UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA - CAMPUS SÃO CARLOS
DEPARTAMENTO DE COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

RODNEY RENATO DE SOUZA SILVA

APRENDIZADO DE MÁQUINA CONSTRUTIVO E CLASSIFICAÇÃO HIERÁRQUICA MULTIRRÓTULO APLICADOS À GERAÇÃO DE MOLÉCULAS

SÃO CARLOS
março/2023

RODNEY RENATO DE SOUZA SILVA

APRENDIZADO DE MÁQUINA CONSTRUTIVO E CLASSIFICAÇÃO HIERÁRQUICA MULTIRRÓTULO APLICADOS À GERAÇÃO DE MOLÉCULAS

Dissertação de Mestrado em Ciência da Computação da Universidade Federal de São Carlos, Centro de Ciências Exatas e de Tecnologia, Campus São Carlos.
Orientador: Prof. Dr. Ricardo Cerri

SÃO CARLOS
março/2023

Rodney Renato de Souza Silva Aprendizado de Máquina Construtivo e Classificação Hierárquica Multirrótulo aplicados à Geração de Moléculas/ Rodney Renato de Souza Silva. – São Carlos, março/2023- 52 p. : il. (algumas color.) ; 30 cm.
Orientador: Prof. Dr. Ricardo Cerri
Dissertação de Mestrado – Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Mestrado em Ciência da Computação, março/2023.
1. Aprendizado de máquina

Rodney Renato de Souza Silva

Aprendizado de Máquina Construtivo e Classificação Hierárquica Multirrótulo aplicados à Geração de Moléculas

Dissertação de Mestrado em Ciência da Computação da Universidade Federal de São Carlos, Centro de Ciências Exatas e de Tecnologia, Campus São Carlos.

Banca Examinadora

Prof. Dr. Ricardo Cerri
Orientador
Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Computação

Profa. Dra. Helena de Medeiros Caseli
Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Computação

Prof. Dr. Denis Mauá
Universidade de São Paulo
Instituto de Matemática e Estatística
Departamento de Computação

São Carlos, _____ de _____ de _____.

Dedico esse trabalho a todos que me inspiram.

Agradecimentos

Infelizmente este período foi marcado por uma pandemia e um desgoverno. Agradeço aos familiares e amigos que me fizeram companhia neste período difícil para todos.

Agradeço ao meu orientador Ricardo, mesmo que nunca tenhamos nos encontrado pessoalmente, soube oferecer ajuda mesmo quando eu não soube pedir.

Agradeço também à CAPES pelo financiamento desta pesquisa.

Por fim agradeço a Alan Turing, ateu e homossexual, pai da ciência da computação (1912-1954), agradeço a todos os grandes que contribuíram e a todos que contribuirão para a construção de um futuro mais brilhante.

Meaning is a Jumper That You Have to Knit Yourself.

- Exurb1a

Resumo

Um dos objetivos da Química Medicinal é descobrir novas moléculas com características de fármacos, o que é desafiador, pois o espaço de busca é discreto, não estruturado e enorme. Nos últimos anos, a computação tem sido usada como ferramenta auxiliar na pesquisa química, e um dos campos da ciência da computação que ganhou visibilidade e foi aplicado em diversas áreas do conhecimento nos últimos anos é o Aprendizado de Máquina. O campo de Aprendizado de Máquina pode ser dividido em várias áreas de estudo. Nesta pesquisa, são abordados dois campos de Aprendizado de Máquina: Aprendizado de Máquina Construtivo e Classificação Hierárquica Multirrótulo. Este trabalho explora como o Aprendizado de Máquina Construtivo pode aprender as regras intrínsecas dos bancos de dados de moléculas e gerar instâncias com características semelhantes a essas. Os métodos de Aprendizado de Máquina Construtivo escolhidos para o estudo podem ser divididos em dois tipos: aqueles que usam a representação molecular SMILES e aqueles que usam grafos para representar moléculas. Considerando as diferentes possibilidades de avaliar os métodos e as moléculas geradas, este trabalho propõe o uso de classificação hierárquica no processo de avaliação. Usando um classificador hierárquico previamente treinado em conjuntos de dados de moléculas, as moléculas geradas são classificadas em uma taxonomia. Dessa forma, a relevância das moléculas geradas para as taxonomias existentes pode ser verificada. Este trabalho também propõe uma medida de dissimilaridade entre dois grupos de moléculas, a distância hierárquica, que leva em consideração a taxonomia das moléculas presentes nesses grupos para determinar a dissimilaridade entre eles.

Palavras-chave: Aprendizado de Máquina, Aprendizado de Máquina Construtivo, Classificação Hierárquica, Criação de Drogas.

Abstract

One of the goals of Medicinal Chemistry is to discover new molecules with drug-like characteristics, which is challenging because the search space is discrete, unstructured, and enormous. In recent years, computation has been used as an auxiliary tool in chemical research, and one of the fields of computer science that has gained visibility and applied in various areas of knowledge in recent years is Machine Learning. The field of Machine Learning can be divided into several areas of study. In this research, two fields of Machine Learning are addressed: Constructive Machine Learning and Hierarchical Multi-label Classification. This work explores how Constructive Machine Learning can learn the intrinsic rules of molecule databases and generate instances with similar characteristics to these. The chosen Constructive Machine Learning methods for the study can be divided into two types, those that use the SMILES molecular representation and the methods that use graphs to represent molecules. Considering the different possibilities for evaluating methods and generated molecules, this work proposes the use of hierarchical classification in the evaluation process. Using a hierarchical classifier previously trained on molecule datasets, the generated molecules are classified into a taxonomy. In this way, the relevance of the generated molecules to existing taxonomies can be verified. This work also proposes a measure of dissimilarity between two groups of molecules, the hierarchical distance, which takes into account the taxonomy of the molecules present in these groups to determine the dissimilarity between them.

Keywords: Machine Learning, Contructive Machine Learning, Hierarchical Classification, Drug Design.

Lista de figuras

Lista de abreviaturas e siglas

AE	Autoencoder
AAE	Adversarial Autoencoder
AUPRC	Area Under Precision Recall Curve
AUROC	Area Under Receiver Operating Characteristic
ChEBI	Chemical Entities of Biological Interest
CML	Constructive Machine Learning
DAG	Directed Acyclic Graph
DRD2	Dopamine Receptor Type 2
GAN	Generative Adversarial Network
GCPN	Graph Convolutional Policy Network
GT4SD	Generative Toolkit for Scientific Discovery
LSTM	Long Short-Term Memory
MDP	Markov Decision Process
MCTS	Monte Carlo Tree Search
ML	Machine Learning
PCA	Principal Component Analysis
QED	Quantitative Estimate of Druglikeness
QSAR	Quantitative Structure-Activity Relationship
RAE	Recurrent Autoencoder
RDF	Resource Description Framework
RL	Reinforcement Learning
RNN	Recurrent Neural Network
SAscore	Synthetic Accessibility Score

SeqGAN	Sequence Generative Adversarial Network
SMILES	Simplified Molecular Input Line Entry System
UMAP	Uniform Manifold Approximation and Projection
VAE	Variational Autoencoder
VRAE	Variational Recurrent Autoencoder

Sumário

1 Introdução

Este capítulo, na Seção 1.1, apresenta o contexto no qual esta pequisa está inserida e suas motivações, identificando o problema a ser investigado. A Seção 1.2 lista os objetivos principais desta pesquisa. Os resultados obtidos com a pesquisa são resumidos na Seção 1.3. Por fim, a Seção 1.4 apresenta uma visão geral dos capítulos seguintes e como o texto está organizado.

1.1 Contexto e motivação

> Desde o início dos tempos o homem tentou controlar as doenças a fim de manter sua sobrevivência. As primeiras informações que concernem a prática da medicina datam do 5º milênio A.C., com os egípcios. Os sacerdotes egípcios uniam a medicina e a teleologia supersticiosa de maneira muito íntima; usavam ervas medicinais e outros produtos como leite, mel, sal ou cerveja, mas nenhum deles era considerado eficaz no tratamento de infecções sem o uso de invocações mágicas (FERREIRA; PAES; LICHTENSTEIN, 2008).

Muitos medicamentos foram descobertos por sorte ao decorrer da história. A descoberta da penicilina é considerada um dos acontecimentos mais marcantes da historia da ciência e foi fruto de um acaso. A penicilina proporcionou a cura de patologias infecciosas para as quais não havia qualquer terapêutica medicamentosa eficaz, salvou, e continua a salvar, milhões de vidas. O impacto de seu descobrimento foi além das vidas salvas, pois atraiu a atenção e investimentos para pesquisas de novos medicamentos (PEREIRA; PITA, 2018). Na década de 60 as pesquisas voltadas para a descoberta de novos medicamentos foi revolucionada graças à equação de Hammett, que permitiu o aparecimento das relações quantitativas entre estrutura e atividade. Desde então essas pesquisas passaram a depender menos do acaso (MONTANARI; PILLI,).

Um dos objetivos da Química Medicinal é a descoberta de novas moléculas com características de fármacos, o que é desafiador, pois o espaço de busca é discreto, não estruturado e enorme. Uma tática para a exploração deste espaço é por força bruta, porém o espaço de busca é muito vasto, o que impossibilita o teste de todas as possibilidades. Polishchuk, Madzhidov e Varnek (2013) estimaram que existem entre 10^{23} e 10^{60} moléculas com tais características que podem ser sintetizadas. Até hoje foram sintetizadas na ordem de 10^{8} moléculas. Estes são alguns dos motivos que fazem com que drogas custem milhões de dólares e levem anos para serem desenvolvidas. Tendo em vista esses problemas, pesquisadores têm direcionado esforços para a criação de ferramentas computacionais para auxiliar no longo processo de descoberta de drogas e potencialmente reduzir os custos de pesquisa e

desenvolvimento. Muitas dessas ferramentas utilizam o conceito de Inteligência Artificial para realizar tarefas consideradas mais subjetivas (LEELANANDA; LINDERT, 2016).

A Inteligência Artificial é uma área da Ciência da Computação que tem como objetivo simular o funcionamento do cérebro humano e suas habilidades, como pensar, compreender, agir, abstrair e criar. A habilidade de criar é uma habilidade notória do ser humano que está presente desde seus primórdios. Essa habilidade se perpetuou ao decorrer da história, permitindo a criação de obras de arte, composições musicais e diversas outras coisas. Com avanços recentes da computação, mais especificamente em Aprendizado de Máquina (do inglês *Machine Learning*, ML), agora é possível construir modelos e algoritmos capazes de compor músicas, pintar obras de arte e escrever textos coesos (FOSTER, 2019).

Técnicas tradicionais de ML têm como objetivo acrescentar uma nova informação a uma instância de dados, seja classificando-a ou inferindo o valor de algum de seus atributos (AWAD; KHANNA, 2015). Apesar de serem muito eficientes executando tarefas objetivas, elas não são capazes de criar novas instâncias. Já o Aprendizado de Máquina Construtivo (do inglês *Constructive Machine Learning*, CML) tem como objetivo, utilizando um conjunto de exemplos de instâncias existentes, criar novas instâncias pertencentes ao domínio estudado, visando que as mesmas atendam a certos critérios ou possuam determinados atributos. ML é utilizado em diferentes áreas para automatizar tarefas objetivas, como por exemplo a identificação de elementos em imagens. Agora CML é utilizado em diversas áreas, como por exemplo na geração de músicas, textos, imagens e, como será abordado nesse trabalho, na geração de moléculas (FOSTER, 2019).

ML passou a ser empregado apenas recentemente no processo de descobrimento de drogas. Apesar de diversos trabalhos se debruçarem nesta questão, a classe de algoritmos de classificação hierárquica ainda não foi empregada para auxiliar neste processo. Os algoritmos de classificação hierárquica tem como objetivo rotular instâncias através de uma taxonomia. Esta classificação pode auxiliar na carência de métricas dos métodos de CML para a criação de moléculas e auxiliar na expansão das bases de dados com taxonomias de moléculas.

1.2 Objetivos

Este trabalho tem seis objetivos principais:

- Criação de uma biblioteca para processar a base de dados ChEBI;
- Criação, treinamento e validação de um classificador hierárquico;
- Geração de moléculas usando métodos de CML;
- Comparação dos grupos gerados usando métricas de dissimilaridade da literatura;

- Classificação das moléculas geradas usando o classificador hierárquico;
- Proposição de uma métrica de dissimilaridade para grupos de moléculas.

1.3 Resumo dos resultados

O classificador hierárquico utilizado neste estudo apresentou um excelente desempenho no conjunto de testes. Os resultados obtidos demonstram a eficácia do modelo proposto em classificar corretamente as moléculas em suas respectivas classes da taxonomia ChEBI. Tais resultados são promissores e indicam que a abordagem hierárquica pode ser uma alternativa viável para a tarefa de classificação de moléculas.

As propriedades das moléculas geradas foram avaliadas e foi constatado que elas apresentaram características satisfatórias, onde a maioria não viola nenhum dos limitantes da regra de cinco. Isso indica que as moléculas têm maior probabilidade de serem desenvolvidas em fármacos bem-sucedidos, uma vez que a regra de cinco é um conjunto de diretrizes utilizadas na descoberta de fármacos para determinar se uma molécula tem as propriedades necessárias para ser um candidato a fármaco.

Os grupos de moléculas foram comparados em relação à diversidade externa e os resultados de dissimilaridade obtidos estão de acordo com a visualização dos grupos de moléculas, onde grupos próximos apresentaram baixa dissimilaridade e grupos distantes apresentaram alta dissimilaridade.

A distância hierárquica foi capaz de expressar as mesmas relações que a métrica encontrada na literatura, a diversidade externa, mas em uma fração do tempo. Isso demonstra sua eficiência e utilidade na análise e comparação de conjuntos de moléculas. Além disso, sua rapidez permite sua aplicação em conjuntos de moléculas muito grandes, o que pode ser útil em pesquisas na área de descoberta de medicamentos e química computacional. Essa abordagem pode ajudar na seleção de conjuntos de moléculas com propriedades específicas, economizando tempo e recursos na síntese e teste de compostos. A métrica proposta se mostrou mais relevante ao comparar grupos de tamanhos similares, isso se deve ao fato de ela não ser contida em um intervalo, podendo variar muito seus valores.

