UNIVERSIDADE FEDERAL DE SÃO CARLOS
DEPARTAMENTO DE COMPUTAÇÃO
ENGENHARIA DE COMPUTAÇÃO

Pedro Henrique Casarotto

Impacto de Uso de Expressões Gênicas com Maior Precisão e com Aplicação de Modelo de Redução de Ruído em Diferentes Embeddings para Geração de Moléculas

São Carlos - SP

2025

Pedro Henrique Casarotto

Impacto de Uso de Expressões Gênicas com Maior Precisão e com Aplicação de Modelo de Redução de Ruído em Diferentes Embeddings para Geração de Moléculas

Trabalho de Conclusão de Curso apresentado ao curso de Ciência da Computação da Universidade Federal de São Carlos, como requisito para a obtenção do título de Bacharel em Engenharia de Computação.

Orientação Prof. Dr. Alan Demétrius Baria Valejo

São Carlos - SP

2025

Dedico este trabalho à minha família, à minha namorada, aos meus amigos, e acima de tudo, a Deus

Agradecimentos

Agradeço aos meus pais, Carlos Henrique Casarotto e Georgia Marcela Zabala Vaz Casarotto, por terem me criado, me forjado na fé, e por terem me dado condições para eu ingressar nesta faculdade. Sem o zelo deles pela minha educação, eu nunca teria chegado aqui. Também agradeço à minha irmã, Mariana Casarotto, por ter sido uma irmã sempre atenciosa. Ela me ajudou durante minha adaptação em São Carlos, e me ajudou diversas vezes a olhar por novas perspectivas.

Agradeço à minha namorada, Ana Luisa Rosa, por todos os momentos incríveis que tive junto à mulher mais linda do mundo. Graças a você, pude levar a vida de modo mais leve, e ganhar forças para continuar minha trajetória sem desistir no caminho. Obrigado pelos seus sorrisos e pela sua presença em minha vida.

Agradeço imensamente ao Professor Doutor Ricardo Cerri, por ter confiado a mim a Iniciação Científica que foi minha introdução à área de Inteligência Artificial, e também por ter me apoiado em ir no meu primeiro evento científico, o BRACIS. É graças a ele que a minha vida mudou para essa área, e que pude, e possso, viver todos os dias meu sonho de criança de mexer com Inteligência Artificial. Agradeço também ao Professor Doutor Alan Demétrius Baria Valejo, o qual aceitou de bom grado o tema da minha pesquisa e me incentivou a me aprofundar nesse tema denso e específico de Inteligência Artificial Generativa.

Agradeço à minha mentora Bruna Zamith Santos por ter me guiado no mundo de Aprendizado de Máquina voltado para aplicações, e também para a academia. Agradeço imensamente ao meu gestor William Giotto por ter acreditado no meu potencial e me ter aberto as portas para o mundo de Aprendizado de Máquina dentro da Amazon. Agradeço, juntamente, a todos os meus colegas da Amazon que me incentivaram no desenvolvimento da minha carreira profissional e científica (Ariel, Arthur, Lucas e Fernanda).

Agradeço também aos meus colegas de ensino fundamental, médio e os da faculdade (os Epaminogansters), não só por terem me incentivado aos conhecimentos árduos e a resolver questões de matemática mais difíceis, mas também por trazerem mais leveza aos meus dias, e por suportarem todas as minhas horríveis piadas. Em especial, agradeço a Gabriel de Ângelo Passos, meu melhor amigo a 9 anos (e o que tem mais paciência, pelo jeito).

Por fim, agradeço a Deus e à Santíssima Virgem Maria por todas as coisa que aconteceram na minha vida e que me levaram a este momento. Nunca imaginei que faria um trabalho deste porte na minha vida toda, e isso é só mais um motivo para louvá-Lo mais e agradeçe-Lo. Portanto, ofereço totalmente a Cristo este trabalho.

"Tenho minhas dúvidas sobre o valor real do montanhismo, de chegar ao topo de tudo e contemplar tudo do alto. Satanás foi o maior dos guias montanheses, quando levou Jesus para o alto de uma montanha extremamente alta e lhe mostrou todos os reinos do mundo. Mas a alegria de Satanás em subir em um pico não é a alegria pela grandeza, mas a alegria de contemplar a pequenez, pelo fato de que todos os homens parecem insetos aos seus pés. É desde o vale que as coisas parecem grandes; é da planície que as coisas parecem altas. Eu sou um filho da planície e não tenho necessidade dequele grande guia montanhês. [...] Agora, vou sentar-me e deixar que as maravilhas e aventuras pousem sobre mim como moscas. Há muitas delas, garanto. O mundo nunca sofrerá com a falta de maravilhas, mas apenas com a falta de se maravilhar"

(G. K. Chesterton)

Resumo

A geração de novas moléculas e compostos moleculares é uma tarefa que possui variados números de aplicações em diversas áreas da ciência, sendo altamente relevante para o desenvolvimento de remédios para doenças que até hoje não apresentam cura. Com o "de novo design", algoritmos de aprendizado de máquina são capazes de criar novas moléculas utilizando Inteligência Artificial Generativa. Entretanto, a expressão gênica utilizada para este processo tem dados com muito ruído e que são difíceis de serem mapeados. Este trabalho busca explorar o modo com que expressões gênicas são representadas, explorando a representação das expressões gênicas por diferentes embeddings, com diferentes pontos decimais de precisão, e com a aplicação de um modelo redutor de ruído. Os resultados obtidos mostram um bom resultado com expressões gênicas de três pontos de precisão, e uma grande variedade na aplicação do modelo de redução de ruído para diferentes embeddings.

Palavras-chave: Geração de moléculas, ruído em expressão gênica, Transformers;

Abstract

The generation of new molecules and molecular compounds is a task that has a wide range of applications in various areas of science, and is highly relevant for the development of drugs for diseases that still have no cure. With the “de novo” design, machine learning algorithms are capable of creating new molecules using Generative Artificial Intelligence. However, the gene expression used for this process has data with a lot of noise and that is difficult to map. This work seeks to explore the way in which gene expressions are represented, exploring the representation of gene expressions by different embeddings, with different decimal points of precision, and with the application of a noise reduction model. The results obtained show a good result with gene expressions of three precision points, and a wide variety in the application of the noise reduction model for different embeddings.

Keywords: Molecule Generation, Gene Expression Noise, Transformers;

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

IA	Inteligência Artificial
GED	Gene Expression Denoiser
CADD	Computer-aided drug design methods
SMILES	Simplified Molecular Input Line Entry System
SELFIES	SELF-referecing Embedded Strings
GAN	Generative Adversarial Network
PLN	Processamento de Linguagem Natural
FFN	Feed-Foward Network
GPT	Generative Pre-trained Transformer
BERT	Bidirectional Encoder Representations from Transformers
TransGEM	Transformer-based model from Gene Expression to Molecules
FAME	Fragment-based conditionAl Molecular gEneration model
CVAE	Conditional Variational Autoencoder
VAE	Variational Autoencoder
NLL	Negative Log-Likelihood
QED	Quantitative Estimate of Drug-likeness
MLP	Multi-Layer Perceptron

Sumário

1 Introdução

O avanço da inteligência artificial ao longo dos anos vem possibilitando diversas descobertas nas mais diferentes áreas científicas. Na área agrícola, a IA já é utilizada para controle de pestes, sistemas de irrigação, e detecção de doenças em plantas (BANNERJEE et al., 2018). Na área médica, a inteligência artificial é capaz de atuar nos mais diversos setores, como cardiologia, gastrologia, neurologia, e tratamento de câncer (BRIGANTI; MOINE, 2020). Na área biomédica, a inteligência artificial vem se destacando em áreas como análise de metabolismo, estudo de proteomas e genomas, e na área molecular (GOMES; ASHLEY, 2023), com avanços científicos relacionados à identificação de genes ativados de doenças específicas e estudos sobre o impacto de pequenas moléculas no metabolismo (EKINS, 2016).

Dentro deste setor, a área de pesquisa e desenvolvimento (P&D) de drogas vem ganhando destaque com o tópico de geração de moléculas. As moléculas, que são estruturas de átomos que podem ser concatenados de diferentes maneiras, são importantes na criação de novos medicamentos para tratamento de doenças que até então não apresentam cura (SWINNEY; LEE, 2020) e a descoberta de novas moléculas poderiam trazer avanços científicos e medicinais importantes. Entretanto, o número possível de moléculas que podem existir é muito grande: enquanto o espaço químico conhecido, incluindo bancos de dados públicos e coleções corporativas, contém cerca de 100 milhões de moléculas, estima-se que, ao considerar apenas regras estruturais básicas, o número de moléculas existentes possíveis chegue até 10^{60}. Isso faz com que a taxa de sucesso nesta área seja relativamente baixa, com pesquisas de longa duração não e que não resultados satisfatórios (PEREIRA et al., 2021).

Além disso, este é um processo caro, pois uma nova molécula custa cerca de 2,6 bilhões de dólares para ser produzida, e demora cerca de 12 anos, na média, para ser concluída (CHAN et al., 2019). Há também um estudo que estima que apenas 5 entre 5000 drogas candidatas passam pelo teste pré-clínico e em testes humanos, e que apenas 1 dessas testadas atingem o mercado (MOUCHLIS et al., 2021a).

