UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

DANIEL MATOS DE CASTRO

Viés racial ou pequeno tamanho amostral? Investigando o impacto de disparidade racial em dados genômicos na análise de sobrevida em câncer

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Profª. Drª. Mariana Recamonde Mendoza

Porto Alegre
2023

"If a man will begin with certainties, he shall end in doubts; but if he will be content to begin with doubts, he shall end in certainties."

— FRANCIS BACON

AGRADECIMENTOS

Obrigado a todos que proporcionaram que eu chegasse até este momento.

RESUMO

Este trabalho tem como objetivo investigar o impacto do viés racial nos dados ômicos sobre o desempenho de modelos preditivos com algoritmos de aprendizado de máquina (AM). Visamos analisar como o desbalanceamento entre grupos raciais nos conjuntos de dados obtidos em bancos de dados públicos, como o The Cancer Genoma Atlas (TCGA), pode levar a enviesar modelos de seleção de genes causais e de predição de sobrevida em câncer, de forma prejudicial para os grupos minoritários. Para alcançar este objetivo, foram conduzidos dois experimentos. O primeiro envolveu a seleção de genes causais a partir de dados de transcriptoma utilizando o modelo de riscos proporcionais de Cox, enquanto o segundo tratou do treinamento de um modelo de AM para análise de sobrevida, utilizando o algoritmo *Random Survival Forest*. Para ambos os experimentos, as instâncias de cada conjunto de dados obtido do TCGA foram segregadas em três subgrupos: *all* (conjunto completo), *major* (instâncias com a raça mais prevalente no conjunto de dados) e *minor* (instâncias com raça diferente da majoritária). Os nossos resultados indicam que a dominância do grupo majoritário sobre o resultado geral constatada na identificação de genes causais pode estar relacionada ao tamanho dos conjuntos de dados envolvidos nos grupos majoritário e minoritário (isto é, número absoluto de instâncias disponíveis para as análises estatísticas e computacionais) e não necessariamente a diferenças genéticas entre os subgrupos. Além disso, o estudo constatou que o impacto da disparidade racial no desempenho do modelo de análise de sobrevida varia dependendo do conjunto de dados. Avaliamos também a aplicação de uma estratégia de balanceamento de *major* e *minor* através de subamostragem aleatória, o que não se mostrou eficaz para a obtenção de um desempenho preditivo mais equilibrado entre os dois subgrupos. Concluímos que trabalhos futuros se fazem necessários para investigar estratégias mais sofisticadas para balancear conjuntos de dados, bem como para analisar o efeito do desbalanceamento entre grupos raciais com outros tipos de dados ômicos. Por fim, é de suma importância aprofundar o estudo sobre o potencial de viés racial nos dados genômicos, a fim de determinar mais claramente a contribuição que cabe ao limitado tamanho amostral e à disparidade racial nos desempenhos preditivos mais baixos observados para os grupos minoritários em modelos de AM treinados com os dados ômicos.

Palavras-chave: TCGA. viés racial. análise de sobrevida. aprendizado de máquina.

Racial bias or small sample size? Investigating the impact of racial disparity on genomic data in cancer survival analysis

ABSTRACT

This work aims to investigate the impact of racial bias in omics data on the performance of predictive models with machine learning (ML) algorithms. We aim to analyze how the imbalance between racial groups in datasets obtained from public databases, such as The Cancer Genome Atlas (TCGA), can lead to bias in models for the selection of causal genes and prediction of survival in cancer in a harmful way for minority groups. To achieve this objective, we conducted two experiments. The first involved the selection of causal genes from transcriptome data using the Cox proportional hazards model, while the second dealt with training an ML model for survival analysis using the *Random Survival Forest* algorithm. For both experiments, the instances of each dataset obtained from the TCGA were segregated into three subgroups: *all* (complete set), *major* (instances with the most prevalent race in the dataset) and textitminor (instances with a different race than the majority). Our results indicate that the dominance of the majority group over the overall result found in the identification of causal genes may be related to the size of the datasets involved in the majority and minority groups (that is, the absolute number of instances available for statistical and computational analysis) and not necessarily to genetic differences between subgroups. Furthermore, the study found that the impact of racial disparity on the performance of the survival analysis model varies depending on the data set. We also evaluated the application of a *major* and *minor* balancing strategy through random subsampling, which did not prove to be effective in obtaining a more balanced predictive performance between the two subgroups. We conclude that future work is needed to investigate more sophisticated strategies for balancing datasets, as well as to analyze the effect of imbalance between racial groups with other types of omic data. Finally, it is essential to further study the potential for racial bias in genomic data to more clearly determine the contribution that the limited sample size and racial disparity make to the lower predictive performances observed for minority groups in models of ML trained with the omic data.

Keywords: TCGA, racial bias, survival analysis, machine learning.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de máquina
API	Application Programming Interface
BRCA	Breast cancer
C-index	Concordance Index
CAAP	Consortium on Asthma among African-ancestry Populations in the Americas
CART	Classification and Regression Trees
COAD	Colon adenocarcinoma
HNSC	Head and Neck Squamous Cell Carcinoma
KIRP	Kidney renal papillary cell carcinoma
LGG	Low-Grade Glioma
LIHC	Liver hepatocellular carcinoma
LUAD	Lung adenocarcinoma
LUSC	Lung squamous cell carcinoma
mRNA	messenger RNA
PAAD	Pancreatic adenocarcinoma
PPMR	Population Proportion of The Major Race
READ	Rectum adenocarcinoma
REST	Representational State Transfer
RNA	Ribonucleic Acid
RSF	Random Survival Forest
TCGA	The Cancer Genoma Atlas

SUMÁRIO

1 INTRODUÇÃO

Nos últimos anos, os estudos genômicos surgiram como uma ferramenta valiosa na pesquisa do câncer. O câncer, por ser uma doença multifatorial, é um processo complexo e sua progressão envolve diversos processos no organismo do paciente. Assim, a geração de grandes conjuntos de dados por tecnologias de alto rendimento, como os dados ômicos, que mensuram em larga escala os níveis de expressão gênica (transcriptoma), de concentração de proteínas (proteoma) ou de aspectos como acessibilidade da cromatina e metilação do DNA (epigenoma), têm ganhado interesse crescente na comunidade científica (JIANG et al., 2022). A análise simultânea de múltiplos tipos de dados ômicos, abordagem conhecida como multi-ômica, permite obter uma compreensão mais abrangente dos mecanismos biológicos subjacentes ao desenvolvimento e progressão do câncer (RAUFASTE-CAZAVIEILLE; SANTIAGO; DROIT, 2022). Adicionalmente, os dados ômicos (individualmente ou combinados) podem ser empregados para identificar marcadores moleculares associados a subtipos específicos de câncer, o que pode auxiliar no diagnóstico e na escolha de tratamento.

Nesse contexto, existem ferramentas importantes, tais como o *The Cancer Genome Atlas* (TCGA), um banco de dados disponível publicamente que contém informações moleculares abrangentes de mais de 11.000 amostras de 33 diferentes tipos de câncer (LIU et al., 2018). A disponibilidade deste conjunto de dados extenso e diverso possibilita análises integrativas em larga escala, as quais podem fornecer *insights* acerca das interações entre diversas características genéticas e a manifestação do câncer. Ademais, os dados do TCGA foram amplamente empregados no desenvolvimento e validação de biomarcadores para diagnóstico, prognóstico e seleção de tratamento. Diversos trabalhos utilizaram métodos de aprendizado de máquina para analisar esses dados e construir modelos preditivos (KOUROU et al., 2015; NICORA et al., 2020)

No entanto, a análise de dados genéticos apresenta desafios significativos, especialmente relacionados ao viés racial e à falta de diversidade amostral. Um estudo realizado por Spratt et al. (2016) analisou dados de 10 tipos de câncer presentes no TCGA e constatou que, nos conjuntos analisados, a população branca estava super-representada em relação à população total dos Estados Unidos (de onde se origina a maior parte das amostras), enquanto indivíduos asiáticos e hispânicos estavam sub-representados. Esses achados revelam a existência de viés racial nos dados analisados, que não refletem com precisão as proporções populacionais encontradas na população em geral.

A não-representatividade dos conjuntos de dados pode levar a conclusões imprecisas e até mesmo perigosas. Por exemplo, se um *dataset* de dados genéticos utilizado para o estudo de uma doença específica é composto principalmente por indivíduos de ascendência europeia, os resultados podem não ser generalizáveis para outras populações, como indivíduos de ascendência africana ou latina. Um estudo conduzido por Kessler et al. (2016) analisou 642 sequências de genoma inteiro do projeto *Consortium on Asthma among African-ancestry Populations in the Americas* (CAAPA), que reúne dados de populações de ascendência africana. O estudo encontrou correlações significativas entre as proporções estimadas de ascendência africana e o número de variantes genéticas por indivíduo em todos os conjuntos de classificação de variantes, exceto um.

Outro estudo, conduzido por Zhang et al. (2019) investigou os perfis genômicos mutacionais em pacientes chinesas com câncer de mama e os comparou com pacientes caucasianas do The Cancer Genome Atlas (TCGA). Os pesquisadores realizaram sequenciamento direcionado em 33 genes relacionados ao câncer de mama em 304 pacientes chinesas com câncer de mama e descobriram que a idade média no momento do diagnóstico na coorte chinesa foi significativamente mais jovem do que na coorte TCGA, além de diferenças genômicas significativas entre pacientes asiáticas e caucasianas, como uma prevalência maior de certas mutações em pacientes chinesas. Estas evidências mostram que o resultado de análises genéticas de pacientes com câncer pode estar correlacionado com características raciais destes indivíduos e o conhecimento adquirido por análises deste tipo em um grupo racial não necessariamente se transfere de forma direta para outros grupos.

O crescimento do uso de algoritmos de aprendizado de máquina, que objetivam aprender a partir de exemplos fornecidos em um conjunto de dados e identificar padrões e relações entre as variáveis, torna ainda mais relevante o problema do viés racial em dados ômicos. Um estudo recente de Dai et al. (2022) demonstrou que o uso de conjuntos de dados do TCGA com grande representação de uma raça específica para treinar modelos de aprendizado de máquina pode resultar no desenvolvimento de modelos com desempenho inferior em grupos raciais minoritários. Assim, é crucial que a questão do viés racial nos dados genômicos, no contexto de aprendizado de máquina, seja investigada.

O objetivo principal deste trabalho é investigar o impacto do desequilíbrio entre grupos raciais em algoritmos de aprendizado de máquina para a identificação de genes causais e a análise de sobrevida. Buscamos compreender como o viés racial pode afetar o desempenho de algoritmos desenvolvidos para essas tarefas e determinar até que ponto

um modelo de análise de sobrevida treinado em um grupo racial pode ser generalizado para outros grupos. Desta forma, esperamos contribuir para o avanço do conhecimento sobre o impacto do viés racial nessas aplicações de aprendizado de máquina.

A estrutura do trabalho é organizada da seguinte forma: no Capítulo 2 são discutidos os principais conceitos teóricos sobre análise de sobrevida e aprendizado de máquina que serão utilizados no presente estudo, fornecendo um material de referência para os leitores que não estão familiarizados com esses conceitos. No Capítulo 3 são revisados alguns trabalhos que abordaram assuntos relacionados ao viés racial ou análise de sobrevida em dados ômicos. No Capítulo 4, é apresentado o problema a ser investigado neste estudo e são descritos os experimentos realizados e a metodologia empregada. Os resultados obtidos nesses experimentos são apresentados no Capítulo 5. Por fim, no Capítulo 6, são discutidos os resultados alcançados e apontados os trabalhos futuros relacionados a este estudo.

2 REFERENCIAL TEÓRICO

Este capítulo introduz os principais conceitos de análise de sobrevida e aprendizado de máquina necessários para o desenvolvimento do trabalho.

2.1 Análise de sobrevida

De acordo com Wang, Li and Reddy (2019), "a análise de sobrevida (*survival analysis*) é um subcampo da estatística cujo objetivo é analisar e modelar dados onde o resultado é o tempo até que um evento de interesse ocorra". As técnicas de análise de sobrevida são aplicáveis nas mais diversas áreas, como saúde, engenharia e finanças. A definição de evento depende diretamente da aplicação. Como exemplos de eventos, podemos citar morte ou remissão (na área da saúde), falha técnica (engenharia), ocorrência de acidente (seguros), etc. No escopo deste trabalho, o evento de interesse é o óbito dos indivíduos participantes. Neste contexto, os métodos de análise de sobrevida buscam modelar uma variável aleatória "tempo até o evento"(*time-to-event*), que é denotada por T.

2.1.1 *Censoring*

A principal característica dos dados comumente utilizados em análise de sobrevida é a presença de observações incompletas ou instâncias nas quais o evento de interesse não ocorreu dentro do período de observação. Nestes casos, o tempo anotado para cada instância é denominado tempo observado. Este tipo de informação incompleta é chamado de *censoring*.

Em geral, existem três tipos de *censoring*:

- *Right-censoring*: o tempo até o evento é maior ou igual ao tempo observado (anotado).
- *Left-censoring*: o tempo até o evento é menor ou igual ao tempo observado.
- *Interval-censoring*: o tempo até o evento está em um intervalo conhecido.

Em aplicações na área da saúde, o *right-censoring* é o mais frequente. No restante deste trabalho, o termo *censoring* sempre se refere a *right-censoring*.

Existem várias origens possíveis para o *censoring*. No contexto de estudos médi-

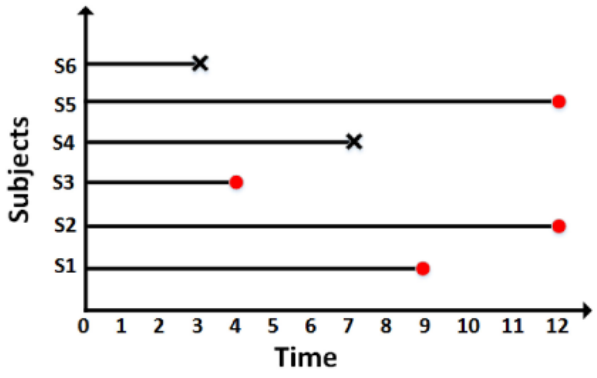
cos, Kleinbaum and Klein (2012) elencam três causas comuns:

- Um sujeito pode não experimentar o evento antes que o estudo termine.
- Perde-se o contato com o sujeito durante o acontecimento do estudo.
- Um sujeito se retira do estudo por causa de morte, quando morte não é o evento de interesse, ou qualquer outra razão, como reação adversa ao medicamento que está sendo estudado.