1.4 Organização do documento

O restante deste documento está organizado da seguinte maneira. O Capítulo 2 tem um papel fundamental na pesquisa, pois ele traz uma fundamentação teórica sobre o tema estudado. Nele, são apresentadas as diferentes representações de moléculas, que são fundamentais para a compreensão da análise molecular. Essas representações incluem desde as mais simples, como a fórmula de linha, até as mais complexas, como as representações em três dimensões. Além disso, o capítulo traz conceitos de aprendizado de máquina, que é a base teórica utilizada

nesta pesquisa para se gerar moléculas e fazer previsões sobre as propriedades moleculares. São apresentados alguns modelos de aprendizado de máquina utilizados em problemas de análise molecular, como redes neurais e árvores de decisão.

Ademais, o Capítulo 2 também inclui uma revisão dos trabalhos relacionados ao tema desta pesquisa, o que é importante para contextualizar o estudo dentro do contexto científico. Essa revisão permite que sejam identificados os principais avanços, bem como as lacunas de conhecimento existentes no tema, além de possibilitar a identificação de oportunidades para novas pesquisas.

O Capítulo 3 é de extrema importância para a compreensão desta pesquisa, pois é nele que serão abordados os aspectos relacionados à proposta, aos materiais e à metodologia utilizados. É neste capítulo que será possível entender como a pesquisa foi conduzida, quais foram os procedimentos adotados e quais foram as ferramentas utilizadas para se chegar aos resultados apresentados. Além disso, o capítulo traz uma contribuição para o campo científico, ao apresentar a métrica de dissimilaridade entre grupos de moléculas proposta por este trabalho, a distancia hierárquica. Essa métrica se mostra relevante para se avaliar a distância entre diferentes grupos de moléculas em uma fração do tempo quando comparada à métrica encontrada na literatura.

O Capítulo 4 apresenta os resultados desta pesquisa. Inicialmente, o capítulo analisa alguns dos descritores das moléculas geradas e visualiza todos os descritores reduzidos em dois espaços diferentes, o que permite uma análise mais clara das características dos grupos de moléculas. Além disso, o capítulo avalia o classificador hierárquico utilizado na pesquisa e apresenta informações sobre as taxonomias dos grupos de moléculas geradas.

Além disso, o Capítulo 4 também discute as diversidades internas e externas dos grupos de moléculas geradas, que são medidas importantes para avaliar o quão dissimilares os grupos de moléculas são. Por fim, o capítulo discute os resultados obtidos através da métrica de dissimilaridade proposta e comparados à diversidade externa. Essa comparação é importante para verificar a eficácia da métrica proposta. Portanto, o Capítulo 4 é uma parte essencial desta pesquisa, uma vez que apresenta todos os resultados obtidos e permite uma análise mais profunda dos dados coletados.

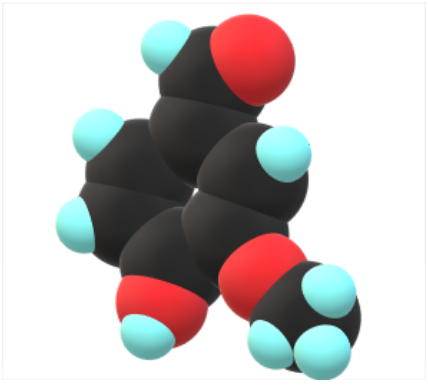
Por fim, o Capítulo 5 relembra alguns pontos abordados pela pesquisa, recapitula os resultados, expõe sua importância e discute os possíveis trabalhos futuros.

2 Fundamentação teórica

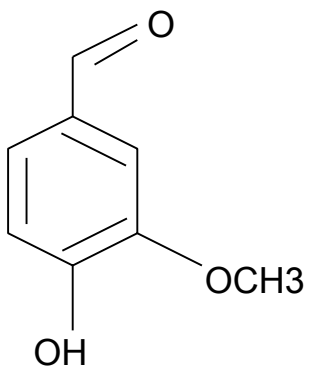
Este capítulo apresenta conceitos teóricos necessários para a realização da pesquisa. Inicialmente alguns tipos de representações de moléculas são discorridos na Seção 2.1. Logo após os conceitos de aprendizado de máquina construtivo e classificação hierárquica também são apresentados nas Seções 2.3 e 2.4 respectivamente. Aqui também são apresentados métodos de aprendizado de máquina utilizados na tarefa de geração de sequências nas Seções 2.3.1, 2.3.2, 2.3.3 e 2.3.4. Os conceitos de semelhança entre moléculas, diversidade interna e externa são desenvolvidos na Seção 2.5. E por fim, na Seção 2.6, os trabalhos relacionados são apresentados.

2.1 Representações de moléculas

Existem diversas maneiras de se representar a estrutura química de uma molécula, como por exemplo as ilustrações não necessariamente computacionais das Figuras 1a e 1b. A Figura 1a é a representação tridimensional da vanilina, onde seus átomos são representados por esferas cuja cor representa um elemento, e suas ligações são representadas pela proximidade dos átomos. Já a Figura 1b é a representação em fórmula de linha da vanilina, que retrata suas ligações com uma linha entre os átomos, e seus átomos com letras. No caso do carbono, esse pode ser representado apenas por um vértice entre duas linhas (VOLLHARDT; SCHORE, 2013).



(a) Representação tridimensional da vanilina.



(b) Vanilina em fórmula de linha.

Retirado de: Desenvolvida pelo autor.

Figura 1 – Representações da vanilina.

Existem também diversas formas de se representar moléculas computacionalmente, sendo que essas representações podem ser bidimensionais ou tridimensionais. Dentre as representações computacionais, a representação ideal, além de capturar as informações essenciais da estrutura

molecular, é também única, implicando que só há uma maneira de representar uma determinada molécula. É também inversível, em que cada representação só pode ser interpretada como uma única molécula, e compacta, minimizando o espaço ocupado para ser armazenada. Essas características, apesar de serem desejadas, não são obrigatórias (ELTON et al., 2019).

Este trabalho aborda duas representações bidimensionais, o Sistema de Registros Moleculares Simplificados em Linha (do inglês *Simplified Molecular Input Line Entry System, SMILES*) e *fingerprints*.

2.1.1 SMILES

SMILES é uma representação inspirada na fórmula de linha e representa as moléculas com uma série de caracteres. Os átomos são representados pelas letras que os simbolizam na tabela periódica, o átomo de hidrogênio é omitido na maioria das vezes. Os elementos que utilizam mais de um caractere para serem representado na tabela periódica são representados entre colchetes na codificação SMILES. As ligações duplas e triplas pelos símbolos $=$ e $\#$ respectivamente, as ligações simples são omitidas e subentendidas pela concatenação de dois símbolos. Apesar de uma cadeia de carácteres ser essencialmente unidimensional, SMILES é capaz de capturar a essência bidimensional das moléculas, usando parênteses para indicar ramificações na estrutura, e pares de números para representar seus ciclos. A representação SMILES não é única, pois existem diversas cadeias que representam a mesma molécula, mas é inversível e densa, usando menos caracteres para moléculas menores. Uma vantagem dessa codificação é a possibilidade da interpretação humana (WEININGER, 1988).

Um exemplo da codificação SMILES é a vanilina, que será explicada em relação à Figura 1b. O primeiro passo da codificação é escolher arbitrariamente um átomo da cadeia. Existem convenções de quais átomos iniciar a cadeia, porém essencialmente qualquer átomo pode ser escolhido. O primeiro átomo escolhido para a codificação da vanilina é o carbono do anel aromático localizado mais acima na imagem. Por se tratar de um anel aromático, os carbonos são representados em letra minuscula e suas ligações duplas entre si são omitidas. Este átomo além de participar de um ciclo, possui uma ramificação. O caractere 1 é usado para representar o inicio do primeiro (e único) ciclo. Nesta primeira ramificação há um átomo de carbono ligado a um oxigênio, o que se traduz para $(C = O)$. Os próximos passos consistem em concatenar os átomos seguintes na cadeia. Como se trata de um ciclo é preciso escolher um sentido, e o sentido horário relativo à figura foi escolhido. O próximo átomo de carbono não possui ramificações, e o subsequente possui uma ramificação com um oxigênio e um carbono, o que se traduz como $cc(OC)$. O próximo carbono também possui uma ramificação e se transcrevem como $c(O)$. Os últimos dois átomos não possuem ramificações e concluem o ciclo. Por tanto, uma das representações válidas na sintaxe SMILES para essa molécula é $c1(C = O)cc(OC)c(O)cc1$.

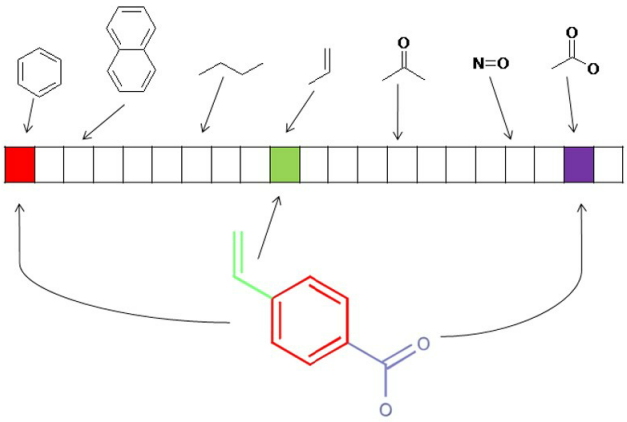
É importante lembrar que a sintaxe da representação deve ser respeitada para que a

cadeia SMILES seja válida. Qualquer caractere que não esteja de acordo com a sintaxe pode tornar a cadeia SMILES inválida.

É possível que uma cadeia SMILES seja inválida mesmo que esteja correta sintaticamente. Isso ocorre quando a estrutura molecular que ela descreve não é possível fisicamente. Por exemplo, uma cadeia SMILES pode descrever um ciclo muito grande que não pode ser formado na realidade devido a restrições estéricas ou outras limitações físicas. Em tais casos, a cadeia SMILES é considerada inválida, pois não representa uma estrutura molecular que possa realmente existir na natureza. Portanto, é importante considerar não apenas a sintaxe da cadeia SMILES, mas também a sua plausibilidade física ao usá-la para descrever estruturas moleculares.

2.1.2 Fingerprints

A representação *fingerprints* utiliza um vetor de tamanho fixo, cada uma de suas posições indica a presença ou ausência de uma estrutura. Ela pode também contar as aparições de cada estrutura molecular. Um exemplo é apresentado na Figura 2, em que a molécula a ser codificada possui 3 estruturas, que correspondem às posições 1, 10 e 20 do vetor. Essas posições assumem o valor 1 enquanto as demais recebem o valor 0. A representação *fingerprints* é única, porém não é inversível, e é geralmente esparsa. Ainda assim é útil como descritor codificando características importantes das moléculas. É útil também para comparar a similaridade entre moléculas. *Fingerprints* podem ser utilizadas para estimar se uma molécula interage com determinados receptores biológicos (CAO; LIANG, 2012).



Retirado de: <https://www.researchgate.net/publication/235919348_manual_for_chemopy>

Figura 2 – Ilustração da codificação de *fingerprints*.

2.1.3 Representação por grafos

A representação por grafos é uma ferramenta poderosa no campo da química. Ela usa vértices para representar átomos e arestas para representar ligações, proporcionando um

alto nível de cobertura do espaço químico. Os grafos também são capazes de capturar várias características químicas, como estereoquímica, aromaticidade e hibridização. Devido à sua natureza intuitiva e concisa, argumenta-se que a representação gráfica é a maneira mais eficaz de comunicar estruturas. Além disso, a representação em grafos permite a aplicação de teoria dos grafos e outros algoritmos matemáticos para analisar e prever propriedades e comportamentos químicos.

Esta representação se mostra ser relevante na tarefa de geração pois durante este processo, as moléculas parcialmente geradas podem ser interpretadas como sub-estruturas, enquanto nas representações baseadas em texto, como por exemplo a representação SMILES, os métodos podem gerar cadeias SMILES invalidas antes de gerar uma válida (YOU et al., 2018).

2.2 Propriedades das moléculas

Diferentes moléculas possuem diferentes propriedades e funções. As propriedades (também chamadas de descritores) mais analisadas no estudo de moléculas são seu peso molecular, solubilidade em água, número de aceptores de ligações de hidrogênio e o número de doadores de ligações de hidrogênio. A área superficial polar, número de ligações rotacionáveis e o número de anéis aromáticos também são descritores usados para a análise (BICKERTON et al., 2012). As moléculas satisfazem o princípio de propriedades similares, ou seja, moléculas com funções e estruturas similares têm vetores de descritores próximos. Este é um ponto chave para a comparação entre moléculas (MITCHELL, 2014a).

Existem diferentes maneiras de identificar uma molécula com características de fármacos. Uma proposta foi a Regra de Cinco de Lipinski (do inglês *Lipinski's Rule of Five*), que consiste em uma série de regras, como por exemplo a limitação do peso a 500 Daltons, ou o logaritmo do coeficiente de partição (solubilidade) a 5. É muito provável que a molécula não tenha uma boa absorção no corpo caso infrinja duas ou mais dessas regras. Porém ao menos 6% das drogas de ingestão oral violam ao menos duas das regras, estas são classificadas igualmente às outras moléculas com baixa absorção. Outra proposta para identificar e avaliar é a Estimativa de Semelhança com Drogas (do inglês *Quantitative Estimate of Druglikeness*, QED). A Regra de Cinco de Lipinski e as propriedades citadas são utilizadas para cálculo do QED, e este cálculo resulta em um número entre 0 e 1, que representa a probabilidade da molécula ser semelhante a drogas (BICKERTON et al., 2012).

Outra maneira de identificar moléculas candidatas é utilizar um modelo de aprendizado de máquina para estimar a Relação Quantitativa Estrutura-Atividade (do inglês *Quantitative Structure-Activity Relationship*, QSAR). QSAR também é um número escalar entre 0 e 1 que representa a probabilidade de uma molécula ser ativa em relação a um receptor biológico (MITCHELL, 2014b).