Nesse contexto é que surge o de novo design. O de novo design consiste em se gerar novos componentes químicos sem a necessidade de ter uma relação a priori, feito somente com base em um conjunto de regras. O termo “de novo” vem de “do início, do começo” o que significa que os componentes são criados sem terem um padrão inicial de geração (MOUCHLIS et al., 2021b). Isso possibilita um processo mais criativo de criação de componentes químicos, incluindo as moléculas.

Um dos métodos mais populares para se criar moléculas é por meio de tipagens

celulares e substâncias química. Dado um tipo celular e uma molécula de teste, aplica-se uma dose determinada de uma substância química específica e, após algum tempo, observar a variação de expressão gênica do tipo celular. Esta variação de expressão gênica (ou expressão genética) é comparada com perfis genéticos que demonstram ter certas características. Com isso, a diferença da expressão genética da molécula atuante mostra quais as características da molécula neste meio aplicado.

No contexto de aprendizado de máquina, um uso muito comum deste sistema é de utilizar uma base de dados – com a LINCS, por exemplo (SUBRAMANIAN et al., 2017) – e treinar um modelo que, dada uma molécula em um tipo celular, substância química e diferença gênica específica, possa criar moléculas que teriam este mesmo perfil nestas mesmas circunstâncias.

Entretanto, um problema patente neste processo é a qualidade dos pontos de expressão gênica (PHAM et al., 2021). Devido à tecnologia utilizada para captar as variações de expressão gênicas, e também pelo fato de elas serem feitas em grande escala, e somente uma vez, existe muito ruído nos dados coletados. Isto é um grande problema na área de aprendizado de máquina para geração de moléculas, pois vários modelos apresentam baixo desempenho por conta da qualidade dos dados (PRAVALPHRUEKUL et al., 2023).

1.1 Objetivo

Este trabalho tem como objetivo explorar a qualidade e representação dos dados de expressão gênica aplicados em diferentes embeddings, com a finalidade de melhorar o processo de criação de moléculas.

1.2 Objetivos específicos

Para atingir o objetivo principal, este trabalho foi dividido em três objetivos específicos:

1. Utilizar dados de perfis de expressões gênicas com valores mais específicos, isto é, com um número maior de casas decimais de representação;
2. Utilizar um modelo de redução de ruído nos dados de perfis de expressões gênicas;
3. Utilizar um modelo de redução de ruído nos dados de perfis de expressões gênicas com valores mais específicos de maior número de casas decimais.

Estes três objetivos específicos serão realizados também com o uso de diferentes embeddings para a representação da expressão gênica.

1.3 Divisão do documento

A seção 2 apresenta uma fundamentação teórica sobre o De Novo Design, representação de moléculas, sobre a arquitetura Transformers e do TransGEM, e sobre o modelo de redução de ruído de expressão gênica GED.

A seção 3 apresenta a revisão da literatura. Em específico, a subseção 1 explora modelos geradores de moléculas, enquanto a subseção 2 apresenta trabalhos relativos à redução de ruído em expressões genéticas.

A seção 4 apresenta a metodologia do trabalho. As subseções abordadas incluem a proposta de trabalho; os hiperparâmetros usados; as bases de dados utilizadas para os algoritmos TransGEM, FAME e do presente estudo; os experimentos realizados; e as medidas de avaliação do resultado.

A seção 5 apresenta os resultados obtidos nos três experimentos, cada um com detalhes específicos, como análise por embeddings, análise por maior valor por tipagem celular, e comparação por pontos de precisão.

A seção 6 apresenta uma discussão dos resultados obtidos, contendo principais descobertas, limitações e trabalhos futuros.

2 Fundamentação Teórica

Este capítulo apresenta os principais conceitos abordados neste trabalho, organizados em seis seções. A primeira explora a geração e descoberta de componentes químicos, com ênfase no funcionamento do *de novo* design. A segunda trata da representação de moléculas e expressões gênicas no ambiente computacional. Na terceira, são discutidos aspectos da Inteligência Artificial Generativa, com foco em embeddings e na arquitetura Transformers. A quarta seção detalha a arquitetura do TransGEM, explicando seu processo de embedding das expressões gênicas. Por fim, a quinta seção aborda a arquitetura FAME e seu modelo de redução de ruído GED.

2.1 Geração e Descoberta de componentes químicos

Dentro da área de biotecnologia, existem alguns métodos utilizados para geração de novos componentes. Um dos métodos mais populares para isso são os métodos de desenho de medicamentos assistidos por computador (CADD, Computer-aided drug design methods). Estes métodos tornaram-se uma ferramenta poderosa no processo de descoberta e desenvolvimento de fármacos e incluem abordagens baseadas em estrutura e em ligantes (MOUCHLIS et al., 2021a). A expansão das ferramentas de estrutura de detecção biológicas, como raios-X, ressonância magnética nuclear e microscopia eletrônica, acelerou mais ainda a descoberta de novas entidades químicas. No entanto, devido à complexidade relacionada à sua aplicação na área humana, como difícil obtenção de padrões biológicos em doenças, e incertezas relacionadas a novos tratamentos médicos, levaram à necessidade do desenvolvimento de métodos mais rigorosos para explorar o vasto espaço químico e facilitar a identificação de novas estruturas moleculares a serem sintetizadas.

2.1.1 De Novo Design

O De Novo Design refere-se à criação de novas entidades químicas que atendem a um conjunto de restrições, utilizando para isso algoritmos computacionais de crescimento, como algoritmos evolutivos, de fragmentação, ou inteligência artificial generativa (SCHNEIDER; SCHNEIDER, 2016). Ela é uma solução criada para a geração de novos componentes sem apresentar relações a priori entre os compostos (MOUCHLIS et al., 2021a): o próprio termo “de novo” significa “do começo, do início”, o que indica a criação de moléculas sem um modelo inicial (DEVI; SATHYA; COUMAR, 2015). Esta estratégia apresenta uma enorme vantagem de criar um espaço de busca mais amplo, visto sua maior liberdade na criação de estruturas inéditas (MOUCHLIS et al., 2021b).

Algoritmos evolutivos são uma possibilidade de aplicar essa estratégia (DEVI; SATHYA; COUMAR, 2015). Esses algoritmos focam em otimização populacional baseados na área de evolução biológica (GOLBERG, 1989) e apresentam diversas abordagens, como algoritmos genéticos, programação genética, programação evolutiva e estratégias evolutivas. No contexto do design de medicamentos, é criada uma população de estruturas ou conformações, e cada membro dessa população é codificado por um cromossomo gerado aleatoriamente. O ciclo começa com a geração de uma população "pai" a partir de uma população inicial criada de forma aleatória (ou estocástica). Cada "pai" passa por transformações aleatórias usando operadores genéticos para gerar uma nova população de estruturas, chamadas "filhos".

Os dois principais operadores genéticos utilizados são mutação e crossover. A mutação introduz novas informações para gerar populações inéditas, enquanto o crossover usa essas informações para criar novos indivíduos na população de estruturas candidatas. Uma função de aptidão é então aplicada para avaliar o score de ligação de cada estrutura "filho" Com base nesse score, uma nova geração de "pais" é selecionada a partir da combinação das populações de "pais" e "filhos" iniciais. Essa nova população de "pais" é usada no próximo ciclo. Esse processo é repetido até que um critério de término seja alcançado (CLARK; WESTHEAD, 1996). Os algoritmos MEGA (KUMAR et al., 2018) e EvoMD (WONG; LUO; CHAN, 2010) são exemplos algoritmos evolucionários com o De Novo Design para geração de moléculas.

Durante o COVID, o De Novo Design foi amplamente utilizado, como em (CHENTHAMARAKSHAN et al., 2020) com o método CogMol, e em (TANG et al., 2022) com o método Q-learning network. Além disso, com o avanço do Aprendizado Profundo, novos algoritmos surgiram para criação de componentes, como Redes Neurais Recorrentes, Redes Neurais Convolucionais, Autoencoders, e IA Generativa (MOUCHLIS et al., 2021b).

2.2 Moléculas e expressão gênica no ambiente de Computação

2.2.1 Conjunto de Dados de Moléculas

O Library of Integrated Network-based Cellular Signatures (LINCS) é um programa fundado para mapear respostas celulares a vários tipos de perturbação em células humanas específicas – como resposta ao contato com uma droga nova, por exemplo (SUBRAMANIAN et al., 2017). A tecnologia utilizada em sua criação é o L1000, o qual é uma tecnologia transcriptômica de baixo custo de de alta eficiência, capaz de gerar milhões de expressões genéticas.

