Aprendizagem de Máquina na Identificação de Regiões Codantes em Sequências de DNA de Fungos Filamentosos

Gustavo Henrique Ferreira Cruz
Universidade Estadual de Maringá
Maringá, Paraná, Brasil
ra109895@uem.br

Josiane Melchiori Pinheiro
Universidade Estadual de Maringá
Maringá, Paraná, Brasil
jmpferreira@uem.br

Gustavo Luiz Furuhata Ferreira
Universidade Estadual de Maringá
Maringá, Paraná, Brasil
gustavo_furuhata@hotmail.com

Vinícius Menossi
Universidade Estadual de Maringá
Maringá, Paraná, Brasil
ra108840@uem.br

Antônio Roberto dos Santos
Universidade Estadual de Maringá
Maringá, Paraná, Brasil
ra102450@uem.br

Sarah Anduca de Oliveira
Universidade Estadual de Maringá
Maringá, Paraná, Brasil
ra115506@uem.br

ABSTRACT

The task of identifying intron and exon regions in genes is a very complex task, and it is necessary to identify certain nucleotide patterns in the gene sequence. This task can be done manually or through software that most often uses genetic alignment techniques, which is not a very effective way for this purpose. In this opportunity for collaboration between biology and computer science using machine learning techniques, the objective was to predict the intron and exon regions in filamentous fungi genes as well to translate the identified regions intro proteic codons. In this paper, the problem was modeled as a supervised learning problem, based on training a set of genes obtained from GenBank that already have the intron and exon regions identified. The machine learning model used in this work was the Condicional Random Fields (CRF). Through the values resulting from the metrics applied to the model, it can be seen that it is possible to achieve a good precision in the task of identifying the intron and exon regions as well the proteic codons. Thus, although there is a need for a greater diversity of database characteristics to support the effectiveness of identifying the splicing sites, this paper gives evidence that it is possible to predict these splicing sites with a good accuracy.

KEYWORDS

Machine Learning, DNA, Regiões Codantes, Conditional Random Fields

1 INTRODUÇÃO

A bioinformática evoluiu muito desde a descoberta e do estudo dos genomas humanos. Essa evolução permitiu realizar a manipulação de diversos dados biológicos por meio de programas computacionais especializados que podem realizar o processamento, armazenamento e distribuição da informação biológica [1].

Uma importante tarefa para entender e analisar o genoma é a identificação de regiões do gene que contém informações para codificar uma proteína. Os genes dos organismos eucariontes são formados por segmentos alternados de regiões codantes, os éxons, e regiões não codantes, os íntrons. Os íntrons são eliminados do gene e os éxons são unidos para produzir a proteína correspondente, em um processo chamado de *splicing*. Grande parte das ferramentas disponíveis nos dias de hoje realiza a identificação de éxons por meio de alinhamento, porém, as repetições no genoma, os erros de leitura e o tamanho curto de segmentos não mapeados tornam a tarefa de alinhamento um tanto complexa e exaustiva [2].

A determinação das regiões de íntrons e éxons é muito importante no diagnóstico de doenças genéticas. Segundo Makal *et al.* [3], 15% das mutações que causam doenças genéticas são originárias de erros de *splicing*.

Nos últimos anos, várias abordagens utilizando Aprendizado de Máquina (AM) tem sido publicadas no sentido de auxiliar na identificação dessas regiões codantes [2–5], muitas delas voltadas para o genoma humano e que treinam modelos de Redes Neurais. Este artigo descreve duas abordagem para modelar o problema de identificação de regiões codantes, especificamente em genes de fungos filamentos, como um problema de aprendizado supervisionado na classificação de sequências. Diferentemente da maioria das abordagens que treinam Redes Neurais, este trabalho treinou modelos baseados no algoritmo *Conditional Random Fields* (CRF), utilizado em problemas como previsão de estrutura secundária de RNA e processamento de linguagem natural [6]. O objetivo foi exatamente verificar a possibilidade de aplicação de modelos baseados em probabilidade para classificar as regiões codantes e não codantes dos genes.

A primeira abordagem utiliza a informação das "sequências de consenso" sendo no contexto deste trabalho a presença das bases GU no início do íntron e AG no seu final. A partir de uma sequência proveniente do GenBank[1], foram geradas todas as possibilidades de íntrons e éxons para esta sequência baseando-se na sequência de consenso. Uma base de dados foi construída com essas informações, na qual as sequências eram a única *feature* do exemplo e cada exemplo poderia ser classificado para uma das seguintes classes: éxon, íntron, ou *neither* - uma mistura de íntrons e éxons.

A segunda abordagem utiliza a informação de que, após o processo de *splicing*, cada sequência de três nucleotídeos, chamados códons, são mapeados para um aminoácido (estrutura base das proteínas). Dessa forma, a partir de uma sequência proveniente do GenBank, foram geradas todas as trincas possíveis da sequência, inclusive por meio da técnica de janela deslizante. Uma base de dados foi construída com essas informações, na qual cada códon, juntamente com seu códon anterior e posterior eram as *features* de

[1] https://www.ncbi.nlm.nih.gov/genbank/

cada exemplo que poderia ser classificado para uma das seguintes classes: III, IIE, IEE, EEE, EEI, EII, para as quais um "I" denota um nucleotídeo que faz parte de um íntron e "E" denota um nucleotídeo que faz parte de um éxon.