Utilizamos C para denotar o tempo anotado para uma instância que apresenta *censoring*, ou seja, que não experienciou o evento de interesse.

A Figura 2.1, elaborada por Wang, Li and Reddy (2019), ilustra a presença de *censoring* em um estudo com seis participantes. Neste cenário fictício, os indivíduos $S4$ e $S6$ experienciaram o evento de interesse nos tempos 7 e 3, respectivamente, enquanto perdeu-se contato com os participantes $S1$ e $S3$ nos tempos 9 e 4, respectivamente. Os indivíduos $S2$ e $S5$ completaram o estudo sem experienciar o evento. Com base nesse cenário, dizemos que $S1$, $S2$, $S3$ e $S5$ apresentam *censoring* e que seus tempos foram registrados como 9, 12, 4 e 12, respectivamente. Já $S4$ e $S6$ não apresentam *censoring*, tendo seus tempos registrados como 7 e 3, respectivamente.

Figura 2.1: Exemplo de dados com censoring



Fonte: Wang, Li and Reddy (2019)

Uma abordagem ingênua para o problema de análise de sobrevida seria remover as instâncias que apresentam *censoring* e tentar modelar o tempo até o evento como uma tarefa de regressão. Porém, as instâncias com *censoring* não podem ser simplesmente descartadas porque contêm informações significativas sobre o problema que podem influenciar a estimativa da função de sobrevida e afetar as conclusões obtidas a partir da análise. Descartar instâncias com *censoring* pode levar a resultados tendenciosos e imprecisos e

reduzir o poder e a eficiência da análise. Ainda, em estudos médicos bem-sucedidos, é esperado que eventos como óbito ou evolução da doença ocorram apenas em uma pequena porcentagem dos sujeitos, o que tornaria inviável utilizar métodos de aprendizado de máquina em um número tão reduzido de instâncias. De fato, Herrmann et al. (2021) reporta que é comum encontrar conjuntos de dados em que menos de 20% das instâncias passam pelo evento de interesse. Dessa forma, faz-se necessária a utilização de métodos apropriados para lidar com o *censoring*.

2.1.2 Estrutura típica de uma instância em análise de sobrevida

Devido a necessidade de distinguir instâncias com *censoring* das demais, geralmente utiliza-se um atributo binário para tal. Wang, Li and Reddy (2019) define uma instância i como uma tripla (X_i, y_i, δ_i) onde:

- X_i é um vetor de atributos (*features*, ou covariáveis na terminologia de análise de sobrevida).
- y_i é o tempo observado.
- δ_i é um atributo binário que indica se a instância apresenta *censoring* (δ_i = 0) ou não (δ_i = 1).

Os atributos δ_i e y_i obedecem a seguinte relação:

$$y_i = \begin{cases} T_i, & \text{se } \delta_i = 1 \\ C_i, & \text{se } \delta_i = 0 \end{cases}$$

Com base nessa estrutura, Wang, Li and Reddy (2019) estabelece que "O objetivo da análise de sobrevida é estimar o tempo até o evento de interesse T_j para uma nova instância j com atributos denotados por X_j."

2.1.3 Função de risco e função de sobrevida

Existem diversas funções que são utilizadas por métodos estatísticos ou de aprendizado de máquina para modelar o tempo até evento. A função de sobrevida, denotada por $S(t)$, é definida como a probabilidade do evento ocorrer após o tempo t. Formalmente,

$$S(t) = P(T > t)$$

Já a função de risco, que denotamos como $h(t)$, mede o risco do evento ocorrer em um tempo t, dado que não ocorreu antes. Formalmente,

$$h(t) = \lim_{\delta t \to 0} \frac{P(t < T < t + \delta t \mid T \geq t)}{\delta t}$$

Alguns métodos de análise de sobrevida estimam a função de risco, enquanto outros modelam a função de sobrevida. Kleinbaum and Klein (2012) define a seguinte relação, que nos permite obter qualquer uma dessas funções a partir da outra:

$$h(t) = -\left[\frac{dS(t)/dt}{S(t)}\right]$$

2.1.4 Estimador de Kaplan-Meier

O método de Kaplan-Meier é um estimador não-paramétrico para a função de sobrevida (KAPLAN; MEIER, 1958), que leva em consideração a presença de *censoring*. Seu cálculo consiste em estimar, para cada ponto no tempo T_j presente no conjunto de dados, a probabilidade de um indivíduo sobreviver além de T_j considerando a razão entre o número de indivíduos que sobreviveram até aquele momento e número de indivíduos em risco naquele momento. Wang, Li and Reddy (2019) apresenta a seguinte relação para o cálculo do Kaplan-Meier:

$$\hat{S}(t) = \prod_{j:T_j<t} p(T_j)$$

onde:

$$p(T_j) = \frac{r_j - d_j}{r_j}$$

sendo r_j o número de instâncias consideradas em risco no tempo T_j, ou seja, que possuem tempo observado maior ou igual a T_j, e d_j o número de instâncias com evento observado no tempo T_j. A cada tempo T_j, o número de indivíduos ainda em risco r_j é calculado recursivamente como $r_j = r_{j-1} - d_{j-1} - c_{j-1}$, sendo c_{j-1} o número de instâncias que apresentam *censoring* entre T_{j-1} e T_j.

Uma limitação deste estimador é não levar em conta atributos de interesse, como idade, sexo, estágio de doença, etc. As únicas informações consideradas são os tempos registrados e a ocorrência de *censoring*.

2.1.5 Modelo de Cox de riscos proporcionais

O modelo de Cox de riscos proporcionais é um método estatístico semi-paramétrico que visa estimar a função de risco através de uma regressão (COX, 1972). Trata-se do método de regressão mais utilizado para problemas de análise de sobrevida (WANG; LI; REDDY, 2019). Como grande vantagem em relação ao Kaplan-Meier, o algoritmo de Cox leva em consideração todas as covariáveis (*features*) conhecidas sobre as instâncias e permite avaliar a contribuição de cada atributo para os resultados observados.

O modelo assume que a função de risco é proporcional às covariáveis e estima os coeficientes de regressão usando um método de verossimilhança parcial (WANG; LI; REDDY, 2019). Pode-se utilizar o algoritmo com atributos contínuos e categóricos (codificados com alguma estratégia como *one-hot encoding*), e não requer nenhuma suposição sobre a distribuição subjacente da variável de tempo até o evento. Seu resultado consiste em um coeficiente para cada *feature* bem como um p-valor, que mede a significância desse coeficiente. Através desse p-valor podemos avaliar o impacto desse atributo na função de risco que está sendo modelada (por exemplo, definindo um limiar de significância $p < 0,01$).

Existem algumas limitações práticas no uso do modelo de Cox. O algoritmo assume que a razão entre o risco de dois indivíduos permanece constante ao longo do tempo e que a ocorrência de *censoring* é não-informativa, o que pode não ser verdadeiro em alguns casos. Além disso, sabe-se que o modelo de Cox não escala de maneira adequada quando aplicado em *datasets* que apresentam alta dimensionalidade (SPOONER et al., 2020), característica comum em dados ômicos. Dessa forma, faz-se necessária a utilização de abordagens mais sofisticadas para estes cenários.

2.2 Aprendizado de máquina

Aprendizado de máquina (AM) é um subcampo da inteligência artificial que se concentra no desenvolvimento de algoritmos e modelos para a realização de tarefas pre-

ditivas e descritivas. Tarefas descritivas são caracterizadas como aquelas em que é necessário extrair padrões de um conjunto de dados sem que haja algum tipo de supervisor externo, enquanto tarefas preditivas objetiva a indução de um modelo através de treinamento em dados rotulados, visando a predição do rótulo de uma nova instância não-rotulada (FACELI et al., 2011). Dizemos que os algoritmos aplicáveis em tarefas preditivas realizam aprendizado supervisionado, enquanto algoritmos aplicáveis a tarefas descritivas realizam aprendizado não-supervisionado. Exemplos de tarefas em aprendizado supervisionado incluem classificação e regressão. A principal diferença entre estas tarefas está na natureza da saída. Na regressão, a saída é um valor numérico contínuo, enquanto na classificação, a saída é um rótulo categórico.

Com o tempo, foram desenvolvidos métodos de análise de sobrevida baseados em aprendizado de máquina. Estes geralmente surgem como adaptações de algoritmos tradicionais de aprendizado supervisionado para considerar o *censoring*, e possuem vantagens em relação aos métodos estatísticos como Kaplan-Meier e Cox, pois podem lidar com conjuntos de dados de alta dimensionalidade, além de conseguirem modelar dependências entre os atributos (WANG; LI; REDDY, 2019). Nas próximas seções revisaremos o algoritmo de aprendizado aplicado no presente trabalho e conceitos relevantes

2.2.1 *Survival trees*

Árvores de sobrevida, ou *survival trees*, são uma adaptação de árvores de regressão tradicionais, para dados que apresentam *censoring* (WANG; LI; REDDY, 2019). Neste trabalho seguimos a abordagem proposta por (LEBLANC; CROWLEY, 1993), que propõe modificações ao algoritmo CART proposto por Breiman (2017) para o contexto de análise de sobrevida.

Segundo essa abordagem, a escolha do atributo de divisão dos nós é feita através de um teste log-rank, que mede a significância da diferença entre as funções de sobrevida estimadas para dois grupos. Para cada atributo x_i gera-se duas partições p_{i1} e p_{i2}, no estilo CART, e calcula-se estimativas da função de sobrevida $\hat{S}_{i1}(t)$ e $\hat{S}_{i2}(t)$ para cada partição. Executando o teste log-rank para as funções estimadas, temos um p-valor que mede a significância da diferença entre elas. Esta significância é interpretada como uma medida de pureza das partições: quanto mais significativa é a diferença, mais puras são as partições. Assim, selecionamos para o nodo em questão o atributo cujo teste log-rank acusou maior significância.

Outra distinção das *survival trees* para árvores de regressão tradicionais é o método de sumarização das instâncias dos nós folha. Enquanto a média é comumente utilizada em árvores de regressão, nas árvores de sobrevida utiliza-se o Kaplan-Meier para gerar uma estimativa da função de sobrevida a partir das instâncias da folha, a qual é reportada como o valor do nó terminal.

2.2.2 Aprendizado *ensemble* e *random survival forests*

Aprendizado *ensemble* é uma técnica frequentemente empregada para se obter melhor desempenho em algoritmos de máquina, e consiste no uso combinado de múltiplos modelos. Espera-se que através da combinação de modelos diversos entre si obtenha-se um desempenho agregado superior ao dos modelos individuais isoladamente (POLIKAR, 2006). Para que isso ocorra, é fundamental que os modelos combinados sejam diversos, isto é, modelem diferentes funções sobre os dados.

Ensembles são particularmente úteis para reduzir a variância encontrada em alguns algoritmos, como os baseados em árvores. *Random Survival Forest* (RSF) consiste em uma técnica de aprendizado ensemble onde um conjunto de *survival trees* é utilizado, ao invés de considerar apenas uma árvore. Este método foi proposto originalmente por Ishwaran et al. (2008) como uma adaptação do algoritmo *Random Forest* de Breiman (2001) para o contexto de análise de sobrevida.

O algoritmo RSF constrói uma floresta de B árvores através da geração de B amostragens com repetição dos dados de treinamento. Cada uma dessas amostragens, também chamadas de *bootstraps*, será utilizado para treinar uma árvore. Durante o treinamento da árvore, considera-se apenas um subconjunto aleatório de atributos para a criação de cada nó interno. Tanto os *bootstraps* como os subconjuntos de atributos são fontes de diversidade, para que os componentes do *ensemble* sejam distintos entre si. Para agregar o resultado de uma predição retornado pelos componentes da RSF utiliza-se a média.

2.2.3 Validação cruzada k-fold

A separação entre dados de treinamento e teste é uma prática fundamental na construção de modelos de AM. A separação permite avaliar o desempenho do modelo em dados não vistos durante o treinamento e, assim, obter uma estimativa mais confiá-

vel do desempenho em novos dados, visto que a avaliação de um modelo nos mesmos dados utilizados em seu treinamento com frequência leva a estimativas otimistas em relação ao desempenho real em dados não vistos anteriormente (FACELI et al., 2011). Uma abordagem ingênua de separação é a divisão *holdout*, na qual os dados são divididos em conjuntos de treinamento e teste uma única vez, seguindo uma proporção pré-definida. No entanto, a divisão *holdout* tem limitações, como a sensibilidade à escolha aleatória dos conjuntos de treinamento e teste e à qualidade da representatividade dos dados selecionados (FACELI et al., 2011). Além disso, por reter uma porção do *dataset* para teste, nem todos os dados acabam sendo utilizados para treino.

A técnica de validação cruzada k-fold (*k-fold cross-validation*) se propõe como uma alternativa que supera as limitações da divisão *holdout*. A validação cruzada é uma técnica estatística que permite avaliar o desempenho do modelo de forma mais precisa e robusta. Nessa técnica, os dados são divididos em k partes iguais, chamadas de *folds*, e o modelo é treinado k vezes, cada vez usando um *fold* diferente como conjunto de teste e os demais como conjunto de treinamento. O desempenho do modelo é então calculado como a média dos desempenhos obtidos em cada iteração. Dessa forma, garantimos que cada instância é utilizada pelo menos uma vez para treino e teste, o que aumenta a confiabilidade das estimativas de desempenho.

Com a validação cruzada tradicional obtemos K estimativas de desempenho. A validação cruzada com repetição é uma técnica que consiste em realizar r repetições da validação cruzada, embaralhando as instâncias a cada repetição para obter diferentes *folds*. Assim, obtemos $r \times k$ estimativas de desempenho, o que torna o resultado agregado uma estimativa mais confiável (REFAEILZADEH et al., 2009). Além disso, o embaralhamento de instâncias entre repetições diminui a chance um particionamento em *folds* enviesado, isto é, que gere um *fold* com muitas instâncias particularmente fáceis ou difíceis.

Outra prática comum é a estratificação dos *folds* para garantir que a distribuição das classes ou rótulos nos conjuntos de treinamento e validação seja representativa do conjunto de dados completo. Ao utilizar a validação cruzada estratificada, o processo de divisão dos *folds* assegura que cada partição contenha uma proporção semelhante de classes em relação ao *dataset* completo(REFAEILZADEH et al., 2009). Esse procedimento é especialmente importante quando lidamos com conjuntos de dados desbalanceados. Quando a estratificação não é utilizada, pode haver a formação de partições que contenham um número desproporcional de amostras de uma classe, o que pode resultar

em estimativas de desempenho do modelo enviesadas.