Por fim, uma característica importante e desejada de uma molécula criada virtualmente é que ela possa ser sintetizada. Para predizer o quão fácil é sintetizar uma molécula é empregada a Pontuação de Acessibilidade Sintética (do inglês *Synthetic Accessibility Score*, SAscore). Essa medida leva em consideração a complexidade da molécula (ERTL; SCHUFFENHAUER, 2009).

2.3 Aprendizado de máquina construtivo

Em Ciência da Computação, ML é um ramo da Inteligência Artificial que, como definido por Awad e Khanna (2015), aplica algoritmos para sintetizar relações intrínsecas em um conjunto de dados e informações. ML tem como principal objetivo prever eventos ou cenários que são desconhecidos pelo computador, para isso é utilizado um modelo que manifesta o aprendizado do algorítimo de ML. Algorítimos de ML podem ser divididos com base no tipo de aprendizado durante a fase de treinamento. Apesar de existirem diversos tipos de aprendizado, aqui serão divididos em dois grupos, os que utilizam aprendizado supervisionado e os que utilizam aprendizado não supervisionado.

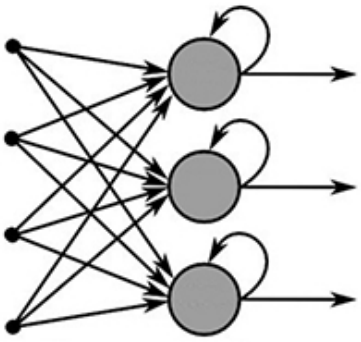
Os métodos de CML são sensivelmente diferentes dos métodos de ML tradicionais. Essa diferença se deve principalmente ao objetivo dos métodos. Os métodos tradicionais têm como objetivo classificar, agrupar ou inferir certas características dos dados, enquanto os de CML têm como objetivo criar instâncias semelhante às da base de dados. Foster (2019) cita algumas restrições que modelos de CML devem seguir. As instâncias criadas não devem ser idênticas às instâncias já existentes, pois não seria possível inferir que o modelo adquiriu a habilidade de criar. E elas não devem ser tão diferentes a ponto de não se parecerem membros do grupo de instâncias da base de dados, pois neste caso não seria possível inferir que o modelo foi capaz de aprender as regras intrínsecas que formam o conjunto alvo. A segunda regra é mais difícil de se cumprir, pois normalmente as características das instâncias são dependentes entre si, e para gerar boas instâncias, o modelo deve ser capaz de sintetizar essas relações e dependências. Os algoritmos de CML se encontram na categoria de algoritmos que utilizam aprendizado não supervisionado, pois durante sua fase de treinamento não necessitam que os dados sejam rotulados.

2.3.1 Redes neurais recorrentes

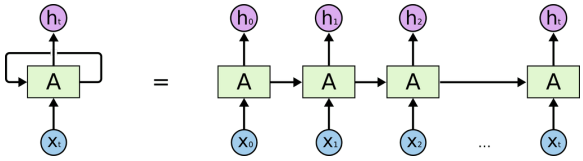
Redes Neurais completamente conectadas, ao processar uma série de dados não levam em consideração uma possível relação entre os dados e sua ordem, processando-os independentemente. Redes Neurais Recorrentes (do inglês *Recurrent Neural Networks*, RNNs) são um tipo de rede neural capaz de processar a ideia de contexto e ordem dos dados, com uma estrutura de estado oculto (do inglês *Hidden State*) que é capaz de armazenar informações que foram processadas, conectando dados processados anteriormente com dados ainda a serem processados. A Figura 3a apresenta o estado oculto com um neurônio alimentando a si mesmo.

A Figura 3b mostra um neurônio A processando dados de maneira sequencial, bem como a influência do estado oculto nas iterações seguintes (GOYAL; PANDEY; JAIN, 2018).

(a) Rede neural recorrente.



(b) Rede neural recorrente através do tempo.

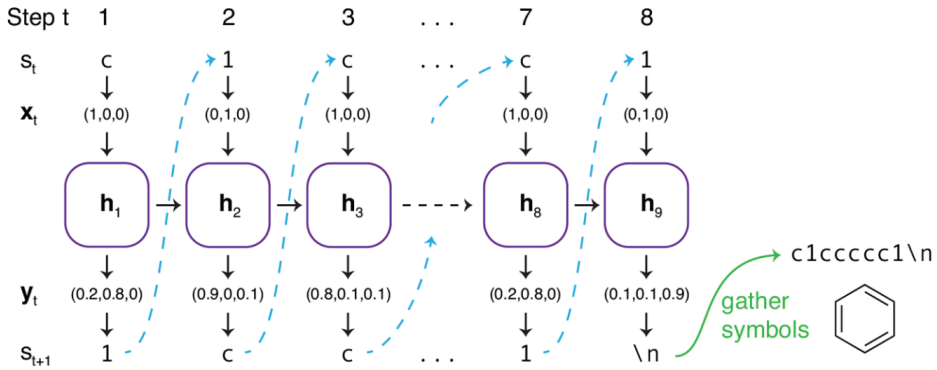


Retirado de: <http://deeplearningbook.com.br/>

Figura 3 – Representações do funcionamento de uma rede neural recorrente.

RNNs podem ser aplicadas para diferentes tarefas, como por exemplo classificação e regressão. Será abordado aqui como elas podem ser empregadas para criar cadeias de caracteres, mais especificamente SMILES que são explicados na Seção 2.1.

Para aplica-las na tarefa de geração é preciso definir um dicionário de todos os elementos que podem ser gerados. As camadas de entrada e saída da RNN são definidas pelo tamanho deste dicionário. Como é possível observar na Figura 4, para dar início ao processo de geração se processa um elemento aleatório presente no dicionário ou o elemento especial que simboliza o início de uma cadeia. Então o modelo gerará como saída as probabilidades de elementos serem os próximos a compor a cadeia. O próximo passo é a escolha do elemento com maior probabilidade ou amostrando um elemento de acordo com as probabilidades geradas pela rede. Por fim o elemento escolhido é adicionado à cadeia, e esta mesma cadeia é processada até que seja gerado um número de elementos pré determinado ou o elemento especial que indica o fim de uma cadeia.

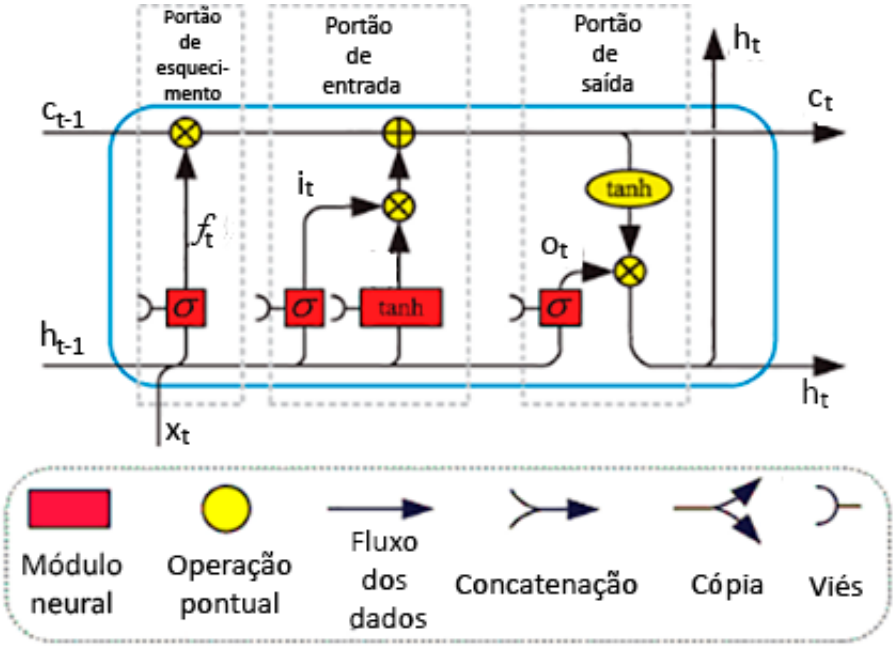


Retirado de: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5785775/>

Figura 4 – Processo de geração de SMILES por uma Rede neural recorrente.

2.3.1.1 Long Short-Term Memory

Long Short-Term Memory (LSTM) são uma adaptação das RNNs comuns. Esta adaptação em geral tem melhor desempenho que as RNNs comuns, pois RNNs comuns sofrem de memória de curto prazo, podendo perder a relação entre informações distantes ao analisarem uma sequência muito longa. A arquitetura das redes LSTM possui três portões que as diferenciam das RNNs comuns. Para melhor compreendimento da Figura 5, que ilustra a arquitetura de uma célula LSTM, é preciso ter o entendimento de seus símbolos: $\otimes$ indica a multiplicação posição a posição dos valores de dois vetores, $\oplus$ se refere à soma entre vetores, e os retângulos vermelhos representam módulos neurais com suas respectivas funções de ativação representadas em seu interior.



Adaptado de: <https://doi.org/10.1162/neco_a_01199>

Figura 5 – Arquitetura LSTM.

Como é possível observar na Figura 5, sempre que uma nova informação entra em uma célula, ela é concatenada com o resultado anterior, esta é copiada e então suas cópias são utilizadas para o cálculo dos portões. Cada portão possui um módulo neural com conjuntos de pesos que são apresentados nas Equações 2.1, 2.2 e 2.3 pelos símbolos W, U e b, sendo esses respectivamente os pesos multiplicados pela entrada, os pesos que multiplicam o resultado anterior e o viés.

Cada portão possui uma função diferente, o primeiro portão é o portão de esquecimento (do inglês *forget gate*). Expresso pela Equação 2.2, este portão é utilizado para alterar o estado oculto da célula, filtrando informações relevantes dos estados anteriores. O segundo portão é o portão de entrada (do inglês *input gate*), estabelecido pela Equação 2.1. Este portão é responsável por filtrar os dados de entrada para que estes sejam incorporados no estado oculto

da célula. A Equação 2.4 apresenta o cálculo do novo estado oculto como a soma de duas partes, a primeira sendo o estado oculto anterior modificado pelo portão de esquecimento, e a segunda parte como a entrada previamente processada por um módulo neural e filtrada pelo portão de entrada. O ultimo portão, como é descrito na Equação 2.3, é o portão de saída (do inglês *output gate*), cuja função é filtrar o novo estado oculto da célula para gerar um novo resultado, assim como expressa a Equação 2.5 (YU et al., 2019).

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{2.1}$$

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{2.2}$$

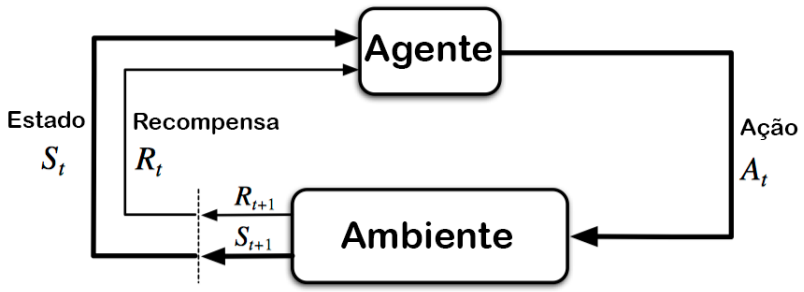
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{2.3}$$

$$c_t = (f_t \otimes c_{t-1}) + (i_t \otimes \sigma_h(W_c x_t + U_c h_{t-1} + b_c)) \tag{2.4}$$

$$h_t = o_t \otimes \sigma_h(c_t) \tag{2.5}$$

2.3.2 Aprendizado por reforço

Aprendizado por Reforço (do inglês *Reinforcement Learning*, RL) é uma área de estudos em ML. A estrutura do RL pode ser dividida entre o agente e o ambiente. O agente possui uma política para tomada de ações, e para cada estado uma política determina uma ação. O ambiente além de fornecer o estado em que o agente se encontra, provê um número escalar para cada ação ou para uma ação específica, esse número escalar é chamado de recompensa. O agente então é capaz de atualizar sua política com base na recompensa a fim de maximizá-la.



Adaptado de: <https://im.perhapsbay.es/kb/policy-gradients-in-a-nutshell>

Figura 6 – Laço de interação entre o agente e o ambiente.

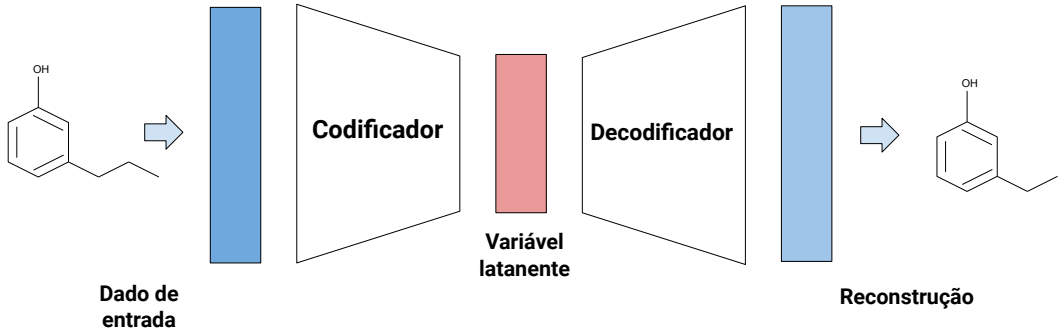
Para aplicar RL junto com RNNs basta considerar a RNN como o agente, a cadeia gerada até o momento como o estado e a recompensa como uma característica da molécula gerada. Neste caso, a política do agente é parametrizada pelos pesos da RNN e definida pela

distribuição de probabilidades do próximo elemento. A ação é amostrada desta distribuição e o estado é alterado, concatenando o elemento escolhido ao fim da estado atual (KAPOOR, 2018).

RL em teoria pode resolver qualquer problema, porém existem dois grandes problemas que envolvem RL, a dificuldade de modelar uma função de recompensa e a instabilidade de treino devido ao grande número de estados e ações possíveis. Uma dificuldade de aplicar RL na geração de SMILES é a dificuldade de atribuir uma recompensa a todas as ações do agente, pois só é possível calcular as propriedades de uma cadeia SMILES completa.