Este conjunto de dados L1000 é capaz de mensurar a expressão gênica de 978 pontos de mRNA, e o valor de 11,350 novos pontos de gene são computacionalmente

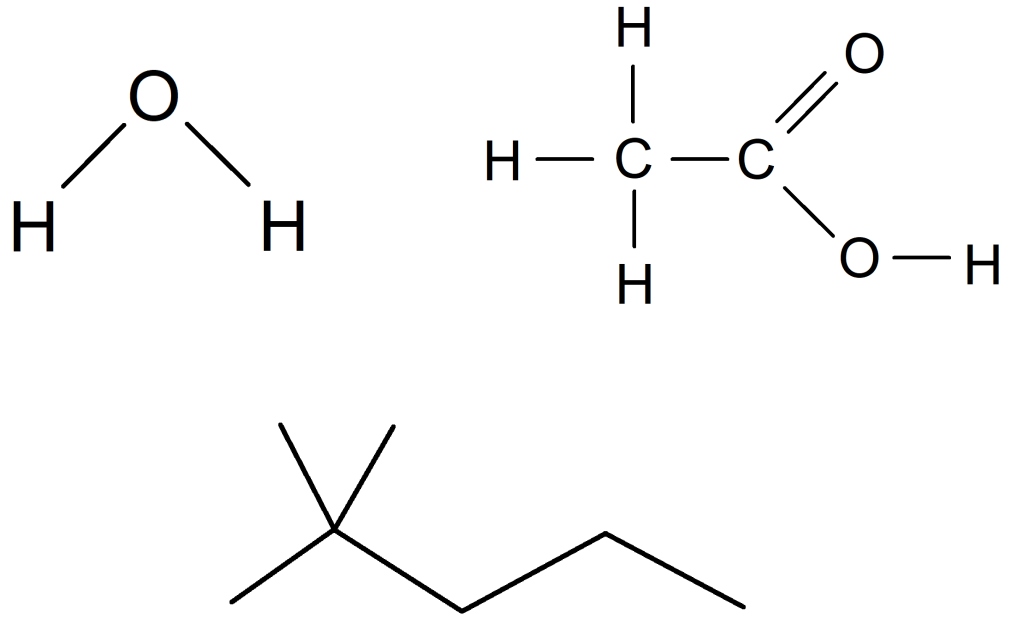
inferidos com base nestes (SUBRAMANIAN et al., 2017) – embora o valor não chegue nem mesmo à metade de todos os pontos de expressões genéticas dos mRNAs existentes.

Entretanto, estes dados apresentam um problema que se refere à grande quantidade de ruído, que é proveniente do próprio método de captura (MOUTSOPOULOS et al., 2021). Este ruído na informação de expressão gênica é difícil de se detectar, pois uma pequena alteração pode ser apenas um ruído, ou uma variação própria do gene (SHA; PHAN; WANG, 2015).

2.2.2 Representação de moléculas

As moléculas apresentam diversos tipos de representação, sendo as mais tradicionais a fórmula molecular (representa a composição química da molécula, como H_2O e $C_6H_{12}O_6$), a fórmula estrutural (ilustra como os átomos são ligados, como o diagramas de Lewis ou representações em bastão), e a nomenclatura IUPAC (nomeação padronizada para identificar moléculas com base na sua estrutura química). A Figura 1 apresenta algumas moléculas na notação de fórmula estrutural.

Figura 1 – Fórmula estrutural plana da água (H_2O), na parte superior esquerda; fórmula estrutural plana do ácido acético (CH_3COOH), na parte superior direita, e fórmula estrutural de esqueleto da 2,2-Dimetilpentano (C_7H_{16}) na parte inferior central.

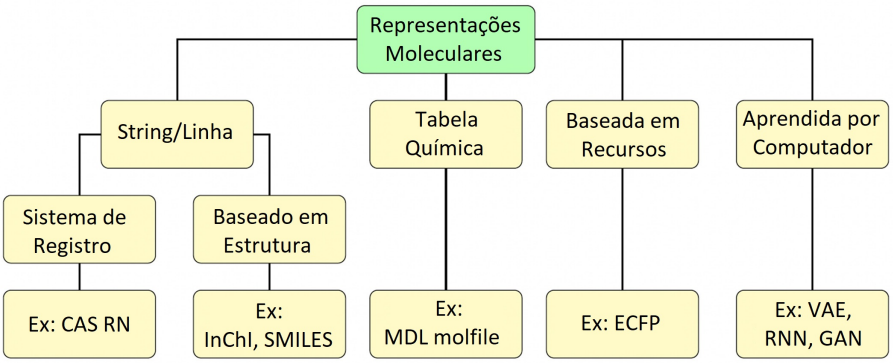


Fonte: Próprio autor.

Em um contexto computacional, há desafios envolvendo a representação de moléculas, haja visto que as moléculas apresentam propriedades as quais devem estar expressas em sua representação. Além disso, é esperado que cada representação de moléculas seja única.

Na área de Aprendizado de Máquina, há uma vasta gama de representações moleculares que podem ser usadas, como ilustrado na Figura 2.

Figura 2 – Tipos de representações moleculares.



Fonte: Adaptado de (WIGH; GOODMAN; LAPKIN, 2022)

As representações mais usadas são por uma string de texto, ou por grafos. Dentre as representações de texto, as mais famosas são a SMILES e a SELFIES.

2.2.2.1 SMILES

SMILES (*Simplified Molecular Input Line Entry System*, ou em potuguês, Sistema de Registros Moleculares Simplificados em Linha) é uma representação de moléculas em linha de texto (WEININGER, 1988). Ela representa os átomos de uma molécula através de caracteres alfanuméricos, com regras de encadeamento de elementos a fim de representar a geometria e conectividade das moléculas.

Nessa modelo, os átomos são representados pelo símbolo químico e as ligações como - para uma ligação simples, = para uma ligação dupla e ♯ para uma ligação tripla. A representação de estereoquímica também é possível com barras simples e barras invertidas.

Embora seja simples e prática, SMILES tem a limitação de que nem todas as moléculas representadas são verdadeiras. Isso faz com que, em modelos que utilizem esta representação, algumas moléculas geradas sejam inválidas. Esta é a representação mais antiga e mais comumente usada.

2.2.2.2 SELFIES

O SELFIES (SELF-referecing Embedded Strings, ou em português, Embedding de String com Auto-referência) é uma representação molecular de gramática livre (KRENN et al., 2020) que busca ser uma evolução da SMILES.

Sua principal inovação está em seu sistema de construção. Ele utiliza tokens que descrevem os átomos, as ligações, e a conectividade molecular, mas com regras de

composição que garantem a validade química da molécula resultante. Essa abordagem elimina o problema de invalidade estrutural, que é comum ao utilizar SMILES para geração de novas moléculas.

A Tabela 1 exibe exemplos de moléculas representadas pelas notações SMILES e SELFIES.

Tabela 1 – Exemplos de Moléculas nas Representações SMILES e SELFIES

Molécula	SMILES	SELFIES
Etanol	`CCO`	`[C][C][O]`
Acetona	`CC(C)=O`	`[C][C][C]=[O]`
Cicloexano	`C1CCCCC1`	`[C][C][C][C][C][C]`
Benzeno	`C1=CC=CC=C1`	`[C]=[C][C]=[C][C]=[C]`
Fenol	`C1=CC=CC=C1O`	`[C]=[C][C]=[C][C]=[C][O]`
Piridina	`C1=CC=NC=C1`	`[C]=[C][C]=[N][C]=[C]`
Ácido fórmico	`O=CO`	`[O]=[C][O]`
Ácido acético	`CC(=O)O`	`[C][C]=[O][O]`
Ureia	`C(=O)(N)N`	`[C]=[O][N][N]`
Anilina	`C1=CC=CC=C1N`	`[C]=[C][C]=[C][C]=[C][N]`
Nitrobenzeno	`C1=CC=CC=C1[N+](=O)[O-]`	`[C]=[C][C]=[C][C]=[C][N+]=[O][O-]`
Glicina	`NCC(=O)O`	`[N][C][C]=[O][O]`

2.2.3 Expressão Gênica

O perfil de expressão gênica (ou expressão genética) de compostos biológicos é um meio usado para caracterizar fenótipos celulares e de organismos. Dada uma perturbação química (isto é, a adição de uma molécula) em uma tipo celular específico, alterações no perfil de expressão gênica da célula podem ocorrer. Esta estratégia já foi amplamente utilizada para descobertas científicas, como descoberta de mecanismos de ação de medicamentos (WEI et al., 2006), identificação dos principais compostos em um objeto químico (HASSANE et al., 2008), reposicionamento de medicamentos (KOSAKA et al., 2013), e previsão de efeitos colaterais de medicamentos (STEGMAIER et al., 2004).

O Connectivity Map (CMap) (LAMB, 2007) é uma plataforma de pesquisa criada para explorar como diferentes compostos químicos poderiam afetar a expressão gênica das células. A plataforma foi capaz de induzir a expressão gênica quimicamente de cinco linhagens celulares humanas diferentes por mais de 1.300 compostos após 6 horas (PHAM et al., 2021). O L1000 é uma extensão do CMap também desenvolvido pelo LINCS que apresenta mais de 1 milhão de de perfis de expressão gênica em resposta de mais de 50 linhagens celulares humanas, com mais de 20.000 compostos.

Entretanto, existem vários problemas importantes com os dados do L1000. Embora o número de perfis de expressão gênica seja muito maior do que o do CMap, há ainda muitos valores de expressão gênicas ausentes na relação de compostos e linhagens celulares, além de existirem centenas de milhões de compostos químicos semelhantes a medicamentos e adquiríveis que são candidatos potenciais a medicamentos (STERLING; IRWIN, 2015), o que torna inviável testar experimentalmente todos os compostos para seus perfis de expressão gênica induzidos quimicamente em várias linhagens celulares.