2.2.4 Otimização de hiperparâmetros com *Grid search*

Algoritmos de AM frequentemente possuem hiperparâmetros, ou seja, parâmetros que não são aprendidos a partir dos dados mas sim definidos antes do processo de treinamento, como a taxa de aprendizado de uma rede neural ou o número de árvores em uma floresta aleatória. Tais parâmetros influenciam diretamente no desempenho do modelo, e geralmente são difíceis de serem determinados *a priori*. Assim, existem técnicas de otimização de hiperparâmetros que tem como objetivo encontrar a combinação ideal de hiperparâmetros que resulte no melhor desempenho do modelo em dados não vistos.

Um dos métodos mais comuns de otimização de hiperparâmetros é o *Grid Search*. Este algoritmo consiste em uma abordagem de força bruta em que se define um espaço de valores de hiperparâmetros a serem avaliados e, em seguida, cada combinação de valores no espaço definido é utilizada para o treinamento e avaliação de um modelo (por exemplo, com validação cruzada). A combinação que produz o melhor desempenho é selecionada como o conjunto ótimo de hiperparâmetros.

2.2.5 *Concordance index* (c-index)

Para avaliar o desempenho de um modelo de análise de sobrevida precisamos utilizar métricas que levem em consideração a presença de *censoring*. Uma escolha comum é o *concordance index* (c-index), que mede a proporção de pares de instâncias no conjunto de teste que tiveram seus tempos até o evento corretamente ordenados. Formalmente, para duas instâncias i e j com tempos observados y_i e y_j e predições $\hat{y_i}$ e $\hat{y_j}$ respectivamente, Wang, Li and Reddy (2019) define o c-index (denotado por c) como:

$$c = P(\hat{y_i} > \hat{y_j} | y_i > y_j)$$

Sendo uma probabilidade, temos que $c \in [0, 1]$. Dizemos que o modelo possui desempenho ideal se $c = 1$ e aleatório se $c = 0,5$.

2.2.6 Coeficiente de Jaccard

O coeficiente de Jaccard, também conhecido como *intersection over union*, é uma medida de similaridade entre dois conjuntos de dados e é definido como o tamanho da interseção dos dois conjuntos dividido pelo tamanho da união dos dois conjuntos. Matematicamente, o coeficiente de Jaccard de dois conjuntos A e B pode ser representado como:

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

O coeficiente de Jaccard varia entre 0 e 1, sendo 0 indicando nenhuma similaridade entre os conjuntos e 1 indicando similaridade perfeita.

3 TRABALHOS RELACIONADOS

Nesta seção vamos descrever trabalhos que abordaram os temas de viés racial e/ou análise de sobrevida em dados ômicos.

3.1 Viés racial em dados ômicos

O estudo realizado por Dai et al. (2022) teve como objetivo analisar o impacto do viés racial nas tarefas de descoberta de genes causais e predição de sobrevida, em que um resultado binário indicando sobrevida ou morte era o objetivo da predição. Foram utilizados dados de expressão de RNA mensageiro (mRNA, de *messenger RNA*) obtidos do TCGA, juntamente com atributos clínicos, como idade, sexo, raça e estágio tumoral. Todas as análises foram conduzidas considerando três subgrupos raciais: *major* (conjunto de instâncias da raça com mais representantes no *dataset* em questão), *minor* (conjunto de instâncias que apresentam raça diferente da majoritária) e *all* (conjunto completo, sem agrupamento por informação racial, composto pela união de *major* e *minor*).

O pipeline utilizado pelos autores consistiu em três etapas. Na etapa de pré-processamento, cada *dataset* foi classificado quanto ao nível de viés utilizando a métrica *Racial Bias Index*. Essa métrica é definida como a proporção de instâncias que pertencem à raça majoritária, sendo denotada pela sigla *PPMR* (*population proportion of the major race*). O viés foi considerado forte se $PPMR \geq 0,75$, moderado se $0,75 > PPMR \geq 0,5$ e fraco se $PPMR < 0,5$. Além dessa classificação, foram descartados os *datasets* que não atendiam a alguns critérios, como apresentar pelo menos 10 instâncias no grupo minoritário. Embora todos os conjuntos de dados tenham sido considerados inicialmente, os autores chegaram a um subconjunto de 6 tipos de câncer considerados representativos após esta etapa de pré-processamento. A Tabela 3.1 apresenta a relação de conjuntos de dados utilizados e seu nível de viés, conforme análise no trabalho original.

Tabela 3.1: Datasets utilizados por Dai et al. (2022)

Abreviação	Nome do câncer	Nível de viés
HNSC	Head and neck squamous cell carcinoma	Alto
PAAD	Pancreatic adenocarcinoma	Alto
LUAD	Lung adenocarcinoma	Moderado
LUSC	Lung squamous cell carcinoma	Moderado
LIHC	Liver hepatocellular carcinoma	Fraco
READ	Rectum adenocarcinoma	Fraco

Fonte: Adaptado de Dai et al. (2022)

Na segunda etapa, foi utilizado o modelo de Cox de riscos proporcionais para identificar os genes mais significativos para cada subgrupo em cada conjunto de dados. Isso foi realizado ao executar o modelo de Cox uma vez para cada gene no *dataset*, utilizando juntamente as informações de sexo, raça, idade e estágio tumoral, e obtendo o p-valor resultante, que foi utilizado como medida de significância daquele gene. Esta análise foi realizada para os subconjuntos *all*, *major* e *minor* separadamente. Os dez genes mais significativos foram selecionados para cada subgrupo.

Para cada subgrupo de cada *dataset*, foram treinados quatro modelos de aprendizado de máquina: florestas aleatórias, máquinas de vetor de suporte, K-vizinhos mais próximos e redes neurais profundas. Foi treinado um modelo de cada tipo em cada subgrupo (*all*, *major* e *minor*), e cada modelo treinado foi avaliado em todos os subgrupos. Os dados de entrada para os modelos foram os genes selecionados na etapa anterior, sendo que cada subgrupo pode utilizar um conjunto potencialmente diferente de genes. O treinamento foi realizado utilizando validação cruzada com cinco *folds*.

A partir da seleção de genes, foi observado que nos conjuntos de dados que apresentavam viés forte ou moderado, os conjuntos de genes mais significativos para o subgrupo *all* possuíam intersecção com os genes mais significativos para o subgrupo *major*, enquanto *all* e *minor* não compartilhavam genes em comum. Segundo os autores, isso indica que "conclusões derivadas da população geral podem ser amplamente aplicáveis à raça majoritária, mas não à raça minoritária"(DAI et al., 2022).

Em relação à tarefa de predição de sobrevida, os autores observaram que os modelos treinados nos conjuntos *all* e *major* apresentaram boa acurácia nesses subgrupos, mas resultados piores no subgrupo *minor*. Além disso, os modelos treinados com o subgrupo *all* de *datasets* com viés fraco ($PPMR < 0.5$) apresentaram bom desempenho

tanto para *major* quanto para *minor*. A partir disso, os autores concluíram que, quando $PPMR > 0.5$, as características dos indivíduos da raça majoritária dominam os modelos preditivos, que acabam se adequando mais a esse subgrupo.

3.2 Análise de sobrevida em dados multi-ômicos utilizando aprendizado de máquina

O artigo de Herrmann et al. (2021) apresenta um grande *benchmark* de algoritmos de análise de sobrevida aplicados a dados multi-ômicos. Os autores compararam 10 métodos de aprendizado de máquina divididos em três categorias, a saber, métodos baseados em *boosting*, métodos baseados em árvores e algoritmos de regressão penalizada. Estes algoritmos foram comparados em relação a dois modelos estatísticos clássicos, o Kaplan-Meier e o modelo de Cox de riscos proporcionais, sendo que o último foi treinado utilizando apenas dados clínicos. A Tabela 3.2 apresenta a relação de algoritmos utilizados, juntamente com a estratégia de otimização de hiperparâmetros empregada para cada algoritmo.

Tabela 3.2: Algoritmos utilizados por Herrmann et al. (2021)

Algoritmo	Estratégia de otimização de hiperparâmetros
Standard Lasso	validação cruzada com 10 *folds*
TS IPF-Lasso	validação cruzada com 10 *folds*
Priority Lasso	validação cruzada com 10 *folds*
GRidge	validação cruzada com 10 *folds*
Priority Lasso	validação cruzada com 10 *folds*
SGL	validação cruzada com 10 *folds*
Model-based boosting (glmboost)	validação cruzada com 10 *folds*
Likelihood-based boosting (CoxBoost)	validação cruzada com 10 *folds*
Random forest	OOB
Block forest	OOB
Cox	-
Kaplan-Meier	-

Fonte: Adaptado de Herrmann et al. (2021)

Os dados utilizados foram obtidos do TCGA, e contemplam 18 tipos de câncer. Cada conjunto de dados apresenta atributos clínicos, *copy nymber variation*, *mirna*, mutação e RNA mensageiro. De forma geral, o modelo de Cox obteve um desempenho superior aos algoritmos de aprendizado de máquina. O algoritmo *block forest* conseguiu um desempenho numericamente superior ao Cox, mas a diferença não foi estatisticamente significativa.

4 METODOLOGIA

Como demonstrado por Dai et al. (2022), modelos de aprendizado de máquina podem apresentar diferenças de desempenho (por exemplo, diferenças de acurácia) a favor de determinados subgrupos raciais. Os autores chegaram a esta conclusão ao gerar três subconjuntos de dados, denominados *all*, *major* e *minor*, que agrupam respectivamente todas as instâncias, somente as instâncias da raça majoritária (mais comum no *dataset*) e todas as instâncias que não pertencem à raça majoritária. Foi observado que modelos preditivos treinados no conjunto *all* apresentavam maior desempenho quando testados em *major* do que em *minor*. Contudo, os autores não investigaram se isso ocorre por diferenças genéticas entre os subgrupos ou meramente por efeito do número de participantes da raça majoritária ser, em geral, consideravelmente maior. Assim, os experimentos conduzidos neste trabalho visam avaliar o impacto da diferença de tamanho entre os conjuntos *major* e *minor* conforme definidos por Dai et al. (2022) nas tarefas de identificação de genes causais e análise de sobrevida, tentando melhor compreender a contribuição do tamanho amostral e do viés racial na geração de resultados com vieses.

4.1 Planejamento dos experimentos

Inicialmente, iremos reproduzir o experimento de identificação de genes causais realizado por Dai et al. (2022) e ampliar a análise para incluir dois novos cenários. No primeiro cenário, chamado de caso aleatório, vamos formar conjuntos *major* e *minor* com o mesmo tamanho dos originais, mas com instâncias selecionadas aleatoriamente a partir do conjunto *all*. No segundo cenário, chamado de caso balanceado, vamos equilibrar o tamanho dos conjuntos *major* e *minor*, mantendo o tamanho de *major* igual ao de *minor* por meio de subamostragens aleatórias do conjunto *major* total. A partir desses resultados, pretendemos determinar se a grande intersecção de genes relevantes encontrados entre os conjuntos *all* e *major* no estudo de Dai et al. (2022) se deve ao tamanho dos conjuntos envolvidos. Caso o tamanho dos subconjuntos seja a causa dessa semelhança, esperamos que essa semelhança seja muito atenuada ou desapareça completamente no caso balanceado e se mantenha no caso aleatório. Por outro lado, caso existam diferenças genéticas significativas entre *major* e *minor* que impactem no resultado, esperamos observar uma diminuição nas intersecções entre *all* e *major* no caso aleatório (visto que estaremos misturando instâncias de todas as raças tanto em *major* como *minor*) e resultados análogos

ao caso não-balanceado no cenário balanceado (visto que estaremos utilizando partições que separam os indivíduos da raça majoritária dos demais em ambos os cenários).

Em seguida, realizaremos o treinamento de uma *random survival forest* segmentando os *datasets* em *all*, *major* e *minor*. Esse experimento será conduzido em dois cenários: o primeiro com conjuntos *major* e *minor* de tamanhos balanceados e o segundo sem modificações, mantendo o desbalanceamento original (cenário não-balanceado). Para cada cenário, realizaremos o treinamento em cada subconjunto e avaliaremos cada modelo treinado em todos os subconjuntos. O objetivo desse experimento é avaliar se o desempenho do algoritmo de AM é afetado pelo desbalanceamento dos subconjuntos *major* e *minor* e, caso ocorra uma diferença de desempenho entre esses subconjuntos, entender se uma simples estratégia de subamostragem aleatória é suficiente para atenuar essa discrepância. Além disso, buscamos compreender até que ponto um modelo treinado em um dos subgrupos pode ser aplicado em outro com desempenho satisfatório. Por exemplo, avaliaremos a possibilidade de treinar um modelo em *major* e aplicá-lo em *minor*. Esta análise se faz relevante pois, se houverem diferenças significativas entre os subgrupos espera-se que modelos treinados em um subgrupo não apresentem desempenho satisfatório nos demais subgrupos. As seções a seguir irão detalhar as etapas envolvidas na metodologia dos experimentos planejados.