As duas abordagens treinaram o modelo CRF, com o objetivo de identificar as regiões codantes (éxons) e não codantes (íntrons) nas sequências de genes de fungos filamentos. Os resultados gerados pelo CRF foram bastante interessantes, mostrando que um modelo baseado em probabilidade também pode gerar bons resultados para esse tipo de problema.

Este artigo está organizado da seguinte forma: a seção 2 descreve os conceitos de íntrons e éxons assim como outros dados biológicos; a seção 3 apresenta uma elucidação sobre os conceitos e técnicas de Aprendizado de Máquina (AM) assim como uma revisão da literatura de trabalhos com problemáticas relacionadas ao domínio desta pesquisa; as seções 4 e 5 apresentam, respectivamente, as duas abordagens escolhidas para o desenvolvimento da pesquisa e os materiais e métodos utilizados; a seção 6 apresenta os resultados obtidos; a seção 7 apresenta as conclusões e a seção 8 as perspectivas futuras para este trabalho.

2 DNA, REGIÕES CODANTES E SÍNTESE DE PROTEÍNAS

Os segmentos do DNA (ácido desoxirribonucleico) que possuem informações genéticas são denominados genes e o restante do DNA é responsável por estruturar e regular essa informação genética.

O DNA é constituído por estruturas que se repetem ao longo de sua estrutura chamadas de nucleotídeos. Dentro dos seres vivos o DNA não é encontrado como uma cadeia simples e sim como um par de moléculas associadas na qual duas cadeias de polinucleotídeos se unem em forma de dupla hélice, sendo que os nucleotídeos são unidos por ligações fosfodiéster. A dupla hélice é ligada por pontes de hidrogênio entre as bases, sendo que as quatro bases encontradas no DNA são a Adenina (A), Citosina (C), Guanina (G) e Timina (T). Essas bases são ligadas de maneira que Adenina (A) se liga à Timina (T) e Citosina (C) se liga à Guanina (G), sendo essa estrutura chamada de par de bases. Com essas ligações em pares, toda informação que uma fita de DNA armazena também é armazenada na outra fita ligada a ela, o que é essencial na duplicação do DNA [7].

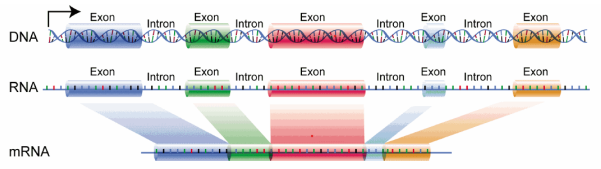

Figura 1: Figura ilustrativa da localização dos éxons, via National Human Genome Research Institute (domínio público).

Para que a realização da síntese de proteínas seja possível, é necessário que um RNA mensageiro (mRNA) seja gerado a partir do DNA. Como no RNA (ácido ribonucleico) não há a existência da Timina (T), ela é substituída por outra base, chamada de Uracila (U). Após o mRNA ser formado, ele sai do núcleo da célula e é liberado no citoplasma até o encontro dos ribossomos, onde ocorre o processo de síntese de proteínas conhecido como tradução [7]. Essa relação entre DNA, RNA e mRNA é ilustrada na Figura 1 .

Algumas partes do mRNA não são codantes, ou seja, algumas partes do mRNA são descartadas e não participam do processo da tradução para a proteína. Essas regiões não codantes são denominadas íntrons e os intervalos codantes que irão fazer parte efetivamente do processo de tradução são denominados éxons. Todos os organismos eucariontes possuem íntrons em suas estruturas genéticas e antes do processo de tradução é necessário remover essas partes do mRNA através de um mecanismo molecular chamado spliceossomo, no processo conhecido como *splicing* de RNA [8].

Segundo Alberts *et al.* [9], há padrões nos nucleotídeos quase invariantes no processo de identificação de íntrons, esses padrões são chamados de "sequências consenso", sendo um desses a presença das bases GU no início do íntron e AG no seu final.

Após o processo de *splicing* ser finalizado no mRNA, é iniciado o processo de tradução. O processo de tradução consiste em traduzir uma sequência de nucleotídeos do mRNA em uma sequência de aminoácidos. Esse processo ocorre em trincas, ou seja, três nucleotídeos específicos, chamados códons, para cada aminoácido [10].

3 APRENDIZADO DE MÁQUINA E A IDENTIFICAÇÃO DE REGIÕES CODANTES

A Inteligência Artificial (IA), mais especificamente o Aprendizado de Máquina (AM), tem sido amplamente utilizado nas mais diversas áreas de conhecimento que precisam analisar grandes volumes de dados, encontrar padrões nesses dados, resolver problemas complexos até pouco tempo dominados pelo ser humano, entre outras. Muitas tarefas específicas de determinadas áreas, que normalmente são/eram feitas de forma manual nos laboratórios de pesquisa, tem sido cada vez mais automatizadas em razão do avanço da IA e do AM em especializar cada vez mais as técnicas de resolução de problemas complexos.