Por fim, devido a diversos problemas experimentais, como a captura dos dados ter sido feita em lote, muitas medições experimentais não são confiáveis. Esses sérios obstáculos limitaram a eficácia e o escopo do uso do conjunto de dados L1000 na descoberta de medicamentos. Prever valores de expressão gênica para experimentos não medidos e não confiáveis é, portanto, necessário (PHAM et al., 2021).

2.3 Inteligência Artificial Generativa

A inteligência artificial generativa tem se consolidado como um dos avanços mais significativos no campo da inteligência artificial nos últimos anos. Esse campo engloba técnicas capazes de gerar novos conteúdos, como textos, imagens, música (ECK; SCHMIDHUBER, 2002) e vídeos (SRIVASTAVA; MANSIMOV; SALAKHUDINOV, 2015), com base em dados de entrada, sem a necessidade de supervisionamento explícito. Em vez de apenas reconhecer ou classificar padrões, a IA generativa visa criar algo novo e original, o que abre uma infinidade de possibilidades em diversas áreas, como arte, design, saúde, e marketing (PERES et al., 2023). O avanço das redes neurais profundas, especialmente as Redes Generativas Adversariais (GANs), tem sido um dos principais fatores para essa revolução, permitindo que sistemas criem outputs realistas e complexos, indistinguíveis dos criados por humanos (FEUERRIEGEL et al., 2024).

Os modelos generativos podem ser agrupados em modelos unimodais e multimodais. Modelos unimodais recebem instruções do mesmo tipo de entrada que sua saída (por exemplo, texto). Por outro lado, modelos multimodais podem receber entradas de diferentes fontes e gerar saídas em várias formas. Modelos multimodais existem em diversas modalidades de dados, como texto, imagem e áudio. Alguns exemplos incluem o Stable Diffusion (ROMBACH et al., 2022) para geração de imagem a partir de texto, o MusicLM (AGOSTINELLI et al., 2023) para geração de música a partir de texto, e o Codex (CHEN et al., 2021) e o AlphaCode (LI et al., 2022) para geração de código a partir de texto.

O processo de treinamento varia consideravelmente entre os diferentes modelos de IA generativa. As GANS, por exemplo, são treinadas através de dois modelos competidores (GOODFELLOW et al., 2014), sendo um criar novas amostras sintéticas enquanto o outro tenta detectar amostras sintéticas das amostras reais de treinamento, de forma

que a distribuição das amostras sintéticas se aproxime da distribuição das amostras de treinamento. De maneira diferente, sistemas como os modelos conversacionais utilizam aprendizado por reforço a partir de feedback humano, qual ocorre em três etapas: primeiro, cria-se um conjunto de dados de demonstração para prompts; depois, os usuários classificam a qualidade de diferentes saídas para um prompt; e, finalmente, aprende-se uma política que gera saídas desejáveis por meio do aprendizado por reforço, de modo que a saída obtenha uma boa classificação durante a avaliação (BROWN et al., 2020).

2.3.1 Embeddings

Os embeddings são vetores densos de tamanho fixo que representam um dado para o modelo (TURIAN; RATINOV; BENGIO, 2010). Ele é crucial na área de Inteligência Artificial Generativa, pois permite transformar uma sequência de uma linguagem humana para uma linguagem numérica eficiente, a fim de que o modelo criado possa entender a sequência. Os embeddings também buscam posicionar elementos similares mais próximos do espaço contínuo, a fim de mostrar maior relação entre objetos similares (MIKOLOV et al., 2013).

Os embeddings se destacaram na área de Processamento de Linguagem Natural (PLN) principalmente com o surgimento de modelos como o Word2Vec (MIKOLOV et al., 2013) e GloVE (PENNINGTON; SOCHER; MANNING, 2014). O primeiro utiliza abordagens como *Skip-gram* e *Continuous Bag of Words (CBOW)* para aprender representações de palavras a partir de grandes quantidades de texto. Já o GloVe baseia-se em estatísticas globais de coocorrência para gerar representações mais interpretáveis. Uma limitação dos embeddings nesta área é a falta de conhecimento contextual, pois palavras polissêmicas possuem a mesma representação, mesmo com contextos diferentes (por exemplo, o modelo não sabe diferenciar a palavra “banco” no contexto financeiro ou no contexto de assento).

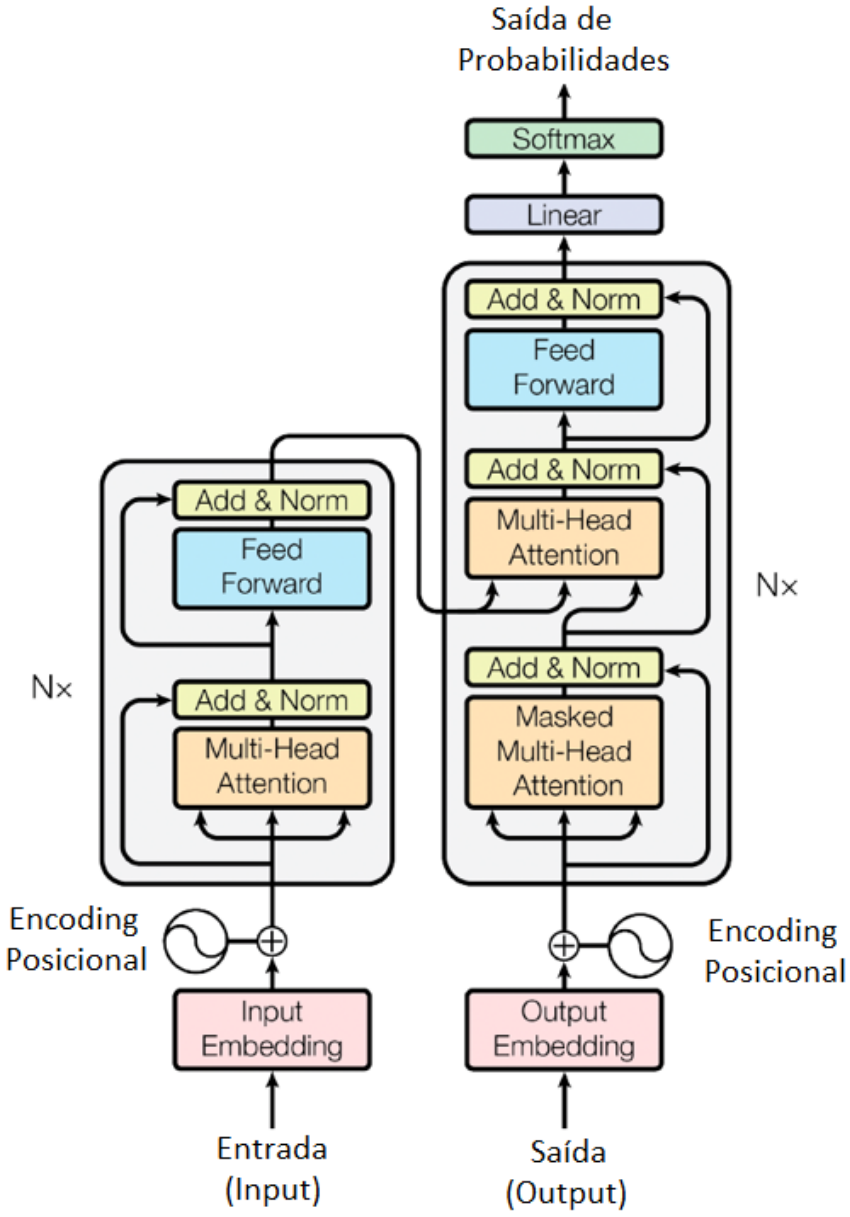
Os embeddings também surgem em contextos diferentes dos de PLN, como na área de imagens e grafos (GROVER; LESKOVEC, 2016). É comum também a aplicação de embeddings em contextos multimodais como de imagem e texto (RADFORD et al., 2021; DESAI; JOHNSON, 2021).

2.3.2 Transformers

O Transformers (VASWANI, 2017) é uma arquitetura de Inteligência Artificial Generativa projetado para processar dados de entrada sequenciais, como a linguagem natural, com aplicações em tarefas como tradução e sumarização de texto. Esta arquitetura revolucionou a área de Inteligência Artificial Generativa pois, seu processamento dos dados de entrada pode ser feito simultaneamente, o que permite um paralelismo significativo nas etapas de treinamento e de inferência (DEVLIN, 2018). A arquitetura é apresentada na

Figura 3.

Figura 3 – Arquitetura Transformers.



Fonte: Adaptado de Vaswani (2017).

O modelo do Transformers é dividido em 3 partes: um codificador, um decodificador, e um gerador. O codificador é responsável por transformar a entrada em uma representação contínua e contextualizada, utilizando para isso camadas de autoatenção e Redes Neurais Feed-Foward (FNN). O decodificador recebe a representação gerada pelo codificador e a utiliza para aprender padrões de predição, por meio do mecanismo de autoatenção mascarada, em que o token atual é predito com base dos tokens anteriores (BROWN et al., 2020). O gerador é a camada final do Transformer, responsável por transformar a saída do decodificador em uma distribuição de probabilidade sobre um determinado vocabulário,

permitindo a seleção do próximo token. Essa seleção pode ser feita de diferentes formas, como argmax, amostragem estocástica ou métodos mais sofisticados como top-k sampling (RADFORD et al., 2019).

