificação multirrótulo, classificação hierárquica e previsão de séries temporais e descoberta de subgrupos e agrupamentos (STRUYF et al., 2018).

3.4 Medidas de Avaliação

A avaliação de classificadores multirrótulo requer medidas diferentes das utilizadas em problemas de classificação simples-rótulo. Diferente da classificação simples-rótulo, na qual uma instância é classificada de maneira errada ou correta, na classificação multirrótulo, uma instância pode ser classificada de maneira parcialmente errada ou parcialmente correta. Esses casos acontecem quando um classificador atribui corretamente a uma instância pelo menos uma das classes a que ela pertence, mas também não atribui à instância uma ou mais classes as quais ela pertence. Pode acontecer também de o classificador atribuir a uma instância uma ou mais classes as quais ela não pertence.

Considere que cada instância de um conjunto multirrótulo é representada por $(\mathbf{x}_i, Y_i)$, em que i = 1...m (sendo m o número total de instâncias), $Y_i \subseteq L$ é o conjunto de

classes reais e $L = \{\lambda_j : j = 1...q\}$ é o conjunto de todas as classes do problema. Dada uma instancia $\mathbf{x}_i$, o conjunto de classes preditas pelo classificador para essa instancia é denominado Z_i. Utilizamos as mesmas medidas que foram utilizadas no trabalho de GODBOLE; SARAWAGI. Elas são apresentadas nas Equações 3.3, 3.4, 3.5 e 3.6.

$$\textit{Acurácia baseada em exemplo} = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{3.3}$$

$$\textit{Precisão baseada em exemplo} = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \cap Z_i|}{|Z_i|} \tag{3.4}$$

$$\textit{Revocação baseada em exemplo} = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \cap Z_i|}{|Y_i|} \tag{3.5}$$

$$\textit{F1 baseada em exemplo} = \frac{1}{m}\sum_{i=1}^{m}\frac{2|Y_i \cap Z_i|}{|Z_i| + |Y_i|} \tag{3.6}$$

4 Desenvolvimento

Este capítulo descreve em detalhes toda a etapa de desenvolvimento do trabalho. Seguindo o que foi proposto no Capítulo 3, o desenvolvimento do projeto foi dividido nas seguintes etapas:

- implementação do algoritmo de PCT semi-supervisionada utilizando o Clus como base;
- criação dos cenários semi-supervisionados utilizando os conjuntos de dados descritos na seção 3.2.2;
- testes utilizando validação cruzada para avaliar a performance dos classificadores utilizando as medidas de avaliação descritas na seção 3.4

4.1 PCT semi-supervisionada

A nova abordagem de CSS proposta para a PLSP neste estudo foi baseada no *framework* PCT e consiste em adaptar o procedimento de indução do Clus para criar um uma PCT semi-supervisionada capaz de usar dados rotulados e não rotulados ao mesmo tempo no processo de aprendizagem. Essa árvore de classificação é a mesma proposta no trabalho de LEVATIć et al., utilizando os mesmos cálculos de impureza, porém aplicada ao problema de PLSP.

A principal vantagem deste método é que ele preserva as características atraentes de se usar uma árvore de decisão, como ser interpretável e ter um processo de treino rápido; ao mesmo tempo que pode fazer uso de dados não rotulados para melhorar o processo de classificação (KOCEV; SLAVKOV; DZEROSKI, 2013). O algoritmo de PCT semi-supervisionada também não apresenta o problema de reforço de erros citado no caso ded SL, pois utiliza ao mesmo tempo os dados rotulados e não rotulados para definir o melhor teste para realizar o *split* a cada nó da PCT, sem assumir como corretas as previsões de instâncias não rotuladas.

Para adaptar o algoritmo do Clus para o ASS, foi necessário adaptar duas seções do algoritmo. O primeiro estava relacionado com o conjunto inicial de exemplos (E), no qual tivemos que modificar o Clus para que ele aceitasse exemplos rotulados e não rotulados. Portanto, nosso conjunto inicial de exemplos passa a ser $E = E_l \cup E_u$, onde E_l representa a parte rotulada do conjunto de treino e E_u representa a parte não rotulada do conjunto de treinamento. A segunda adaptação estava relacionada ao cálculo da impureza, métrica genérica que representa a homogeneidade do conjunto de dados de um nó, no

procedimento *BestTest*. No Clus, o cálculo da impureza para um conjunto de dados é tratado pelas heurísticas existentes. Portanto, foi necessário criar uma nova heurística que leve em consideração os dados rotulados e não rotulados no conjunto de dados do nó. A nova impureza é composta por duas partes, uma parte supervisionada e outra não supervisionada. A influência de cada parte é controlada pelo parâmetro ω, melhor explicado nos parágrafos abaixo. A impureza é medida da seguinte forma:

$$\text{Impureza}_{\text{SSL}}(E) = \underbrace{\frac{\omega}{T}.\sum_{i=1}^{T}\text{Impureza}(E_l, Y_i)}_{\text{Parte supervisionada}} + \underbrace{\frac{1-\omega}{D}.\sum_{i=1}^{D}\text{Impureza}(E, X_i)}_{\text{Parte não supervisionada}} \tag{4.1}$$

onde E representa o conjunto de exemplos, Y_i representa cada um dos atributos T alvo (classes), X_i representa cada um dos D atributos descritivos e ω é um peso com valor entre 0 e 1 usado para controlar quanta supervisão o algoritmo terá. Se ω for igual a 1, o método funciona usando apenas a parte rotulada dos conjuntos de dados, comportando-se como um algoritmo supervisionado. Por outro lado, se ω for igual a 0, o algoritmo se comporta como um algoritmo não supervisionado, usando apenas informações dos atributos descritivos para definir o melhor teste. Qualquer valor entre 0 e 1 significa que o algoritmo irá utilizar dados rotulados e não rotulados para encontrar o melhor teste.

Este nível controlado de supervisão é uma das principais vantagens deste método, pois ao realizar CSS, existe sempre o perigo de que os dados não rotulados afetem negativamente o desempenho do classificador. Com esse tipo de método adaptativo, podemos garantir que a PCT pode atingir um desempenho preditivo melhor ou igual quando comparada à sua contraparte supervisionada, pois no caso dos dados rotulados sempre piorarem o desempenho da classificação, podemos definir o valor de ω como 1 e o algoritmo irá se comportar como a versão supervisionada da PCT.

No algoritmo de PCT original, a principal medida de impureza utilizada é a Impureza de Gini (IG). IG é uma métrica famosa usada em árvores de decisão para calcular o quanto um conjunto é puro para encontrar o melhor *split*. A IG pode ser calculada para um conjunto de dados E e para uma variável de destino específica T da seguinte forma:

$$\text{Gini}(E, Y) = 1 - \sum_{i=1}^{C} p_i^2 \tag{4.2}$$

onde C é o número de valores possíveis para o atributo alvo (classe) Y, por exemplo: em uma classificação binária, C = 2. E p_i é a probabilidade a priori da classe c_i.

Para criarmos uma heuristica semi-supervisionada, foi necessário modificar o cálculo de impurezas do CLUS. A impureza sobre os atributos alvo é calculada usando apenas os

exemplos rotulados (E_l) e será chamada de impureza rotulada, enquanto a impureza sobre os atributos descritivos é calculada usando todos os exemplos rotulados e não rotulados (E) e será nomeada de impureza não rotulada.

A impureza rotulada para um conjunto rotulado E_l sobre o atributo alvo Y_i é calculada da seguinte forma:

$$\text{Impureza}(E_l, Y_i) = \frac{\text{Gini}(E_l, Y_i)}{\text{Gini}(E_l^{treino}, Y_i)} \tag{4.3}$$

Onde E_l^{treino} representa todas as seções rotuladas do conjunto de dados de treino. Basicamente a parte rotulada do conjunto de dados do nó raíz.

A impureza não rotulada é calculada apenas sobre os atributos descritivos, que, ao contrário das classes, podem assumir valores numéricos ou nominais. Portanto, precisamos definir uma impureza para cada caso. A impureza não rotulada para um conjunto de dados parcialmente rotulado E sobre o atributo descritivo X_i é calculada da seguinte forma:

$$\underbrace{\text{Impureza}(E, X_i) = \frac{\text{Gini}(E, X_i)}{\text{Gini}(E^{treino}, X_i)}}_{\text{Para atributos nominais}} \tag{4.4} \qquad \underbrace{\text{Impureza}(E, X_i) = \frac{\text{Var}(E, X_i)}{\text{Var}(E^{treino}, X_i)}}_{\text{Para atributos numéricos}} \tag{4.5}$$

onde E^{treino} representa todo o conjunto de treino contido no nó raiz da PCT. A equação 4.4 representa a impureza para atributos descritivos nominais, calculando a IG sobre os atributos descritivovs nominais ao invés das classes e a equação 4.5 para os atributos descritivos numéricos. A variância do atributo i-ésimo no conjunto E sobre o atributo descritivo X_i é calculada da seguinte forma:

$$\text{Var}(E, X_i) = \frac{\sum_{j=1}^{N}(x_i^j)^2 - \frac{1}{N}\cdot(\sum_{j=1}^{N} x_i^j)^2}{N} \tag{4.6}$$

Após a adição da nova heurística de ASS ao Clus, ele deve ser capaz de lidar com conjuntos de dados contendo instâncias rotuladas e não rotuladas, utilizando o peso ω para definir o nível de supervisão.