Um problema que pode ser resolvido por aprendizado de máquina normalmente se enquadra em um dos três tipos de problemas:

(i) Aprendizado de máquina supervisionado: quando os dados necessários para treinar um modelo possuem informações de rótulo ou classe de cada exemplo;
(ii) Aprendizado de máquina não-supervisionado: quando os dados necessários para treinar um modelo não possuem informações de rótulo ou classe dos exemplos, mas possuem informações suficientes para que sejam criadas métricas de comparação entre os exemplos;
(iii) Aprendizado por reforço: quando os dados necessários para treinar um modelo não possuem informações de rótulo ou classe dos exemplos diretamente, mas é possível "experimentar" determinadas ações para as quais o sistema fornecerá um resultado positivo (recompensa) ou não. Com base nesses resultados, o sistema pode aprender quais valores de entrada resultam em boas recompensas.

Este trabalho aborda o problema de identificação de íntrons e éxons em sequências de DNA de fungos filamentos como um problema de aprendizado de máquina supervisionado. Os dados utilizados para treinar os modelos propostos possuem informações

sobre a classe/rótulo correto para cada um dos exemplos da base de dados, como será descrito na seção 4.

O *Conditional Random Fields* (CRF) é uma classe de algoritmos de aprendizado supervisionado que buscam resolver problemas de classificação utilizando informações de dados vizinhos (ou dados adjacentes) como contexto para realizar essas predições [11]. Uma das principais características dos CRFs é classificar dados sequenciais, justamente por utilizar informações do contexto, como classificações e características de dados adjacentes, para aumentar a quantidade de informações que o modelo treinado tem para realizar melhores predições. Como este trabalho tem as sequências genéticas como material de estudo, é proposta a utilização do CRF como algoritmo de AM para gerar os modelos deste trabalho.

O CRF é derivado do Modelo Oculto de Markov (*Hidden Markov Model* - HMM) e do Modelo de Markov de Entropia Máxima (*Maximum Entropy Markov Model* - MEMM). O HMM é um modelo característico por calcular a probabilidade de um estado futuro a partir somente do estado atual e não do conjunto de estados passados e anteriores. O MEMM é um modelo discriminativo que combina características do HMM e de um modelo chamado Modelo de Entropia Máxima (*Maximum Entropy Method* - MEM) e se caracteriza por ser discriminativo e inspirado no classificador de entropia máxima. O CRF difere-se desses dois algoritmos pois para determinar a classe de um exemplar, leva em consideração também a classe anterior e a posterior do exemplar. Além disso, o CRF realiza maximizações para todas as classificações de forma conjunta, ao contrário do MEMM que realiza esse processo para cada transição [12].

Culotta, Kulp e McCallum [13] descrevem a utilização do CRF na predição de genes, sendo que a escolha dessa técnica se deve ao fato de que CRFs podem naturalmente incorporar características arbitrárias e não independentes da entrada sem fazer suposições de independência condicional entre os recursos. A base de dados utilizada foi obtida do GenBank, *release* 105, 1998, e consiste em um conjunto de 450 genes humanos (mais de 5 milhões de bases). Os dados foram divididos aleatoriamente em 70% de treinamento e 30% de teste. O aprendizado foi realizado com três variantes de CRF criadas para treinamento com conjuntos de *features* diferentes. Entre as três variantes do CRF, o melhor resultado obteve o valor de 0,86 na medida F1.

Rätsch *et al.* [14] descreve a utilização da técnica de Máquina de Vetor de Suporte (SVM) para modelar e prever como o processo de *splicing* atua. Os autores utilizaram somente o genoma do nematoide *Caenorhabditis elegans* para o processo de aprendizagem, uma vez que este é um dos organismos mais estudados em biologia experimental, pois apresenta uma facilidade de manipulações genéticas e experimentais. A escolha do SVM se deve ao fato de que essa técnica já foi utilizada anteriormente com considerável sucesso em uma variedade de campos, incluindo biologia computacional. A base de dados utilizada foi obtida do Wormbase 3, versão WS120, que consiste em repositório de dados biológicos de *Caenorhabditis elegans*. As taxas de acerto na identificação de íntrons e éxons foi de 100%, considerando que 87% (regiões codificadoras e não traduzidas) e 95% (somente regiões codificadoras) de todos os genes foram testados em várias avaliações fora da amostra.

O estudo de Makal, Ozyilmaz e Palavaroglu [3], utiliza Redes Neurais para encontrar e determinar as regiões de transição de íntrons para éxons (*acceptor splice sites*) ou de éxons para íntrons (*donor splice sites*), as chamadas regiões de *splicing*. Os autores, ao experimentarem o uso de três implementações de Redes Neurais (*Multi-layer Perceptron* (MLP), *Radial Basis Function* (RBF) e *Generalized Regression Neural Networks* (GRNN)), obtiveram precisões entre 89,35% e 91,23%.

Outras abordagens utilizam diversos tipos de Redes Neurais, como Redes Neurais Recorrentes e Redes Neurais Convolucionais [4, 5, 15], para tentar identificar as regiões de *splicing* em sequências de DNA dos mais diversos organismos e espécies. Este trabalho utilizou o algoritmo CRF, na tentativa de uma abordagem diferente, baseada em probabilidade e no Modelo Oculto de Markov, para treinar um modelo específico para fungos filamentosos.

4 ABORDAGENS PROPOSTAS PARA MODELAR O PROBLEMA

Este trabalho descreve duas abordagens utilizadas para modelar o problema de identificação de íntrons e éxons como um problema de aprendizagem supervisionada. As duas abordagens treinaram o modelo CRF para identificar as regiões de íntrons e éxons em dois fungos filamentosos, o *Colletotrichum* e o *Diaporthe*.