Uma característica crucial da arquitetura é o embedding posicional. Como a entrada do modelo é analisada ao mesmo tempo, um mecanismo de posição dos tokens é extremamente necessário para que o modelo entenda a ordem desse dado de entrada. Isso é feito por meio do embedding posicional, o qual injeta uma informação de posição para cada token antes de aplicar o mecanismo de atenção. Essa informação é comumente obtida por meio dos embeddings sinusoidais, os quais adicionam uma função senoide e cossenoide aos tokens, com diferentes frequências em cada posição, a fim de indicar temporalidade. O vetor de posição P para um token na posição i pode ser gerado pelas seguintes fórmulas:

$$P(i, 2j) = \sin\left(\frac{i}{10000^{2j/d}}\right), \tag{2.1}$$

e

$$P(i, 2j+1) = \cos\left(\frac{i}{10000^{2j/d}}\right), \tag{2.2}$$

onde i é a posição do token, j é o índice da dimensão do vetor de posição, e d é o número de dimensões dos embeddings. Esse método permite que o modelo capture relações de posição tanto locais (vizinhança imediata de tokens) quanto globais (distâncias mais longas entre os tokens), de maneira eficiente e contínua.

Modelos baseados em Transformers tem obtido resultados muito significativos de melhora e de desempenho. Atualmente, as arquiteturas mais famosas de PLN, o GPT (Generative Pre-trained Transformer) (RADFORD et al., 2019) e o BERT (Bidirectional Encoder Representations from Transformers) (DEVLIN, 2018), utilizam esta arquitetura.

2.3.2.1 Mecanismo de Atenção

O conceito de atenção tem origem na área de neurociência, a qual afirma que o cérebro humano dá um peso maior a informações mais importantes o qual está a informação buscada em relação ao contexto todo – como um texto ou uma imagem (VASWANI, 2017). Nesse ínterim, o mecanismo de atenção busca imitar essa característica cerebral e fazer com que a máquina consiga dar peso maior a características mais importantes.

Esse mecanismo é implementado através do mapeamento de uma consulta (Query Q) a um conjunto de chaves (Keys K) e seus respectivos valores (Values V). Para cada termo Q, avalia-se a influência que K exerce sobre ele, ponderando sua relação com V. A fórmula matemática que representa essa operação é dada por:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{2.3}$$

onde d_k representa a dimensão de K. O termo $\sqrt{d_k}$ é utilizado para evitar que os valores das pontuações de atenção cresçam excessivamente, o que poderia prejudicar a obtenção de gradientes estáveis durante o treinamento do modelo.

2.3.2.2 Mecanismo de atenção múltipla

Uma das inovações fundamentais da arquitetura Transformers é o mecanismo de atenção múltipla (*Multi-Head Attention*), que permite ao modelo capturar diferentes relações de atenção simultaneamente. Enquanto a atenção simples foca em uma única perspectiva dos dados, a atenção múltipla utiliza vários cabeçalhos (heads), cada um aprendendo padrões distintos de interação entre os tokens.

Cada cabeçalho processa uma projeção diferente das matrizes Q, K e V, permitindo que o modelo identifique relações em diferentes espaços semânticos. Os resultados dessas cabeças de atenção são então concatenados e transformados por uma matriz de projeção final W^O, conforme descrito na equação abaixo:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{2.4}$$

$$\text{onde head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.5}$$

2.4 TransGEM

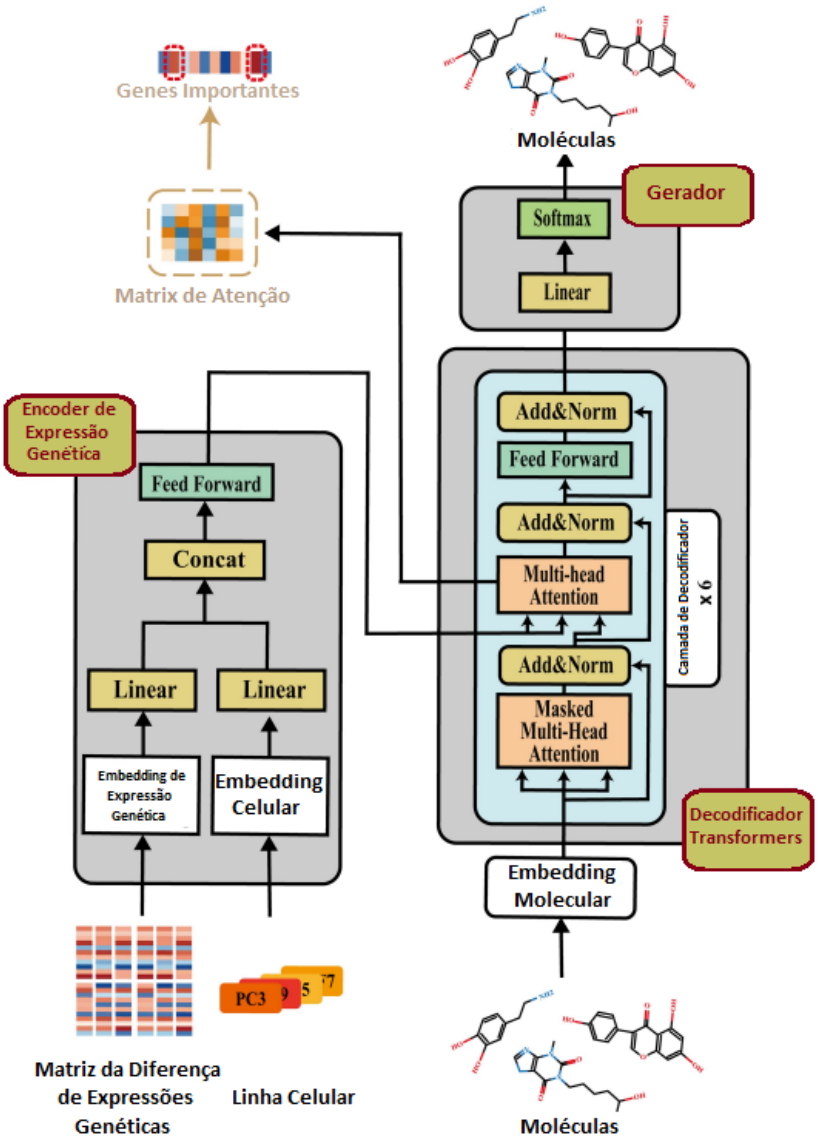
O algoritmo TransGEM (Transformer-based model from Gene Expression to Molecules) (LIU et al., 2024) é uma arquitetura baseada no modelo Transformers com a finalidade de gerar moléculas relacionadas a uma variação específica de sua expressão gênica. Sua motivação principal é de gerar moléculas específicas para determinadas alterações de sequência de genes, a fim de caracterizar mudanças fenotípicas.

Assim como a arquitetura Transformers, a arquitetura do TransGEM conta com três principais componentes: o codificador de expressão gênica, o decodificador, e o gerador. Essa geração conta com três principais dados de entrada: a molécula, a alteração de expressão gênica na célula afetada pela molécula, e o tipo da molécula. A Figura 4 apresenta a arquitetura do modelo.

2.4.1 Codificador da Expressão Gênica

O codificador da expressão gênica contém duas informações: a linhagem celular, e a expressão genética por si mesma. Após cada uma delas passarem por sua camada de

Figura 4 – Arquitetura do TransGEM.



Fonte: Adaptado de (LIU et al., 2024)

embedding, elas são unidas e dadas como entrada para o modelo decodificador TransGEM, conforme a fórmula 2.6.

$$G_o = FNN(linear(Concat(v_c, v_e)), \tag{2.6}$$

na qual v_c representa o embedding da linhagem celular, e v_e representa a expressão gênica.

A linhagem celular é inserida no modelo através de one-hot encodding através da seguinte fórmula:

$$v_c = Linear(\text{One-hot}(C)), \tag{2.7}$$

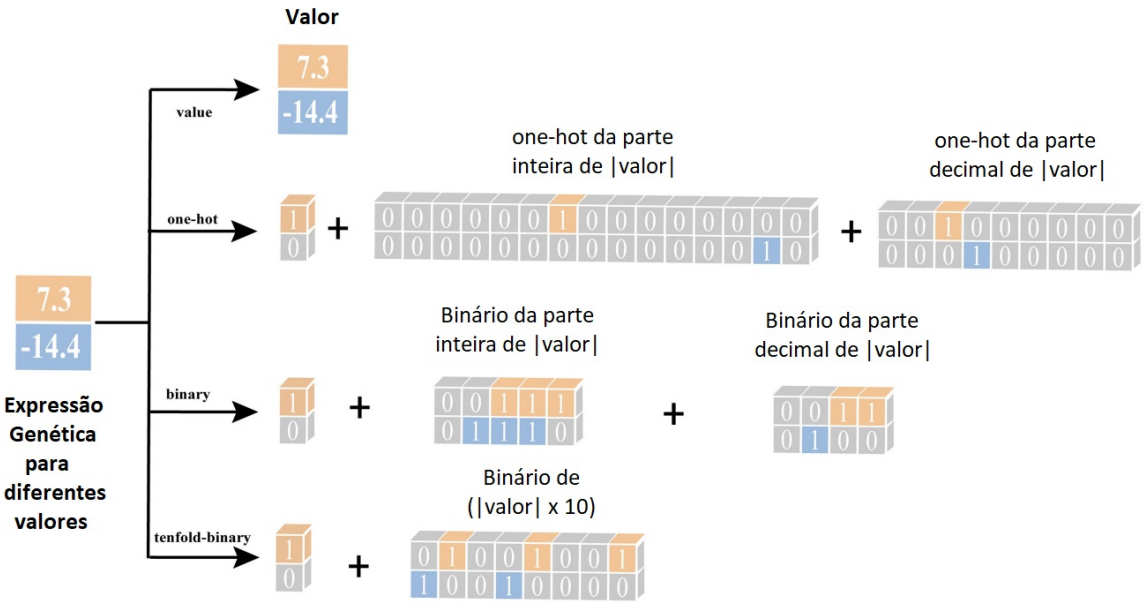
onde $Linear(*)$ representa a uma camada de linearização, e One-hot(*) é uma camada de one-hot encodding.