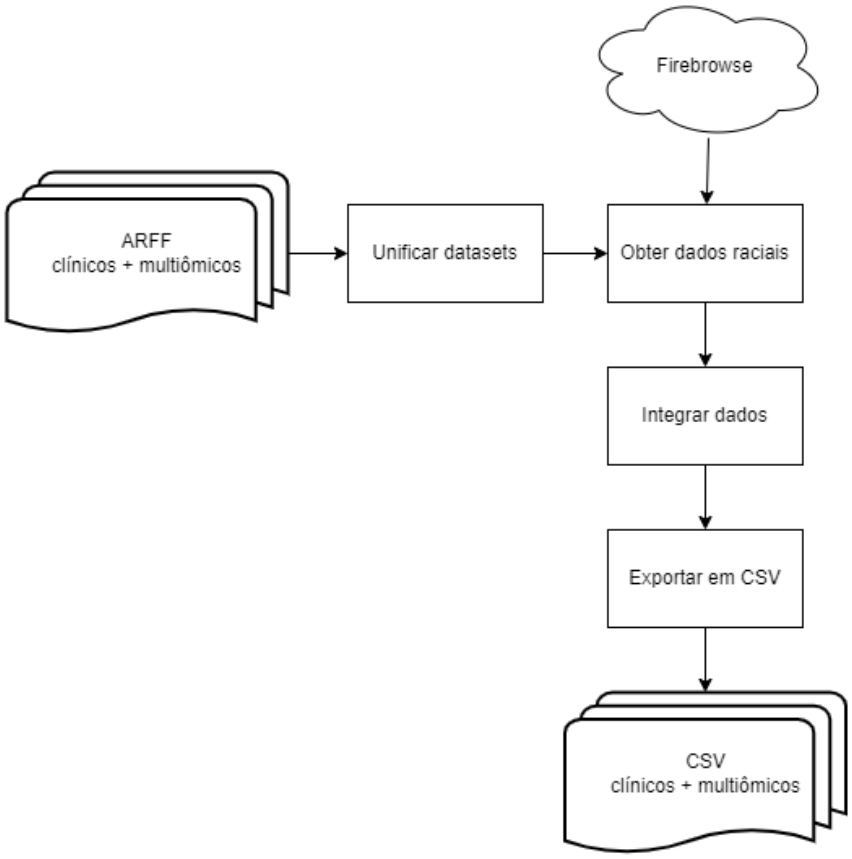
4.2 Conjuntos de dados

Os dados utilizados neste estudo foram obtidos do TCGA, utilizando os mesmos *datasets* empregados no grande estudo de *benchmark* conduzido por Herrmann et al. (2021). No entanto, foi necessário incluir um atributo com informações raciais de cada indivíduo, que não estava disponível nos conjuntos de dados fornecidos pelo autor. Essas informações são coletadas através de entrevistas com os participantes e podem ser acessadas na plataforma *FireBrowse*, mantida pelo *Broad Institute of MIT and Harvard*.

O *FireBrowse* fornece uma API REST que permite a consulta sob demanda de sua base de dados (Broad Institute TCGA Genome Data Analysis Center, 2016), filtrando pelo identificador atribuído pelo TCGA ao paciente em questão. Para isso, implementamos um pipeline de pré-processamento que consulta as informações raciais necessárias de cada instância e exporta um novo *dataset* unificado. Durante esse processo, outras tarefas foram realizadas para facilitar análises posteriores, como a conversão de formatos de arquivo (os conjuntos de dados estavam disponíveis em arquivos *.arff*, que apresentavam

baixa performance de leitura) e a unificação de arquivos (cada *dataset* estava dividido em múltiplos arquivos). A Figura 4.1 apresenta o pipeline completo.

Figura 4.1: Pipeline de processamento inicial



Fonte: O Autor

Cada conjunto de dados apresenta dados clínicos (como raça e estágio tumoral) e os níveis de expressão de mRNA (isto é, o transcriptoma). Selecionamos alguns *datasets* representativos para a execução dos experimentos. Para isso, calculamos a métrica PPMR sugerida por Dai et al. (2022), que mede a proporção entre os subconjuntos *major* e *minor*. Com essa métrica, escolhemos conjuntos que apresentem variados níveis de desbalanço. Os selecionados foram LIHC (*liver hepatocellular carcinoma*), KIRP (*kidney renal papillary cell carcinoma*), COAD (*colon adenocarcinoma*), BRCA (*breast cancer*), LUAD (*lung adenocarcinoma*), LUSC (*lung squamous cell carcinoma*) e LGG (*low-grade glioma*). A Tabela 4.1 apresenta a contagem de genes para cada *dataset* selecionado, a contagem de instâncias de cada grupo racial, bem como os valores encontrados de PPMR para cada conjunto de dados.

Tabela 4.1: Contagem de genes, instâncias por grupo racial e classificação de nível de viés

dataset	#genes	american indian or alaska native	asian	black or african american	white	NaN	#all	#major	#minor	PPMR
LGG	22297	1	5	18	385	10	409	385	24	0.941
LUSC	23524	0	7	25	294	92	326	294	32	0.902
LUAD	23681	0	6	45	332	43	383	332	51	0.867
BRCA	22694	1	16	137	571	10	725	571	154	0.788
COAD	22210	1	0	42	119	29	162	119	43	0.735
KIRP	32525	2	3	39	116	7	160	116	44	0.725
LIHC	20994	0	71	6	79	3	156	79	77	0.507

4.3 Identificação de genes causais

Dai et al. (2022) propõem uma estratégia para a identificação de genes causais utilizando o p-valor gerado pelo algoritmo de Cox, a qual foi adotada no presente trabalho. Essa análise é implementada por meio do Algoritmo 1. O conjunto de dados é dividido em *all*, *major* e *minor* usando a função $particionar\_conjunto$ e, para cada subconjunto gerado, o modelo de Cox é executado utilizando os dados clínicos de idade, sexo, raça, estágio tumoral e gene. Esse processo é repetido para cada gene presente no *dataset*, resultando em um p-valor que mede a significância do gene no resultado da regressão de Cox. No trabalho original (DAI et al., 2022), os 10 genes com os p-valores mais significativos em cada subconjunto são selecionados.

Algorithm 1: Identificação de genes causais

Data: $D(idade, sexo, raca, estagio\_tumoral, gene_1, gene_2, ...gene_m)$
Result: 10 genes mais significativos para cada um dos subconjuntos all, major e minor.

$genes \leftarrow \emptyset$;
foreach $sub \in \{all, major, minor\}$ **do**
  $D_{sub} \leftarrow particionar\_conjunto(D, sub)$;
  $gps \leftarrow \emptyset$;
  foreach $i \in 1...m$ **do**
    $ps \leftarrow Cox(D\_sub(idade, sexo, raca, estagio\_tumoral, gene_i))$;
    $p_{gene_i} \leftarrow ps[5]$;
    $gps \leftarrow gps \cup \{(gene_i, p_{gene_i})\}$;
  end
  $top\_genes \leftarrow \{g|(g,p) \in gps \wedge p$ é um dos 10 menores p-valores de $gps\}$;
  $genes \leftarrow genes \cup \{(sub, top\_genes)\}$;
end
return $genes$;

Com o objetivo de avaliar o impacto do tamanho dos conjuntos *major* e *minor* no resultado, geramos diferentes cenários experimentais através de implementações alterna-

tivas da função $particionar\_conjunto$, utilizada no Algoritmo 1. Foram implementadas as seguintes variantes:

- Particionamento por raça não-balanceado (Algoritmo 2): conjunto *major* composto por todas as instâncias que apresentam a raça majoritária e *minor* agrupando todas as demais. É a abordagem utilizada por Dai et al. (2022).

Algorithm 2: Particionar conjunto - particionamento por raça não-balanceado

Data: $D(idade, sexo, raca, estagio\_tumoral, gene_1, gene_2, ...gene_m)$ e sub
Result: Subconjunto de D de acordo com a partição especificada em sub

```
raca_majoritaria ← valor mais comum para raça em D;
switch sub do
    case all do
        return D;
    end
    case major do
        return {i|i ∈ D ∧ i[raca] = raca_majoritaria};
    end
    otherwise do
        /*minor*/
        return {i|i ∈ D ∧ i[raca] ≠ raca_majoritaria};
    end
end
```

- Particionamento aleatório (Algoritmo 3): Os tamanhos de *major* e *minor* são os mesmos da abordagem anterior (ou seja, desbalanceados), porém as instâncias que compõem cada conjunto são selecionadas aleatoriamente.
- Particionamento por raça balanceado (Algoritmo 4): Nesta abordagem geramos o conjunto *minor* por critério racial, conforme o Algoritmo 2, porém o *major* é gerado amostrando instâncias da raça majoritária de forma a resultar em *major* e *minor* de tamanhos iguais.

Nos cenários aleatório e balanceado diferentes amostragens aleatórias podem levar a diferentes p-valores. Para minimizar esse possível viés amostral, repetimos cada cenário 10 vezes com sementes aleatórias diferentes. As análises, detalhadas em sequência, são realizadas para cada uma das 10 execuções e o resultado médio é considerado.

Para analisar as semelhanças entre os conjuntos de genes selecionados utilizamos duas abordagens. Assim como Dai et al. (2022), plotamos os conjuntos gerados em diagramas de Venn para uma análise visual das intersecções entre os genes selecionados para *all*, *major* e *minor*. Além disso, utilizamos o coeficiente de Jaccard para obter uma

Algorithm 3: Particionar conjunto - particionamento aleatório

Data: $D(idade, sexo, raca, estagio\_tumoral, gene_1, gene_2, ...gene_m)$ e sub.

Result: Subconjunto de D de acordo com a partição especificada em sub.

$raca\_majoritaria \leftarrow$ valor mais comum para raça em D;
$tam\_major \leftarrow |\{i | i \in D \wedge i[raca] = raca\_majoritaria\}|$;
$cjt\_major \leftarrow$ amostre tam_major instâncias de D;
$cjt\_minor \leftarrow D \setminus cjt\_major$;
switch *sub* **do**
    case *all* **do**
        return D;
    end
    case *major* **do**
        return cjt_major;
    end
    otherwise do
        /*minor*/
        return cjt_minor;
    end
end

Algorithm 4: Particionar conjunto - particionamento por raça balanceado

Data: $D(idade, sexo, raca, estagio\_tumoral, gene_1, gene_2, ...gene_m)$ e sub.

Result: Subconjunto de D de acordo com a partição especificada em sub.

$raca\_majoritaria \leftarrow$ valor mais comum para raça em D;
$cjt\_minor \leftarrow \{i | i \in D \wedge i[raca] \neq raca\_majoritaria\}$;
$cjt\_major \leftarrow$ amostre $|cjt\_minor|$ instâncias de D;
$cjt\_all \leftarrow cjt\_major \cup cjt\_minor$;
switch *sub* **do**
    case *all* **do**
        return cjt_all;
    end
    case *major* **do**
        return cjt_major;
    end
    otherwise do
        /*minor*/
        return cjt_minor;
    end
end

medida quantitativa da semelhança entre os conjuntos de genes.

4.4 Aprendizado de máquina

Para avaliar o impacto do desbalanço entre os conjuntos *major* e *minor* nos algoritmos de aprendizado de máquina, criamos dois cenários de treinamento para o algoritmo RSF. Em um dos cenários utilizamos o particionamento desbalanceado conforme o Algoritmo 2 e no outro o particionamento balanceado conforme o Algoritmo 4.

4.4.1 Pré-processamento

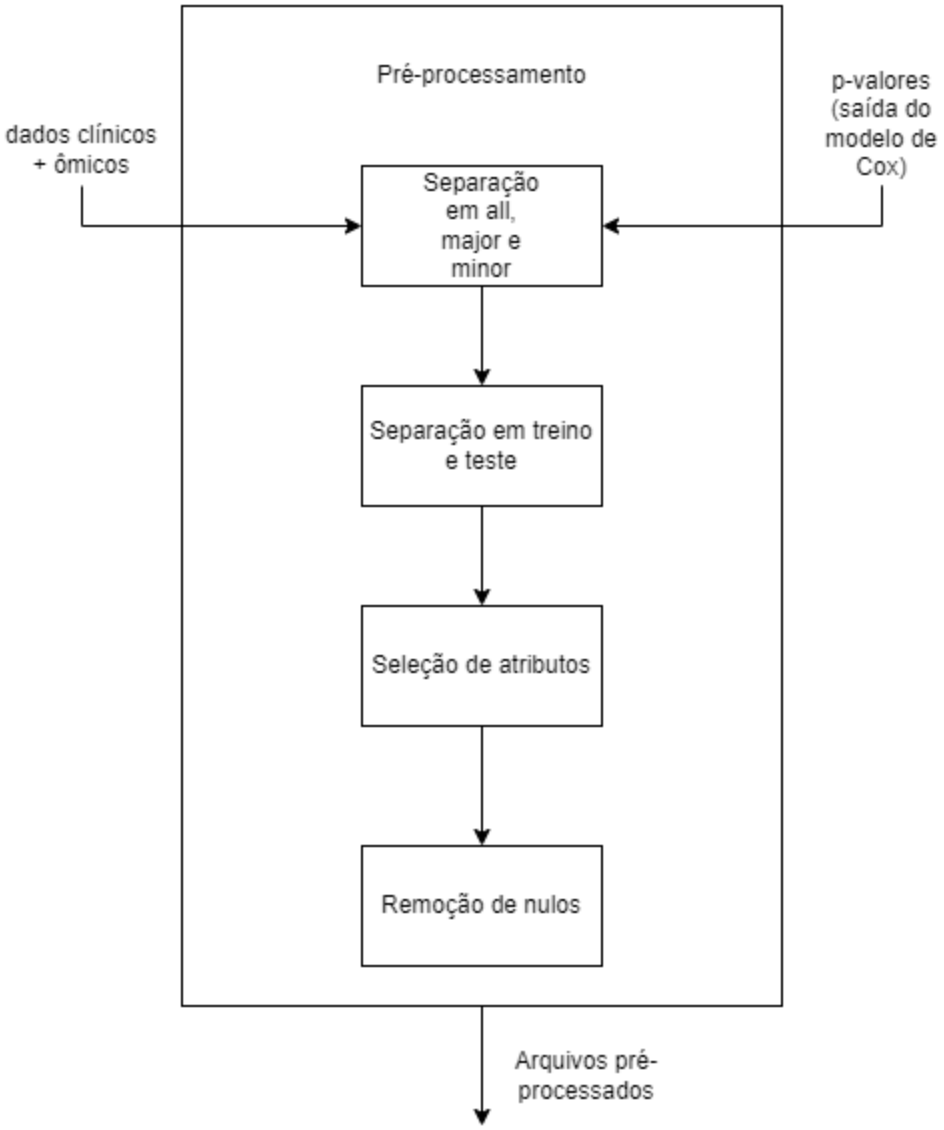
Implementamos um pipeline de pré-processamento para preparar os dados para o treinamento. A Figura 4.2 exibe as etapas realizadas.

A primeira etapa consistiu em realizar a separação em *all*, *major* e *minor* para ambos os cenários, seguindo os algoritmos 2 e 4. É importante ressaltar que, semelhantemente ao que ocorre na identificação de genes causais, a subamostragem realizada no cenário balanceado pode gerar conjuntos enviesados, particularmente mais fáceis ou mais difíceis que o original. Por isso, geramos 20 versões diferentes para cada *dataset* neste caso balanceado. Os valores reportados são sempre as médias dos 20 treinamentos resultantes. Ao término desta etapa temos então 21 conjuntos divididos em *all*, *major* e *minor* para cada *dataset* (um conjunto não-balanceado, 20 balanceados).