As bases de dados construídas para as duas abordagens possuem dados provenientes do GenBank, um banco de dados público, amplamente utilizado pelos cientistas que trabalham com sequências genéticas, pois possui anotações de sequências de nucleotídeos dos mais diversos seres vivos. Mantido e gerenciado pelo *National Center for Biotechnology Information*, o NCBI, o GenBank conta com milhares de sequências das mais diversas espécies, incluindo os fungos *Colletotrichum* e *Diaporthe*.

4.1 Abordagem 1 - Sequências de consenso

A primeira abordagem utiliza a informação das "sequências de consenso", ou seja, a presença das bases GU no início do íntron e AG no final. A partir de uma sequência proveniente do GenBank, foram geradas todas as possibilidade de íntrons e éxons para essa sequência baseando-se na sequência de consenso.

A Figura 2 mostra como uma sequência é processada e quais possibilidades de íntrons e éxons foram geradas.

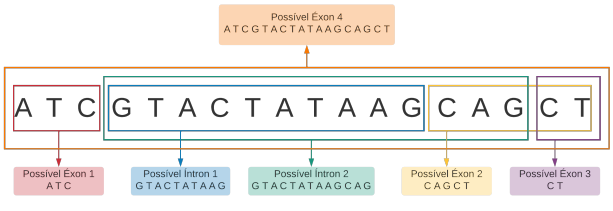

Figura 2: Exemplo de extração das possibilidades de íntrons e éxons.

Com base nas informações do GenBank de quais sequências realmente representam os íntrons e os éxons naquela sequência de origem e das possíveis sequências geradas (Figura 2), foram criados os exemplos que irão compor a base de dados para treinar o modelo CRF. Cada exemplo que compõe a base é representado por uma única *feature* - a própria sequência candidata a íntron ou éxon, e a classificação dessa sequência, de acordo com as informações da

sequência de origem no GenBank. Cada exemplo pode ser classificado em uma das seguintes classes: éxon, íntron, ou *neither* - uma mistura de nucleotídeos de íntrons e éxons.

A quantidade de exemplos gerados que seriam classificados como *neither* é bem maior do que a quantidade de exemplos de íntrons e éxons. Dessa forma, para manter a quantidade de exemplos de cada classe equilibrada com as demais na base de dados, foram escolhidos aleatoriamente exemplos *neither* em uma quantidade próxima da quantidade de íntron/éxons.

É importante destacar que treinar o classificador é apenas um passo no processo todo de identificação de íntrons e éxons no contexto deste trabalho. Mesmo após o classificador treinado e em operação é necessário saber quais partes da sequência desconhecida devem ser submetidas ao classificador (os possíveis íntrons ou éxons). Por isso, as sequências de consenso são importantes, pois elas trazem evidências de possíveis inícios e fins de íntrons. Dessa forma, o processamento pelo qual cada uma das sequências do GenBank foi submetido para construir a base de dados é o mesmo processo que as sequências desconhecidas submetidas pelos usuários ao processo de identificação de íntrons e éxons deverá passar antes de submeter as sequências candidatas a íntrons e éxons ao classificador já treinado.

Além disso, após a classificação das sequências candidatas, é preciso "remontar" a sequência original submetida pelo usuário para validar a classificação resultante do modelo treinado e determinar a região de *splicing*. Por isso, para avaliar o desempenho da abordagem como um todo, não basta apenas coletar as medidas obtidas durante o treinamento do modelo para as classificações das sequências candidatas. É necessário também "remontar" as sequências de entrada, com base no resultado do classificador para as sequências candidatas e comparar as regiões de *splicing* da sequência remontada com as sequências do conjunto de teste vindas do Genbank, para as quais já se sabe onde estão as regiões de *splicing*. Os resultados dessas medidas são mostrados na seção 6.

Para construir a base de dados desta abordagem foram recuperadas 7.005 amostras (sequências) do fungo *Colletotrichum*, na região do gene que produz a proteína Actina e 2.259 amostras (sequências) do fungo *Diaporthe* em duas regiões que produzem a proteína Beta Tubulina e Actina, como mostra a Tabela 1.

As sequências recuperadas do Genbank deram origem à base de dados com a quantidade de exemplos descrita na Tabela 2.

Tabela 1: Dados obtidos do Genbank para construção da base de dados

Fungo	Proteína	# amostras	Tamanho (MB)
Colletotrichum	Actina	7.005	16,2
Diaporthe	Beta Tubulina	1.821	6,2
Diaporthe	Actina	155	0,4
Total		9.114	22,8

4.2 Abordagem 2 - Trincas de nucleotídeos

A segunda abordagem utiliza a informação de que, após o processo de *splicing*, cada sequência de três nucleotídeos, chamados códons, são mapeados para um aminoácido (estrutura base das proteínas).

Os códons de nucleotídeos são fundamentais na tradução dos éxons para as proteínas. No entanto, é importante ressaltar que a identificação da região que um determinado nucleotídeo pertence (íntron ou éxon) e a tradução de uma trinca codificante para proteína são processos de natureza diferentes. O conceito de trinca foi utilizado nesta abordagem como forma de modelar o problema, baseado em uma quantidade mínima de nucleotídeos que pudesse ser coerente com algum processo biológico posterior, no caso, a tradução. Mas a trinca utilizada nesta abordagem não possui influência no processo de tradução propriamente dito.