A expressão gênica apresenta quatro tipos de embeddings $f_{emb}(e_i)$ possíveis: *Values*, *One-hot*, *Binary* e *Tenfold-Binary*. O tipo *Values* (1) é o qual o valor da expressão gênica é passado normalmente, sem nenhuma alteração; o tipo *One-hot* (2) é utilizando one-hot encoder – com uma parte representando o valor inteiro, e outra representando o valor após a virgula; o tipo *Binary* (3) é tornando a expressão gênica binária – também com uma parte representando o valor inteiro, e outra o valor após a virgula); e o tipo *Tenfold-Binary* (4) é representado pela multiplicação do valor de expressão gênica por dez, seguida pela binarização do número. Após aplicada a função específica de embedding, o valor é concatenado e gera o embedding final:

$$v_e = Linear(f_{emb}(E)) \tag{2.8}$$

No trabalho de Liu et al. (2024), a diferença de expressão gênica apresenta somente um ponto decimal. A Figura 5 ilustra o funcionamento do embedding para cada entrada de expressão gênica específica:

Figura 5 – Representação de cada Embedding do TransGEM.



Fonte: Adaptado de Liu et al. (2024).

A forma matemática com que cada valor de $f_{emb}(e_i)$ é montado para a instância de expressão gênica e_i é apresentada a seguir para esses quatro tipos de embedding:

1. *Values*: Valor puro da expressão gênica

$$f_{emb}(e_i) = e_i \tag{2.9}$$

2. *One-hot*: One-hot encodding dos valores inteiros e decimais

$$emb_1 = \begin{cases} 1, & \text{se } e_i > 0 \\ 0, & \text{se } e_i \leq 0 \end{cases} \tag{2.10}$$

$$emb_2 = \text{One-hot}\left(\lfloor e_i \rfloor\right) \tag{2.11}$$

$$emb_3 = \text{One-hot}((|e_i \cdot \lfloor e_i \rfloor|) \times 10) \tag{2.12}$$

$$f_{emb}(e_i) = \text{Concat}(emb_1, emb_2, ..., emb_{d+1}) \tag{2.13}$$

3. *Binary*: Forma binária dos valores inteiros e decimais

$$emb_1 = \begin{cases} 1, & \text{se } e_i > 0 \\ 0, & \text{se } e_i \leq 0 \end{cases} \tag{2.14}$$

$$emb_2 = \text{Pad}(\text{Binário}\left(\lfloor e_i \rfloor\right), 5) \tag{2.15}$$

$$emb_3 = \text{Pad}(\text{Binário}(|e_i \cdot \lfloor e_i \rfloor|) \times 10^2), 4) \tag{2.16}$$

$$f_{emb}(e_i) = \text{Concat}(emb_1, emb_2, emb_3) \tag{2.17}$$

4. *Tenfold-Binary*: Binarização de e_i vezes 10

$$emb_1 = \begin{cases} 1, & \text{se } e_i > 0 \\ 0, & \text{se } e_i \leq 0 \end{cases} \tag{2.18}$$

$$emb_2 = \text{Pad}(\text{Binário}(e_i \times 10, 9)) \tag{2.19}$$

$$f_{emb}(e_i) = \text{Concat}(emb_1, emb_2) \tag{2.20}$$

Os autores apresentam que os diferentes tipos de embedding para expressão gênica foram capazes de melhorar o desempenho do modelo, principalmente em relação à unicidade das moléculas geradas, e da divisão interna delas. A Tabela 2 apresenta os resultados obtidos com o algoritmo do TransGEM para cada um dos embeddings utilizados. As métricas apresentadas na tabela serão mais detalhadas na seção 4.

Tabela 2 – Resultados obtidos pelo algoritmo TransGEM.

Embedding	**Validade**	**Novidade**	**Unicidade**	**Div. Interna**
Values	99.9%	100%	59.9%	46.9%
One Hot	100%	100%	54.0%	63.4%
Binary	100%	100%	63.6%	71.0%
Tenfold-Binary	100%	100%	84.9%	78.9%

2.4.2 Decodificador do Transformers

O decodificador é utilizado para integrar os resultados advindos do embedding da molécula com os da expressão do gene. Ele apresenta N camadas de decodificação que utilizam em N máscaras de multi-head self-attention, seguido por uma FFN. A fórmula 2.5 é utilizada tendo V_N como a união do embedding da molécula com os da expressão do gene expressão gênica.

O embedding da molécula V_{N-1} passa pelo mecanismo de atenção e gera V'_{N-1} (Equação 2.21). Após isso, V'_{N-1} e a expressão gênica codificada G_0 passam por outro mecanismo de atenção de N camadas (Equação 2.22.

$$V'_{N-1} = V_{N-1} + \text{Attention}(V_{N-1}, V_{N-1}, V_{N-1}) \tag{2.21}$$

$$V'_N = V'_{N-1} + \text{Attention}(V'_{N-1}, G_0, G_0) \tag{2.22}$$

Por fim, V'_N é atualizado para V_N com base em uma Rede FFN:

$$V_N = V'_N + \text{FNN}(V'_N) \tag{2.23}$$

2.4.3 Gerador

O gerador consiste na última fase da arquitetura, em que as moléculas são finalmente geradas. As moléculas M' geradas são obtidas pela linearização de V_N, seguida de uma camada do Softmax, como mostrado na seguinte equação:

$$M' = \text{Softmax}(\text{Linear}(V_N)) \tag{2.24}$$

2.4.4 Função de Erro

A divergência de Kullback–Leibler (KL), também conhecida como entropia relativa (KULLBACK; LEIBLER, 1951), é uma medida estatística usada para quantificar a diferença entre duas distribuições de probabilidade P (a distribuição verdadeira) e Q (a distribuição estimada). Ela é amplamente utilizada em aprendizado de máquina para avaliar o quão bem a distribuição de probabilidade prevista Q aproxima a distribuição verdadeira P.

Divergência KL da molécula gerada $\hat{M}$ para a molécula original M é dada pela seguinte fórmula:

$$D_{KL}(M, \hat{M}) = \sum_{i=0}^{L} P(i) \log \frac{P(i)}{Q(i)}, \tag{2.25}$$

onde P(i) é a probabilidade verdadeira do i-ésimo evento, e Q(i) é a probabilidade prevista do i-ésimo evento.

Essa medida é assimétrica, o que significa que $D_{KL}(P||Q) \neq D_{KL}(Q||P)$. Ela avalia a ineficiência de assumir Q como a distribuição verdadeira quando a distribuição real é P.

Em aprendizado de máquina, particularmente em tarefas de classificação e modelos generativos, a divergência KL é usada para comparar a distribuição alvo P com a distribuição prevista Q do modelo (LIU; YAO, 2022).

2.5 FAME

O Modelo de Geração de Moléculas Condicional baseado em Fragmentos (FAME, Fragment-based conditionAl Molecular gEneration model) foi desenvolvido por Pham, Xie e Zhang (2022), e é um algoritmo de aprendizado de máquina baseado em grafos para gerar moléculas a partir da variação genéticas com base no Autocodificador Variacional Condicional (CVAE, Conditional Variational Autoencoder).

O CVAE é uma variação dos Autocodificador Variacional (VAE). O VAE apresenta um codificador que diminui a dimensionalidade do dado de entrada X, e o decodificador, o qual amplia este espaço até o tamanho da saída. O CVAE é uma variação na qual um rótulo Y é colocado no dado reduzido, fazendo com que o decodificador gere o dado do rótulo Y como base no dado X.

No algoritmo do FAME, a molécula, na forma de SMILES, é representada em forma de grafo. Este grafo passa por uma a rede de grafos isomórfica de 5 camadas. A variação genética é passada como rótulo para o modelo, fazendo com que uma molécula nova baseada nessa expressão de gene seja criada.