2.3.3 Variational Autoencoder

O *Autoencoder* (AE), como é possível observar na Figura 7, é uma estrutura dividida em duas partes, o codificador e o decodificador. O objetivo do codificador é transformar um dado de entrada, reduzindo sua dimensionalidade e representando somente suas características mais importantes. O resultado dessa codificação é chamado de variável latente (do inglês *latent variable*), e a função do decodificador é reconstruir o dado de entrada usando essa variável latente, o que na prática não acontece perfeitamente pois o processo apresenta perdas. No entanto um bom modelo pode ser capaz de recriar uma boa aproximação. Tanto o codificador quanto o decodificador podem ser implementados com estruturas de Redes Neurais, como por exemplo RNNs e Redes Neurais Convolucionais.



Retirado de: Desenvolvida pelo autor.

Figura 7 – Estrutura do *autoencoder*.

Os AEs têm tamanhos fixos de entrada e saída fixa, por isso são capazes de processar apenas dados de tamanhos pré-definidos. Para processar sequências de tamanho indefinido é utilizada uma adaptação dos AEs, o *Recurrent AutoEncoder*(RAE). Nessa rede, o codificador e o decodificador são RNNs. O codificador processa o dado de entrada, porém descarta suas saídas, usando como variável latente seu estado oculto. O decodificador por sua vez incorpora esta variável latente em seu estado oculto e produz uma cadeia caracterizada por este estado oculto (SUSIK, 2021).

Quando se trata de gerar novos dados, os AEs não são adequados. Assim, uma possível abordagem é navegar no espaço latente com valores aleatórios ou próximos aos valores de dados conhecidos, e aplicar o decodificador a esse valor, gerando assim uma nova instância, porém os AEs não garantem que a mesma seja relevante e não seja apenas ruído. Os codificadores automáticos variacionais (do inglês *Variational AutoEncoder*, VAE) têm seu funcionamento muito semelhante aos AEs, porém são melhores na tarefa de geração. Ao codificar os dados, ao invés do codificador ter como saída uma variável latente, ele tem como saída uma distribuição normal de variáveis latentes, representada por duas variáveis de mesma dimensão do espaço latente, a primeira descreve a média e a segunda o desvio padrão da distribuição, e então uma amostra dessa distribuição é decodificada (DOERSCH, 2016).

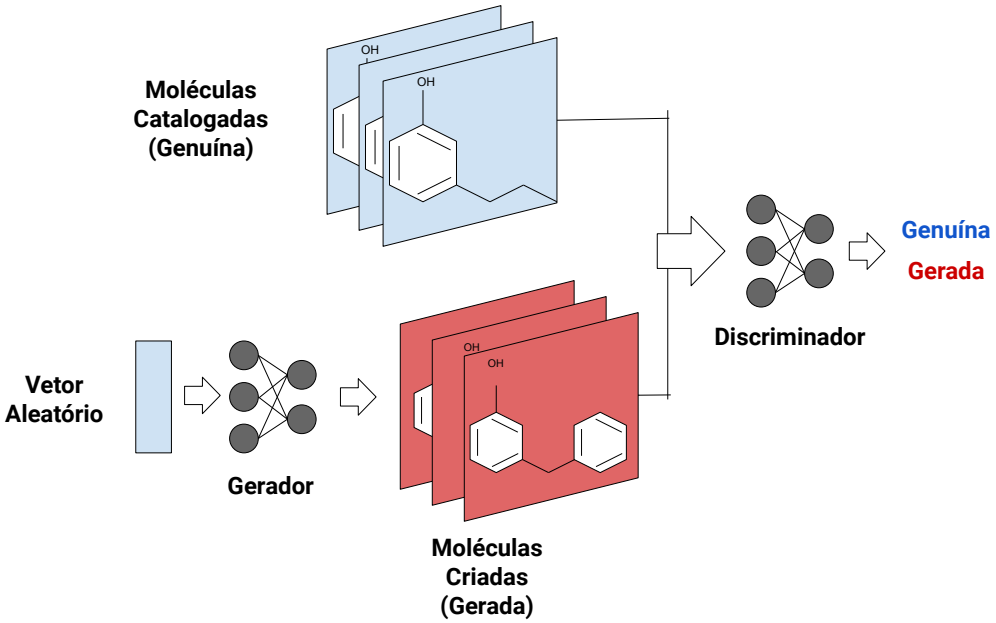
Para a tarefa de geração de SMILES, os *Variational Recurrent AutoEncoders* (VRAEs) são mais adequados pois combinam os conceitos dos VAEs e dos RAEs para gerar sequências de tamanho variável (BOWMAN et al., 2015).

2.3.4 Redes neurais adversárias generativas

As Redes Adversárias Generativas (do inglês *Generative Adversarial Networks*, GANs), como ilustra a Figura 8, são divididas em duas partes, o gerador e o discriminador, onde o gerador cria novas instâncias de dados a partir de um vetor aleatório (também chamado de variável latente), e essas instâncias geradas são colocadas junto a outras instâncias já existentes. Então o discriminador as classifica como genuína (rotulando com o valor 1) ou gerada (rotulando com o valor 0), com o objetivo de minimizar o erro de classificação. O gerador por sua vez tem como objetivo maximizar este erro, gerando instâncias cada vez mais semelhantes às existentes (GOODFELLOW et al., 2014). A função da Equação 2.6 remete aos objetivos do gerador e do discriminador. Esta função avalia o desempenho do discriminador e é dividida em duas partes, a que diz respeito à classificação dos dados genuínos e a classificação dos dados gerados. Ambas as partes são acompanhadas pelo simbolo da expectância, que retrata o valor médio esperado ao realizar-se diversas amostragens. A primeira parte é descrita por $\mathbb{E}_{x\sim p_{data}(x)}[logD(x)]$ onde $p_{data}(x)$ indica a base de dados e x é uma instância amostrada desta base. $D(x)$ indica a predição do discriminador em relação à instância x. A segunda parte é expressa por $\mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))]$, sendo $p_z(z)$ o espaço latente e z uma variável amostrada deste espaço. $G(z)$ expressa uma instância criada pelo gerador e esta é avaliada pelo discriminador.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))] \tag{2.6}$$

Alguns problemas surgem ao aplicar GANs para a geração de sequências. O primeiro se deve ao fato de que GANs são projetadas para trabalhar com dados contínuos. Com dados contínuos é possível calcular o gradiente do erro e atualizar os pesos do gerador, porém quando



Retirado de: Desenvolvida pelo autor.

Figura 8 – Redes neurais adversarias generativas.

se trata de dados discretos não há um cálculo direto do gradiente. Outro problema é que GANs tradicionais são projetadas para gerar dados com tamanho fixo, o que não corresponde com a natureza das cadeias SMILES.

Para a solução destes problemas, são empregadas as Redes Adversárias Generativas de Sequências (do inglês *Sequence Generative Adversarial Networks*, SeqGANs). SeqGANs se diferem das GANs tradicionais, pois seu gerador é uma RNN e é tratado como um agente de aprendizado por reforço. Para utilizar o gerador como um agente de aprendizado por reforço, todos os elementos gerados até um determinado momento são interpretados como o estado, o próximo elemento a ser gerado é entendido como a ação do agente, e a classificação do discriminador é utilizada como recompensa (NAIR, 2018; YU et al., 2017).

Um problema de utilizar aprendizado por reforço ao gerar SMILES é que uma recompensa só pode ser dada ao gerar uma sequência completa. SeqGANs utilizam a Busca em Árvore de Monte Carlo (do inglês *Monte Carlo Tree Search*, MCTS) para estimar uma recompensa para trechos de sequências e recompensar o modelo mais de uma vez por cadeia gerada. MCTS cria uma árvore de busca onde cada nó possui um valor associado e corresponde a um simbolo, e a profundidade da árvore se relaciona ao tamanho da cadeia. A árvore se inicia apenas com o nó raiz e cresce gradualmente ao repetir os 4 passos: seleção, expansão, simulação e retro-propagação. O passo de seleção consiste em percorrer a árvore da raiz até um nó folha. A Expansão diz respeito ao processo de aumentar o tamanho da árvore, adicionando filhos ao nó escolhido no passo anterior. Então a simulação ou uma avaliação do quão apto esse novo

nó é. E por fim esse valor é retro-propagado para atualizar os valores dos nós que constituíram o caminho percorrido até este nó (YANG et al., 2017).

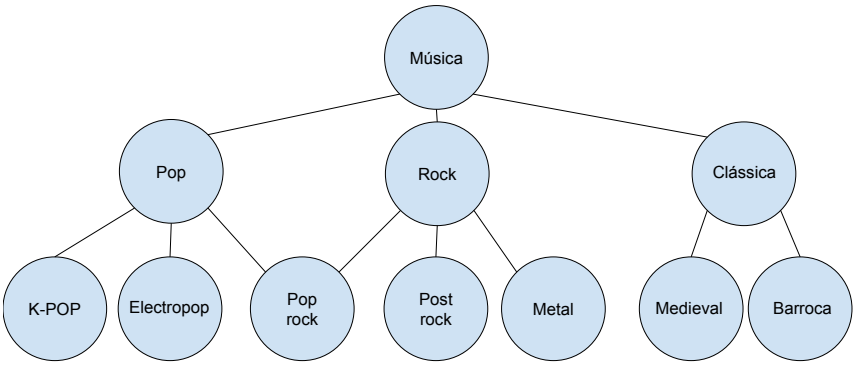
2.3.5 Redes Neurais de Grafo

As redes neurais de grafo são uma classe de modelos de aprendizado profundo que são projetados para trabalhar com dados estruturados em forma de grafos. Ao contrário das redes neurais tradicionais, que operam em dados estruturados em forma de matrizes ou vetores, as redes neurais de grafo são projetadas para trabalhar diretamente com dados em forma de grafos, permitindo que elas capturem a estrutura de interconexões complexas entre diferentes elementos em um conjunto de dados (YOU et al., 2018).

2.4 Classificação hierárquica

Os algoritmos de ML são tradicionalmente de classificação binária ou multi-classe, onde todos os rótulos estão em um mesmo nível. Esse tipo de classificação também é conhecida como classificação plana. Esses algoritmos são muito úteis, porém não abordam todos os problemas de classificação da vida real. Esses problemas podem se tratar de problemas de classificação hierárquica, onde os rótulos estão organizados em uma hierarquia e os dados são classificados através de uma taxonomia.

As taxonomias podem ser representadas por meio de uma árvore ou por meio de um Grafo Acíclico Direcionado (do inglês *Directed Acyclic Graph*, DAG). Um exemplo de taxonomia é a classificação de músicas. A Figura 9 ilustra uma DAG que constitui uma taxonomia de gêneros musicais, onde gêneros possuem sub-gêneros, e sub-gêneros podem estar contidos em mais de um gênero. Por utilizar rótulos, os algoritmos de classificação hierárquica se encontram na categoria de algoritmos que utilizam aprendizado supervisionado.

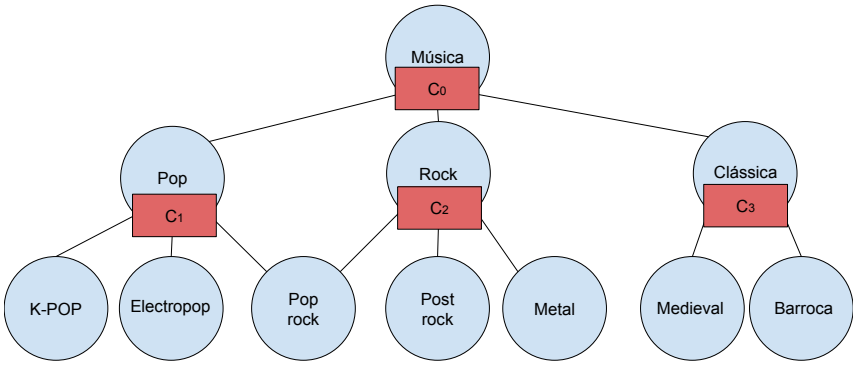


Retirado de: Desenvolvido pelo autor.

Figura 9 – Exemplo de taxonomia de gêneros musicais.

Para realizar a classificação hierárquica é possível utilizar duas abordagens, a global e a local. Uma estratégia muito comum baseada na abordagem local consiste em percorrer a

hierarquia a partir do vértice raiz com o auxílio de diversos classificadores. Esses classificadores estão presentes em cada vértice e classificam o próximo rótulo na hierarquia, assim como retratado na Figura 10. Cada classificador é treinado apenas com os dados referentes à classes que pode prever. O classificador no primeiro nível é treinado com todo o conjunto de dados. Quanto mais profundo um classificador se encontra na hierarquia, mais especifico e menor é o conjunto de dados usado para o seu treinamento. Caso se trate de uma classificação hierárquica completa, esse processo se repete até que se atinja um vértice folha. Caso se trate de uma classificação hierárquica parcial, o processo pode ser interrompido caso um dos classificadores classifique os possíveis próximos rótulos como inadequados. A abordagem local pode ser computacionalmente cara dependendo do tamanho da taxonomia, pois quanto mais vértices a árvore ou DAG tiver, mais classificadores serão precisos para a classificação. Outro ponto negativo da abordagem local é a propagação de erros, se um classificador cometer um erro em uma fase inicial do processo, os classificadores seguintes estarão mais suscetíveis a cometer erros. A abordagem de classificação global utiliza um único classificador mais complexo. Este classificador global considera toda a hierarquia de uma vez para realizar a classificação (SILLA; FREITAS, 2011).



Retirado de: Desenvolvido pelo autor.

Figura 10 – Exemplo de abordagem local para classificação hierárquica.