Nesta abordagem, a partir de uma sequência proveniente do GenBank, foram geradas todas as trincas possíveis de cada sequência, inclusive por meio da técnica de janela deslizante, como mostra a Figura 3. Uma base de dados foi construída com informações, na qual cada códon, juntamente com os códons anterior e posterior distintos, são as únicas *features* do exemplo e cada exemplo poderia ser classificado para uma das seguintes classes: III, IIE, IEE, EEE, EEI, EII, para as quais um "I" denota um nucleotídeo que faz parte de um íntron e "E" denota um nucleotídeo que faz parte de um éxon.

Figura 3: Exemplo de extração de trincas por Janela Deslizante.

A Figura 3 mostra a forma como foi feita a extração das trincas (códons) que compuseram os exemplos da base de dados. A primeira trinca comporta os três primeiros nucleotídeos; a segunda trinca inicia-se no segundo nucleotídeo e vai até o quarto; as demais trincas seguem esse mesmo padrão de extração, sempre reutilizando os dois últimos nucleotídeos da trinca anterior e anexando o próximo nucleotídeo da cadeia. Essa estratégia foi adotada para que na etapa de predição pelo modelo treinado das trincas formadas, dos nucleotídeos no geral fizessem parte de três trincas e, consequentemente, tivessem três classificações. Isso faz com que a classificação definida para cada nucleotídeo leve em conta a classificação que mais aparecer ("I" ou "E") nas três classificações obtidas para o nucleotídeo. Os únicos nucleotídeos que não fazem parte de três trincas são: o primeiro e o último, que fazem parte apenas de uma trinca, e portanto possuem apenas uma classificação; e o segundo e o penúltimo, que fazem parte de duas trincas, tendo sua classificação definida por arbitrariedade, observando a sua classificação dentro da trinca que tenha sido montada primeiro.

As sequências oriundas do GenBank que foram utilizadas para a extração das trincas são um subconjunto das sequências utilizadas na primeira abordagem, coletadas em um primeiro momento da pesquisa. Após a primeira coleta, a base de dados da primeira abordagem foi atualizada, enquanto esta, continuou com os dados originais.

5 MATERIAIS E MÉTODOS

Além dos dados provenientes do GenBank, da modelagem do problema como um problema de aprendizado supervisionado e do uso do CRF, as duas abordagens foram implementadas utilizando a linguagem Python e as bibliotecas sklearn, crfsuite, pickle e random.

As bibliotecas sklearn e crfsuite contam com funções, modelos e técnicas de treinamento já implementados, bem como funções de métricas. Por meio delas foi possível treinar e expor os resultados dos modelos *Conditional Random Fields* gerados. A biblioteca pickle foi utilizada para o gerenciamento dos arquivos gerados pelos módulos do algoritmo, isto é, leitura e gravação, e a biblioteca random foi usada para auxiliar em atividades que necessitassem de escolhas aleatórias.

6 RESULTADOS

As características das bases de dados resultantes das duas abordagens são mostradas a seguir em conjunto com os resultados obtidos para cada fungo/modelo. É importante notar, que para cada fungo e região do gene escolhido foi gerado um modelo diferente para obter os resultados descritos nesta seção.

6.1 Abordagem 1 - Sequências de consenso

Para o fungo *Colletotrichum*, utilizando a base de dados com 7.005 amostras de sequências para a região do gene que produz a proteína Actina, foram gerados 47.020 exemplos para treinar e testar o modelo, que produziram os resultados descritos na Tabela 2.

Tabela 2: Medidas obtidas pelo modelo treinado utilizando a base com 7.005 amostras do fungo *Colletotrichum* e gene da Actina.

Rótulo	Quantidade	% exemplos	F1
Intron	13.017	26,55%	0,96
Exon	19.663	40,11%	0,84
Neither	16.340	33,33%	0,75
Total	47.020	100%	

Tabela 3: Medidas obtidas pelo modelo treinado utilizando a base com 1.821 amostras do fungo *Diaporthe* e gene da Beta Tubulina.

Rótulo	Quantidade	% exemplos	F1
Intron	5.990	28,46%	0,84
Exon	8.040	38,20%	0,67
Neither	7.015	33,33%	0,21
Total	21.045	100%	

Para o fungo *Diaporthe*, utilizando a base de dados com 1.821 amostras de sequências para a região do gene que produz a proteína Beta Tubulina, foram gerados 21.045 exemplos para treinar e testar o modelo, que produziram os resultados apresentados na Tabela 3, e, utilizando a base de dados com 155 amostras de sequências para a região do gene que produz a proteína Actina, fora gerados 1.156 exemplos para treinar e testar o modelo, que produziram os resultados apresentados na Tabela 4.

Tabela 4: Medidas obtidas pelo modelo treinado utilizando a base com 155 amostras do fungo *Diaporthe* e gene da Actina.

Rótulo	Quantidade	% exemplos	F1
Intron	308	26,64%	0,79
Exon	463	40,05%	0,68
Neither	385	33,33%	0,27
Total	1.156	100%	

Os exemplos criados para compor as bases de dados de cada fungo/proteína foram divididos aleatoriamente em 90% de treino e 10% de teste. Não foi utilizada validação cruzada nesta abordagem.