Cada subconjunto gerado é dividido em treino e teste utilizando uma proporção de 60% dos dados para treino e 40% para teste. Após realizarmos esta divisão para *major* e *minor*, tomamos a união do *major* e *minor* de teste como *all* de teste, e uma operação análoga é realizada para gerar o *all* de treino. A Figura 4.3 exibe um esquema de divisão dos dados para o cenário não-balanceado, ao passo que na Figura 4.4 temos uma representação da separação dos dados no cenário balanceado (com subamostragem do conjunto *major*). Em ambos os cenários, realizamos a divisão de treino e teste estratificando pelo atributo *status*, que indica se a instância apresenta *censoring* ou não. Esta medida é necessária para garantir uma distribuição do *censoring* semelhante nos conjuntos de treino e teste.

Para a seleção de atributos, adotou-se uma abordagem semelhante a Dai et al. (2022), baseada nos p-valores resultantes da identificação de genes causais. Os 10 genes

Figura 4.2: Pré-processamento

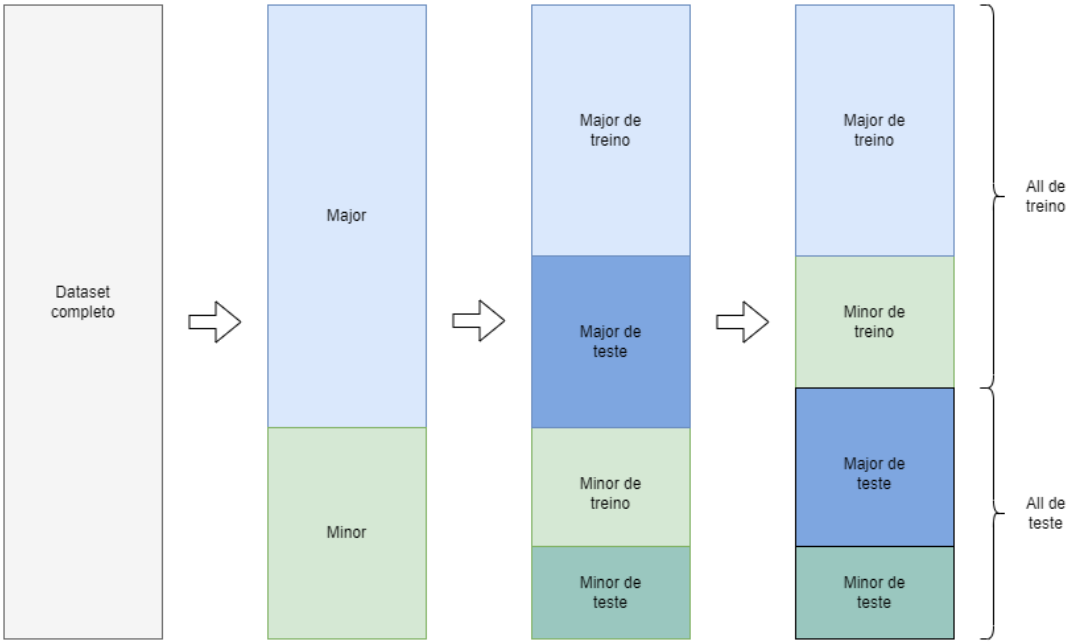


Fonte: O Autor

mais significativos reportados pelo algoritmo de identificação de genes causais foram utilizados como atributos para cada subconjunto. No caso balanceado, os genes utilizados foram obtidos pelas médias dos p-valores de cada gene nas 10 execuções do algoritmo de Cox. Dado que o treinamento e avaliação serão realizados em subconjuntos diferentes (por exemplo, treinamento em *major* e teste em *minor*), é importante enfatizar que a seleção dos atributos é feita de tal maneira que os atributos utilizados para o treinamento e teste sempre correspondem aos selecionados para o conjunto de treinamento em questão.

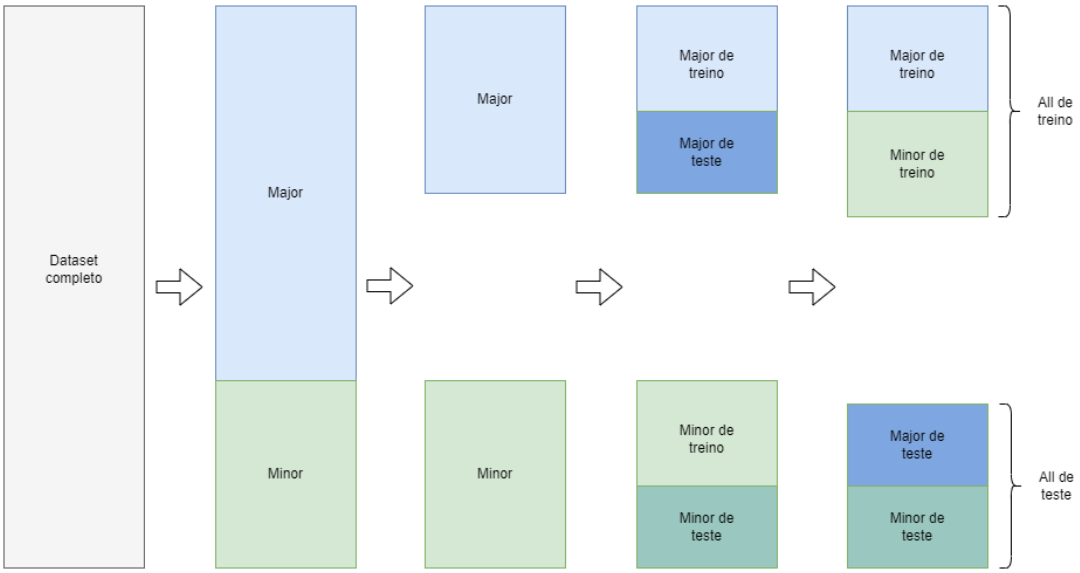
Por fim, durante o pré-processamento, foram removidas todas as instâncias que

Figura 4.3: Esquema da separação em treino e teste no caso não-balanceado



Fonte: O Autor

Figura 4.4: Esquema da separação em treino e teste no caso balanceado



Fonte: O Autor

apresentavam valores faltantes (nulos).

4.4.2 Treinamento e avaliação

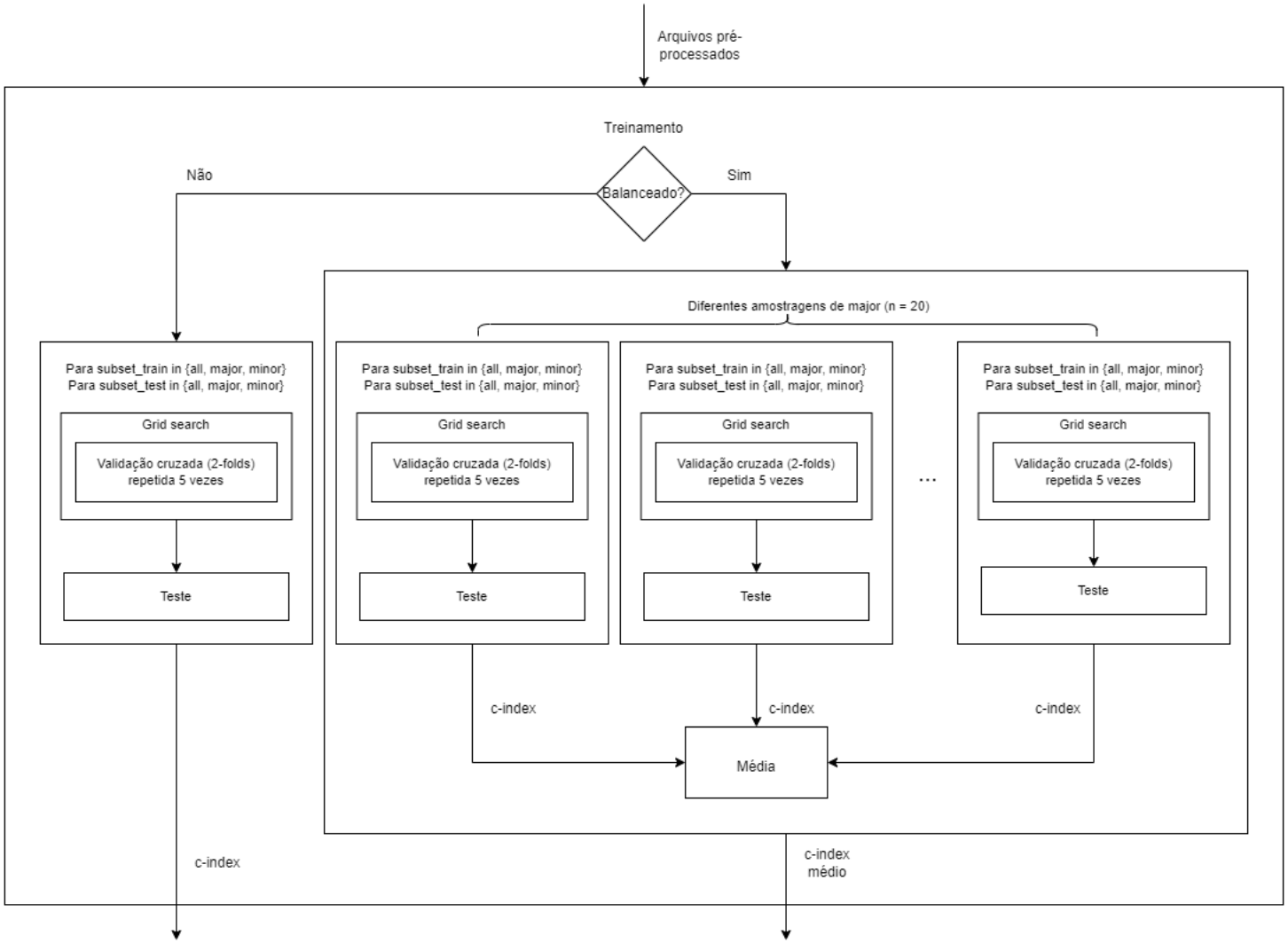
Como desejamos realizar treinamento em todos os subconjuntos (*all*, *major* e *minor*) e avaliar cada modelo também em cada subconjunto, o processo de treinamento e avaliação descrito a seguir ocorre para cada combinação de subconjunto de treino e subconjunto de avaliação. A Figura 4.5 exibe o *setup* utilizado para treinamento e avaliação.

A RSF foi treinada utilizando validação cruzada, com os dados de treinamento sendo divididos em 2 *folds*. A opção por um pequeno número de *folds* deve-se ao limitado tamanho amostral de algumas raças presentes nos *datasets*. A fim de mitigar possíveis vieses decorrentes de uma divisão que produzisse *folds* particularmente fáceis ou difíceis, a validação cruzada foi repetida 5 vezes, com as instâncias sendo embaralhadas a cada repetição. Além disso, foi realizada uma divisão estratificada pelo atributo *status* para garantir uma distribuição aproximadamente igual de *censoring* em todos os *folds*.

Conduzimos uma etapa de otimização de hiperparâmetros, a qual foi realizada utilizando a técnica *grid search*. O hiperparâmetro otimizado foi o $n\_estimators$, que representa o número de árvores no *ensemble*, e foram explorados valores no conjunto $\{50, 100, 150\}$. Após a identificação dos melhores hiperparâmetros com *grid search*, estes foram utilizados para treinar o modelo com todos os dados de treinamento. Em seguida, o modelo resultante foi avaliado com os dados de teste previamente separados.

No caso balanceado, o processo de treinamento e avaliação descrito foi repetido 20 vezes, utilizando uma das diferentes versões geradas com subamostragens do conjunto *major* em cada uma das iterações. Para cada um dos 20 modelos gerados, o C-index, métrica de desempenho utilizada, foi calculado e um valor médio foi obtido a partir dos resultados.

Figura 4.5: Esquema de treinamento e avaliação do modelo de AM



Fonte: O Autor

4.5 Detalhes de implementação

A implementação do projeto foi realizada inteiramente na linguagem de programação Python, versão 3.10.0. Para executar tarefas como separação de dados em treino e teste, validação cruzada, treinamento e validação, utilizamos o ferramental disponibilizado pela biblioteca scikit-learn (v1.1.2) (PEDREGOSA et al., 2011). O algoritmo RSF foi implementado através da biblioteca scikit-survival (v0.18.0) (PÖLSTERL, 2020), que disponibiliza uma série de algoritmos de análise de sobrevida com uma API compatível com o scikit-learn. Para o modelo de Cox, utilizamos a biblioteca lifelines (v0.27.3) (DAVIDSON-PILON, 2019). Ademais, outras bibliotecas de apoio utilizadas incluem pandas (v1.4.3) (TEAM, 2022) (MCKINNEY, 2010), numpy (v1.3.2) (HARRIS et al., 2020), matplotlib (v3.5.3) (HUNTER, 2007) e matplotlib-venn (v0.11.7) (TRETYAKOV, 2022).

5 RESULTADOS

Este capítulo apresenta os resultados dos experimentos detalhados no Capítulo 4.

5.1 Identificação de genes causais

5.1.1 Análise de diagramas de Venn

Na Figura 5.1 temos uma representação em diagramas de Venn dos conjuntos de genes selecionados para os subgrupos *all*, *major* e *minor* de cada *dataset* analisado.

No caso não-balanceado, observamos um padrão muito semelhante ao identificado por Dai et al. (2022). A intersecção entre os conjuntos *all* e *major* é, em geral, maior que a intersecção entre *all* e *minor*. Isso indica que ao criar um modelo que usa como atributos de entrada os genes mais relevantes da população total podemos estar selecionando genes que não são informativos para grupos minoritários, gerando um viés que prejudique estes indivíduos. O conjunto COAD é o único que não apresenta alguma diferença entre as interseções *all-major* e *all-minor* utilizando esta estratégia de particionamento.