4.1.1 Experimentos semi-supervisionados

Para melhor avaliar a performance dos classificadores, realizamos diversos testes nos classificadores dos conjunto de dados descrito na seção 3.2.2 utilizando uma estratégia de validação cruzada (*5-fold cross-validation*). Esse procedimento divide o conjunto de dados em 5 partições contendo 20 % dos dados originais. Para cada partição, o algoritmo é treinado usando as outras 4 partições (conjunto de treino). A partição não utilizada no

treino é denominada conjunto de testes e é mantida à parte dos dados de treinamento para avaliar o desempenho indutivo do classificador. O desempenho transdutivo também foi avaliado nos experimento avaliado, rotulando as instâncias das 4 partições utilizada no treino em cada teste. Os testes são realizados 1 vez considerando cada partição como o conjunto de teste e as demais como conjunto de treino, calculando todas as métricas de avaliação descritas na seção 3.4. O resultado final é a média das medidas das avaliações calculadas para cada partição.

Cada conjunto de treino é então dividido em partes rotuladas e não rotuladas. Seguindo a recomendação de (WANG et al., 2010), não mantemos a proporção de classe em conjuntos rotulados e não rotulados, porque o principal objetivo da CSS é explorar o uso de dados não rotulados para melhorar o desempenho preditivo. A seleção de instâncias a serem usadas no conjunto rotulado foi feita considerando os *labelsets* existentes em cada conjunto de dados, utilizado a estratégia *stratified* do método *create_holdout_partition*() do pacote mldr da linguagem R [1].

Essa partição é necessária, pois o objetivo destes experimentos é entender se o classificador criado é capaz de utilizar dados não rotulados para melhorar o desempenho da classificação para casos de proteínas com pouquíssimos dados rotulados. Não seria possível avaliar o aprendizado indutivo, nem o transdutivo, caso usássemos conjuntos de dados que verdadeiramente tem pouquíssimas instâncias rotuladas. Dessa forma, esses experimentos foram propostos para simular um cenário de ASS, a partir de conjunto de dados rotulados.

A partir desta divisão entre parte rotulada e parte não rotulada, as classes das instâncias marcadas como não rotuladas foram substituídas por "?", que no Clus significa que a instância tem uma classe que não sabemos o rótulo e devemos ignorar. Dirante essa divisão, garantimos que cada parte rotulada do conjunto de treino tenha pelo menos uma instância representativa para cada classe.

Para testar múltiplos cenários de CSS para cada conjunto de dados, usamos diferentes proporções ao dividir o conjunto de treino em rotulado e não rotulado. A metodologia utilizada para isso funcionou da seguinte forma: considere C como o número de classes de um conjunto de dados E. Para cada conjunto de dados, a parte rotulada do conjunto de treinamento foi definida com instâncias C*α. O parâmetro α é um valor natural que controla o número de instâncias no conjunto rotulado a partir do número de classes. Usamos 3 valores diferentes para α: 5, 10 e 20.

Inspirado no trabalho de VENS et al., consideramos duas abordagens para os a classificação utilizando as PCTs. O primeiro, denominado **local**, é semelhante a um processo de *Binary-Relevance*, no qual treinamos um classificador para cada classe do conjunto multirrótulo e então combinamos os resultados para ter a classificação final.

[1] <https://www.rdocumentation.org/packages/utiml/versions/0.1.4/topics/create_holdout_partition>

A segunda abordagem é chamada **global**, na qual usamos apenas uma PCT capaz de lidar com todos os rótulos de uma vez, onde o classificador utiliza uma média da IG de todas as classes para o cálculo de impureza. Essas abordagens já estavam previamente implementadas no próprio Clus.

Por fim, variamos o parâmetro ω, para cada teste, a fim de validar o melhor nível de supervisão para cada experimento. utilizamos 6 valores diferentes de ω: 0,0, 0,2, 0,4, 0,6, 0,8 e 1,0. Assim, foi possível validar se as instâncias não rotuladas realmente ajudaram ou não na classificação.

Vale notar que os métodos de ASS podem apresentar desempenhos muito diferentes em conjuntos de dados diferentes. Isso acontece porque o sucesso de um algoritmo de ASS é dependente do domínio da aplicação (LEVATIć et al., 2017; CHAWLA; KARAKOULAS, 2011). Por isso, fizemos experimentos em 9 conjuntos de dados diferentes para avaliar melhor se o nosso método realmente conseguiu extrair informações relevantes dos dados não rotulados.

5 Resultados

A partir das descrições dos capítulos 3 e 4, realizamos os experimentos de validação cruzada para todos os conjuntos de dados, calculado as métricas de avaliação multirrótulo para todos os casos.

As tabelas a seguir apresentam as medidas de avaliação de acurácia e F1 para todos os experimentos. Precisão e Revocação também foram calculadas em nossos experimentos, mas estão omitidas na monografia, a fim de condensar melhor os resultados. Além disso, como medida F1 é a média harmônica de ambas as medidas, os valores de Precisão e Revocação estão refletidos nela.

Por fim, a melhor métrica de avaliação dentro os valores de ω foi destacada em vermelho. Dessa forma é possível ver facilmente os experimentos em que o algoritmos conseguiu utilizar os dados não rotulados de forma benéfica no processo de classificação. Se o melhor resultado não for no caso de $\omega = 1.0$, então a PCT semi-supervisionada conseguiu melhorar a classificação utilizando os dados não rotulados. Caso o melhor experimento tenha sido o caso de $\omega = 1$, então o algoritmo não conseguiu tirar proveito dos dados não rotulados de forma que a performance da classificação ultrapassasse a do classificador PCT supervisionado.

5.1 Resultados Obtidos

Tabela 6 – Resultados para os experimentos com o conjunto de dados de Vírus - Abordagem local

Virus_10 - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**81,32%**	**84,71%**	**85,74%**	**88,23%**
0,8	78,76%	82,71%	82,38%	85,56%
0,6	78,78%	82,92%	81,39%	84,93%
0,4	77,04%	81,21%	80,42%	83,84%
0,2	73,92%	78,94%	77,10%	81,68%
0,0	70,20%	75,44%	73,34%	78,32%
Virus_10 - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**89,64%**	**92,26%**	**90,40%**	92,64%
0,8	88,89%	91,54%	90,30%	92,60%
0,6	88,06%	90,58%	90,22%	**92,65%**
0,4	86,28%	88,91%	89,18%	91,54%
0,2	82,67%	86,34%	86,05%	89,16%
0,0	75,67%	79,97%	80,10%	83,89%
Virus_10 - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**90,72%**	**92,92%**	**91,71%**	**93,72%**
0,8	90,48%	92,70%	91,53%	93,54%
0,6	89,52%	91,80%	91,16%	93,16%
0,4	89,57%	91,78%	90,71%	92,78%
0,2	88,65%	91,15%	90,23%	92,46%
0,0	79,38%	83,06%	82,21%	85,57%
Virus_FT - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	84,08%	86,87%	**86,24%**	88,79%
0,8	80,81%	83,77%	83,74%	86,39%
0,6	81,85%	84,81%	84,57%	87,22%
0,4	**84,39%**	**87,50%**	86,23%	**89,05%**
0,2	79,69%	83,42%	82,13%	85,60%
0,0	45,09%	53,95%	46,41%	55,49%
Virus_FT - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	89,88%	92,19%	**90,23%**	**92,56%**
0,8	**89,94%**	**92,28%**	89,94%	92,29%
0,6	77,34%	83,73%	77,46%	83,74%
0,4	77,75%	84,20%	78,12%	84,34%
0,2	77,17%	83,64%	77,17%	83,59%
0,0	38,18%	42,04%	38,69%	42,42%
Virus_FT - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	89,63%	91,98%	90,51%	92,48%
0,8	89,57%	92,04%	**91,24%**	**93,16%**
0,6	**89,91%**	**92,29%**	90,03%	92,21%
0,4	88,80%	91,49%	89,31%	91,78%
0,2	88,51%	91,27%	89,27%	91,69%
0,0	0,00%	0,00%	0,00%	0,00%

Tabela 7 – Resultados para os experimentos com o conjunto de dados de Vírus - Abordagem global

Virus_10 - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**75,11%**	**79,32%**	**77,48%**	**81,39%**
0,8	69,38%	74,08%	72,51%	77,12%
0,6	69,86%	74,74%	72,94%	77,72%
0,4	68,58%	73,76%	71,56%	76,60%
0,2	68,68%	73,46%	71,71%	76,37%
0,0	70,20%	75,44%	73,34%	78,32%
Virus_10 - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**83,14%**	**86,43%**	**86,91%**	**89,69%**
0,8	80,56%	84,20%	83,61%	86,96%
0,6	79,52%	83,30%	83,89%	87,22%
0,4	78,40%	82,57%	82,35%	86,01%
0,2	77,46%	81,73%	81,65%	85,26%
0,0	75,67%	79,97%	80,10%	83,89%
Virus_10 - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**88,12%**	**90,94%**	**89,30%**	**91,82%**
0,8	85,02%	88,15%	86,93%	89,62%
0,6	83,49%	86,86%	87,00%	89,99%
0,4	82,81%	85,98%	86,26%	89,12%
0,2	82,71%	85,82%	86,57%	89,39%
0,0	79,38%	83,06%	82,21%	85,57%
Virus_FT - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**75,70%**	**80,64%**	**77,54%**	**82,11%**
0,8	75,26%	80,42%	77,24%	82,03%
0,6	69,27%	73,51%	72,35%	76,59%
0,4	68,25%	72,59%	71,21%	75,63%
0,2	56,88%	63,04%	60,61%	66,57%
0,0	45,09%	53,95%	46,41%	55,49%
Virus_FT - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**82,97%**	**86,82%**	**84,92%**	**88,18%**
0,8	80,68%	84,50%	81,24%	84,78%
0,6	82,76%	86,51%	83,49%	86,89%
0,4	76,77%	81,59%	78,28%	82,67%
0,2	69,18%	74,97%	71,52%	77,21%
0,0	38,18%	42,04%	38,69%	42,42%
Virus_FT - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	87,39%	90,12%	**89,03%**	**91,46%**
0,8	**87,90%**	**90,79%**	88,34%	91,18%
0,6	87,48%	90,48%	87,99%	90,67%
0,4	85,96%	89,36%	87,04%	89,99%
0,2	82,67%	86,53%	84,51%	88,01%
0,0	0,00%	0,00%	0,00%	0,00%