É importante notar nos resultados apresentados que a medida F1 é sempre maior para a classe íntron, o que evidencia as sequências de consenso de início e fim de íntron.

Foi ainda criada outra métrica de desempenho, calculando quantos dos exemplos da própria base o modelo seria capaz de achar todos os íntrons, removê-los e remontar a sequência resultante somente com os éxons remanescentes. Os resultados desta métrica são descritos na Tabela 5.

Tabela 5: Acurácia obtida pela comparação das sequências resultantes após a retirada dos íntrons.

Fungo	Proteína	% Acurácia
Colletrotrichum	Actina	82,50%
Diaporthe	Beta Tubulina	27,30%
Diaporthe	Actina	100%

Por último, foi utilizada a ferramenta BLAST[2] do GenBank, que foi capaz de identificar a sequência resultante da remontagem com uma compatibilidade de 95% nas sequências geradas pelo modelos do *Colletotrichum* para a proteína Actina. Para o *Diaporthe*, os resultados obtidos foram inconclusivos, apresentando compatibilidade inferior à 40%, sendo que os resultados da ferramenta eram, erroneamente, outros fungos correspondentes das sequências.

Com base nos resultados obtidos podemos concluir que a abordagem gerou resultados que se mostraram satisfatórios para o fungo *Colletotrichum* no gene da proteína Actina, enquanto que para o fungo *Diaporthe* as medidas não foram tão boas, isso é, para o gene da proteína Beta Tubulina a precisão foi consideravelmente baixa e, para o gene da proteína Actina, a precisão de 100% é um reflexo da baixa quantidade de exemplos disponíveis neste caso (apenas 155 sequências foram obtidas do GenBank). Isso pode ser uma evidência de que o CRF pode não lidar bem com sequências maiores, como são as sequências da Beta Tubulina. Para contornar este problema foi então projetada a segunda abordagem, na qual o tamanho da sequência deixa de ser um fator importante para o modelo, pois os exemplos da base são sempre do mesmo tamanho.

[2]O BLAST é um algoritmo usado na comparação de informações biológicas, como sequências de aminoácidos ou nucleotídeos de sequências de DNA.

6.2 Abordagem 2 - Trincas de nucleotídeos

Antes da divisão das sequências em trincas foram separadas 20% das sequências oriundas do GenBank para o teste da abordagem. Isso foi necessário para poder ter uma correspondências das trincas que vieram de cadas sequência depois. Não foi utilizada validação cruzada nesta abordagem.

A partir das 5.200 sequências da base de dados coletada para o fungo *Colletotrichum*, no gene da proteína Actina, foram geradas 262.104 trincas para treinar e testar o modelo, que produziram os seguintes resultados, mostrados na Tabela 6:

Tabela 6: Métricas do modelo treinado para o fungo *Colletotrichum*, gene da proteína Actina.

Rótulo	Quantidade	% exemplos	F1
EEE	84.859	32,38%	0,79
IEE	2.029	0,77%	0,99
EEI	1.979	0,76%	0,99
IIE	2.029	0,77%	0,99
EII	1.980	0,76%	0,99
III	169.228	64,56%	0,89
Total	262.104	100%	0,86

É importante notar que a medida F1 é maior para as regiões de *splicing*, ou seja, para as trincas que possuem as regiões de transição de íntrons para éxons (*acceptor splice sites*) ou de éxons para íntrons (*donor splice sites*), mesmo com essas classes possuindo uma quantidade menor de trincas. Isso mostra que pode realmente existir um padrão nas sequências de nucleotídeos dessa regiões, como por exemplo, as sequências de consenso. Para o fungo *Diaporthe* no gene da proteína Beta Tubulina, foram obtidas do GenBank, 2.109 sequências, das quais foram geradas 246.442 trincas para treino e teste, que produziram os resultados descritos na Tabela 7:

Tabela 7: Métricas do modelo treinado para o fungo *Diaporthe*, gene da proteína Beta Tubulina.

Rótulo	Quantidade	% exemplos	F1
EEE	112.436	45,63%	0,39
IEE	1.510	0,61%	0,98
EEI	1.233	0,5%	0,96
IIE	1.509	0,61%	0,98
EII	1.233	0,5%	0,97
III	128.521	52,15%	0,73
Total	246.442	100%	0,72

Podemos notar na Tabela 7 que o padrão de maior medida F1 para as sequências de transição se repetem.

Como esta abordagem particionou as sequências oriundas do GenBank em trincas, é necessário, após a classificação das trincas, reconstruir as sequências para identificar os íntrons e éxons. Dessa forma, foi decidida a classificação "I" ou "E" para cada nucleotídeo de cada sequência, de acordo com a estratégia discorrida na Seção 4.2 e, com as sequências formadas por "I"s e "E"s, foi possível definir o início e o fim de cada íntron e cada éxon.