2.5 Métricas de diversidade molecular

A comparação entre duas moléculas pode ser realizada pela similaridade de Tanimoto, que determina o quão similares duas moléculas são a partir das *fingerprints* de duas moléculas a e b. Como é possível ver na Equação 2.7, a similaridade de Tanimoto é definida pela razão da interseção e a união dos dois vetores, sendo por definição um número entre 0 e 1.

$$T_s(a,b) = \frac{a \cap b}{a \cup b} \tag{2.7}$$

Essa métrica pode ser reinterpretada como a distância de Tanimoto, definida como o complemento da similaridade de Tanimoto na Equação 2.8.

$$T_d(a,b) = 1 - T_s(a,b) \tag{2.8}$$

Comparar conjuntos de moléculas é útil para validar o desempenho dos geradores. Pode-se comparar o conjunto gerado consigo mesmo para determinar sua diversidade interna, e comparar o conjunto de moléculas gerado pelo modelo e o conjunto usado no treinamento do modelo para determinar sua diversidade externa. A diversidade interna é definida pela média das distâncias de todas as moléculas entre si. A Equação 2.9 mostra como é realizado o cálculo da diversidade interna de um conjunto A de moléculas.

$$D_I = \frac{1}{|A|^2} \sum_{(x,y) \in A \times A} T_d(x,y) \tag{2.9}$$

A diversidade externa indica o quão diferentes são dois conjuntos de moléculas A e B. Como é possível observar na Equação 2.10, é definida pela média das distâncias entre todas as moléculas do conjunto A com as moléculas do conjunto B.

$$D_E(A,B) = \frac{1}{|A||B|} \sum_{(x,y) \in A \times B} T_d(x,y) \tag{2.10}$$

Um problema a ser considerado ao utilizar a diversidade externa é a diferença entre os tamanhos dos conjuntos A e B, quanto maior essa diferença, menor a confiança dessa métrica. Uma alternativa é comparar a diversidade interna dos conjuntos A e B (BENHENDA, 2018).

2.6 Trabalhos relacionados

Foi realizado um levantamento de trabalhos recentes que empregaram CML na geração de moléculas, discorrendo também sobre as dificuldades envolvendo funções de recompensa, representações computacionais das moléculas, e os métodos de avaliação dos métodos. Apesar de existirem diversos trabalhos na área de CML, definir o estado da arte é uma tarefa difícil pois os trabalhos usam métricas diferentes para avaliar seu desempenho. Nesta seção serão apresentados alguns trabalhos que usam CML para a geração de moléculas. Também serão mencionados trabalhos que utilizam CML para outros fins que aplicam os modelos citados (ELTON et al., 2019).

Em (BJERRUM; THRELFALL, 2017) RNNs foram empregadas para geração de moléculas, assim como explicado na Seção 2.3.1. Para o treinamento foram utilizadas duas bases de dados. Para a análise, dois conjuntos moléculas foram geradas e suas propriedades foram estimadas. Foi realizada uma comparação entre as propriedades das moléculas geradas e das

moléculas das bases de dados, deixando evidente as semelhanças entre as distribuições dos dados.

Gupta et al. (2018) também utilizaram RNNs para a tarefa de geração, porém é sensivelmente diferente do trabalho anterior pois utiliza a técnica de *transfer learning* empregada em uma RNN, onde a RNN é treinada em uma base de dados mais genérica, e depois retreinada em outra base mais especifica. A base CHEMBL22 foi utilizada para o treinamento prévio, e outras três bases de moléculas com interações especificas para o refinamento dos modelos. Para a avaliação das moléculas geradas foi empregado a Análise de Componentes Principais (do inglês *Principal Component Analysis*, PCA). A PCA foi ajustada às propriedades das moléculas da base de dados, e as dimensões das propriedades das moléculas foram reduzidas para realizar a comparação. Foi utilizada também a similaridade de Tanimoto para determinar vizinhos mais próximos de moléculas catalogadas .

Além de utilizar métodos vistos nos trabalhos anteriores, Segler et al. (2018) utilizaram um modelo TPM para prever se uma molécula gerada é ativa ou não em relação a um alvo biológico. Este modelo preditor foi utilizado para filtrar a base de dados original, e então aperfeiçoar a RNN, treinando-a em uma base de dados com apenas moléculas consideradas ativas em relação ao alvo. Este modelo foi utilizado também no aumento artificial dos dados, onde moléculas geradas previstas como ativas foram adicionadas à base de dados de treino para o retreinamento da RNN.

O RL tem sido objeto de investigação extensiva nos últimos anos devido às suas potenciais aplicações em diversas áreas, incluindo a descoberta de medicamentos. Em um estudo recente, a combinação de RL e QSAR foi explorada como função de recompensa para identificar moléculas ativas contra o Receptor de Dopamina Tipo 2 (do inglês *Dopamine Receptor Type 2*, DRD2) (OLIVECRONA et al., 2017). O estudo utilizou uma RNN e modelo QSAR para gerar estruturas que são previstas como ativas contra DRD2. Notavelmente, os resultados mostraram que mais de 95% das estruturas geradas pela RNN foram previstas como ativas contra DRD2, demonstrando o potencial do RL em auxiliar os esforços de descoberta de medicamentos. Os achados deste estudo tem implicações significativas para o desenvolvimento de novos medicamentos pois demonstra que um modelo QSAR pode ser usado como função objetivo para treinar agentes.

Uma contribuição importante é a aplicação de MCTS na tarefa de geração de SMILES. MCTS é comentada na Seção 2.3.4 e é o mecanismo que permite recompensar o agente mais de uma única vez por molécula gerada, recompensando também durante a geração da molécula e não apenas no fim (YANG et al., 2017).

Kotsias et al. (2020) também empregaram RNNs para a geração condicional das moléculas. Durante a fase de treinamento os descritores de uma molécula são processadas por uma rede neural completamente conectada e então incorporadas ao estado oculto da RNN, que tenta reconstruir esta molécula. Depois é esperado que a RNN aprenda a projetar moléculas

com propriedades pré-determinadas. O trabalho também realizou geração condicionada às *fingerprints* de uma molécula, gerando assim outra molécula semelhante à molécula escolhida .

Blaschke et al. (2018), Gómez-Bombarelli et al. (2018a) utilizaram VRAES para a geração de moléculas. Foi um estudo em que as VRAES preservaram o princípio da similaridade em seu espaço latente, onde moléculas semelhantes são codificadas próximas no espaço latente. O espaço latente mantém a propriedade de similaridade e é contínuo, e essas características são utilizadas para explorar este espaço. A interpolação entre moléculas conhecidas é uma das técnicas utilizadas. Com o auxílio de modelos preditivos, propriedades de moléculas são previstas, como por exemplo a semelhança com drogas ou um modelo QSAR. Com auxílio desses modelos, a Busca Bayesiana é empregada para buscar neste espaço as moléculas que maximizem as características previstas pelo modelo preditivo.

VRAEs também são empregadas para a geração condicional de moléculas. Durante sua fase de treinamento as VRAEs codificam moléculas em conjunto com suas propriedades. Como resultado o modelo pode criar moléculas com propriedades pré determinadas (LIM et al., 2018).

Como explicado na Seção 2.3.4, não existe um cálculo de gradiente direto de GANs quando se trata de gerar sequências. O Aprendizado por Reforço é empregado para viabilizar o treinamento de GANs e controlar as propriedades das moléculas geradas (GUIMARAES et al., 2017; NEIL et al., 2018).

Os Autoencoders Variacionais (VAEs) e as Redes Adversárias Generativas (GANs) são modelos generativos populares em aprendizado de máquina. Recentemente, pesquisadores estenderam esses modelos para criar os Autoencoders Adversários (AAEs) (KADURIN et al., 2017). Os AAEs visam melhorar a criação de instancias semelhantes aos da base de dados incorporando o treinamento adversarial. Isso significa que a rede geradora do AAE é treinada para enganar a rede discriminadora, enquanto a rede discriminadora é treinada para distinguir entre amostras reais e falsas, análogo ao treinamento das GANs. Em um estudo comparativo, os autores mostraram que os VAEs têm uma cobertura do espaço molecular maior do que os AAEs. Dada a comparação de coberturas, foi introduzido um novo hiper-parâmetro ao AAE que permite trocar a cobertura do conjunto de dados original pela qualidade da reconstrução ou vice-versa. Essa inovação levou a resultados comparáveis aos dos VAEs, enquanto também permite mais flexibilidade na escolha entre cobertura e qualidade.

As Redes Neurais de Políticas Convolucionais para Grafos (do inglês *Graph Convolutional Policy Network*, GCPN) foram propostas para a geração de moléculas em 2018 (YOU et al., 2018). Este modelo permite que o processo de geração seja guiado por objetivos específicos, restringindo o espaço de saída com base em regras químicas fundamentais. Para a geração direcionada por objetivos, foram utilizados a representação em grafo, o aprendizado por reforço e o treinamento adversarial, que foram estendidos e combinados em uma estrutura unificada. A GCPN é uma ferramenta útil para a geração de moléculas com propriedades específicas. Ela permite que os usuários especifiquem os objetivos desejados para a molécula a ser gerada e,

em seguida, gera uma molécula que atenda a esses objetivos. Essa abordagem é especialmente útil em química medicinal e na descoberta de medicamentos, onde os pesquisadores buscam criar moléculas com propriedades específicas, como eficácia contra uma doença em particular.

A GCPN utiliza a representação em grafo para modelar as moléculas, permitindo que sejam tratadas como objetos estruturados e mantendo as informações de conectividade entre os átomos. O aprendizado por reforço é usado para guiar o processo de geração, onde o modelo é treinado para maximizar uma recompensa que está diretamente relacionada aos objetivos desejados. Além disso, o treinamento adversarial é utilizado para melhorar a qualidade das amostras geradas, garantindo que elas sigam as regras químicas fundamentais.

No geral, a GCPN é uma abordagem promissora para a geração de moléculas com propriedades específicas. Com mais desenvolvimento e refinamento, ela tem o potencial de se tornar uma ferramenta valiosa para a química medicinal e a descoberta de medicamentos, além de ter aplicações em outras áreas da química e da ciência dos materiais.

3 Proposta e Metodologia

Este capítulo apresenta a proposta e a metodologia deste trabalho. A Seção 3.1 discorre sobre as propostas deste trabalho. Na Seção 3.2 é feita uma breve descrição dos materiais que serão utilizados neste trabalho, apresentando o ferramental e a base de dados a serem utilizados. A Seção 3.3 é dedicada à descrição detalhada dos passos metodológicos que são seguidos nesta pesquisa. São apresentados os métodos e técnicas que foram utilizados para a análise dos dados e a validação dos resultados obtidos.

3.1 Propostas

Esta pesquisa se propõe a execução de sete principais tarefas:

- Geração de moléculas usando métodos de CML;
- Criação de uma biblioteca para processar a base de dados ChEBI;
- Criação de um classificador hierárquico treinado na base de dados ChEBI;
- Classificação das moléculas geradas na taxonomia;
- Comparação dos grupos de moléculas gerados usando a diversidade de Tanimoto, métrica usada na literatura;
- proposição de uma métrica de dissimilaridade entre grupos de moléculas, a distancia hierárquica;
- Validação da métrica proposta através de uma comparação com a métrica encontrada na literatura.

A pesquisa propõe inicialmente o estudo e levantamento de métodos e algoritmos de CML utilizados na geração de moléculas, visando encontrar os métodos mais adequados, com maior desempenho e identificar suas principais características, o uso dos métodos para a geração de moléculas também é proposto. Também propomos a utilização da classificação hierárquica de diversas formas. Como visto na Seção 2.6, outros trabalhos utilizam modelos QSAR para avaliar o quão adequada é uma molécula, porém nenhum deles utiliza classificação hierárquica para classificar novas moléculas. Essa classificação pode ser um indicador da aptidão de novas moléculas. Ela também pode ser utilizada para aprimorar bases de dados, como por exemplo a *Chemical Entities of Biological Interest* (ChEBI), classificando moléculas catalogadas em outras bases ou moléculas geradas por métodos de CML. A taxonomia de uma molécula obtida através da classificação hierárquica é fundação para a medida proposta de dissimilaridade entre grupos

de moléculas, a distancia hierárquica. Para a avaliação do desempenho dos métodos serão utilizadas as métricas mais comuns na literatura e a distancia hierárquica. E para a avaliação da distância hierárquica, ela será comparada com a métrica de dissimilaridade estabelecida na literatura, a distância de Tanimoto.

3.2 Materiais

Nesta seção são descritos os materiais a serem utilizados nessa pesquisa.

GT4SD Toolkit: O Kit de ferramentas generativo para descoberta científica (do inglês *Generative Toolkit for Scientific Discovery*, GT4SD) é uma plataforma abrangente que lida com moléculas e realiza cálculos de propriedades. Ele vem com uma ampla gama de modelos pré-treinados e uma estrutura para treinar modelos de CML. Essa ferramenta é especificamente projetada para ajudar pesquisadores a gerar novas descobertas e obter insights sobre propriedades moleculares. O GT4SD fornece ferramentas eficientes para entender estruturas químicas complexas e prever suas propriedades (TEAM, 2022).

Sistema Clus: Clus (CLUS, 2008) é um sistema de árvore de decisão e indução de regras baseado em Árvores de Clusterização Preditiva (do Inglês Predictive Clustering Trees). Ele unifica a clusterização e a modelagem preditiva, e lida com tarefas de predição complexas, como classificação hierárquica multirrótulo.

ChEBI: A base de dados Chemical Entities of Biological Interest (ChEBI) (DEGTYARENKO et al., 2007) é uma das mais completas fontes de informações sobre moléculas, contendo dados sobre sua estrutura, como a representação SMILES e informações de descritores. Esta base de dados se destaca das outras pois possui uma taxonomia detalhada de moléculas. Com isso, ChEBI permite uma análise mais precisa sobre as propriedades químicas e biológicas das moléculas. A base ChEBI utiliza o formato *Web Ontology Language* (OWL) (ANTONIOU; HARMELEN, 2004), uma linguagem padrão para a descrição de ontologias na Web Semântica.

3.3 Metodologia

A metodologia deste trabalho contempla a preparação das bases de dados, treinamento de diversos modelos construtivos, treinamento de modelos de classificação hierárquica, avaliação dos modelos classificadores e dos modelos construtivos. As etapas de desenvolvimento são listadas a seguir e descritas nas próximas seções.