Tabela 8 – Resultados para os experimentos com o conjunto de dados de Viridiplantae - Abordagem local

Viridiplantae_10 - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**82,02%**	**85,66%**	**82,99%**	**86,45%**
0,8	74,13%	77,66%	74,40%	78,03%
0,6	72,81%	76,50%	73,05%	76,71%
0,4	70,20%	74,22%	70,41%	74,53%
0,2	69,91%	74,05%	69,88%	74,10%
0,0	69,09%	73,23%	69,00%	73,19%
Viridiplantae_10 - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**85,07%**	**88,58%**	**85,34%**	**88,75%**
0,8	80,15%	83,80%	80,48%	84,13%
0,6	78,02%	81,72%	77,92%	81,80%
0,4	75,99%	79,71%	76,42%	80,14%
0,2	74,98%	78,72%	75,43%	79,20%
0,0	74,93%	78,56%	75,21%	78,88%
Viridiplantae_10 - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**86,24%**	**89,32%**	**86,93%**	**89,94%**
0,8	85,18%	88,31%	85,62%	88,68%
0,6	82,67%	85,95%	83,11%	86,30%
0,4	80,06%	83,58%	80,70%	84,19%
0,2	79,43%	82,74%	79,96%	83,30%
0,0	78,36%	81,78%	78,48%	81,96%
Viridiplantae_FT - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**82,39%**	**85,13%**	**82,63%**	**85,41%**
0,8	78,13%	81,33%	78,37%	81,56%
0,6	70,04%	73,67%	70,43%	74,07%
0,4	69,87%	73,56%	70,25%	73,96%
0,2	67,57%	72,28%	68,24%	72,81%
0,0	0,00%	0,00%	0,00%	0,00%
Viridiplantae_FT - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	85,08%	87,57%	85,43%	87,77%
0,8	**85,30%**	**87,83%**	**85,68%**	**88,07%**
0,6	85,05%	87,66%	85,42%	87,91%
0,4	85,11%	87,70%	85,58%	88,03%
0,2	84,77%	87,72%	84,92%	87,77%
0,0	31,64%	34,82%	32,34%	35,42%
Viridiplantae_FT - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	88,50%	90,83%	**88,95%**	91,18%
0,8	**88,77%**	**91,02%**	89,02%	**91,22%**
0,6	88,31%	90,59%	88,67%	90,90%
0,4	87,84%	90,18%	88,24%	90,49%
0,2	87,53%	89,98%	87,62%	90,06%
0,0	43,17%	46,78%	44,43%	48,10%

Tabela 9 – Resultados para os experimentos com o conjunto de dados de Viridiplantae - Abordagem global

Viridiplantae_10 - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**74,63%**	**78,06%**	**75,25%**	**78,68%**
0,8	70,61%	74,46%	70,78%	74,69%
0,6	69,70%	73,63%	69,55%	73,57%
0,4	69,07%	73,21%	68,96%	73,17%
0,2	69,09%	73,23%	69,00%	73,19%
0,0	69,09%	73,23%	69,00%	73,19%
Viridiplantae_10 - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**79,58%**	**83,14%**	**79,93%**	**83,39%**
0,8	75,26%	78,99%	75,33%	79,15%
0,6	75,00%	78,67%	75,27%	78,94%
0,4	75,16%	78,79%	75,41%	79,05%
0,2	74,88%	78,56%	75,19%	78,89%
0,0	74,93%	78,56%	75,21%	78,88%
Viridiplantae_10 - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**83,28%**	**86,26%**	**83,64%**	**86,55%**
0,8	79,94%	83,31%	80,44%	83,89%
0,6	78,58%	82,01%	79,05%	82,51%
0,4	78,46%	81,94%	78,66%	82,16%
0,2	77,91%	81,46%	78,18%	81,74%
0,0	78,36%	81,78%	78,48%	81,96%
Viridiplantae_FT - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**72,24%**	**76,07%**	**73,33%**	**76,97%**
0,8	71,69%	75,63%	72,34%	76,10%
0,6	68,49%	72,42%	69,44%	73,35%
0,4	62,85%	67,99%	63,60%	68,70%
0,2	59,13%	64,62%	60,28%	65,76%
0,0	0,00%	0,00%	0,00%	0,00%
Viridiplantae_FT - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	80,03%	83,18%	80,83%	83,80%
0,8	**81,11%**	**84,26%**	**81,88%**	**84,90%**
0,6	80,30%	83,74%	81,18%	84,45%
0,4	78,26%	82,43%	78,81%	82,86%
0,2	70,87%	75,06%	71,80%	76,03%
0,0	31,64%	34,82%	32,34%	35,42%
Viridiplantae_FT - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**85,70%**	**88,43%**	**86,10%**	**88,60%**
0,8	84,79%	87,72%	85,34%	88,07%
0,6	83,89%	87,02%	84,49%	87,44%
0,4	83,54%	86,70%	84,34%	87,33%
0,2	80,85%	84,48%	81,95%	85,33%
0,0	43,17%	46,78%	44,43%	48,10%

Tabela 10 – Resultados para os experimentos com o conjunto de dados de Fungi - Abordagem local

Fungi_10 - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**83,66%**	**87,07%**	**83,34%**	**86,70%**
0,8	75,32%	80,58%	76,03%	81,04%
0,6	73,25%	78,65%	73,81%	78,94%
0,4	70,03%	76,06%	70,61%	76,35%
0,2	68,48%	74,25%	69,34%	74,91%
0,0	67,42%	73,20%	68,27%	73,86%
Fungi_10 - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**86,79%**	**89,80%**	**87,37%**	**90,38%**
0,8	80,74%	84,63%	81,28%	85,06%
0,6	81,02%	84,70%	81,62%	85,30%
0,4	77,38%	81,27%	77,93%	81,83%
0,2	75,27%	79,15%	75,84%	79,71%
0,0	74,11%	77,78%	74,76%	78,46%
Fungi_10 - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**84,80%**	87,64%	**85,51%**	**88,32%**
0,8	84,75%	**87,82%**	85,04%	88,17%
0,6	82,92%	86,07%	83,63%	86,67%
0,4	81,86%	85,22%	82,31%	85,56%
0,2	81,47%	84,80%	82,00%	85,27%
0,0	77,99%	81,45%	78,41%	81,81%
Fungi_FT - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	83,22%	86,05%	83,69%	86,40%
0,8	84,26%	87,28%	**84,70%**	**87,60%**
0,6	83,95%	86,99%	84,22%	87,18%
0,4	83,64%	86,72%	83,79%	86,75%
0,2	**84,45%**	**87,35%**	84,35%	87,22%
0,0	20,60%	22,42%	21,44%	23,29%
Fungi_FT - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	87,50%	89,75%	87,25%	89,55%
0,8	**89,38%**	**91,65%**	**88,85%**	**91,19%**
0,6	88,73%	91,16%	88,29%	90,78%
0,4	88,29%	90,79%	87,73%	90,28%
0,2	87,37%	89,88%	86,87%	89,44%
0,0	0,00%	0,00%	0,00%	0,00%
Fungi_FT - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	89,74%	91,75%	90,34%	92,25%
0,8	**90,04%**	**92,06%**	**90,67%**	**92,60%**
0,6	89,64%	91,73%	90,31%	92,31%
0,4	88,96%	91,02%	89,39%	91,40%
0,2	89,10%	91,25%	89,73%	91,77%
0,0	10,97%	12,54%	11,02%	12,47%

Tabela 11 – Resultados para os experimentos com o conjunto de dados de Fungi - Abordagem global