2.5.1 Redução de ruido de expressão gênica

Uma outra contribuição desse trabalho foi o modelo de redução de ruido de expressão gênica (GED, Gene Expression Denoising model). Este modelo tem como objetivo reduzir o ruído presente no dado de variação genética. Para isso, os autores utilizaram uma função objetivo contrastiva para mapear perfis de expressão gênica em um espaço de hiperesfera unitária. Essa abordagem força o agrupamento das representações de expressão gênica de um mesmo composto químico enquanto, simultaneamente, as afasta das representações de outros compostos químicos nesse espaço, ajudando, assim, a reduzir o ruído nos dados de expressão gênica.

Em particular, um perfil de expressão gênica y_l, induzido por um composto químico c_l, é projetado para o espaço de hiperesfera unitária como $\hat{y}_l = \text{NORM}(\text{PROJ}(y_l))$, onde PROJ é a rede de projeção e NORM é o operador de normalização que garante que a representação aprendida esteja no espaço dimensional dos embeddings. Em seguida, a função objetivo contrastiva é aplicada:

$$L_{CL} = -\sum_{l=1}^{L} \frac{1}{|P(l)|} \sum_{p \in P(l)} \log \frac{\exp\left(\hat{y}_l \cdot \hat{y_p}/\rho\right)}{\sum_{a \in \mathbb{A}(l)} \exp\left(\hat{y}_l \cdot \hat{y}_a/\rho\right)}, \tag{2.26}$$

na qual $\mathbb{A}(l) \equiv \{1, 2, \ldots, L\} \setminus \{l\}$ é o conjunto de todos os índices exceto l, $\mathbb{P}(l) \equiv \{p \in \mathbb{A}(i) : c_p = c_l\}$ é o conjunto de índices de toda a expressão gênica induzida pelo composto c_l, e $\hat{y}_a$ e $\hat{y}_p$ são as projeções de perfil de expressão gênica nos os conjuntos $\mathbb{A}(l)$ e $\mathbb{P}(l)$, respectivamente.

Além disso, a rede PROJ foi modelada como uma FFN com conexões de atalhos entre as camadas, como mostra a equação 2.27. Aqui, a entrada na camada de número o é a concatenação das camadas anteriores ocultas.

$$h_o = FFN_o(CONCAT(h_1, h_2, ..., h_{o-1})) \tag{2.27}$$

O GED foi criado contendo 64 camadas FFN, com taxa de crescimento de 16%, com uma entrada de tamanho 978, e uma saída de tamanho 64.

3 Revisão da Literatura

Esta seção apresenta uma revisão da literatura sobre o tema; em específico, uma seção aborda os algoritmos geradores de moléculas, enquanto a outra discute sobre trabalhos cujo objetivo é reduzir o ruído de expressões gênicas.

3.1 Geradores de Moléculas

No trabalho de Méndez-Lucio (MÉNDEZ-LUCIO et al., 2020), os autores desenvolveram uma arquitetura baseada em GANs. Esta rede apresenta dois discriminadores, um para avaliar a associação entre moléculas geradas e perfis de expressão gênica, e outro para verificar sua autenticidade. O modelo apresentou um baixo percentual de moléculas válidas.

O algoritmo BiAAE (*Bidirectional Adversarial Autoencoder*, em português, Autoencoders Adversiáis Bidirecionais), desenvolvido em Shayakhmetov et al. (2020), é um gerador de moléculas baseado em Autoencoders Adversiáis Supervisionados. O algoritmo apresenta três redes neurais: um codificador, um gerador, e um descriminador; o codificador faz um mapeamento da molécula de entrada em uma representação latente, e o gerador reconstrói a molécula com base na diferença de expressão gênica. Por fim, o discriminador é treinado para distinguir se a molécula gerada foi válida ou não, e seu feedback é usado para melhorar o desempenho do gerador. Os autores descobriram que duas perdas discriminadoras adicionais ajudam o modelo a aprender uma parte compartilhada mais expressiva e garantem que todas as três partes sejam mutuamente independentes, apresentando ótimos resultados.

O modelo PaccMannRL (BORN et al., 2021) é um modelo multimodal de aprendizado por reforço que utiliza um modelo de previsão de sensibilidade a medicamentos anticâncer como função de recompensa. O modelo é dividido em duas partes: um gerador condicional de moléculas, composto por dois VAEs separados (KINGMA, 2013), e um modelo crítico (PaccMann) que faz a avaliação das predições. O modelo foi inovador por gerar compostos anticancerígenos mesmo sem ter sido expressamente treinado com esses dados (somente o crítico que foi treinado desse modo); entretanto, algumas propriedades secundárias para ser um bom composto anticancerígeno – como solubilidade em água, similaridade com fármacos existentes, e facilidade de síntese – não foram diretamente otimizadas.

Em Liao et al. (2023), os autores buscam a criação de um modelo com base em scaffolds. Os scaffolds (“esqueleto”, em português) é uma estrutura central de uma molécula

que é usada no estudo da construção de novas moléculas. Os autores utilizam duas etapas para isso: primeiro, é gerado um conjunto de scaffords apenas com átomos de carbono, por meio de uma VAE (KINGMA, 2013); após isso, os scaffolds gerados são melhorados por meio da substituição de átomos e regiões de borda por meio de um Transformers (VASWANI, 2017). O modelo foi capaz de obter resultados melhores que os da literatura, e também foi capaz de criar regras de transformações de scaffolds automaticamente.

Em Mazuz et al. (2023) é proposto o modelo Taiga, o qual é um modelo de geração de moléculas baseado em Transformers (VASWANI, 2017) com aprendizado por reforço. O modelo é dividido em duas etapas: na primeira, o modelo aprende a fazer o embedding das moléculas em um espaço vetorial; na segunda etapa, o modelo otimizado o espaço vetorial para gerar moléculas com as características específicas desejadas. A otimização é feita com a métrica de QED (Quantitative Estimate of Drug-likeness, ou estimativa quantitativa da semelhança de drogas). O modelo apresentou resultados melhores que os do estado-da-arte para a métrica de QED.

O modelo CogMol (CHENTHAMARAKSHAN et al., 2020) é um framework criado no contexto de combate ao SARS-CoV-2 (popularmente conhecido como Covid-19). Seu acrônimo vem de Geração de Moléculas Controladas (Controlled Generation of Molecules, em inglês) e tem por objetivo não só criar moléculas farmacêuticas de alta afinidade com proteínas virais inéditas, mas também de obter um alto valor de seletividade off-target (minimizar iterações indesejadas com outras proteínas do organismo). Para isso, o framework utiliza uma VAE (KINGMA, 2013) pré-treinada baseada em SMILES (WEININGER, 1988), e uma amostragem eficiente de seleção de atributos que guiam as propriedades desejadas da molécula. Os resultados foram muito positivos, com alto número drogas geradas com baixa toxidade prevista, e alta viabilidade sintética. Além disso, o modelo tem aplicação para qualquer proteína viral inédita, não só a do SARS-CoV-2. O modelo DockingGA (GAO et al., 2024) propõe o uso de algoritmos genéticos em união com a arquitetura Transformers (VASWANI, 2017). Isso é feito por meio de uma política de aprendizado parametrizado pelo Transformers que captura os dados complexos da molécula de entrada; em seguida, simulações de Docking são geradas com os dados complexos recebidos, para serem simuladas em diferentes cenários e, com base em um alvo, produzirem novas moléculas. O algoritmo obteve 100% de novidade em seus resultados.

O modelo Gex2SGen (Expressão Genética para Geração de SMILES, Gene Expression 2 SMILES Generation, em inglês), desenvolvido em Das et al. (2023), é um modelo que utiliza duas VAE pré-treinadas no seu processo e geração; entretanto, ele se destaca por ser um modelo o qual recebe como entrada um perfil de expressão gênica, e retornar uma molécula em formato SMILES (WEININGER, 1988). Para isso, os autores treinaram uma VAE que aprende a representar expressões gênicas em um espaço latente (p-VAE); em seguida, outra rede é criada para transformar dados deste espaço latente em moléculas

do tipo SMILES (SMILES-VAE); por fim, os modelos são combinados, e a saída de um se torna a entrada do outro. Embora o algoritmo tenha sido capaz de gerar moléculas com o perfil gênico desejado, o trabalho foi prejudicado pela falta de dados de alta qualidade de expressão gênica, e pela falta de variedade de dosagens, tipagens celulares, e tempo de reação da molécula na célula.

O modelo BiCEV (PRAVALPHRUEKUL et al., 2023) também utiliza um autoencoder que recebe alterações de expressões gênicas, e retorna moléculas SMILES geradas. O modelo apresentou altas taxas de validade, novidade, e diversidade interna. Entretanto, novamente o modelo foi afetado pela qualidade dos dados de treinamento de expressões gênicas, as quais apresentavam muito ruído.

A tabela 3 apresenta o resultados obtido por alguns dos modelos comentados.

Tabela 3 – Desempenho de alguns dos modelos dos modelos citados.

Modelo	**Validade**	**Novidade**	**Unicidade**	**Div. Int.**
CGAN	8.5%	-	-	-
BiAAE	76.0%	-	-	-
FAME	83.8%	99.8%	86.6%	86.7%
PaccMannRL	84%–93%	-	-	-
BiCEV	95.7%	100%	98.2%	77.0%
Gex2SGen	45%	88%	95%	-
TransGEM	100%	100%	84.9%	78.9%

3.2 Redução de ruído de expressões genéticas

Em (JEON et al., 2024), é apresentado o Denoiseit, um modelo de aprendizado de máquina que busca remover potencias outliers, ao invés de selecionar genes com maior informação genética. O algoritmo foi capaz de diminuir o nível de ruído técnico presente na expressão gênica, enquanto manteve os genes de maior relevância biológica comparados com os métodos existentes.