Nas sequências remontadas, formadas por "I"s e "E"s, poderia haver ruídos, ou seja, uma quantidade muito pequena de "I"s cercada por uma grande quantidade de "E"s, ou o caso inverso, situações que poderiam caracterizar uma falha de predição. Para identificar esses possíveis ruídos e amenizá-los, foi criada uma função, denominada ***aliasing***, que buscava sequências na situação acima, de tamanho menor ou igual a 6[3], e as invertia, isto é, o nucleotídeo classificado como "I" se tornava "E" e vice-versa. Dessa forma, caso essas pequenas subsequências realmente fossem ruídos e tivessem sido classificadas de forma errônea, as medidas de precisão tenderiam a melhorar.

Após a remontagem das sequências, as mesmas foram comparadas às classificações correspondentes, presentes na base de dados utilizada para a pesquisa. Para obter a métrica F1 das sequências remontadas, foi utilizado o método *flat classification report*, da biblioteca crfsuite, comparando os íntrons e éxons encontrados na sequência remontada, com os íntrons e éxons que vieram anotados na base de dados para a sequência correspondente. Dessa forma, foi possível obter os seguintes resultados na identificação dos íntrons e éxons de fato, descritos nas Tabelas 8 e 9:

Tabela 8: Métricas das sequências remontadas para fungo *Colletotrichum*.

Rótulo	Quantidade	F1	F1 + *aliasing*
Éxon	90.746	0,87	0,92
Íntron	90.746	0,93	0,96
Total	264.184	0,91	0,94

Tabela 9: Métricas das sequências remontadas para o fungo *Diaporthe*.

Rótulo	Quantidade	F1	F1 + *aliasing*
Éxon	115.708	0,77	0,80
Íntron	131.576	0,79	0,82
Total	247.284	0,78	0,81

7 CONCLUSÃO

Com o objetivo de contribuir para a área de biotecnologia e genética, este artigo descreve duas abordagens baseadas no modelo CRF de aprendizado supervisionado, para a identificação de regiões de *splicing* em sequências de DNA de fungos filamentos.

Na primeira abordagem foram treinados dois modelos do CRF, uma para cada fungo selecionado. Os resultados do modelo treinado para o fungo *Colletotrichum* (Actina) foram satisfatórios, obtendo uma acurácia de 82,5% nas comparações com os resultados gerados pelo modelo em relação às sequências corretas do GenBank. O modelo treinado para o fungo *Diaporthe* (Beta Tubulina), não obteve resultados tão promissores, atingindo uma acurácia, para o mesmo teste aplicado ao *Colletotrichum*, de 27,3%. As diferenças nas acurácias se devem ao fato de que, para cada fungo, a proteína selecionada é de uma região diferente do gene, sendo que

[3]Outros limiares foram testados, mas os melhores resultados foram obtidos com o limiar igual a 6.

as sequências da Beta Tubulina no fungo *Diaporthe* são maiores quando comparadas às sequências da Actina no *Colletotrichum*. As medidas de desempenho geradas pelo algoritmo de treinamento do modelo mostraram sempre valores maiores da medida F1 para a classificação dos íntrons, reforçando a afirmação de Alberts *et al.* [9] sobre as sequências de consenso.

Com a análise dos resultados obtidos podemos notar que o modelo CRF treinado para o fungo *Colletotrichum* (proteína Actina) apresentou bons resultados na identificação das regiões não codantes, gerando assim, sequências codantes com ótima precisão. Por outro lado, o modelo CRF treinado para o fungo *Diaporthe* (proteína Beta Tubulina), de maneira geral, não apresentou bons resultados, mostrando uma possível evidência de que o CRF pode não lidar bem com sequências maiores, como são as sequências da Beta Tubulina. Para contornar o problema do possível tamanho da sequência não ter bons resultados no CRF, foi então projetada a segunda abordagem, baseada em trincas.

Na segunda abordagem, foram treinados dois modelo do CRF, um para cada fungo mencionado, recebendo como entrada partes de sequências de DNA representadas por trincas de nucleotídeos. Essas trincas foram obtidas das sequências por meio da técnica de janela deslizante e levavam consigo a trinca distinta anterior e a trinca distinta posterior para formar um exemplo para entrada do algoritmo.

Utilizando 5.200 sequências para teste, o modelo para o fungo *Colletotrichum* obteve 0,86 para a medida F1, no geral. Para o fungo *Diaporthe*, com 2.109 sequências para teste, o modelo obteve 0,72 para a medida F1, também no geral.

Após a classificação da trincas pelo modelo treinado correspondente, as sequências foram então remontadas com suas respectivas trincas classificadas para a identificação das regiões de *splicing*. Além disso, foi criada uma função chamada *aliasing* com o objetivo de remover possíveis ruídos nas sequências remontadas. Para testar a efetividade dessa função, os resultados para a identificação de íntrons e éxons foram coletados tanto com o auxílio dessa função, quanto sem ela.

Para o fungo *Colletotrichum* foi observado o valor de 0,91 de medida F1 geral sem a utilização da função *aliasing* e 0,94 de F1 utilizando a função. Para o fungo *Diaporthe*, os resultados obtidos foram um pouco inferiores em relação ao outro fungo, de acordo com a medida F1: 0,78 sem a função *aliasing* e 0,81 com sua utilização.

De forma geral, podemos avaliar que a abordagem baseada em trincas obteve melhores resultados do que a abordagem baseada nas sequências de consenso, principalmente para o fungo *Diaporthe* na região da Beta Tubulina, para o qual as sequências são maiores. Como já mencionado, isso pode ser uma evidência de que o CRF não trabalha bem com sequências maiores.