Fungi_10 - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**77,55%**	**81,64%**	**77,72%**	**81,70%**
0,8	69,65%	74,90%	70,42%	75,49%
0,6	69,60%	74,87%	70,41%	75,49%
0,4	68,08%	73,70%	68,86%	74,31%
0,2	68,08%	73,70%	68,86%	74,31%
0,0	67,42%	73,20%	68,27%	73,86%
Fungi_10 - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**81,27%**	**84,61%**	**81,59%**	**84,82%**
0,8	77,21%	81,20%	77,70%	81,63%
0,6	76,71%	80,28%	77,33%	80,84%
0,4	75,70%	79,42%	76,22%	79,94%
0,2	75,36%	78,95%	75,76%	79,38%
0,0	74,11%	77,78%	74,76%	78,46%
Fungi_10 - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**82,86%**	**86,16%**	**83,24%**	**86,50%**
0,8	82,27%	85,42%	82,88%	86,01%
0,6	81,67%	84,82%	81,99%	85,11%
0,4	80,72%	84,10%	80,94%	84,30%
0,2	78,65%	82,23%	79,04%	82,55%
0,0	77,99%	81,45%	78,41%	81,81%
Fungi_FT - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	75,33%	79,22%	75,59%	79,50%
0,8	**79,30%**	**82,98%**	**79,32%**	**82,91%**
0,6	77,01%	81,36%	76,99%	81,21%
0,4	72,84%	77,20%	73,44%	77,61%
0,2	57,18%	63,14%	57,19%	63,03%
0,0	20,60%	22,42%	21,44%	23,29%
Fungi_FT - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**81,59%**	**84,82%**	**81,51%**	**84,66%**
0,8	80,58%	83,48%	80,21%	83,10%
0,6	80,59%	83,84%	80,52%	83,73%
0,4	77,24%	81,41%	76,84%	80,96%
0,2	72,08%	77,09%	72,40%	77,42%
0,0	0,00%	0,00%	0,00%	0,00%
Fungi_FT - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**86,50%**	**89,09%**	**87,34%**	**89,81%**
0,8	85,65%	88,45%	86,50%	89,12%
0,6	84,99%	87,95%	85,87%	88,63%
0,4	82,56%	86,01%	84,02%	87,15%
0,2	78,98%	83,51%	79,68%	84,02%
0,0	10,97%	12,54%	11,02%	12,47%

Tabela 12 – Resultados para os experimentos com o conjunto de dados Scene

Scene - local - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	29,54%	33,38%	30,60%	34,73%
0,8	39,38%	45,47%	40,07%	45,81%
0,6	41,08%	46,87%	42,52%	48,08%
0,4	**43,14%**	**48,57%**	**43,22%**	**48,41%**
0,2	38,99%	44,07%	40,46%	45,27%
0,0	36,74%	41,74%	38,94%	44,15%
Scene - local - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	33,01%	38,22%	34,95%	39,98%
0,8	39,53%	45,54%	42,35%	48,05%
0,6	**41,54%**	**47,59%**	**44,40%**	**50,03%**
0,4	40,43%	45,63%	42,89%	47,77%
0,2	39,14%	43,84%	42,00%	46,62%
0,0	38,97%	43,92%	40,22%	45,39%
Scene - local - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	34,75%	39,70%	38,06%	42,77%
0,8	40,28%	45,15%	44,82%	49,70%
0,6	41,70%	46,36%	46,00%	50,82%
0,4	41,21%	45,65%	46,22%	50,59%
0,2	42,34%	**46,94%**	47,93%	**52,33%**
0,0	**43,12%**	46,87%	**48,17%**	52,25%
Scene - global - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	40,90%	43,01%	42,13%	44,23%
0,8	**44,73%**	**48,09%**	45,21%	48,34%
0,6	42,99%	46,62%	**45,57%**	**49,42%**
0,4	39,00%	42,40%	40,71%	44,01%
0,2	36,49%	40,68%	38,16%	42,31%
0,0	36,74%	41,74%	38,94%	44,15%
Scene - global - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	42,09%	44,77%	45,63%	48,34%
0,8	45,88%	49,11%	**48,43%**	**51,65%**
0,6	45,68%	48,84%	45,98%	49,01%
0,4	**46,87%**	**50,07%**	48,27%	51,41%
0,2	40,86%	44,58%	43,23%	46,97%
0,0	38,97%	43,92%	40,22%	45,39%
Scene - global - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	45,70%	48,14%	49,02%	51,68%
0,8	46,71%	49,82%	49,88%	52,82%
0,6	46,50%	49,87%	50,22%	53,45%
0,4	**48,36%**	**51,07%**	**51,04%**	**53,78%**
0,2	44,68%	48,21%	49,46%	53,18%
0,0	43,12%	46,87%	48,17%	52,25%

Tabela 13 – Resultados para os experimentos com o conjunto de dados Emotions

Emotions - local - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	39,87%	50,11%	40,93%	50,80%
0,8	38,86%	48,95%	42,28%	51,72%
0,6	38,85%	49,70%	42,56%	52,31%
0,4	38,83%	49,15%	42,11%	51,43%
0,2	40,20%	51,32%	**44,12%**	**54,85%**
0,0	**41,74%**	**52,47%**	43,93%	54,47%
Emotions - local - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	38,47%	48,41%	46,52%	55,32%
0,8	38,36%	48,92%	45,85%	55,30%
0,6	40,52%	50,59%	47,83%	56,97%
0,4	40,30%	50,24%	**48,37%**	**57,60%**
0,2	42,55%	**52,79%**	48,19%	57,49%
0,0	**42,96%**	52,70%	46,32%	56,27%
Emotions - local - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**41,50%**	**51,32%**	54,73%	62,57%
0,8	40,56%	50,39%	53,63%	61,67%
0,6	40,62%	51,04%	54,64%	63,20%
0,4	39,41%	49,26%	55,69%	63,79%
0,2	39,78%	49,75%	**56,17%**	**64,41%**
0,0	40,39%	50,20%	52,14%	61,48%
Emotions - global - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	40,65%	50,30%	42,87%	51,77%
0,8	40,96%	50,07%	44,34%	53,01%
0,6	40,41%	50,13%	44,32%	53,73%
0,4	41,08%	51,23%	44,24%	53,85%
0,2	40,96%	51,58%	**44,77%**	**54,97%**
0,0	**41,74%**	**52,47%**	43,93%	54,47%
Emotions - global - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	42,13%	51,14%	46,52%	54,39%
0,8	42,65%	51,34%	47,93%	56,29%
0,6	**44,77%**	**53,52%**	48,14%	56,52%
0,4	43,85%	52,50%	**48,51%**	**57,10%**
0,2	41,67%	50,77%	47,10%	56,42%
0,0	42,96%	52,70%	46,32%	56,27%
Emotions - global - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	41,07%	49,76%	55,44%	62,58%
0,8	44,06%	52,81%	**56,07%**	**63,62%**
0,6	43,73%	52,53%	55,05%	62,83%
0,4	**45,57%**	**54,25%**	55,61%	63,35%
0,2	41,21%	49,68%	54,44%	62,59%
0,0	40,39%	50,20%	52,14%	61,48%

Tabela 14 – Resultados para os experimentos com o conjunto de dados Yeast

Yeast - local - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	33,79%	46,92%	35,03%	47,73%
0,8	33,10%	46,13%	34,78%	47,36%
0,6	32,82%	45,93%	34,64%	47,26%
0,4	33,97%	46,83%	35,69%	48,02%
0,2	35,28%	48,12%	36,74%	49,09%
0,0	**37,72%**	**50,31%**	**38,70%**	**51,02%**
Yeast - local - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	34,91%	47,73%	39,42%	51,57%
0,8	34,75%	47,43%	39,29%	51,42%
0,6	34,71%	47,37%	38,86%	50,81%
0,4	34,39%	46,89%	38,55%	50,37%
0,2	34,58%	47,18%	38,50%	50,36%
0,0	**38,24%**	**50,30%**	**40,03%**	**51,97%**
Yeast - local - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	36,34%	49,17%	44,76%	**56,23%**
0,8	36,41%	49,15%	**44,83%**	56,18%
0,6	36,94%	49,65%	44,54%	55,85%
0,4	35,54%	48,14%	44,57%	55,84%
0,2	35,43%	47,69%	43,98%	55,13%
0,0	**39,59%**	**51,71%**	44,12%	55,98%
Yeast - global - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	36,31%	48,46%	37,31%	49,07%
0,8	35,78%	48,02%	37,37%	49,35%
0,6	36,82%	49,19%	**38,70%**	50,80%
0,4	36,98%	49,25%	38,57%	50,56%
0,2	36,67%	49,01%	38,05%	50,06%
0,0	**37,72%**	**50,31%**	**38,70%**	**51,02%**
Yeast - global - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	36,79%	48,35%	40,19%	51,44%
0,8	36,07%	47,37%	39,30%	50,24%
0,6	37,83%	49,41%	**40,34%**	51,64%
0,4	37,47%	49,34%	40,03%	51,87%
0,2	36,96%	48,73%	39,43%	51,05%
0,0	**38,24%**	**50,30%**	40,03%	**51,97%**
Yeast - global - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	38,88%	50,29%	45,01%	55,67%
0,8	38,16%	49,63%	**45,25%**	55,65%
0,6	37,99%	49,41%	44,31%	54,88%
0,4	38,06%	49,28%	44,57%	55,32%
0,2	38,82%	50,62%	44,58%	55,93%
0,0	**39,59%**	**51,71%**	44,12%	**55,98%**

5.2 Análise Preliminar dos Resultados

Podemos ver que a performance do classificador foi muito bem para as bases de dados do Mulan, tendo a maioria dos melhores resultados em experimentos com nível de

supervisão semi-supervisionado ($\omega! = 0$ e $\omega! = 1$). Isso demonstra que o algoritmo realmente é capaz de extrair mais informações dos dados não rotulados para melhorar a performance do classificador. Também podemos notar nestes testes, que quanto mais instâncias rotuladas, maior é a performance de classificadores com ω maior. Esse resultado está conforma esperado, dado que quanto mais instâncias rotuladas, mais a versão supervisionada do algoritmo melhora o aprendizado e menos menos instância não rotuladas estão presentes para ser possível melhorar a classificação supervisionada.