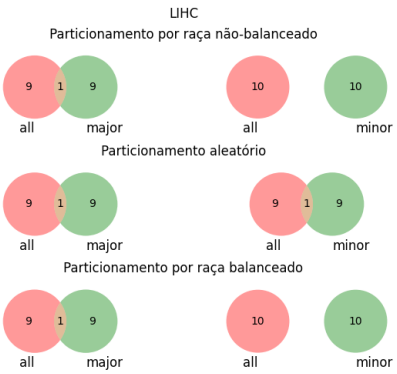
Para o cenário em que o particionamento é feito de forma aleatória mantendo os tamanhos de *major* e *minor*, observamos um comportamento semelhante, com interseções *all-major* maiores que intersecções *all-minor*. Os *datasets* BRCA, COAD, LUSC e LGG inclusive apresentaram um aumento da intersecção *all-major* neste caso. Já para KIRP houve uma diminuição da intersecção, ao passo que para LIHC e LUAD os valores de intersecção *all-major* se mantiveram os mesmos. Ao analisar a intersecção *all-minor*, observamos que LIHC apresentou um aumento em uma unidade neste cenário, enquanto todos os demais se mantiveram iguais em relação ao caso não-balanceado.

Porém, ao realizar o particionamento por raça balanceando os tamanhos de *major* e *minor*, observamos que as diferenças entre intersecções *all-major* e *all-minor* desaparecem ou são muito atenuadas.

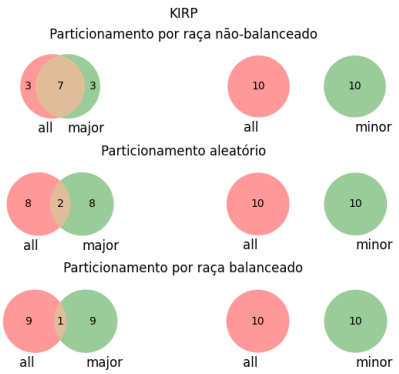
Esses resultados em conjunto indicam que a dominância do *major* sobre *all* observada no caso não-balanceado se deve principalmente à diferença de tamanho entre os conjuntos *major* e *minor*, e não a diferenças significativas nos dados utilizados. Essa ideia é reforçada quando observamos que os *datasets* mais desbalanceados (BRCA, LUAD, LUSC e LGG) foram os que apresentaram maior semelhança entre o caso não-balanceado e o caso aleatório.

Figura 5.1: Diagramas de Venn dos conjuntos de genes selecionados para *all*, *major* e *minor*

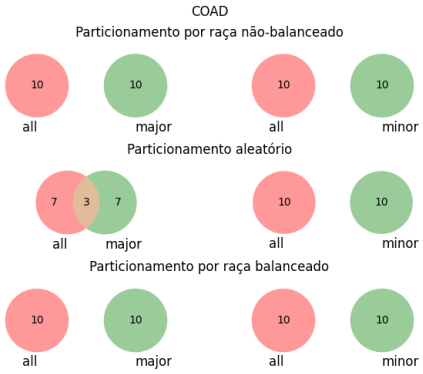
(a) LIHC



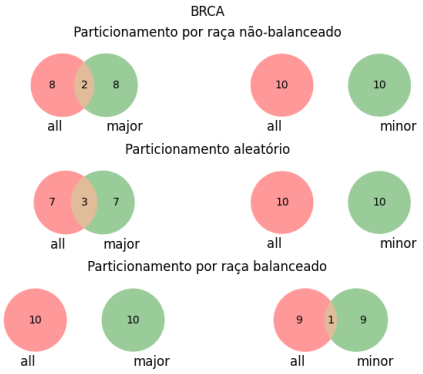
(b) KIRP



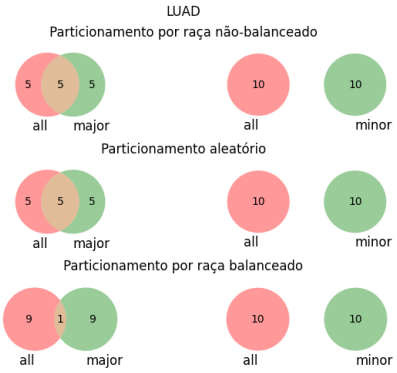
(c) COAD



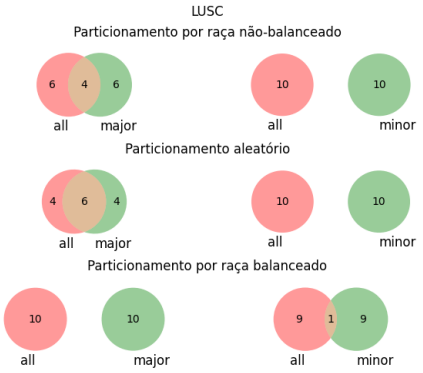
(d) BRCA



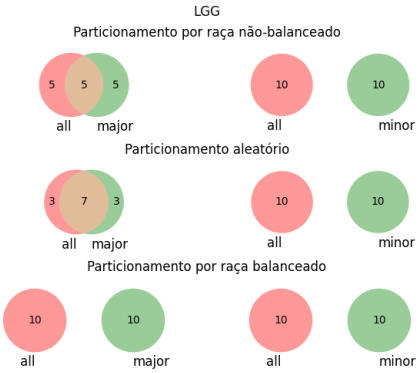
(e) LUAD



(f) LUSC



(g) LGG



Fonte: O Autor

É importante ressaltar que os valores de intersecção para os casos aleatório e balanceado são a média de 10 execuções arredondada para o inteiro mais próximo (para obter um valor de intersecção inteiro). A distribuição dos valores de intersecção encontrados em cada execução do cenário aleatório é apresentada na Figura 5.2. É possível observar que, em geral, há pouca variação nos valores de intersecção *all-minor*, com exceção do conjunto de dados LIHC. De acordo com a métrica PPMR, esse conjunto apresenta o menor nível de viés, com apenas uma diferença de 3 instâncias entre *major* e *minor*. Assim, o resultado similar entre as intersecções *all-major* e *all-minor* de LIHC no caso aleatório é esperado.

A distribuição das intersecções do cenário balanceado está exibida na Figura 5.3. Percebe-se que, de forma geral, houve menor variabilidade nas intersecções *all-major*. Por outro lado, LUSC e BRCA demonstram um aumento na variabilidade da intersecção *all-minor*, que se deve ao fato de que, na execução balanceada, o conjunto *all* varia consideravelmente entre diferentes execuções (visto que *all* é obtido pela união do *major* subamostrado e do *minor*). Coincidentemente, LUSC e BRCA foram os casos que apresentaram uma intersecção média *all-minor* maior no caso balanceado em relação ao caso não-balanceado. Com base nestas considerações, provavelmente este aumento se deve a um viés causado pela amostragem e não reflete algum padrão real nos dados. Possivelmente um número maior de execuções causaria uma diluição desta diferença, que então se aproximaria do caso não-balanceado.

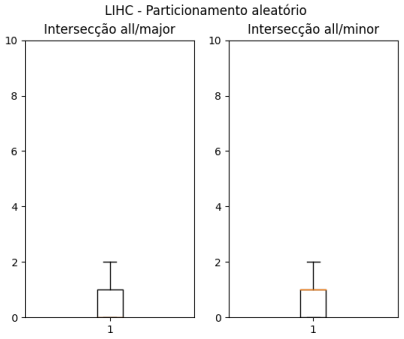
5.1.2 Análise de coeficientes de Jaccard

As tabelas 5.1, 5.2 e 5.3 apresentam os coeficientes de Jaccard para os cenários não-balanceado, aleatório e balanceado, respectivamente. O coeficiente de Jaccard mede a similaridade entre dois conjuntos e varia entre 0 e 1, onde 0 significa conjuntos totalmente diferentes e 1 indica conjuntos iguais. Para os cenários aleatório e balanceado, reportamos os valores de média e desvio padrão dos coeficientes calculados em cada uma das 10 execuções.

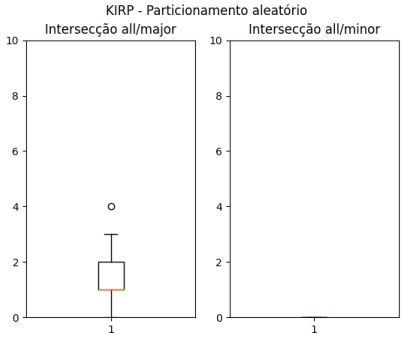
Observamos no *dataset* LIHC que a similaridade entre os conjuntos *all-major* e *all-minor* é quase a mesma em todos os casos. Isto provavelmente se deve ao fato do LIHC ser naturalmente quase balanceado. No entanto, no *dataset* KIRP, a similaridade entre os conjuntos *all* e *major* é uma ordem de magnitude menor nos casos aleatório e balanceado em comparação com o não-balanceado. Além disso, o caso balanceado de

Figura 5.2: Boxplots dos valores de intersecção para 10 execuções do particionamento aleatório

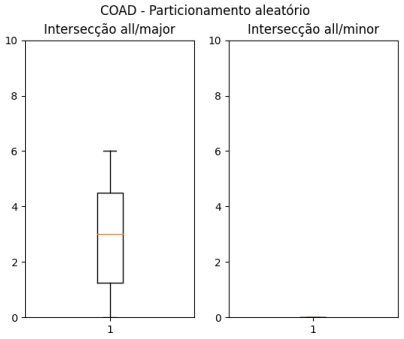
(a) LIHC



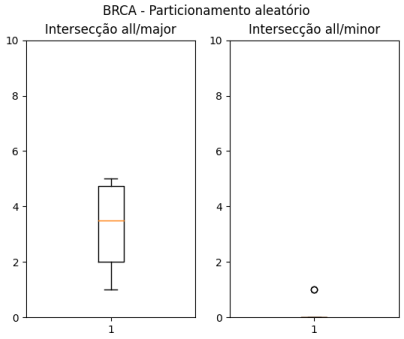
(b) KIRP



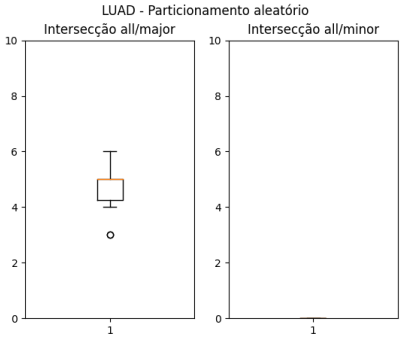
(c) COAD



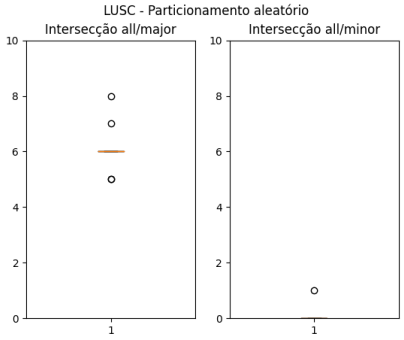
(d) BRCA



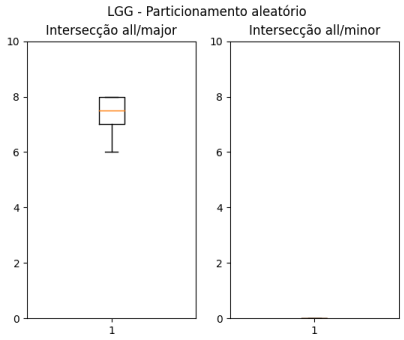
(e) LUAD



(f) LUSC



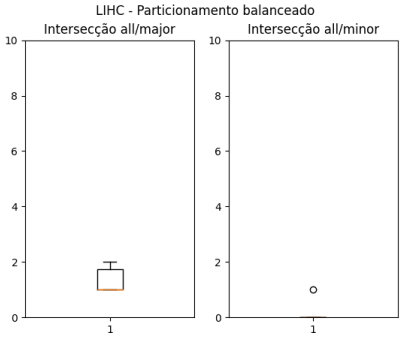
(g) LGG



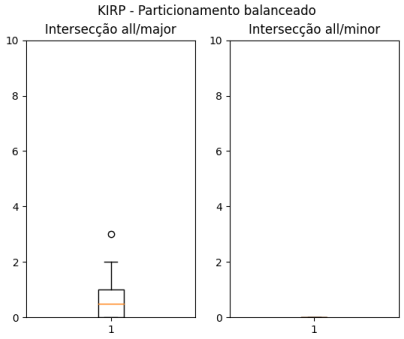
Fonte: O Autor

Figura 5.3: Boxplots dos valores de intersecção para 10 execuções do particionamento por raça balanceado

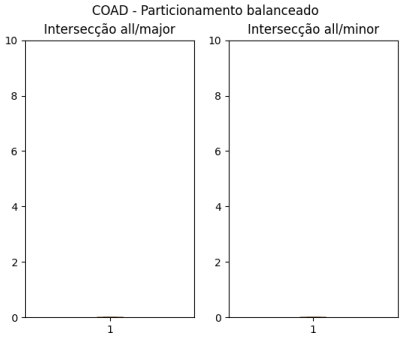
(a) LIHC



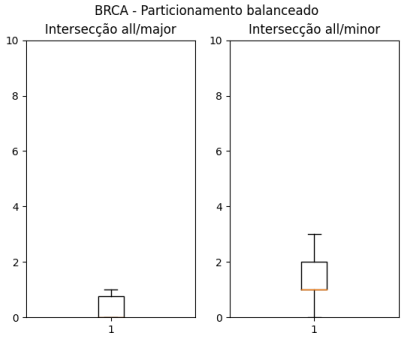
(b) KIRP



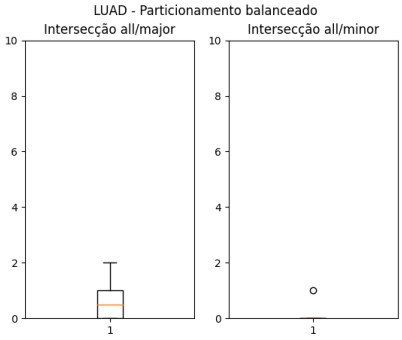
(c) COAD



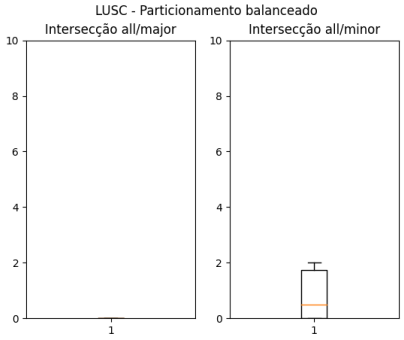
(d) BRCA



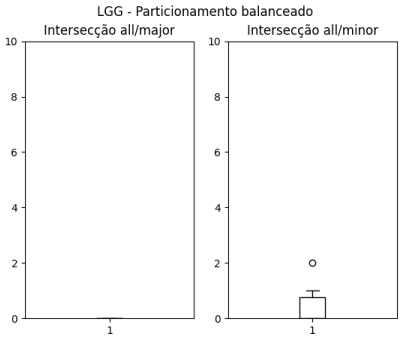
(e) LUAD



(f) LUSC



(g) LGG



Fonte: O Autor

KIRP apresenta ainda menos semelhança entre *all-major* do que o cenário aleatório.