Além disso, os resultados da abordagem baseada em trincas se mostraram bastante satisfatórios quando comparados a outras abordagens baseadas em Redes Neurais para o mesmo problema, o que mostra que o CRF pode ser uma boa opção de modelo para identificação de sequências de íntrons e éxons, pelo menos no contexto dos fungos filamentosos. Todo o código desenvolvido, os módulos, arquivos do GenBank utilizados e arquivos de configuração estão disponíveis em: https://github.com/GustavoHCruz/RNACodingRegions.

8 TRABALHOS FUTUROS

Com base nas análises feitas com especialistas na área de biotecnologia, pretende-se primeiramente, treinar modelos com base em dados de fungos filamentos nas regiões do gene da mesma proteína, possibilitando a comparação mais direta dos resultados obtidos. Além disso, pretende-se treinar um modelo mais geralista, com os dados de mais de um fungo e disponibilizar um ferramenta online, que possam ser útil principalmente para os profissionais da área de biotecnologia (que auxiliaram neste trabalho) além de outros profissionais interessados.

AGRADECIMENTOS

Agradecemos aos acadêmicos e docentes dos cursos de Graduação e Pós-Graduação em Biotecnologia da UEM, em especial o Professor João Alencar Pamphile (*in memoriam*) e ao Professor Júlio Cesar Polônio, pelos esclarecimentos necessários e específicos da área no desenvolvimento deste trabalho. Agradecemos também as contribuições dos professores membros das bancas dos Trabalhos de Conclusão de Curso dos acadêmicos Antônio Roberto dos Santos e Gustavo Luiz Furuhata Ferreira, bem como ao Programa de Iniciação Científica da UEM, no qual foram desenvolvidos os projetos dos acadêmicos Gustavo Henrique Ferreira Cruz, Vinícius Menossi e Sarah Anduca de Oliveira.

REFERÊNCIAS

[1] Thais de Paulo Lehugeur and Hugo Christiano Soares Melo. Bioinformática aplicada no desenvolvimento de novos fármacos. *Psicologia e Saúde em debate*, 4 (Suppl1):55–55, 2018.

[2] Chong Chu, Xin Li, and Yufeng Wu. Splicejumper: a classification-based approach for calling splicing junctions from rna-seq data. *BMC bioinformatics*, 16(17):S10, 2015.

[3] S Makal, L Ozyilmaz, and S Palavaroglu. Neural network based determination of splice junctions by roc analysis. *Training*, 98(100):91–99, 2008.

[4] Aparajita Dutta, Kusum Kumari Singh, and Ashish Anand. Deep learning models for identification of splice junctions across species. *bioRxiv*, 2021.

[5] Jasper Zuallaert, Fréderic Godin, Mijung Kim, Arne Soete, Yvan Saeys, and Wesley De Neve. Splicerover: interpretable convolutional neural networks for improved splice site prediction. *Bioinformatics*, 34(24):4180–4188, 2018.

[6] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[7] James D Watson, Tania A Baker, Stephen P Bell, Alexander Gann, Michael Levine, and Richard Losicke. *Biologia molecular do gene.* Artmed Editora, 7 edition, 2015.

[8] Louise T Chow, James M Roberts, James B Lewis, and Thomas R Broker. A map of cytoplasmic rna transcripts from lytic adenovirus type 2, determined by electron microscopy of rna: Dna hybrids. *Cell*, 11(4):819–836, 1977.

[9] Bruce Alberts, Alexander Johnson, Julian Lewis, David Morgan, Martin Raff, Keith Roberts, Peter Walter, John Wilson, and Tim Hunt. *Biologia molecular da célula.* Artmed Editora, 6 edition, 2017. ISBN 9788582714225.

[10] Mark Ridley. *Evolução.* Artmed Editora, Porto Alegre, Brazil, 3 edition, 2006.

[11] Aditya Prasad. Conditional random fields explained, 2019. URL https://towardsdatascience.com/conditional-random-fields-explained-e5b8256da776.

[12] Alexandre Cassimiro Andreani. Predição estruturada aplicada à detecção de estrutura retórica. Master's thesis, Universidade Estadual de Maringá, 2017.

[13] Aron Culotta, David Kulp, and Andrew McCallum. Gene prediction with conditional random fields. *University of Massachusetts, Amherst, Tech. Rep. UM-CS-2005-028*, 2005.

[14] Gunnar Rätsch, Sören Sonnenburg, Jagan Srinivasan, Hanh Witte, Klaus-R Müller, Ralf-J Sommer, and Bernhard Schölkopf. Improving the caenorhabditis elegans genome annotation using machine learning. *PLoS Computational Biology*, 3(2): e20, 2007.

[15] Somayah Albaradei, Arturo Magana-Mora, Maha Thafar, Mahmut Uludag, Vladimir B. Bajic, Takashi Gojobori, Magbubah Essack, and Boris R. Jankovic. Splice2deep: An ensemble of deep convolutional neural networks for improved splice site prediction in genomic dna. *Gene*, 763:100035, 2020. ISSN 0378-1119. doi: https://doi.org/10.1016/j.gene.2020.100035. URL https://www.sciencedirect.com/science/article/pii/S2590158320300097. Articles initially published in Gene: X 5, 2020.