Com relação aos experimentos com as bases de dados de proteína, podemos ver que na maioria dos casos o melhor resultado foi quando o nível de supervisão era máximo ($\omega = 1$), indicando que o algoritmos não foi capaz de melhorar a classificação com os dados não rotulados. É possível ver que para bases de dados de FT, o algoritmo conseguiu melhorar levemente o resultado para alguns casos, em comparação ao algoritmos puramente supervisionado. Mesmo nesses casos o aumento não é muito significativo, apenas no caso da base de dados Fungi_FT com $\omega = 0,2$, onde tivemos um aumento de mais de 1% em todas as medidas.

Um ponto não usual destes resultados é que, mesmo com pouquíssimas instâncias rotuladas, as métricas de avaliação para os experimentos com conjuntos de dados de proteínas foram muito altas. Não vemos o mesmo padrão aparecendo nas bases do Mulan, que tiveram métricas de avaliação bem mais baixas como o esperado. Isso é mais inusitado ainda quando vemos que nos experimentos com $\alpha = 5$, temos aproximadamente 30 instâncias rotuladas de Vírus (4,65% do total da base), 60 instâncias rotuladas de Viridiplantae (1,14% do total da base) e 50 instâncias rotulas de Fungi (0,91% do total da base) nestes experimentos.

Uma hipótese levantada para explicar esse ponto é que temos diversos tipos de termos da GO, cada um representando um conceito [1]. Por exemplo, temos o tipo ***involved_in***, que anota que uma proteínas está envolvida em algum processo biológico, ou então ***enables*** que anota que uma proteína permite que algum processo biológico ocorra. O possível problema é que dentro da GO temos a ontologia de componentes celulares, existem anotações de localizações com o tipo de termo ***located_in***. Esse tipo de termo aparece anotado em uma proteína quando temos evidências de que ela está presente em algum componente celular, ou seja, temos um atributo descritivo que representa quase o mesmo conceito do atributo alvo. Isso explicaria o motivo das métricas terem atingido resultados altos, mesmo com pouquíssimas proteínas, dado que esses atributos do tipo ***located_in*** sempre serão ótimos candidatos a serem o melhor atributo para particionar o conjunto de dados de um nó. Além disso, como o objetivo desse método é melhorar a classificação para taxonomias que não tem muitas proteínas anotadas, é esperado que essas não tenham poucos ou nenhum termo da GO do tipo ***located_in***, dado que não

[1] <http://geneontology.org/docs/go-annotation-file-gaf-format-2.2/>

temos experimentos que comprovem essa localização.

Para confirmar essa hipótese, uma nova bateria de testes foi rodada para os conjuntos de dados de proteínas, agora removendo todos os termos da GO to tipo ***located_in*** desses conjuntos de dados. Abaixo podemos ver os resultados desses experimentos. Novamente, os melhores resultados de cada experimento estarão destacados em vermelho.

Tabela 15 – Resultados para os experimentos com o conjunto de dados de Vírus - **sem termos located_in** - Abordagem local

Virus_10 - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	33,10%	38,90%	35,99%	41,68%
0,8	29,20%	35,82%	33,83%	39,78%
0,6	34,49%	42,19%	**38,31%**	45,58%
0,4	33,86%	41,59%	36,82%	44,25%
0,2	**35,48%**	**43,39%**	37,64%	45,28%
0,0	35,31%	43,31%	37,91%	**45,67%**
Virus_10 - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	36,80%	42,37%	**41,18%**	47,38%
0,8	37,08%	43,37%	41,02%	47,17%
0,6	34,87%	40,88%	39,32%	45,59%
0,4	35,42%	41,97%	39,46%	46,05%
0,2	36,67%	43,01%	39,98%	46,74%
0,0	**39,27%**	**46,19%**	41,14%	**48,40%**
Virus_10 - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**43,00%**	**49,49%**	**51,75%**	**57,62%**
0,8	40,83%	46,80%	49,47%	55,35%
0,6	42,00%	48,13%	50,00%	55,84%
0,4	41,53%	47,69%	49,76%	55,95%
0,2	42,50%	48,58%	50,00%	56,18%
0,0	40,25%	46,58%	47,76%	53,90%
Virus_FT - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	34,06%	40,82%	38,71%	44,81%
0,8	33,86%	40,99%	39,54%	45,79%
0,6	32,19%	39,10%	36,68%	42,70%
0,4	36,67%	43,85%	39,15%	**45,37%**
0,2	38,15%	**45,33%**	39,21%	45,48%
0,0	**40,29%**	44,17%	**40,06%**	43,72%
Virus_FT - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	36,36%	41,87%	42,35%	48,28%
0,8	37,89%	43,47%	41,55%	47,43%
0,6	39,04%	44,91%	42,56%	48,68%
0,4	41,04%	**47,25%**	43,62%	49,92%
0,2	**40,00%**	46,54%	**43,26%**	**50,10%**
0,0	39,13%	44,56%	40,18%	45,62%
Virus_FT - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	41,44%	48,02%	53,90%	59,75%
0,8	42,62%	49,57%	**54,68%**	**60,71%**
0,6	**44,54%**	**51,34%**	54,56%	60,38%
0,4	42,20%	49,20%	53,17%	59,22%
0,2	43,54%	50,74%	51,98%	58,10%
0,0	42,08%	48,24%	49,53%	55,57%

Tabela 16 – Resultados para os experimentos com o conjunto de dados de Vírus - **sem termos located_in** - Abordagem global

Virus_10 - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**36,10%**	42,99%	**39,71%**	**46,33%**
0,8	33,65%	41,20%	36,68%	44,04%
0,6	33,27%	41,29%	36,60%	44,16%
0,4	34,13%	41,91%	37,02%	44,42%
0,2	35,91%	**43,90%**	38,50%	46,18%
0,0	35,31%	43,31%	37,91%	45,67%
Virus_10 - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**44,16%**	**49,80%**	**44,04%**	**50,24%**
0,8	35,64%	41,10%	38,73%	44,84%
0,6	36,67%	43,14%	39,29%	46,20%
0,4	37,40%	44,09%	39,50%	46,54%
0,2	37,10%	43,86%	39,87%	47,08%
0,0	39,27%	46,19%	41,14%	48,40%
Virus_10 - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**45,00%**	**51,76%**	**53,35%**	**59,79%**
0,8	43,46%	50,45%	51,61%	57,86%
0,6	40,91%	47,43%	47,98%	54,01%
0,4	40,51%	46,98%	48,69%	54,64%
0,2	39,25%	45,40%	48,53%	54,43%
0,0	40,25%	46,58%	47,76%	53,90%
Virus_FT - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**44,15%**	**51,45%**	**46,38%**	**53,03%**
0,8	42,41%	50,47%	44,14%	51,68%
0,6	41,18%	47,03%	42,30%	47,59%
0,4	39,61%	46,27%	40,29%	45,94%
0,2	38,07%	44,43%	38,87%	44,49%
0,0	40,29%	44,17%	40,06%	43,72%
Virus_FT - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	44,22%	50,92%	46,66%	53,53%
0,8	44,10%	51,00%	**47,73%**	**54,47%**
0,6	**45,13%**	**52,15%**	47,42%	54,42%
0,4	40,55%	47,25%	43,60%	50,47%
0,2	44,44%	51,34%	45,91%	53,24%
0,0	39,13%	44,56%	40,18%	45,62%
Virus_FT - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	44,37%	50,67%	53,06%	59,04%
0,8	45,99%	52,85%	54,31%	60,66%
0,6	**46,18%**	**52,91%**	52,69%	58,92%
0,4	43,91%	51,15%	**54,53%**	**60,94%**
0,2	40,92%	47,76%	49,77%	56,20%
0,0	42,08%	48,24%	49,53%	55,57%

Tabela 17 – Resultados para os experimentos com o conjunto de dados de Viridiplantae - **sem termos located_in** - Abordagem local

Viridiplantae_10 - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**26,11%**	**30,05%**	**27,74%**	**31,76%**
0,8	23,96%	26,94%	24,59%	27,51%
0,6	24,82%	27,57%	25,44%	28,13%
0,4	25,07%	27,84%	25,66%	28,35%
0,2	25,08%	27,85%	25,67%	28,36%
0,0	25,13%	27,90%	25,71%	28,38%
Viridiplantae_10 - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**32,08%**	**36,72%**	**32,77%**	**37,24%**
0,8	25,64%	28,77%	25,90%	29,16%
0,6	24,66%	27,67%	24,73%	27,78%
0,4	24,33%	27,25%	24,73%	27,69%
0,2	24,01%	26,89%	24,55%	27,46%
0,0	24,07%	26,93%	24,48%	27,33%
Viridiplantae_10 - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**35,64%**	**39,97%**	**38,61%**	**43,15%**
0,8	29,28%	33,03%	32,35%	36,14%
0,6	29,10%	32,65%	31,50%	35,24%
0,4	28,62%	32,09%	31,12%	34,75%
0,2	28,67%	32,10%	30,97%	34,57%
0,0	29,11%	32,61%	30,81%	34,46%
Viridiplantae_FT - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	25,40%	29,35%	26,16%	30,18%
0,8	**27,64%**	32,08%	28,18%	32,67%
0,6	**27,64%**	**32,75%**	**28,39%**	**33,50%**
0,4	25,57%	29,92%	26,08%	30,46%
0,2	27,58%	32,61%	28,32%	33,29%
0,0	11,31%	13,63%	11,54%	13,93%
Viridiplantae_FT - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	31,42%	35,57%	32,11%	36,32%
0,8	30,69%	34,80%	30,93%	35,02%
0,6	31,63%	35,80%	31,65%	35,73%
0,4	**31,77%**	**36,31%**	**32,28%**	**36,92%**
0,2	30,21%	34,73%	30,95%	35,40%
0,0	10,10%	11,67%	10,89%	12,48%
Viridiplantae_FT - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	35,40%	39,69%	37,38%	41,78%
0,8	36,22%	40,69%	38,06%	42,61%
0,6	**36,50%**	**41,02%**	38,03%	42,69%
0,4	35,79%	40,39%	37,77%	42,48%
0,2	35,59%	40,66%	**38,08%**	**43,08%**
0,0	20,07%	22,60%	21,29%	23,93%