- Pré-processamento da base de dados ChEBI. Seção 3.3.1;
- Geração de moléculas usando métodos de CML. Seção 3.3.2;

- Criação e treinamento dos modelos de classificação hierárquica. Seção 3.3.3;
- Avaliação do modelo de classificação hierárquica. Seção 3.3.3;
- Avaliação das moléculas geradas e comparação dos resultados através das diferentes métricas encontradas na literatura. Seção 3.3.4;
- Classificação das moléculas geradas empregando os modelos de classificação hierárquica, com o objetivo de avaliar a classificação das novas moléculas na taxonomia. Seção 3.3.4;
- Comparação dos grupos gerados usando a distância hierárquica. Seção 3.3.5.

3.3.1 Pré-processamento da base de dados ChEBI

A base ChEBI se encontra no padrão RDF, que não é um padrão desejado para o treinamento de classificadores hierárquicos. Para treinar o classificador hierárquico é preciso formatar os dados, onde cada linha representa uma molécula. Nesta linha seus descritores são impressos seguidos de uma série de rótulos descrevendo sua posição na taxonomia. Para esta tarefa uma biblioteca foi desenvolvida para transformar os dados da estrutura baseada em RDF para a estrutura intermediária de DAGs e também para a estrutura requerida pelo classificador.

O DAG da base de dados ChEBI é formada por três ontologias, e cada um de seus nós representa um rótulo da ontologia, todos os seus rótulos pertencem a ao menos uma dessas ontologias e todos os rótulos herdam significado semântico de seus pais. A primeira ontologia diz respeito à informação estrutural da molécula, representando uma entidade química, classificando entidades moleculares, partes das mesmas e substâncias químicas. A segunda ontologia diz respeito às funções moleculares, que é um comportamento particular que um material pode exibir. A terceira ontologia expressa partículas atômicas menores que um átomo, como nêutrons e prótons.

A Tabela 1 apresenta algumas informações estatísticas sobre a base ChEBI original. Como pode ser visto, o DAG contém muitos nós e a maioria deles são folhas. Existem 129192 moléculas catalogadas em diferentes nós do DAG, 3842 nós do DAG têm conexões com as ontologias estruturais e de funções, e 2875 desses nós são folhas. Muitas moléculas do conjunto de dados ChEBI são classificadas em mais de um nó e muitos nós estão associados a poucas moléculas.

Tabela 1 – Estatísticas originais do grafo de taxonomia ChEBI

	Grafo estrutural	Grafo funcional	Grafo de particulas
Altura	30	16	6
Numero de nós	12332	5416	36
Numero de folhas	8428	3699	21
Nó com mais pais	CHEBI_9648	CHEBI_9648	CHEBI_36338
Maior numero de pais	15	15	2
Nó com mais filhos	CHEBI_50860	CHEBI_25212	CHEBI_36338
Maior numero de filhos	487	122	6

Para melhorar o desempenho do classificador hierárquico, uma poda foi realizada no DAG original da ChEBI. Apenas moléculas com informações de função foram utilizadas, já que essas são mais relevantes para a descoberta de medicamentos, e apenas nós com pelo menos 100 moléculas anotadas foram considerados. Com isso, o grafo de partículas e muitos nós dos grafos estruturais e de funções foram desconsiderados, obtendo um DAG final com 35032 moléculas catalogadas. A Tabela 2 mostra as estatísticas do DAG pré-processado após a poda.

Tabela 2 – Estatísticas do grafo podado da taxonomia ChEBI

	Grafo estrutural	Grafo funcional
Altura	17	12
Número de nós	278	166
Número de folhas	87	75
Nó com mais filhos	CHEBI_29067	CHEBI_33575
Maior numero de filhos	12	12
Nó com mais pais	CHEBI_29067	CHEBI_33575
Maior número de pais	6	6
folha com mais elementos	CHEBI_61379	CHEBI_35610
Mair número de elementos	753	1666

3.3.2 Geração de moléculas

Dos trabalhos revisados, seis métodos construtivos foram escolhidos para a tarefa de geração de moléculas. A primeira metade deles utiliza a representação SMILES, sendo estes ORGAN (SANCHEZ-LENGELING et al., 2017), VAE (GÓMEZ-BOMBARELLI et al., 2018b), AAE (KADURIN et al., 2017), e a segunda metade utiliza grafos para representar as moléculas, sendo MoLeR (MAZIARZ et al., 2021), GCPN (YOU et al., 2018) e GraphAF (SHI et al., 2020). Para cada método são geradas 2000 moléculas.

3.3.3 Modelo de classificação hierárquica.

Para classificação hierárquica, a abordagem global foi escolhida. O classificador escolhido foi o classificador hierárquico multirrótulo Clus-HMC (VENS et al., 2008) do sistema Clus (CLUS, 2008), treinado utilizando os descritores das moléculas presentes na base de dados ChEBI.

Para a avaliação foram usadas duas métricas, a Área sob a Curva Característica de Operação do Receptor (do inglês *Area Under Receiver Operating Characteristic*, AUROC) e a Área sob a Curva de Precisão-Revocação (do inglês *Area Under Precision Recall Curve*, AUPRC) para avaliar o classificador.

A AUROC plota a taxa de verdadeiros positivos contra a taxa de falsos positivos em diferentes limiares. A AUROC mede a capacidade de um classificador de distinguir entre classes positivas e negativas. É calculada como a área sob a curva ROC, é contínua e sua área varia de 0.5 a 1. Um classificador com AUROC de 1 distingue perfeitamente entre classes positivas e negativas, enquanto um classificador com AUROC de 0.5 não é melhor do que um palpite aleatório. A AUROC é amplamente utilizada em aprendizado de máquina porque

é insensível ao desequilíbrio de classes e fornece uma avaliação robusta mesmo quando a distribuição de instâncias positivas e negativas é altamente desequilibrada (MELO; DUBITZKY; WOLKENHAUER, 2013).

A AUPRC resume a relação entre precisão e revocação. A precisão mede quantas das instâncias positivas previstas são realmente positivas, enquanto o revocação mede quantas das instâncias positivas verdadeiras foram previstas corretamente. A AUPRC é calculada traçando a precisão em relação à revocação para diferentes limiares e computando a área sob essa curva. Um alto valor de AUPRC indica um bom desempenho, pois significa que o classificador é capaz de alcançar tanto alta precisão quanto alta revocação. A AUPRC é mais informativa do que usar apenas precisão ou a revocação, fornecendo uma imagem mais abrangente do desempenho do classificador, especialmente em distribuições desequilibradas de instâncias positivas e negativas (DAVIS; GOADRICH, 2006).

3.3.4 Avaliação das moléculas geradas

Uma forma de avaliar o desempenho dos modelos construtivos é através da comparação das distribuições dos descritores das moléculas geradas. A análise dos descritores relacionados à regra de cinco e do descritor QED são bons guias para saber se as moléculas geradas são boas candidatas a serem fármacos. Quando muitos descritores são avaliados, é possível aplicar um redutor de dimensionalidade nas propriedades para condensar a comparação. Outras métricas são a diversidades interna e externa, que como explicadas na Seção 2.5, são utilizadas para identificar o quão diverso é um grupo de moléculas e comparar a similaridade entre dois grupos distintos de moléculas.

A avaliação proposta neste trabalho consiste em classificar hierarquicamente as moléculas geradas e verificar a pertinência destas em relação às taxonomias existentes.

3.3.5 Distância hierárquica proposta

A distância hierárquica proposta usa informações de taxonomia para comparar diferentes grupos de moléculas. Para cada conjunto de moléculas é criado um DAG, e no cálculo um deles é escolhido como referência. Cada nó do DAG armazena quantas moléculas são classificadas nele, e os nós são ordenados para construir a primeira distribuição do DAG. O segundo DAG usa os mesmos rótulos na mesma ordem que o primeiro DAG para construir sua distribuição de maneira análoga, como um dos DAGs é escolhido para referencia. Finalmente, a distância entre as distribuições de moléculas é medida usando a Distância de Wasserstein (VALLENDER, 1974), o que pode ser útil para comparar grandes grupos de moléculas porque sua complexidade é muito menor do que a diversidade externa. A Equação 3.1 apresenta a distância de Wasserstein duas distribuições u e v. As distribuições usadas para calcular a distância de Wasserstein são as distribuições das moléculas dos DAGs em seus nós.

$$W_d(u,v) = \inf_{\pi \in \Gamma(u,v)} \int_{\mathbb{R}\times\mathbb{R}} |x-y| \mathrm{d}\pi(x,y) \tag{3.1}$$

A distância hierárquica é assimétrica. Como precisamos escolher um dos DAGs como referência, os resultados podem ser diferentes dependendo do DAG escolhido. No entanto, nossos resultados no Capítulo 4 mostraram que essas diferenças não foram significativas.

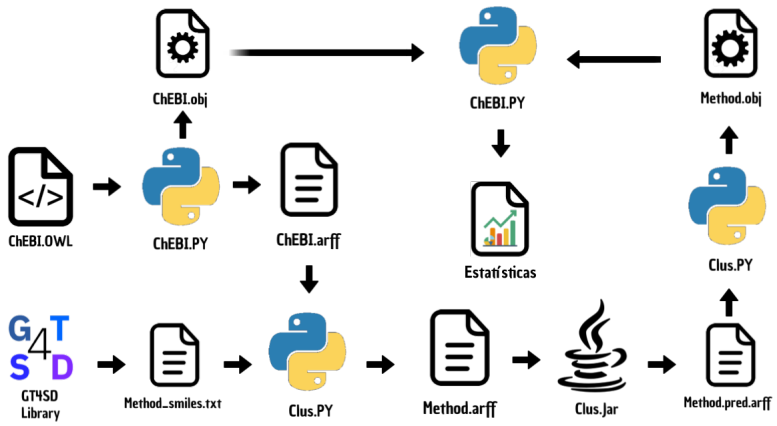
Para validar a distância hierárquica, duas metodologias foram escolhidas. A primeira usando a visualização das moléculas, escolhendo um par de grupos de moléculas cujo descritores reduzidos se sobreponham e então calculando a distância hierárquica deste par para verificar se esta é baixa. Um par de grupos de moléculas que não se sobrepõem também é escolhido para verificar se a distância hierárquica é alta. A segunda metodologia é a comparação com a métrica de dissimilaridade encontrada na literatura, a diversidade externa, e verificar se a distância hierárquica exprime as mesmas relações que a diversidade externa.

3.3.6 Visão geral da metodologia

A Figura 11 Apresenta uma visão geral dos passos da metodologia. Primeiro, o conjunto de dados ChEBI, originalmente em formato OWL (ChEBI.OWL), é pré-processado em um DAG. Este DAG é podado e salvo em dois arquivos: ChEBI.arff e ChEBI.obj. As moléculas são geradas usando a biblioteca GT4SD e, para cada um de seus métodos, é criado o respectivo arquivo Method.arff contendo as moléculas geradas. Essas moléculas serão classificadas pelo sistema Clus, que utiliza o arquivo ChEBI.arff para treinar o classificador hierárquico multirrótulo Clus-HMC e faz previsões nas moléculas presentes nos arquivos Method.arff. As previsões são armazenadas no arquivo Method.pred.arff e convertidas para o formato Method.obj, que é usado juntamente com o ChEBI.obj para gerar os resultados discutidos no Capítulo 4.

A biblioteca desenvolvida pode ser encontrada no *github* [1]

[1] <https://github.com/yendorr/gt4sd>



Retirado de: Desenvolvido pelo autor

Figura 11 – Ilustração da metodologia utilizada por este trabalho.

4 Resultados

O capítulo em questão tem como objetivo apresentar uma série de informações relevantes sobre as moléculas geradas e suas propriedades. Para isso, ele se inicia com uma descrição sucinta de algumas das propriedades mais importantes dessas moléculas. Em seguida, são apresentadas visualizações comprimidas dos descritores das moléculas, o que permite que o leitor possa ter uma visão geral sobre os grupos de moléculas geradas. Além disso, este capítulo também traz informações sobre as métricas de desempenho do classificador hierárquico utilizado. Essas métricas são de extrema importância para avaliar a capacidade do modelo em classificar as moléculas geradas de acordo com suas características.
Outra informação relevante presente neste capítulo é a apresentação das estatísticas dos DAGs dos grupos de moléculas geradas. Essas estatísticas fornecem uma visão sobre a estrutura dessas moléculas e podem ajudar a identificar possíveis padrões. Em seguida, são apresentadas estatísticas dos grupos de moléculas por classes, o que permite uma análise mais específica sobre cada uma dessas classes e suas características.
Por fim, o capítulo aborda a distância hierárquica e como ela pode ser usada para medir a dissimilaridade entre os grupos de moléculas geradas. Essa medida é comparada com a diversidade externa, o que permite avaliar se as relações entre os grupos de moléculas geradas são mantidos.

Para cada molécula, foram calculados 21 descritores, incluindo o peso molecular, relacionado ao número e tipo de átomos na molécula, o $PlogP$, relacionado à solubilidade da molécula em água, e todas as propriedades apresentadas na Seção 2.2. A Tabela 3 apresenta informações estatísticas sobre alguns dos descritores mais importantes para cada conjunto de moléculas gerado. O conjunto de moléculas gerado pelo modelo AAE se difere dos outros conjuntos, tendo em geral média e variância mais altas para os descritores de peso molecular e $PlogP$. É possível inferir que muitas moléculas neste conjunto violam pelo menos uma regra da regra RO5. Por outro lado, o conjunto gerado pelo GCPN apresentou os valores e variações mais baixos para essas duas propriedades.

Tabela 3 – Estatísticas de três propriedades dos grupos de moléculas.