No *dataset* COAD, os conjuntos *all* e *major* são totalmente diferentes em ambos os casos não-balanceado e balanceado, apresentando um valor de 0,179 no caso aleatório. Ou seja, não houve diferenças ao balancear *major* e *minor*, mas houve uma piora no caso aleatório.

Ao analisarmos o *dataset* BRCA, notamos que a comparação *all-major* no cenário aleatório apresenta uma similaridade que corresponde ao dobro daquela observada no não-balanceado. Por outro lado, a semelhança entre *all-major* no caso balanceado é uma ordem de magnitude menor em relação aos demais conjuntos. Além disso, a similaridade entre os conjuntos *all-minor* é pequena em todos os cenários.

No *dataset* LUAD, a comparação *all-major* apresenta valores semelhantes nos casos não-balanceado e aleatório, mas é uma ordem de magnitude menor no caso balanceado. Por sua vez, os conjuntos *all* e *minor* são totalmente diferentes nos casos não-balanceado e aleatório, e apresentam uma pequena similaridade de 0,005 no caso balanceado.

Considerando o *dataset* LUSC, podemos ver que os conjuntos *all* e *major* apresentam maior semelhança no caso aleatório em comparação com o não-balanceado, mas são totalmente distintos no caso balanceado. Já a comparação *all-minor* apresenta um coeficiente quase zero (0,005) no caso aleatório e zero nos demais casos.

No *dataset* LGG, os conjuntos *all* e *major* são mais semelhantes no caso aleatório do que no não-balanceado, enquanto são totalmente diferentes no cenário balanceado. A comparação *all-minor* apresenta um valor de 0,02 para o caso balanceado e zero para os demais casos.

Por fim, no *dataset* LGG, os conjuntos *all* e *major* são mais semelhantes no caso aleatório do que no não-balanceado, enquanto são totalmente diferentes no cenário balanceado. A comparação *all-minor* apresenta um valor de 0,02 no caso balanceado e zero nos demais casos.

Em geral, podemos concluir que o balanceamento dos dados pode afetar significativamente a similaridade entre os conjuntos de genes selecionados para *all*, *major* e *minor*. Em todos os *datasets* com exceção do LIHC (que já é quase balanceado), o balanceamento reduziu a similaridade entre os conjuntos *all-major* ou produziu resultados semelhantes aos obtidos nos conjuntos de dados não-balanceados. Em diversos casos, a semelhança entre *all* e *major* foi reduzida em até uma ordem de magnitude. Esta observação reforça a hipótese de que a alta similaridade entre *all* e *major* e a baixa similaridade entre *all* e

minor se devem ao tamanho dos conjuntos envolvidos (sendo *minor* tipicamente menor que *major*).

Tabela 5.1: Coeficientes de Jaccard - cenário não-balanceado

Dataset	Conjuntos comparados	C. Jaccard
LIHC	all-major	0.053
LIHC	all-minor	0.0
KIRP	all-major	0.539
KIRP	all-minor	0.0
COAD	all-major	0.0
COAD	all-minor	0.0
BRCA	all-major	0.111
BRCA	all-minor	0.0
LUAD	all-major	0.333
LUAD	all-minor	0.0
LUSC	all-major	0.25
LUSC	all-minor	0.0
LGG	all-major	0.333
LGG	all-minor	0.0

Tabela 5.2: Coeficientes de Jaccard - cenário aleatório

Dataset	Conjuntos comparados	C. Jaccard - média	C. Jaccard - desvio padrão
LIHC	all-major	0.033	0.044
LIHC	all-minor	0.037	0.035
KIRP	all-major	0.091	0.070
KIRP	all-minor	0.0	0.0
COAD	all-major	0.179	0.140
COAD	all-minor	0.0	0.0
BRCA	all-major	0.200	0.109
BRCA	all-minor	0.011	0.021
LUAD	all-major	0.313	0.084
LUAD	all-minor	0.0	0.0
LUSC	all-major	0.444	0.092
LUSC	all-minor	0.005	0.016
LGG	all-major	0.592	0.081
LGG	all-minor	0.0	0.0

5.2 Aprendizado de máquina

Durante o experimento que visava avaliar o impacto do viés racial em aprendizado de máquina, nos deparamos com uma limitação prática. É necessário ter pelo menos duas instâncias sem *censoring* no conjunto de teste para calcular o c-index, entretanto, esta condição não foi alcançada em todos os casos. Em alguns conjuntos de dados, não

Tabela 5.3: Coeficientes de Jaccard - cenário balanceado

Dataset	Conjutos comparados	C. Jaccard - média	C. Jaccard - desvio padrão
LIHC	all-major	0.070	0.027
LIHC	all-minor	0.005	0.016
KIRP	all-major	0.045	0.057
KIRP	all-minor	0.0	0.0
COAD	all-major	0.0	0.0
COAD	all-minor	0.0	0.0
BRCA	all-major	0.016	0.024
BRCA	all-minor	0.072	0.052
LUAD	all-major	0.032	0.036
LUAD	all-minor	0.005	0.016
LUSC	all-major	0.0	0.0
LUSC	all-minor	0.044	0.048
LGG	all-major	0.0	0.0
LGG	all-minor	0.022	0.036

havia instâncias suficientes para permitir a divisão em *all*, *major* e *minor* para treino e teste, conforme descrito no Capítulo 4, e ainda manter a restrição de duas instâncias sem *censoring* em todos os conjuntos de teste durante a validação cruzada. Devido a isso, o experimento com algoritmos de aprendizado de máquina foi restrito apenas aos conjuntos de dados que permitiam essas divisões, a saber, LIHC, LUAD, BRCA e LUSC.

Para cada *dataset*, treinamos um modelo em cada subgrupo (*all*, *major* e *minor*), e cada modelo foi testado em cada subgrupo. No cenário balanceado, foram realizadas 20 execuções, considerando diferentes subamostragens do *major* conforme detalhado no Caqpítulo 4. Ao total, foram treinados e avaliados 756 modelos, compreendendo 4 *datasets*, cada um com 3 subgrupos em cada cenário (balanceado e não-balanceado), e 20 repetições do cenário balanceado.

A Figura 5.4 apresenta os resultados obtidos no treinamento da *Random Survival Forest*, e a Tabela 5.4 relaciona a média e desvio padrão obtidos nas 20 execuções do cenário balanceado (os gráficos do cenário balanceado exibem a média), com tons de verde mais escuros indicando c-index mais próximo de 1. De imediato podemos notar que os diferentes *datasets* apresentaram diferentes padrões de resultado. Isso é de certa forma esperado, pois estamos considerando dados que se referem a doenças distintas e com diferentes níveis de desbalanceamento entre os subgrupos analisados.

Ao analisarmos o treinamento no *dataset* LIHC, no treino em *all* notamos um desempenho equilibrado em *all*, *major* e *minor* no caso não-balanceado, e praticamente o mesmo no caso balanceado. No treinamento em *major*, o melhor desempenho é observado ao testar em *major*, o que é esperado, e não há grandes alterações ao balancear

os conjuntos. Já o modelo treinado em *minor* apresenta um desempenho superior ao ser testado em *minor*, e embora a diferença entre *major* e *minor* tenha diminuído no caso balanceado, ainda existe uma diferença considerável. Em geral, podemos concluir que a diferença quase inexistente entre *major* e *minor* no treinamento em *all* é consistente com a observação de que esses conjuntos são naturalmente equilibrados no caso do LIHC. Além disso, o fato de o modelo apresentar um melhor desempenho em *major* ao treinar em *major* e em *minor* ao treinar em *minor* era esperado, sendo apenas uma constatação de que o modelo apresenta melhor poder preditivo ao ser testado em dados semelhantes aos usados em seu treinamento.

No treinamento do *dataset* BRCA em *all*, observamos que o modelo apresentou um desempenho melhor em *minor* do que em *major*, tanto no cenário não-balanceado quanto no balanceado. Ao treinar em *major*, os valores foram equilibrados em *major* e *minor* no caso não-balanceado, e houve um melhor desempenho no *major* ao balancear os conjuntos. Além disso, o modelo treinado no *minor* de BRCA apresentou um desempenho superior ao ser testado em *minor*, o que era esperado. Embora os valores de *major* e *minor* se tornem mais próximos no caso balanceado, ainda há uma diferença considerável entre eles. De forma geral, o BRCA parece apresentar uma diferença entre *major* e *minor*, em que o *minor* parece ser particularmente mais fácil. É interessante notar que o resultado mais equilibrado do BRCA foi obtido ao treinar o modelo em *major*.

No *dataset* LUAD considerando o conjunto *all*, verificamos um maior score em *major* no caso não-balanceado. Esse é o primeiro *dataset* a apresentar o mesmo padrão identificado por Dai et al. (2022) de dominância do *major* sobre o modelo. No caso balanceado, houve uma queda do *major* e um aumento do *minor*. Esse aumento do *minor* é consistente com a suposição de que o modelo é dominado pelo *major*. A queda do *major* para um patamar inferior ao verificado no caso não-balanceado possivelmente se deve a subamostragem não-representativa. Os mesmos padrões observados no treinamento em *all* ocorreram ao treinar em *major*. No treinamento em *minor*, constatamos um desempenho superior em *minor* em ambos os cenários em comparação com o *major*.

Por último, observamos que LUSC quando treinado em *all* apresenta um desempenho em *minor* levemente maior que em *major*, não ocorrendo mudanças consideráveis ao balancear os conjuntos. No treinamento em *major*, no caso não-balanceado, obtemos valores iguais para *major* e *minor*, sendo o resultado mais equilibrado para este *dataset*. Ao efetuar o balanceamento dos conjuntos nota-se uma queda no desempenho do *major*, possivelmente por causa de subamostragens não-representativas durante o processo de

treinamento e avaliação do modelo. Já no treinamento em *minor* o modelo se saiu melhor em *minor* considerando ambos os cenários. De forma geral, a estratégia de balanceamento dos conjuntos não causou grandes alterações no tocante à diferença entre *major* e *minor* para este conjunto de dados.

Podemos levantar alguns pontos a partir desta comparação entre os dois cenários. O desempenho da RSF nos diferentes subgrupos varia conforme o conjunto de dados utilizado, sendo que em alguns casos obtemos resultados mais equilibrados do que em outros, não havendo um padrão claro de dominância de um grupo específico. A estratégia de subamostragem aleatória não parece ser eficaz para atenuar significativamente as discrepâncias entre *major* e *minor*, e apresenta um ponto negativo a se considerar, que é a queda de desempenho do conjunto amostrado causada por eventuais subamostragens não-representativas. Como podemos ver na Tabela 5.4, existe grande variabilidade entre os desempenhos obtidos nas diferentes amostragens. Assim, podemos concluir que estratégias de balanceamento mais sofisticadas são necessárias para garantir que o conjunto subamostrado seja representativo do original.

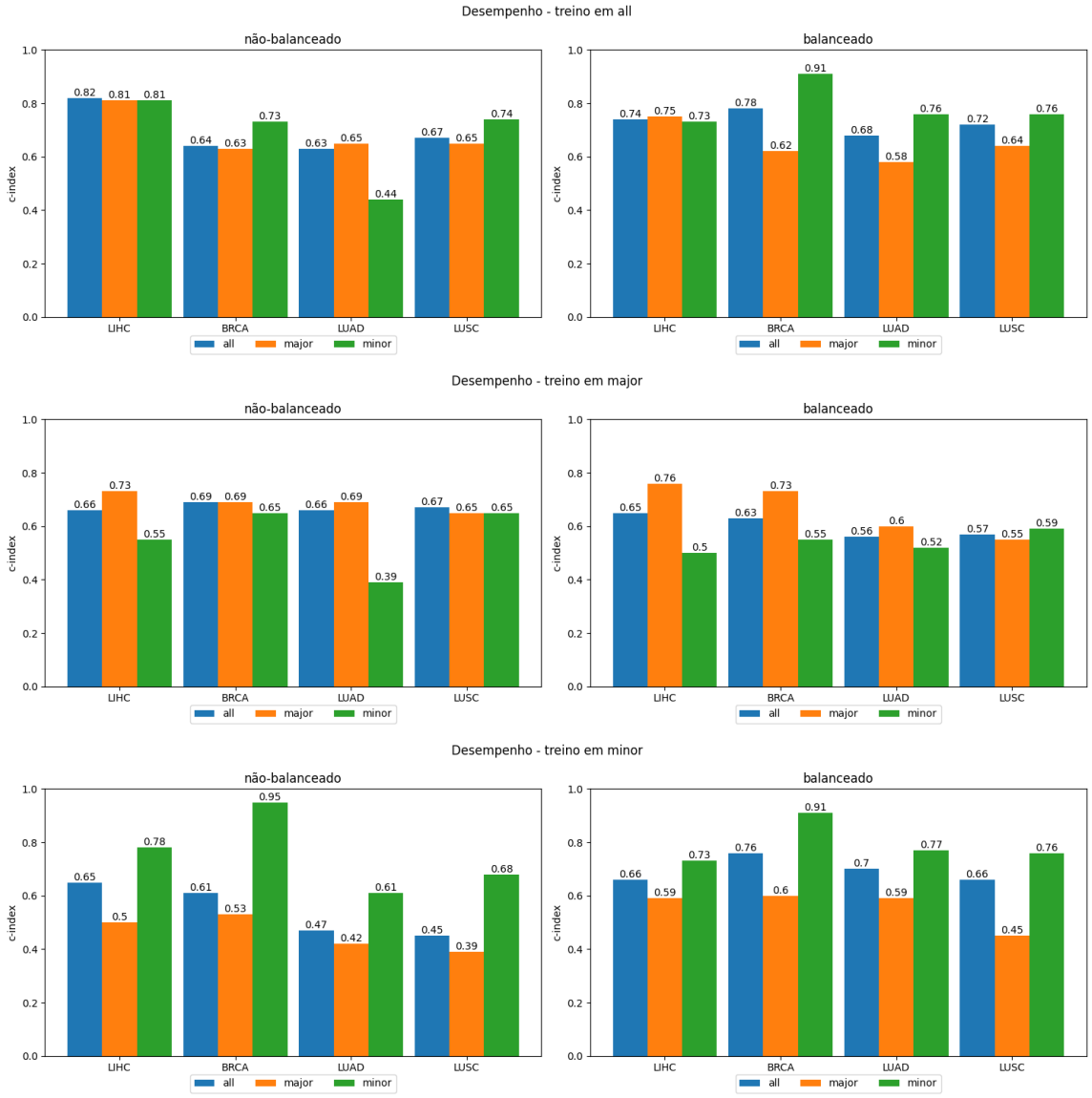
Outro objetivo deste experimento é avaliar se um modelo treinado em um subgrupo pode ser satisfatoriamente utilizado em outro. Consideramos o desempenho satisfatório se o score obtido supera o desempenho de um modelo aleatório, ou seja, se possui c-index maior que 0,5. Além disso, para esta avaliação consideraremos apenas o cenário não-balanceado.