Tabela 18 – Resultados para os experimentos com o conjunto de dados de Viridiplantae - **sem termos located_in** - Abordagem global

Viridiplantae_10 - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**27,33%**	**30,33%**	**28,68%**	**31,68%**
0,8	25,00%	27,68%	25,57%	28,18%
0,6	25,11%	27,87%	25,70%	28,37%
0,4	25,11%	27,87%	25,71%	28,38%
0,2	25,11%	27,87%	25,71%	28,38%
0,0	25,13%	27,90%	25,71%	28,38%
Viridiplantae_10 - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**34,02%**	**38,15%**	**33,68%**	**37,65%**
0,8	25,05%	28,08%	25,11%	28,12%
0,6	24,58%	27,69%	25,06%	28,11%
0,4	24,09%	27,00%	24,66%	27,58%
0,2	24,06%	26,89%	24,52%	27,36%
0,0	24,07%	26,93%	24,48%	27,33%
Viridiplantae_10 - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**36,75%**	**40,16%**	**39,42%**	**42,93%**
0,8	28,85%	32,09%	31,16%	34,59%
0,6	29,16%	32,61%	31,25%	34,80%
0,4	28,96%	32,51%	31,00%	34,68%
0,2	28,91%	32,40%	30,84%	34,51%
0,0	29,11%	32,61%	30,81%	34,46%
Viridiplantae_FT - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	26,32%	30,70%	27,53%	31,94%
0,8	26,23%	30,09%	27,06%	31,12%
0,6	26,55%	31,14%	27,27%	31,92%
0,4	**28,84%**	**33,01%**	**28,97%**	**33,21%**
0,2	28,02%	32,70%	28,20%	33,00%
0,0	11,31%	13,63%	11,54%	13,93%
Viridiplantae_FT - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	31,58%	35,21%	32,34%	36,06%
0,8	**32,14%**	**35,71%**	**32,57%**	**36,18%**
0,6	30,96%	34,97%	31,42%	35,49%
0,4	30,53%	34,52%	31,42%	35,49%
0,2	30,37%	34,28%	31,03%	35,01%
0,0	10,10%	11,67%	10,89%	12,48%
Viridiplantae_FT - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	35,33%	38,85%	36,77%	40,45%
0,8	35,31%	39,17%	37,40%	41,23%
0,6	35,80%	39,50%	37,91%	41,63%
0,4	**37,22%**	**41,20%**	**38,84%**	**42,89%**
0,2	32,74%	37,13%	34,76%	39,29%
0,0	20,07%	22,60%	21,29%	23,93%

Tabela 19 – Resultados para os experimentos com o conjunto de dados de Fungi - **sem termos located_in** - Abordagem local

Fungi_10 - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**44,49%**	**49,78%**	**45,19%**	**50,23%**
0,8	39,62%	43,82%	39,87%	43,93%
0,6	39,90%	43,81%	40,10%	43,91%
0,4	40,16%	44,09%	40,36%	44,19%
0,2	39,98%	43,91%	40,01%	43,84%
0,0	39,88%	43,86%	39,91%	43,80%
Fungi_10 - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**48,24%**	**53,49%**	**49,50%**	**54,86%**
0,8	41,06%	46,17%	41,99%	47,12%
0,6	40,98%	45,74%	42,27%	47,00%
0,4	39,73%	44,43%	41,06%	45,73%
0,2	39,62%	44,26%	40,97%	45,62%
0,0	40,19%	44,96%	41,20%	45,94%
Fungi_10 - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**51,84%**	**56,85%**	**53,78%**	**58,68%**
0,8	45,83%	50,19%	46,99%	51,35%
0,6	43,76%	47,87%	45,01%	49,17%
0,4	41,79%	45,72%	42,79%	46,76%
0,2	42,00%	45,91%	43,01%	46,95%
0,0	41,66%	45,39%	42,42%	46,28%
Fungi_FT - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	42,23%	46,86%	42,47%	47,24%
0,8	42,50%	48,07%	42,31%	47,90%
0,6	42,70%	48,52%	42,60%	48,48%
0,4	**44,33%**	**49,85%**	**44,13%**	**49,84%**
0,2	41,85%	47,53%	41,86%	47,57%
0,0	12,39%	15,33%	12,61%	15,52%
Fungi_FT - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	48,54%	53,50%	49,14%	54,11%
0,8	48,71%	54,28%	49,60%	55,19%
0,6	48,97%	54,79%	49,79%	55,62%
0,4	**51,21%**	**57,12%**	**51,71%**	**57,69%**
0,2	50,61%	56,51%	51,21%	57,08%
0,0	0,00%	0,00%	0,00%	0,00%
Fungi_FT - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	52,39%	57,20%	53,70%	58,61%
0,8	52,90%	57,69%	54,00%	58,96%
0,6	**53,58%**	58,33%	**54,33%**	59,25%
0,4	53,05%	58,37%	53,73%	59,12%
0,2	53,22%	**58,84%**	53,80%	**59,40%**
0,0	17,58%	19,97%	18,14%	20,49%

Tabela 20 – Resultados para os experimentos com o conjunto de dados de Fungi - **sem termos located_in** - Abordagem global

Fungi_10 - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**43,80%**	**48,92%**	**43,49%**	48,64%
0,8	39,98%	43,91%	40,01%	**43,84%**
0,6	39,99%	43,92%	40,01%	**43,84%**
0,4	39,99%	43,92%	40,01%	**43,84%**
0,2	39,99%	43,92%	40,01%	**43,84%**
0,0	39,88%	43,86%	39,91%	43,80%
Fungi_10 - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**52,65%**	**58,02%**	**53,26%**	**58,52%**
0,8	39,37%	44,09%	40,66%	45,39%
0,6	39,83%	44,54%	41,12%	45,82%
0,4	40,04%	44,79%	41,21%	45,94%
0,2	40,26%	45,03%	41,23%	45,97%
0,0	40,19%	44,96%	41,20%	45,94%
Fungi_10 - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**55,00%**	**59,43%**	**55,67%**	**60,15%**
0,8	42,13%	46,19%	43,40%	47,45%
0,6	41,79%	45,63%	42,78%	46,69%
0,4	41,76%	45,60%	42,84%	46,77%
0,2	41,92%	45,70%	42,68%	46,58%
0,0	41,66%	45,39%	42,42%	46,28%
Fungi_FT - $\alpha = 5$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**44,14%**	**48,42%**	**44,57%**	**48,84%**
0,8	41,47%	47,53%	42,49%	48,65%
0,6	41,65%	47,27%	42,16%	47,82%
0,4	36,63%	42,21%	37,13%	42,69%
0,2	33,46%	40,17%	33,98%	40,51%
0,0	12,39%	15,33%	12,61%	15,52%
Fungi_FT - $\alpha = 10$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**52,69%**	**57,74%**	**53,34%**	**58,49%**
0,8	50,34%	54,90%	50,77%	55,55%
0,6	49,54%	54,22%	49,98%	54,75%
0,4	47,06%	52,33%	47,61%	52,86%
0,2	42,55%	48,43%	42,54%	48,43%
0,0	0,00%	0,00%	0,00%	0,00%
Fungi_FT - $\alpha = 20$				
W	**Acc indutiva**	**F1 indutiva**	**Acc transdutiva**	**F1 transdutiva**
1,0	**54,74%**	**59,27%**	**55,97%**	**60,45%**
0,8	54,27%	58,71%	55,07%	59,45%
0,6	53,75%	58,00%	54,69%	59,01%
0,4	52,80%	57,59%	54,40%	59,16%
0,2	46,98%	52,02%	47,10%	52,07%
0,0	17,58%	19,97%	18,14%	20,49%

5.3 Análise dos Resultados Finais

Com esses novos experimentos, as métricas de avaliação se comportaram de maneira esperada, melhorando a classificação com relação ao classificador supervisionado ($\omega = 1$) na maioria dos casos. Analisando de forma geral os resultados podemos notas o seguinte:

- **com relação às PCTs globais vs locais**: podemos observar que as PCTs globais em geral atingiram métricas melhores, porém com um aumento bem pouco significativo e que não é constante. Dessa forma podemos dizer que essa variação não altera o resultado do classificador de forma significativa;
- **com relação à variação do valor de** α: podemos observar que para todas as bases, quanto maior o valor de α, maior são as métricas de avaliação do experimento. Esse comportamento é esperado, dado que quanto maior o α, mais instâncias rotuladas temos para dar base para uma maior precisão da classificação rotuladas. Além disso, era esperado que conforme o α aumentasse, mais o melhor resultado tenderia à um nível de supervisão maior (ω maior). Esse comportamento é esperado, dado que quando maior o α, mais instância rotuladas temos, melhorando a precisão da classificação supervisionada e menos instâncias não rotulas temos, diminuindo a possibilidade dessas instâncias ajudarem no processo de classificação, porém não foi possível ver essa relação de forma clara. Provavelmente mais experimentos com mais valores de α serão necessários para fazer essa análise;
- **com relação aos conjuntos de dados de proteínas 10 vs FT**: podemos observar que classificadores treinados em conjuntos de dados de proteínas com FT atingem resultados maiores que classificadores treinados em conjuntos de dados binários. Vemos aumentos expressivos principalmente nas métrica indutivas dos classificadores virais;
- **com relação aos conjuntos de dados de proteínas com e sem termos do tipo located_in**: vemos que a hipótese apresentada na seção 5.2 parece estar correta e podemos observar que classificadores treinados em bases sem termos do tipo **located_in** apresentam resultados expressivamente mais baixos, porém representam melhor a realidade de taxonomias que possuem poucas proteínas com LSP anotada.