	Peso		**Plog P**		**QED**	
	Média	**Desvio padrão**	**Média**	**Desvio padrão**	**Média**	**Desvio padrão**
AAE	490.55	237.56	6.06	5.79	0.43	0.25
ORGAN	381.42	103.86	3.39	1.75	0.57	0.20
VAE	385.91	106.86	3.43	1.79	0.56	0.21
GCPN	252.56	61.98	2.75	1.25	0.73	0.12
GraphAF	262.61	138.57	2.28	1.62	0.52	0.17
MoLeR	392.27	102.99	3.49	1.88	0.52	0.21

Para visualizar as múltiplas dimensões das moléculas, foram aplicados dois métodos de redução de dimensionalidade: PCA (JOLLIFFE, 2005) e Projeção e Aproximação Uniforme de

Variedade (do inglês *Uniform Manifold Approximation and Projection* , UMAP) (MCINNES; HEALY; MELVILLE, 2018). O PCA foi aplicado em um conjunto de dados com 12 mil vetores (6 métodos x 2 mil moléculas), cada um com 21 dimensões. A PCA foi também aplicado em conjunto do UMAP. A UMAP que rearranja os dados com base em sua proximidade. Esse rearranjo pode ser útil para visualizar dados sobrepostos e possíveis agrupamentos de dados.

A Figura 12 ilustra como as moléculas estão distribuídas nos espaços PCA e UMAP. Podemos ver que o grupo mais destacado de moléculas são os grupos gerados pelos métodos AAE e GraphAF, enquanto os outros quatro grupos parecem ser muito semelhantes entre si. Apesar de que muitos dos pontos que representam as moléculas no espaço PCA se sobrepõe, é possível saber onde cada grupo está localizado em relação ao outro. No espaço UMAP, os pontos raramente se sobrepõem, mas é difícil definir uma região para cada grupo de moléculas.

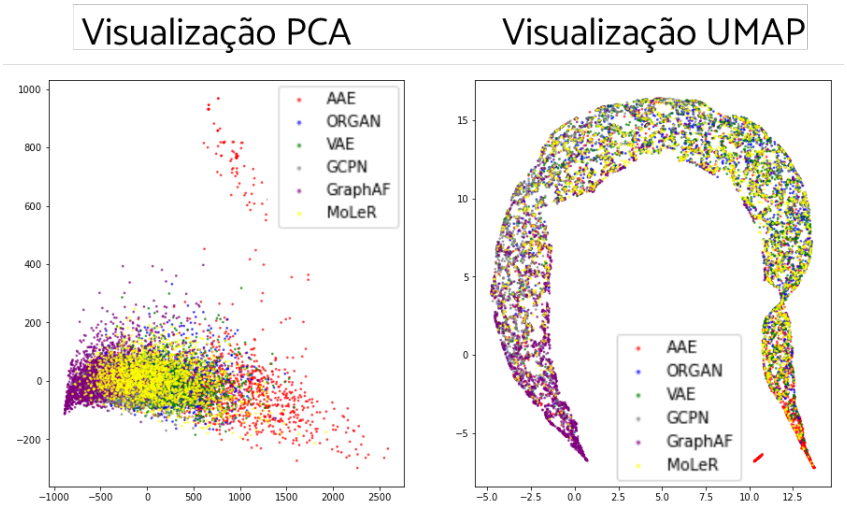


Figura 12 – Visualização bidimensional dos descritores das moléculas geradas por todos os métodos.

O classificador hierárquico utilizado neste estudo foi treinado com uma estratégia de validação cruzada de 10 pastas (10-fold cross-validation), utilizando os descritores moleculares das 35032 moléculas da base de dados ChEBI. Esse processo de validação cruzada é amplamente utilizado para avaliar a capacidade de generalização de um modelo de classificação e minimizar o risco de overfitting. Como resultado desse treinamento, os valores médios de AUPRC e AUROC foram obtidos e estão apresentados na Tabela 4. É importante destacar que esses valores foram considerados bastante elevados, demonstrando que o classificador hierárquico apresentou uma boa capacidade de discriminação entre as diferentes classes de moléculas presentes na base de dados. Contudo, vale ressaltar que a média simples de AUPRC pode não ser um indicador adequado em situações de desequilíbrio de classes, visto que essa métrica pode ser influenciada pela predominância de uma determinada classe na base de dados.

As moléculas geradas foram classificadas em classes da taxonomia ChEBI podada usando um limiar (do inglês *threshold*) de 0.8. Para comparar grupos de moléculas, um DAG foi criado para cada conjunto de moléculas geradas e cada DAG armazena as informações

Tabela 4 – Resultados do classificador hierárquico multirrótulo na base de dados ChEBI.

Nome da métrica	**Valor da métrica**
Média AUROC	0.8931
Média simples AUPRC	0.3541
Média ponderada AUPRC	0.7766

da taxonomia das moléculas de seu respectivo grupo. A Tabela 5 apresenta informações estatísticas sobre os DAGs que expressam a informação de funções das moléculas, e a Tabela 6 apresenta informações estatísticas sobre os DAGs que expressam a informação estrutural. Embora as informações fornecidas nas tabelas possam ser medidas individualmente para cada molécula, mostramos os resultados para grupos de moléculas a fim de ter uma visão geral das propriedades referentes às moléculas geradas por cada método. É possível observar nas tabelas que os métodos ORGAN, VAE e MoLeR possuem estatísticas próximas.

Tabela 5 – Estatísticas dos DAGs funcionais dos grupos de moléculas

	AAE	ORGAN	VAE	GCPN	GrapgAF	MoLeR
Altura	9	10	10	11	10	11
Número de nós	56	65	72	44	46	69
Número de folhas	16	19	22	11	13	16
Maior número de filhos	4	5	5	4	4	5
Maior número de pais	3	5	5	4	4	5
Folha com mais elementos	19	8	6	4	4	8

Tabela 6 – Estatísticas dos DAGs estruturais dos grupos de moléculas

	AAE	ORGAN	VAE	GCPN	GrapgAF	MoLeR
Altura	14	15	15	15	15	15
Número de nós	115	142	141	101	111	123
Número de folhas	18	22	22	13	14	17
Maior número de filhos	8	9	8	7	7	7
Maior número de pais	3	5	5	4	4	5
Folha com mais elementos	6	8	6	4	2	8

A Tabela 7 mostra algumas classes e a porcentagem dos grupos de moléculas gerados que se enquadram nelas. Essas classes incluem uma função bioquímica qualquer, Metabólitos e sub-classes de metabólitos, Alcaloide e uma sub-classe de alcaloide. Com a classificação hierárquica proposta, é possível reduzir o número de moléculas, o que permite uma busca mais localizada e focada em determinadas classes ou características químicas.

A diversidade interna e externa foram usadas para verificar a similaridade dos grupos de moléculas. A diagonal principal da Tabela 8 mostra a diversidade interna das moléculas,

Tabela 7 – Porcentagens por grupos de moléculas classificadas em classes da taxonomia

	AAE (%)	ORGAN (%)	VAE (%)	GCPN (%)	GraphAF (%)	MoLeR (%)
Função bioquimica qualquer	7,45	8,5	8,6	19,15	33,9	6,25
Metabólito	7.4	8.5	8.55	19.15	33.9	6.2
Metabólito de mamífero	0.95	0.0	0.1	0.0	0.3	0.1
Metabólito humano	0.05	0.0	0.0	0.0	0.05	0.1
Alcaloide	0.8	1.0	0.55	0.45	0.05	0.45
Alcaloide harmala	0.25	0.4	0.15	0.0	0.0	0.15

enquanto a diversidade externa é mostrada nas outras células. Três observações principais podem ser feitas: i) os conjuntos de moléculas GraphAF e GCPN têm a maior diversidade interna; ii) as moléculas geradas por GraphAF e GCPN têm as maiores diversidades externas; e iii) as moléculas geradas por AAE e VAE têm a menor diversidade externa. Essas observações são úteis para interpretar as relações entre os conjuntos de moléculas e como base para verificar se a proposta de distância hierárquica também é capaz de capturar essas relações.

Tabela 8 – Diversidade interna e externa entre os grupos de moléculas

	AAE	ORGAN	VAE	GCPN	GaphAF	MoLeR
AAE	0.719	0.727	0.726	0.790	0.843	0.739
ORGAN	0.727	0.728	0.727	0.790	0.842	0.740
VAE	0.726	0.727	0.726	0.790	0.842	0.739
GCPN	0.790	0.790	0.790	0.822	0.869	0.796
GaphAF	0.843	0.842	0.842	0.869	0.895	0.847
MoLeR	0.739	0.740	0.739	0.796	0.847	0.749

Devido à dificuldade de visualizar grupos pouco excepcionais de moléculas na Figura 12, a Figura 13 mostra apenas as moléculas geradas pelos métodos ORGAN e VAE. Elas foram escolhidas para validar como a distância hierárquica proposta difere dois conjuntos semelhantes de moléculas. Podemos ver que os grupos são próximos, portanto, espera-se um valor de distância hierárquica pequeno.

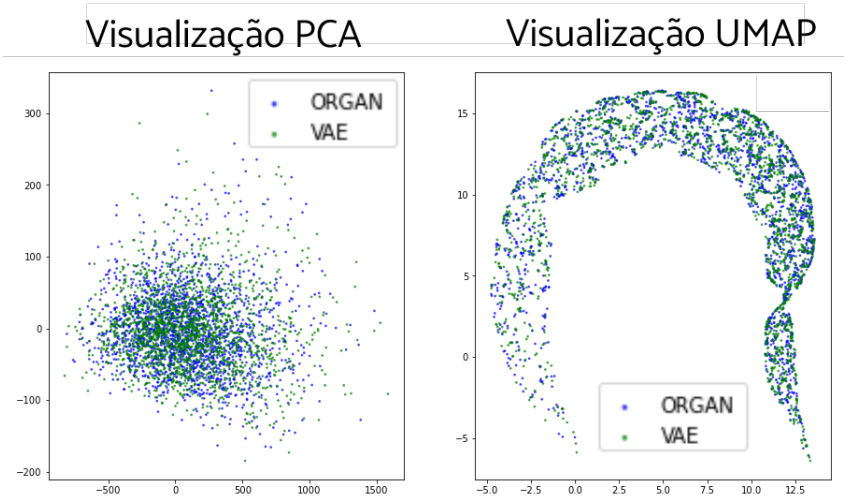


Figura 13 – Visualização bidimensional dos descritores das moléculas geradas pelos métodos ORGAN e VAE.

A Figura 14 mostra as distribuições utilizadas para calcular a distância hierárquica, as distribuições das moléculas sobre os nós do DAG considerando os métodos ORGAN e VAE. As distribuições são muito semelhantes, o que deve resultar em uma pequena distância hierárquica.

A Figura 15 mostra a visualização dos descritores das moléculas geradas por AAE e GraphAF. Eles foram escolhidos para validar como a distância hierárquica difere dois conjuntos de moléculas dissimilares. Eles não estão tão próximos um do outro, e uma alta distância hierárquica é esperada.

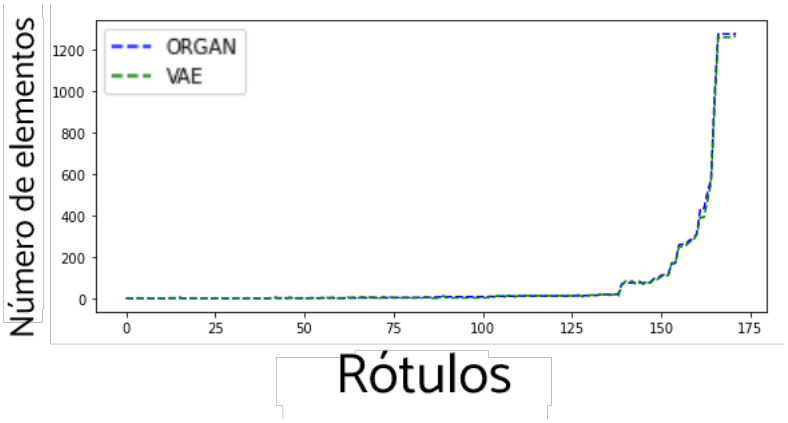


Figura 14 – Comparação das distribuições de moléculas geradas pelos métodos ORGAN e VAE.

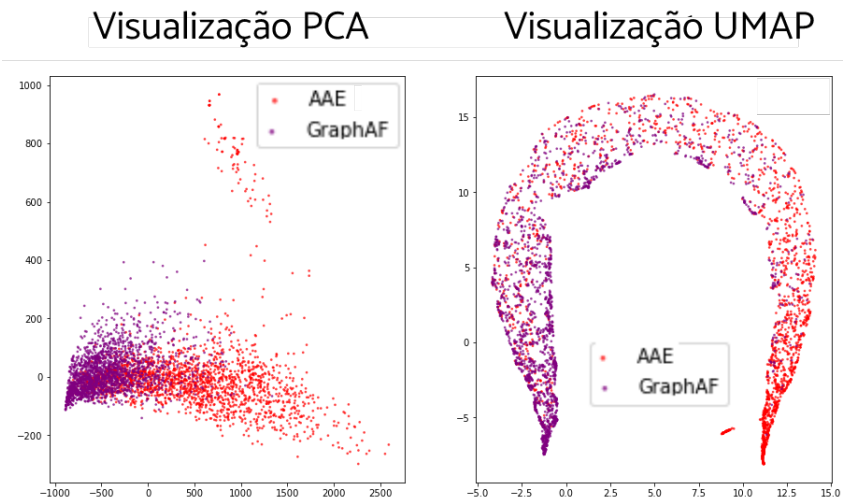


Figura 15 – Visualização bidimensional dos grupos de moléculas geradas pelos métodos AAE e GraphAF.

Para atestar a diferença entre esses dois grupos de moléculas, comparamos suas distribuições. A Figura 16 mostra as distribuições das moléculas sobre os nós DAG. Vemos uma discrepância sensível separando-as, o que deve resultar em uma alta distância hierárquica.