Os resultados novamente variam para cada conjunto de dados. Para LIHC, um modelo treinado em *major* e testado em *minor* se sai levemente melhor que um modelo aleatório (c = 0,55), enquanto ao treinar em *minor* e testar em *major* obtemos o mesmo *score* de um modelo aleatório (c = 0,5).

Para BRCA, ao treinar em *major* e testar em *minor* ainda obtemos um desempenho melhor que o de um modelo aleatório (c = 0,65), porém obtemos um valor muito inferior ao treinar em *minor* e testar em *major* (c = 0,53). Logo, para BRCA poderia ser viável treinar um modelo em *major* e aplicá-lo em *minor*, ainda mais quando consideramos que o treinamento em *major* obteve resultados muito equilibrados entre *major* e *minor*.

Em LUAD, observamos que nem mesmo ao treinar em *all* obtemos um desempenho em *minor* que seja satisfatório (c = 0,44), o que só conseguimos ao treinar em *minor* (c = 0,61). Porém, treinando em *minor* obtemos *score* insatisfatório ao testar em *major*. Para LUAD, portanto, nenhum modelo se mostrou adequado a todos os subgrupos específicos.

Figura 5.4: Resultados do treinamento do modelo de aprendizado de máquina



Fonte: O Autor

Tabela 5.4: Scores da RSF - cenário balanceado

Dataset	Conj. treino	Conj. teste	C-index (média)	C-index (desvio padrão)
LIHC	all	all	0.743	0.073
LIHC	all	major	0.752	0.055
LIHC	all	minor	0.725	0.153
LIHC	major	all	0.648	0.060
LIHC	major	major	0.755	0.059
LIHC	major	minor	0.496	0.128
LIHC	minor	all	0.661	0.063
LIHC	minor	major	0.587	0.097
LIHC	minor	minor	0.732	0.141
BRCA	all	all	0.782	0.051
BRCA	all	major	0.621	0.119
BRCA	all	minor	0.910	0.046
BRCA	major	all	0.634	0.100
BRCA	major	major	0.732	0.128
BRCA	major	minor	0.554	0.132
BRCA	minor	all	0.756	0.072
BRCA	minor	major	0.597	0.126
BRCA	minor	minor	0.907	0.046
LUAD	all	all	0.680	0.106
LUAD	all	major	0.580	0.186
LUAD	all	minor	0.763	0.116
LUAD	major	all	0.562	0.102
LUAD	major	major	0.596	0.162
LUAD	major	minor	0.516	0.150
LUAD	minor	all	0.701	0.115
LUAD	minor	major	0.592	0.183
LUAD	minor	minor	0.775	0.150
LUSC	all	all	0.725	0.073
LUSC	all	major	0.635	0.186
LUSC	all	minor	0.760	0.113
LUSC	major	all	0.572	0.140
LUSC	major	major	0.548	0.210
LUSC	major	minor	0.595	0.151
LUSC	minor	all	0.659	0.057
LUSC	minor	major	0.446	0.217
LUSC	minor	minor	0.765	0.080

Para LUSC, temos um desempenho satisfatório ao treinar em *major* e testar em *minor* (c = 0,65), porém insatisfatório ao treinar em *minor* e testar em *major* (c = 0,39). Assim como ocorreu com BRCA, o desempenho mais equilibrado ocorre ao treinar em *major*.

Evidentemente, em uma aplicação real, utilizaríamos a totalidade de dados disponíveis (ou seja, treino e teste em *all*). Neste caso, todos os modelos atingiram um desempenho satisfatório.

6 CONCLUSÃO

Com o advento de bases de dados públicas como o TCGA, a pesquisa a respeito de algoritmos de aprendizado de máquina na área de dados ômicos tem crescido significativamente. No entanto, sabemos que os dados presentes nesses bancos não necessariamente constituem uma amostra representativa da população em geral no que tange a características como raça, com grupos minoritários sendo sub-representados em alguns casos (SPRATT et al., 2016). Como resultado, os modelos resultantes dos algoritmos de AM treinados com tais *datasets* podem apresentar vieses que levem a resultados piores para determinados grupos, como minorias raciais. É fundamental, portanto, investigar o impacto do viés racial nos dados ômicos em algoritmos de aprendizado de máquina.

Neste trabalho, conduzimos experimentos para entender como modelos de seleção de genes causais e análise de sobrevida podem ser afetados pela disparidade entre grupos raciais. Conduzimos experimentos com dados de expressão de mRNA de diferentes tipos de câncer obtidos do TCGA, considerando três grupos para cada *dataset*: *all* (composto por todas as instâncias), *major* (composto pelas instâncias que apresentam a raça mais comum no conjunto de dados) e *minor* (composto pelas instâncias que apresentam raça diferente da majoritária).

Na tarefa de identificação de genes causais utilizamos o modelo de Cox de riscos proporcionais para obter um p-valor associado a cada gene, que mede sua significância no resultado da regressão. Selecionamos os 10 genes mais significativos para cada subgrupo e analisamos as semelhanças entre os conjuntos obtidos. O objetivo deste experimento era entender se as semelhanças entre os conjuntos de genes selecionados para *all* e *major* observadas no artigo de Dai et al. (2022) se davam por razões genéticas de fato ou por um efeito de tamanho amostral (visto que *major* é tipicamente muito maior que *minor*). Para tanto, executamos três cenários: o cenário não-balanceado, com *major* e *minor* sendo separados por critério racial conforme dito anteriormente, o cenário aleatório, com *major* e *minor* de mesmos tamanhos do cenário não-balanceado mas compostos por instâncias selecionadas aleatoriamente do conjunto completo, e o cenário balanceado, em que é feita uma subamostragem aleatória do conjunto *major* para gerar *major* e *minor* de mesmo tamanho.

Observamos que ao efetuar o balanceamento dos conjuntos a semelhança entre os conjuntos de genes selecionados para *all* e *major* diminui consideravelmente. Assim, concluímos que esta dominância do *major* sobre o resultado geral se dá por causa do ta-

manho dos conjuntos envolvidos e não por uma eventual diferença genética nos dados utilizados. É importante ressaltar que, embora o balanceamento de conjuntos tenha atenuado a discrepância entre major e minor, essa técnica não é uma solução definitiva para o problema do baixo número de amostras de raças minoritárias. Precisamos ser mais representativos e promover uma maior diversidade nos bancos ômicos, buscando incluir grupos minoritários de forma mais justa.

O segundo experimento consistiu no treinamento de uma *Random Survival Forest*, que é um modelo de AM para análise de sobrevida. Efetuamos o treinamento em cada subgrupo e testamos cada modelo também em cada subgrupo. Neste experimento consideramos os cenários não-balanceado e balanceado. Utilizamos o c-index como métrica de desempenho. Com este experimento buscamos avaliar o quanto o desbalanço entre *major* e *minor* impacta no desempenho de um modelo de análise de sobrevida e se a estratégia de balancear os conjuntos com uma subamostragem aleatória de *major* causaria alguma alteração significativa nos resultados.

Os resultados obtidos revelam que, nesta tarefa de análise de sobrevida, o efeito do desbalanceamento depende diretamente do conjunto de dados utilizado. Se considerarmos o modelo treinado em *all* no cenário não-balanceado, vemos que utilizando LIHC não há diferenças significativas entre *major* e *minor*, enquanto BRCA e LUSC apresentam desempenho melhor em *minor* e LUAD se sai melhor em *major*. Também constatamos que a subamostragem aleatória não é uma técnica eficaz para se obter um modelo que performe de modo aproximadamente igual em *major* e *minor*, além de levar a uma piora considerável do desempenho em alguns casos. Possivelmente esta queda de performance se dá por conta de uma subamostragem não-representativa.

Também avaliamos até que ponto seria possível aplicar o modelo treinado em um subgrupo em outro e ainda se obter desempenho superior a um modelo aleatório. Verificamos que LIHC, BRCA e LUSC treinados em *major* poderiam ser utilizados de forma satisfatória em *minor*, enquanto nenhum modelo treinado em um subgrupo de LUAD mostrou desempenho satisfatório em todos os demais subgrupos.

Os experimentos conduzidos apresentam, obviamente, algumas limitações. Para a tarefa de análise de sobrevida apenas um algoritmo foi considerado (*Random Survival Forest*). Trabalhos futuros poderiam investigar se outros algoritmos apresentam resultados semelhantes. Ainda, estratégias de balanceamento mais sofisticadas poderiam ser consideradas, para gerar conjuntos balanceados que sejam amostras representativas de fato. Além disso, é relevante estender a análise para outros tipos de dados ômicos.

REFERÊNCIAS

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, p. 5–32, 2001.

BREIMAN, L. Classification and regression trees. Routledge, 2017.

Broad Institute TCGA Genome Data Analysis Center. **Analysis-ready standardized TCGA data from Broad GDAC Firehose 2016_01_28 run**. [S.l.]: Broad Institute of MIT and Harvard, 2016.

COX, D. R. Regression models and life-tables. **Journal of the Royal Statistical Society: Series B (Methodological)**, Wiley Online Library, v. 34, n. 2, p. 187–202, 1972.

DAI, B. et al. Racial Bias Can Confuse AI for Genomic Studies. **Oncologie**, v. 24, n. 1, 2022.

DAVIDSON-PILON, C. lifelines: survival analysis in python. **Journal of Open Source Software**, The Open Journal, v. 4, n. 40, p. 1317, 2019. Available from Internet: <https://doi.org/10.21105/joss.01317>.

FACELI, K. et al. Inteligência artificial: uma abordagem de aprendizado de máquina. 2011.

HARRIS, C. R. et al. Array programming with NumPy. **Nature**, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, sep. 2020. Available from Internet: <https://doi.org/10.1038/s41586-020-2649-2>.

HERRMANN, M. et al. Large-scale benchmark study of survival prediction methods using multi-omics data. **Briefings in bioinformatics**, Oxford University Press, v. 22, n. 3, p. bbaa167, 2021.

HUNTER, J. D. Matplotlib: A 2D graphics environment. **Computing in Science & Engineering**, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007.

ISHWARAN, H. et al. Random survival forests. 2008.

JIANG, P. et al. Big data in basic and translational cancer research. **Nature Reviews Cancer**, Nature Publishing Group UK London, v. 22, n. 11, p. 625–639, 2022.

KAPLAN, E. L.; MEIER, P. Nonparametric estimation from incomplete observations. **Journal of the American statistical association**, Taylor & Francis, v. 53, n. 282, p. 457–481, 1958.

KESSLER, M. D. et al. Challenges and disparities in the application of personalized genomic medicine to populations with african ancestry. **Nature communications**, Nature Publishing Group UK London, v. 7, n. 1, p. 12521, 2016.

KLEINBAUM, D. G.; KLEIN, M. Survival analysis a self-learning text. Springer, 2012.

KOUROU, K. et al. Machine learning applications in cancer prognosis and prediction. **Computational and Structural Biotechnology Journal**, Elsevier, v. 13, p. 8–17, 2015.

LEBLANC, M.; CROWLEY, J. Survival trees by goodness of split. **Journal of the American Statistical Association**, Taylor & Francis, v. 88, n. 422, p. 457–467, 1993.

LIU, J. et al. An integrated TCGA pan-cancer clinical data resource to drive high-quality survival outcome analytics. **Cell**, Elsevier, v. 173, n. 2, p. 400–416, 2018.

MCKINNEY Wes. Data Structures for Statistical Computing in Python. In: WALT Stéfan van der; MILLMAN Jarrod (Ed.). **Proceedings of the 9th Python in Science Conference**. [S.l.: s.n.], 2010. p. 56 – 61.

NICORA, G. et al. Integrated multi-omics analyses in oncology: a review of machine learning methods and tools. **Frontiers in Oncology**, Frontiers Media SA, v. 10, p. 1030, 2020.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

POLIKAR, R. Ensemble based systems in decision making. **IEEE Circuits and systems magazine**, IEEE, v. 6, n. 3, p. 21–45, 2006.

PÖLSTERL, S. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. **Journal of Machine Learning Research**, v. 21, n. 212, p. 1–6, 2020. Available from Internet: <http://jmlr.org/papers/v21/20-729.html>.

RAUFASTE-CAZAVIEILLE, V.; SANTIAGO, R.; DROIT, A. Multi-omics analysis: Paving the path toward achieving precision medicine in cancer treatment and immuno-oncology. **Frontiers in Molecular Biosciences**, Frontiers Media SA, v. 9, 2022.

REFAEILZADEH, P. et al. Cross-validation. **Encyclopedia of database systems**, Springer, v. 5, p. 532–538, 2009.

SPOONER, A. et al. A comparison of machine learning methods for survival analysis of high-dimensional clinical data for dementia prediction. **Scientific reports**, Springer, v. 10, n. 1, p. 1–10, 2020.

SPRATT, D. E. et al. Racial/ethnic disparities in genomic sequencing. **JAMA oncology**, American Medical Association, v. 2, n. 8, p. 1070–1074, 2016.

TEAM, T. pandas development. **pandas-dev/pandas: Pandas 1.4.3**. Zenodo, 2022. Available from Internet: <https://doi.org/10.5281/zenodo.6702671>.

TRETYAKOV, K. **KONSTANTINT/matplotlib-venn: Area-weighted Venn-diagrams for python/matplotlib**. 2022. Available from Internet: <https://github.com/konstantint/matplotlib-venn>.

WANG, P.; LI, Y.; REDDY, C. K. Machine learning for survival analysis: A survey. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 51, n. 6, p. 1–36, 2019.

ZHANG, G. et al. Characterization of frequently mutated cancer genes in Chinese breast tumors: a comparison of Chinese and TCGA cohorts. **Annals of translational medicine**, AME Publications, v. 7, n. 8, 2019.