Por fim, podemos verificar que a PCT semi-supervisionada desenvolvida neste trabalho foi capaz de extrair informações dos dados não rotulados de forma a melhorar a predição de LSP para conjuntos de dados de proteínas com poucas PRs.

6 Conclusão

Primeiramente, a implementação do algoritmo proposto nesse projeto foi um grande desafio técnico que proporcionou um grande aprendizado ao autor sobre diversos temas relacionados à aprendizado de máquina. O projeto também irá servir de base para futuras pesquisas na área de aprendizado semi-supervisionado para PLSP.

No que diz respeito ao algoritmo de *predictive clustering tree* semi-supervisionado implementado, é possível notar, a partir dos experimentos realizados, que o mesmo foi capaz de utilizar dados rotulados e não rotulados no processo de aprendizagem, conseguindo superar sua versão supervisionado em diversos cenários semi-supervisionados.

O algoritmo também foi desenvolvido de forma flexível, dado que temos um parâmetro ω capaz de controlar o nível de supervisão da árvore de acordo com a necessidade. Se ω for igual a 1, a árvore de decisão atua exatamente como uma PCT supervisionado. Se ω for igual a 0, o algoritmo usa apenas os atributos descritivos dos dados rotulados e não rotulados para construir a árvore, sem as informações sobre os rótulos.

6.0.1 Trabalhos Futuros e Possíveis Melhorias

A implementação atual possui alguns débitos técnicos que impedem o uso do classificador em alguns cenários. Eles são:

- atualmente o algoritmo só aceita conjuntos de dados com atributos descritivos inteiramente nominais ou inteiramente numéricos. Isso impede que o algoritmo seja utilizando com bases de dados que possuam atributos nominais e numéricos ao mesmo tempo;
- quando $\omega < 1$, a cada *split* é necessário calcular o *BestTest* considerando todos os atributos descritivos. Esse processo é muito mais custoso computacionalmente, dado que ele tem complexidade exponencial e precisa percorrer todos os atributos descritivos, que em alguns conjuntos de dados podem chegar na casa dos milhares, como é o caso das bases de dados de proteínas das taxonomias Viridiplantae e Fungi. Isso não afetou tanto os experimentos deste trabalho, pois como estávamos simulando um cenário com pouquíssimas instâncias rotuladas, o algoritmo conseguiu chegar em um conjunto de dados homogêneo rapidamente, dado que o critério de parada da PCT é calculado apenas em cima das classes das PRs. Porém para casos com mais não tão poucas instâncias rotuladas, seria necessário pensar em alguma otimização, dado que a complexidade temporal cresceria exponencialmente;

Como possíveis trabalhos futuros, poderíamos tratar os seguintes pontos:

- utilizar a PCT criada em um algoritmo de *random forest* e de *boosting* para investigar se a classificação poderia ser melhorada ainda mais. Esse algoritmo já está implementado internamente no Clus atualmente e poderia ser facilmente estendido para a PCT semi-supervisionada criada. Além disso, com o *random forest*, seria possível verificar o ranking de atributos e entender melhor qual atributo descritivo mais influencia na decisão dos classificadores;
- investigar com mais detalhes a influência do nível de supervisão do algoritmo controlado pelo peso ω, realizando experimentos com passos menores, dado que o passo utilizado nesse trabalho foi de 0,2;
- realizar uma comparação contra algoritmos do estado da arte de classificação supervisionada para para averiguar se o classificador semi-supervisionado desenvolvido é capaz de utilizar os dados não rotulados para superar até mesmo esses classificadores.

Referências

ALTSCHUL, S. F. et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucl Acids Res*, v. 25, p. 3389–3402, 1997. Citado na página 26.

BLOCKEEL, H.; DZEROSKI, S.; GRBOVIĆ, J. Simultaneous prediction of multiple chemical parameters of river water quality with tilde. In: ŻYTKOW, J. M.; RAUCH, J. (Ed.). *Principles of Data Mining and Knowledge Discovery*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999. p. 32–40. Citado na página 33.

BLOCKEEL, H.; RAEDT, L. D.; RAMON, J. Top-down induction of clustering trees. In: *Proceedings of the Fifteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998. (ICML '98), p. 55–63. ISBN 1-55860-556-8. Citado 3 vezes nas páginas 31, 32 e 33.

BLUM, T.; BRIESEMEISTER, S.; KOHLBACHER, O. MultiLoc2: integrating phylogeny and Gene Ontology terms improves subcellular protein localization predictiony. *BMC Bioinformatics*, v. 10, p. 274, 2009. Citado na página 26.

BOUTELL, M. et al. Learning multi-label scene classification. *Pattern Recognition*, v. 37, p. 1757–1771, 09 2004. Citado na página 28.

BOUTELL, M. R. et al. Learning multi-label scene classification. *Pattern Recognition*, v. 37, n. 9, p. 1757–1771, 2004. Citado na página 16.

BREIMAN, L. et al. *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984. Citado 2 vezes nas páginas 29 e 32.

CAMPBELL, C. Handbook of Molecular and Cellular Methods in Biology and Medicine. In: _____. Vienna, Austria, Austria: Physica Verlag Rudolf Liebing KG, 2001. p. 331–335. ISBN 3-7908-1367-2. Radial basis function networks 1: recent developments in theory and applications. Citado na página 15.

CAO, J. et al. Mining proteins with non-experimental annotations based on an active sample selection strategy for predicting protein subcellular localization. *PLOS One*, 2013. Citado na página 17.

CARAGEA, C. et al. Semi-supervised prediction of protein subcellular localization using abstraction augmented markov models. *BMC Bioinformatics*, v. 11, n. Suppl 8, p. S6, 2010. Citado na página 17.

CERRI, R. et al. Reduction strategies for hierarchical multi-label classification in protein function prediction. *BMC Bioinformatics*, v. 17, n. 1, p. 373, 2016. Citado na página 16.

CHARTE, F. et al. Addressing imbalance in multilabel classification: Measures and random resampling algorithms. *Neurocomputing*, v. 163, p. 3 – 16, 2015. Citado na página 28.

CHAWLA, N. V.; KARAKOULAS, G. I. Learning from labeled and unlabeled data: An empirical study across techniques and domains. *CoRR*, abs/1109.2047, 2011. Citado na página 39.

CHEN, F. et al. OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups. *Nucleic Acids Res*, v. 34, n. Database issue, p. D363 – 368, 2006. Citado na página 26.

CHOU, K. C.; SHEN, H. B. Hum-PLoc: a novel ensemble classifier for predicting human protein subcellular localization. *Biochem Biophys Res Commun*, v. 347, p. 150–157, 2006. Citado 2 vezes nas páginas 26 e 27.

CHOU, K. C.; SHEN, H. B. Large-scale predictions of gram-negative bacterial protein subcellular locations. *J Proteome Res*, v. 5, p. 3420–3428, 2006. Citado 2 vezes nas páginas 26 e 27.

CHOU, K. C.; SHEN, H. B. Predicting eukaryotic protein subcellular location by fusing optimized evidence-theoretic K-Nearest neighbor classifiers. *J Proteome Res*, v. 5, p. 1888–1897, 2006. Citado 2 vezes nas páginas 26 e 27.

CHOU, K. C.; SHEN, H. B. Euk-mPLoc: a fusion classifier for large-scale eukaryotic protein subcellular location prediction by incorporating multiple sites. *J Proteome Res*, v. 6, p. 1728–1734, 2007. Citado na página 26.

CHOU, K.-C.; SHEN, H.-B. Plant-mPLoc: A Top-Down Strategy to Augment the Power for Predicting Plant Protein Subcellular Localization. *PLoS One*, v. 5, n. 6, p. e11335, 2010. Citado na página 15.

CLARE, A.; KING, R. D. Knowledge discovery in multi-label phenotype data. In: *5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD2001)*. [S.l.]: Springer, 2001. (LNAI, v. 2168), p. 42–53. Citado na página 16.

CUI, Q. et al. Esub8: A novel tool to predict protein subcellular localizations in eukaryotic organisms. *BMC Bioinformatics*, v. 5, p. 66, 2004. Citado na página 15.

DEMšAR, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, v. 7, p. 1–30, 2006. Citado na página 33.

DZEROSKI, S. et al. Analysis of time series data with predictive clustering trees. In: *Proceedings of the 5th International Conference on Knowledge Discovery in Inductive Databases*. Berlin, Heidelberg: Springer-Verlag, 2007. p. 63–80. Citado na página 33.