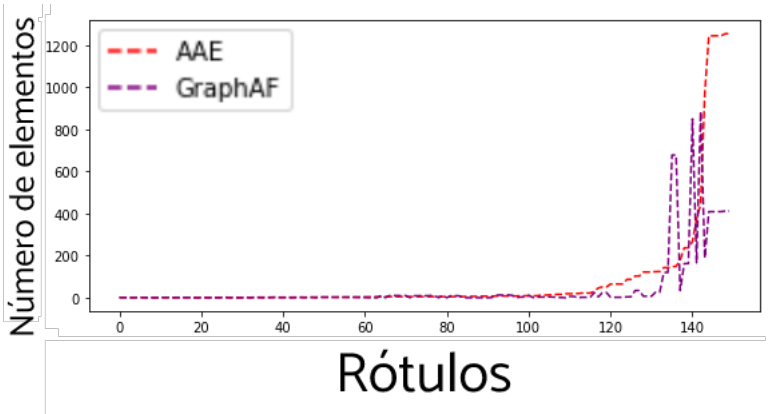


Figura 16 – Comparação das distribuições de moléculas geradas pelos métodos AAE e GraphAF.

A Tabela 9 apresenta a distância hierárquica para cada par de grupos de moléculas. A

partir dos resultados, podemos separar os grupos de moléculas em dois conjuntos. O primeiro conjunto contém os grupos de moléculas geradas pelos métodos GraphAF e GCPN, mostrando uma distância relativa pequena entre si e grandes distâncias para os outros grupos. O segundo conjunto é formado pelos outros grupos de moléculas, com alta similaridade entre si e grandes distâncias para os grupos do primeiro conjunto. Os resultados da Tabela 9 estão em consonância com os da Tabela 8, exceto que os valores de diversidade externa são limitados entre 0 e 1. A distância hierárquica varia de acordo com os tamanhos dos DAGs e a dissonância entre os conjuntos de moléculas, podendo passar de 1.

Tabela 9 – Distância hierárquica entre os grupos de moléculas geradas.

	AAE	ORGAN	VAE	GCPN	GaphAF	MoLeR
AAE	0.00	11.70	9.92	41.62	44.05	8.08
ORGAN	13.08	0.00	3.06	48.96	54.46	9.03
VAE	11.36	3.02	0.00	46.20	52.09	8.24
GCPN	34.07	35.37	32.40	0.00	16.22	32.91
GaphAF	40.81	43.81	40.72	18.35	0.00	40.72
MoLeR	8.44	8.62	7.77	43.08	47.66	0.00

5 Considerações Finais e Trabalhos Futuros

Este trabalho apresentou um estudo sobre Aprendizado de Máquina Construtivo aplicado ao design de moléculas novas semelhantes a medicamentos. moléculas foram geradas utilizando seis métodos diferentes do kit de ferramentas GT4SD e estas foram comparadas com medidas de diversidade interna e externa. Também foi empregado um classificador hierárquico multirrótulo para classificar as moléculas em uma taxonomia e comparamos grupos de moléculas com base em sua classificação.

O conhecido conjunto de dados ChEBI foi utilizado para treinar o classificador hierárquico, uma vez que contém informações de taxonomia sobre uma variedade de moléculas e é muito menor do que outras bases de dados de moléculas. O classificador hierárquico proposto pode ser usado para expandir o conjunto de dados ChEBI classificando novas moléculas em sua taxonomia. Também pode ser usado na descoberta de medicamentos, classificando as moléculas geradas para reduzir o número de candidatos e selecionar as moléculas com um papel específico.

Com as informações da taxonomia das moléculas, é possível comparar grupos de moléculas. A distância hierárquica proposta obteve resultados semelhantes aos da diversidade externa, mas em uma fração do tempo. A diversidade externa tem uma faixa limitada entre 0 e 1, enquanto nossa distância hierárquica pode variar com base no tamanho e diversidade dos grupos comparados. Os experimentos mostraram que a distância hierárquica é mais significativa quando calculada para grupos de moléculas de tamanho semelhante.

Como trabalho futuro, a distância hierárquica poderia ser estendida a outros contextos e campos de pesquisa, para comparar grupos de diferentes moléculas que podem ser classificadas em uma taxonomia. Além disso, as moléculas selecionadas podem ser usadas para ajustar seus geradores, a fim de criar um modelo de geração de domínio específico.

Referências

ANTONIOU, G.; HARMELEN, F. v. Web ontology language: Owl. In: *Handbook on ontologies*. [S.l.]: Springer, 2004. p. 67–92.

AWAD, M.; KHANNA, R. *Efficient learning machines: theories, concepts, and applications for engineers and system designers*. [S.l.]: Springer Nature, 2015.

BENHENDA, M. Can ai reproduce observed chemical diversity? *bioRxiv*, Cold Spring Harbor Laboratory, p. 292177, 2018.

BICKERTON, G. R.; PAOLINI, G. V.; BESNARD, J.; MURESAN, S.; HOPKINS, A. L. Quantifying the chemical beauty of drugs. *Nature chemistry*, Nature Publishing Group, v. 4, n. 2, p. 90–98, 2012.

BJERRUM, E. J.; THRELFALL, R. Molecular generation with recurrent neural networks (rnns). *arXiv e-prints*, p. arXiv–1705, 2017.

BLASCHKE, T.; OLIVECRONA, M.; ENGKVIST, O.; BAJORATH, J.; CHEN, H. Application of generative autoencoder in de novo molecular design. *Molecular informatics*, Wiley Online Library, v. 37, n. 1-2, p. 1700123, 2018.

BOWMAN, S. R.; VILNIS, L.; VINYALS, O.; DAI, A. M.; JOZEFOWICZ, R.; BENGIO, S. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

CAO, D.; LIANG, Y. User guide for chemopy 1.0. 2012.

CLUS. 2008. Disponível em: <https://dtai.cs.kuleuven.be/clus/>.

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

DEGTYARENKO, K.; MATOS, P. D.; ENNIS, M.; HASTINGS, J.; ZBINDEN, M.; MCNAUGHT, A.; ALCÁNTARA, R.; DARSOW, M.; GUEDJ, M.; ASHBURNER, M. Chebi: a database and ontology for chemical entities of biological interest. *Nucleic acids research*, Oxford University Press, v. 36, n. suppl_1, p. D344–D350, 2007.

DOERSCH, C. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

ELTON, D. C.; BOUKOUVALAS, Z.; FUGE, M. D.; CHUNG, P. W. Deep learning for molecular design—a review of the state of the art. *Molecular Systems Design & Engineering*, Royal Society of Chemistry, v. 4, n. 4, p. 828–849, 2019.

ERTL, P.; SCHUFFENHAUER, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, Springer, v. 1, n. 1, p. 1–11, 2009.

FERREIRA, M. V. C.; PAES, V. R.; LICHTENSTEIN, A. Penicilina: oitenta anos. *Revista de Medicina*, v. 87, n. 4, p. 272–276, 2008.

FOSTER, D. *Generative deep learning: teaching machines to paint, write, compose, and play*. [S.l.]: O'Reilly Media, 2019.

GÓMEZ-BOMBARELLI, R.; WEI, J. N.; DUVENAUD, D.; HERNÁNDEZ-LOBATO, J. M.; SÁNCHEZ-LENGELING, B.; SHEBERLA, D.; AGUILERA-IPARRAGUIRRE, J.; HIRZEL, T. D.; ADAMS, R. P.; ASPURU-GUZIK, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, ACS Publications, v. 4, n. 2, p. 268–276, 2018.

GÓMEZ-BOMBARELLI, R.; WEI, J. N.; DUVENAUD, D.; HERNÁNDEZ-LOBATO, J. M.; SÁNCHEZ-LENGELING, B.; SHEBERLA, D.; AGUILERA-IPARRAGUIRRE, J.; HIRZEL, T. D.; ADAMS, R. P.; ASPURU-GUZIK, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, ACS Publications, v. 4, n. 2, p. 268–276, 2018.

GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. Generative adversarial nets. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2014. p. 2672–2680.

GOYAL, P.; PANDEY, S.; JAIN, K. Deep learning for natural language processing. *New York: Apress*, Springer, 2018.

GUIMARAES, G. L.; SANCHEZ-LENGELING, B.; OUTEIRAL, C.; FARIAS, P. L. C.; ASPURU-GUZIK, A. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *arXiv preprint arXiv:1705.10843*, 2017.

GUPTA, A.; MÜLLER, A. T.; HUISMAN, B. J.; FUCHS, J. A.; SCHNEIDER, P.; SCHNEIDER, G. Generative recurrent networks for de novo drug design. *Molecular informatics*, Wiley Online Library, v. 37, n. 1-2, p. 1700111, 2018.

JOLLIFFE, I. Principal component analysis. *Encyclopedia of statistics in behavioral science*, Wiley Online Library, 2005.

KADURIN, A.; NIKOLENKO, S.; KHRABROV, K.; ALIPER, A.; ZHAVORONKOV, A. druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Molecular pharmaceutics*, ACS Publications, v. 14, n. 9, p. 3098–3104, 2017.

KAPOOR, S. *Policy Gradients in a Nutshell*. 2018. <https://towardsdatascience.com/policy-gradients-in-a-nutshell-8b72f9743c5d>. Acessado em: 08 de maio de 2021.

KOTSIAS, P.-C.; ARÚS-POUS, J.; CHEN, H.; ENGKVIST, O.; TYRCHAN, C.; BJERRUM, E. J. Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. *Nature Machine Intelligence*, Nature Publishing Group, v. 2, n. 5, p. 254–265, 2020.

LEELANANDA, S. P.; LINDERT, S. Computational methods in drug discovery. *Beilstein journal of organic chemistry*, Beilstein-Institut, v. 12, n. 1, p. 2694–2718, 2016.

LIM, J.; RYU, S.; KIM, J. W.; KIM, W. Y. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *Journal of cheminformatics*, BioMed Central, v. 10, n. 1, p. 1–9, 2018.

MAZIARZ, K.; JACKSON-FLUX, H.; CAMERON, P.; SIROCKIN, F.; SCHNEIDER, N.; STIEFL, N.; SEGLER, M.; BROCKSCHMIDT, M. Learning to extend molecular scaffolds with structural motifs. *arXiv preprint arXiv:2103.03864*, 2021.

MCINNES, L.; HEALY, J.; MELVILLE, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

MELO, F.; DUBITZKY, W.; WOLKENHAUER, O. *Encyclopedia of systems biology*. [S.l.]: Springer New York New York, NY, USA:, 2013.

MITCHELL, J. B. Machine learning methods in chemoinformatics. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, Wiley Online Library, v. 4, n. 5, p. 468–481, 2014.

MITCHELL, J. B. Machine learning methods in chemoinformatics. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, Wiley Online Library, v. 4, n. 5, p. 468–481, 2014.

MONTANARI, C. A.; PILLI, R. A. *Química Medicinal*. <https://siteantigo.portaleducacao.com.br/conteudo/artigos/farmacia/quimica-medicinal/249>. Acessado em: 29 de setembro de 2021.

NAIR, P. *SeqGAN: GANs for sequence generation*. 2018. <https://medium.com/ai-club-iiitb/seqgan-gans-for-sequence-generation-5c74a84cd230>. Acessado em: 08 de maio de 2021.

NEIL, D.; SEGLER, M.; GUASCH, L.; AHMED, M.; PLUMBLEY, D.; SELLWOOD, M.; BROWN, N. Exploring deep recurrent models with reinforcement learning for molecule design. 2018.

OLIVECRONA, M.; BLASCHKE, T.; ENGKVIST, O.; CHEN, H. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, Springer, v. 9, n. 1, p. 48, 2017.

PEREIRA, A. L.; PITA, J. R. Alexander fleming (1881-1955): da descoberta da penicilina (1982) ao prémio nobel (1945). *História: revista da Faculdade de Letras da Universidade do Porto*, v. 6, 2018.

POLISHCHUK, P. G.; MADZHIDOV, T. I.; VARNEK, A. Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of computer-aided molecular design*, Springer, v. 27, n. 8, p. 675–679, 2013.

SANCHEZ-LENGELING, B.; OUTEIRAL, C.; GUIMARAES, G. L.; ASPURU-GUZIK, A. Optimizing distributions over molecular space. an objective-reinforced generative adversarial network for inverse-design chemistry (organic). 2017.

SEGLER, M. H.; KOGEJ, T.; TYRCHAN, C.; WALLER, M. P. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, ACS Publications, v. 4, n. 1, p. 120–131, 2018.

SHI, C.; XU, M.; ZHU, Z.; ZHANG, W.; ZHANG, M.; TANG, J. Graphaf: a flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382*, 2020.

SILLA, C. N.; FREITAS, A. A. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, Springer, v. 22, n. 1, p. 31–72, 2011.

SUSIK, R. Recurrent autoencoder with sequence-aware encoding. In: SPRINGER. *International Conference on Computational Science*. [S.l.], 2021. p. 47–57.

TEAM, G. *GT4SD (Generative Toolkit for Scientific Discovery)*. 2022. Disponível em: <https://github.com/GT4SD/gt4sd-core>.

VALLENDER, S. Calculation of the wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications*, SIAM, v. 18, n. 4, p. 784–786, 1974.

VENS, C.; STRUYF, J.; SCHIETGAT, L.; DŽEROSKI, S.; BLOCKEEL, H. Decision trees for hierarchical multi-label classification. *Machine learning*, Springer, v. 73, n. 2, p. 185–214, 2008.

VOLLHARDT, P.; SCHORE, N. E. *Química Orgânica-: Estrutura e Função*. [S.l.]: Bookman Editora, 2013.

WEININGER, D. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, ACS Publications, v. 28, n. 1, p. 31–36, 1988.

YANG, X.; ZHANG, J.; YOSHIZOE, K.; TERAYAMA, K.; TSUDA, K. Chemts: an efficient python library for de novo molecular generation. *Science and technology of advanced materials*, Taylor & Francis, v. 18, n. 1, p. 972–976, 2017.

YOU, J.; LIU, B.; YING, Z.; PANDE, V.; LESKOVEC, J. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in neural information processing systems*, v. 31, 2018.

YU, L.; ZHANG, W.; WANG, J.; YU, Y. Seqgan: Sequence generative adversarial nets with policy gradient. In: *Thirty-first AAAI conference on artificial intelligence*. [S.l.: s.n.], 2017.

YU, Y.; SI, X.; HU, C.; ZHANG, J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, v. 31, n. 7, p. 1235–1270, 07 2019. ISSN 0899-7667. Disponível em: <https://doi.org/10.1162/neco_a_01199>.