ELISSEEFF, A.; WESTON, J. A kernel method for multi-labelled classification. In: . [S.l.: s.n.], 2001. v. 14, p. 681–687. Citado na página 28.

ELISSEEFF, A.; WESTON, J. Kernel Methods for Multi-labelled Classification and Categorical Regression Problems. In: *Advances in Neural Information Processing Systems*. [S.l.]: MIT Press, 2001. p. 681–687. Citado na página 16.

FINN, R. D.; CLEMENTS, J.; EDDY, S. R. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res*, v. 39, n. suppl 2, p. W29 – 37, 2011. Citado na página 26.

GLORY, E.; MURPHY, R. F. Automated subcellular location determination and high-throughput microscopy. *Developmental cell*, v. 12, n. 1, p. 7–16, 2007. Citado na página 15.

GODBOLE, S.; SARAWAGI, S. Discriminative methods for multi-labeled classification. In: *8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2004. p. 22–30. Disponível em: <http://www.springerlink.com/content/maa4ag38jd3pwrc0>. Citado na página 34.

GONÇALVES, T.; QUARESMA, P. A preliminary approach to the multilabel classification problem of portuguese juridical documents. In: *EPIA*. [S.l.: s.n.], 2003. p. 435–444. Citado na página 16.

HAWKINS, D. M. The Problem of Overfitting. *J. Chem. Inf. Comput. Sci.*, v. 44, p. 1–12, 2004. Citado na página 17.

HUNTLEY, R. P. et al. The GOA database: gene ontology annotation updates for 2015. *Nucleic Acids Res*, v. 43, p. D1057–D1063, 2015. Citado na página 28.

JUNCKER, A. S. et al. Sequence-based feature prediction and annotation of proteins. *Genome Biology*, v. 10, p. 206, 2009. Citado na página 22.

KARALIC, A.; PIRNAT, V. Significance level based multiple tree classification. In: *Informatica*. [S.l.: s.n.], 1991. v. 15, n. 5, p. 12. Citado na página 16.

KOCEV, D.; SLAVKOV, I.; DZEROSKI, S. Feature ranking for multi-label classification using predictive clustering trees. In: . [S.l.: s.n.], 2013. Citado 2 vezes nas páginas 32 e 35.

LAUSER, B.; HOTHO, A. Automatic multi-label subject indexing in a multilingual environment. In: *Proc. of the 7th European Conference in Research and Advanced Technology for Digital Libraries, ECDL 2003*. [S.l.]: Springer, 2003. v. 2769, p. 140–151. Citado na página 16.

LEVATIć, J. et al. Semi-supervised classification trees. *J. Intell. Inf. Syst.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 49, n. 3, p. 461–486, dez. 2017. Citado 3 vezes nas páginas 24, 35 e 39.

LI, L.; STOECKERT, C. J. J.; ROOS, D. S. OrthoMCL: Identification of Ortholog Groups for Eukaryotic Genomes. *Genome Res*, v. 13, n. 9, p. 2178–2189, 2003. Citado na página 26.

LUO, X.; ZINCIR-HEYWOOD, N. A. Evaluation of two systems on multi-class multi-label document classification. In: *International Syposium on Methodologies for Intelligent Systems*. [S.l.: s.n.], 2005. p. 161–169. Citado na página 16.

MEI, S.; FEI, W.; ZHOU, S. Gene ontology based transfer learning for protein subcellular localization. *BMC Bioinformatics*, v. 12, p. 44, 2011. Citado na página 26.

MISTRY, J. et al. Challenges in homology search: HMMER3 and convergent evolution of coiled-coil regions. *Nucleic Acids Res*, v. 41, n. 12, p. e121, 2013. Citado na página 26.

PISE, N. N.; KULKARNI, P. A survey of semi-supervised learning methods. In: *2008 International Conference on Computational Intelligence and Security*. [S.l.: s.n.], 2008. v. 2, p. 30–34. Citado na página 24.

PRAKASH, V. J.; NITHYA, L. M. A survey on semi-supervised learning techniques. *CoRR*, abs/1402.4645, 2014. Citado na página 24.

QIN, Y. et al. Research progress on semi-supervised clustering. *Cogn Comput*, v. 11, p. 599–612, 2019. Citado na página 23.

QUINLAN, J. R. *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0. Citado na página 21.

REY, S.; GARDY, J. L.; BRINKMAN, F. S. Assessing the precision of high-throughput computational and laboratory approaches for the genome-wide identification of protein subcellular localization in bacteria. *BMC Genomics*, v. 6, p. 162, 2005. Citado na página 15.

SADARANGANI, A.; JIVANI, A. A survey of semi-supervised learning. *International Journal of Engineering Sciences & Research Technology*, v. 5, n. 10, p. 138–143, 2016. Citado 2 vezes nas páginas 23 e 24.

SHEN, H.; CHOU, K. Virus-mploc: a fusion classifier for viral protein subcellular location prediction by incorporating multiple sites. *J Biomol Struct Dyn.*, v. 28, n. 2, p. 175–186, 2010. Citado na página 17.

SHEN, X. et al. Multilabel machine learning and its application to semantic scene classification. In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. [S.l.: s.n.], 2003. v. 5307, p. 188–199. Citado na página 16.

STRUYF, J.; DZEROSKI, S. Constraint based induction of multi-objective regression trees. In: *Proceedings of the 4th International Conference on Knowledge Discovery in Inductive Databases*. Berlin, Heidelberg: Springer-Verlag, 2006. (KDID'05), p. 222–233. ISBN 3-540-33292-8, 978-3-540-33292-3. Citado na página 33.

STRUYF, J.; DZEROSKI, S. Clustering trees with instance level constraints. In: *Proceedings of the 18th European Conference on Machine Learning*. Berlin, Heidelberg: Springer-Verlag, 2007. (ECML '07), p. 359–370. Citado na página 33.

STRUYF, J. et al. Clus: User's manual. 06 2018. Citado na página 33.

TAN, P.-N. et al. *Introduction to Data Mining, (First Edition)*. USA: Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN 0321321367. Citado na página 29.

The UniProt Consortium. UniProt: the universal protein knowledgebase. *Nucleic Acids Research*, v. 45, n. Issue D1, p. D158–D169, 2017. Citado na página 25.

TRIGUERO, I.; GARCIA, S.; HERRERA, F. Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowl. Inf. Syst.*, Springer-Verlag New York, Inc., New York, NY, USA, v. 42, n. 2, p. 245–284, fev. 2015. Citado 2 vezes nas páginas 23 e 24.

TROCHIDIS, K. et al. Multi-label classification of music into emotions. In: . [S.l.: s.n.], 2008. v. 2011, p. 325–330. Citado na página 28.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. P. Mining Multi-label Data. In: MAIMON, O.; ROKACH, L. (Ed.). *Data Mining and Knowledge Discovery Handbook*. 2nd. ed. [S.l.]: Springer, 2010. p. 667–685. Citado na página 21.

TSOUMAKAS, G. et al. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, v. 12, p. 2411–2414, 2011. Citado na página 28.

VAPNIK, V. N. *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995. ISBN 0-387-94559-8. Citado na página 21.

VENS, C. et al. Decision trees for hierarchical multi-label classification. *Machine Learning*, Kluwer Academic Publishers, Hingham, MA, USA, v. 73, p. 185–214, 2008. ISSN 0885-6125. Citado 4 vezes nas páginas 16, 32, 33 e 38.

WAN, S.; MAK, M. W. Machine Learning for Protein Subcellular Localization Prediction. In: _____. 4. ed. Berlin, Germany, Germany: De Gruyter, 2015. p. 7–26. ISBN 3-7908-1367-2. Citado 3 vezes nas páginas 25, 26 e 27.

WAN, S.; MAK, M. W.; KUNG, S. Y. mGOASVM: Multi-label protein subcellular localization based on gene ontology and support vector machines. *BMC Bioinformatics*, v. 13, p. 290, 2012. Citado 3 vezes nas páginas 16, 17 e 22.

WAN, S.; MAK, M. W.; KUNG, S. Y. GOASVM: a subcellular location predictor by incorporating term-frequency gene ontology into the general form of Chou's pseudo-amino aci composition. *J Theor Biol*, v. 323, p. 40 – 48, 2013. Citado na página 22.

WANG, Y. et al. Semi-supervised learning based on nearest neighbor rule and cut edges. *Knowledge-Based Systems*, v. 23, n. 6, p. 547 – 554, 2010. ISSN 0950-7051. Citado na página 38.

XU, Q. et al. Semi-supervised protein subcellular localization. *BMC Bioinformatics*, v. 10, n. Suppl 1, p. S47, 2009. Citado na página 17.

ZDOBNOV, E. M.; APWEILER, R. InterProScan – an integration platform for the signature-recognition methods in InterPro. *Bioinformatics*, v. 17, p. 847–848, 2001. Citado na página 26.

ZHANG, M.-L.; ZHOU, Z.-H. A k-Nearest Neighbor Based Algorithm for Multi-label Classification. In: THE IEEE COMPUTATIONAL INTELLIGENCE SOCIETY. [S.l.], 2005. v. 2, p. 718–721 Vol. 2. Citado na página 16.

ZHU, X. Semi-supervised learning literature survey. v. 2, 07 2008. Citado na página 23.

ZHU, X.; GOLDBERG, A. B. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, v. 3, n. 1, p. 1–130, 2009. Citado na página 23.