Em (MOUTSOPOULOS et al., 2021) é apresentado o noisyR, que aplica um filtro adaptativo baseado em uma análise da variação da distribuição do sinal, e busca com isso trazer maior consistência informacional entre amostras e réplicas de sequenciamento gênico. O modelo permite detectar limiares de ruído específicos para cada amostra, e removê-los. Os resultados mostram que o noisyR reduz significativamente os efeitos estocásticos, melhorando a precisão de análises de expressão diferencial e a inferência de redes regulatórias, especialmente em dados de células únicas.

O modelo OutSingle (Detector de Outliers usando Decomposição de valores Únicos, Outlier detection using Singular Value Decomposition, em inglês) é um método para detectar outliers em dados de contagem, como a expressão gênica de RNA (SALKOVIC et al., 2023). Ele é composto por dois passos: primeiro, os dados são transformados em escala logarítmica, e z-scores específicos são calculados para cada gene; após isso, o modelo aplica um método para controlar variáveis de confusão utilizando SVD (*singular value decomposition*, em português, decomposição em valores singulares) e um OHT (*optimal hard threshold*, em português, limiar rígido ótimo) para eliminar ruídos (GAVISH; DONOHO, 2014). Este processo traz inovação ao OutSingle em relação aos seus predecessores e torna o processo determinístico, mais eficiente e mais rápido.

Em (BHUKYA, 2021), o autor utiliza autoencoders profundos para redução de ruído de expressão gênica. Sua arquitetura conta com uma camada de codificação, três camadas ocultas, e uma camada de decodificação. O modelo teve uma baixa taxa de melhora em relação aos seus pares.

Em (XIE et al., 2017), uma união do Perceptron Multicamada (Multi Layer Perceptron) e do decodificador automático de eliminação de ruído empilhado (Stacked Denoising Auto-encoder) é realizada, formando o método MLP-SAE (*Multilayer Perceptron with Stacked Denoising Auto-encoder*, em português, Perceptron Multicamada com decodificador automático de eliminação de ruído empilhado). Este método utiliza o auto-encoder para gerar, a partir do gene de entrada x, uma gene de saída x' que é similar a x, mas que apresenta um menor ruído. Ele é utilizado como uma etapa de pré-treino das camadas de MLP, o que melhora desempenho da descida do gradiente durante o treinamento supervisionado. O modelo foi capaz de performar melhor do que seus pares que utilizavam Random Forest e Regressão Linear com Lasso.

Em (JEON et al., 2022) é apresentado um algoritmo que tem por objetivo expandir o número de expressões de gene, a fim de gerar mais informação. O modelo tem como entrada a marcação de 978 expressões genéticas e apresenta como saída 23.614 sequências de expressões genéticas. A transformação é feita utilizando uma rede adversária generativa cíclica (CycleGAN) e uma rede neural totalmente conectada. Esta estratégia gerou bons valores de correlação de Pearson e RMSE.

4 Metodologia

O objetivo desse capítulo é informar sobre a proposta de trabalho, a base de dados, os experimentos, os hiperparâmetros, e as medidas de avaliação utilizadas neste trabalho. O código-fonte da aplicação encontra-se disponível no repositório do *GitHub*, Casarotto (2025), acessível em: <https://github.com/PedroHCasarotto/TCC>.

4.1 Proposta do Trabalho

Este trabalho tem como objetivo explorar o uso dos embeddings de expressão gênica do algoritmo do TransGEM e buscar formas de tornar a representação mais fidedigna e precisa, a fim de aumentar o desempenho do modelo de geração de moléculas. Em específico, o trabalho busca isso de três formas: explorando o uso de variações genéticas mais precisas, utilizando o modelo de Redução de Ruido GED de expressão gênica, e utilizando o modelo de Redução de Ruido de expressão gênica GED em conjunto com variações genéticas mais precisas.

4.1.1 Explorando o uso de variações genéticas mais precisas

Conforme apresentado na secção 2, o algoritmo do TransGEM apresenta quatro tipo de embeddings para a geração de moléculas. Neste trabalho, uma adaptação deles foi realizada, de modo com que fossem capazes de suportar até seis pontos decimais de expressão gênica.

As alterações feitas em cada tipo de embedding foram as seguintes, sendo d o número de casas decimais da expressão gênica e, e_{max} o valor de expressão gênica máximo e e_i a i-nésima posição de expressão genética:

1. *Values*: Valor puro da expressão gênica; sem alterações.

$$f_{emb}(e_i) = e_i \tag{4.1}$$

2. *One-hot*: One-hot encodding dos valores inteiros e decimais, tendo cada casa decimal seu próprio embedding.

$$emb_1 = \begin{cases} 1, & \text{se } e_i > 0 \\ 0, & \text{se } e_i \leq 0 \end{cases} \tag{4.2}$$

$$emb_{d+1} = \text{One-hot}((|e_i \cdot \lfloor e_i \rfloor|) \times 10^d) \tag{4.3}$$

$$f_{emb}(e_i) = \text{Concat}(emb_1, emb_2, ..., emb_{d+1}) \tag{4.4}$$

3. *Binary*: Forma binária dos valores inteiros e decimais, tendo cada casa decimal seu próprio embedding.

$$emb_1 = \begin{cases} 1, & \text{se } e_i > 0 \\ 0, & \text{se } e_i \leq 0 \end{cases} \tag{4.5}$$

$$emb_2 = \text{Pad}(\text{Binário}\,(\lfloor e_i \rfloor), 4) \tag{4.6}$$

$$emb_{d+2} = \text{Pad}\left(\text{Binário}\left(\left\lfloor 10^d \times e_i \right\rfloor \mod 10\right), 4\right) \tag{4.7}$$

$$f_{emb}(e_i) = \text{Concat}(emb_1, emb_2, ..., emb_{d+2}) \tag{4.8}$$

4. *Tenfold-Binary*: Binarização de e_i vezes d.

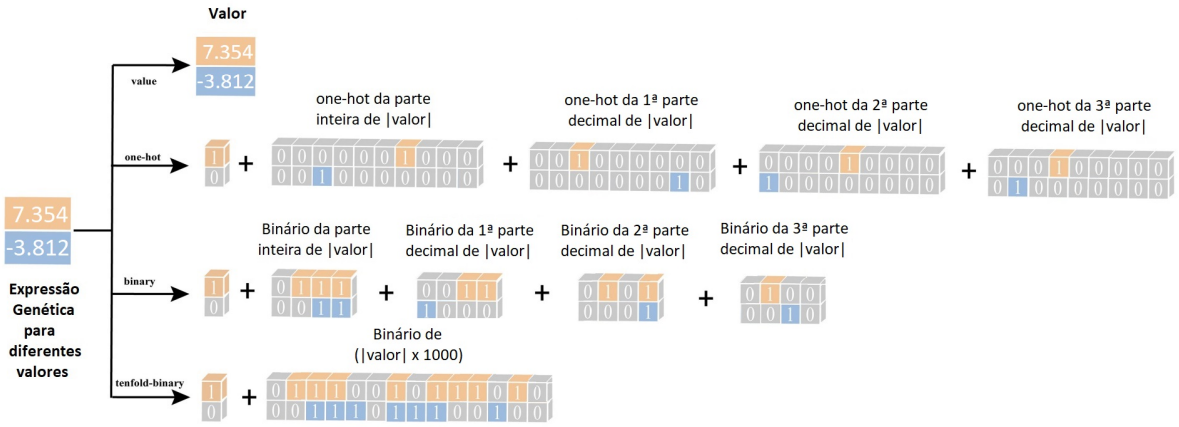
$$emb_1 = \begin{cases} 1, & \text{se } e_i > 0 \\ 0, & \text{se } e_i \leq 0 \end{cases} \tag{4.9}$$

$$emb_2 = \text{Pad}(\text{Binário}(e_i \times 10^d, (e_{max} \times 10^d))) \tag{4.10}$$

$$f_{emb}(e_i) = \text{Concat}(emb_1, emb_2) \tag{4.11}$$

A Figura 6 apresenta dois exemplos similares ao mostrado na Figura 5, só que considerando três pontos de precisão e um intervalo de expressão gênica de até o valor 10 para valores inteiros.

Figura 6 – Representação de cada Embedding do TransGEM aplicada à proposta.



Fonte: Próprio autor.

4.1.2 Utilizando o modelo de Redução de Ruido de expressão gênica

No algoritmo FAME (PHAM; XIE; ZHANG, 2022), o modelo de Redução de Ruído aplicado às expressões gênicas antes do modelo principal é capaz de melhorar o desempenho de geração de moléculas; em específico, o algoritmo foi capaz de diminuir a probabilidade logarítmica negativa (NLL, *Negative Log-Likelihood*), mostrando melhora no aprendizado da distribuição, e o FCD (*Fréchet ChemNet Distance*), que evidencia associações mais fortes de moléculas com perfis de expressão gênica.

Devido a isso, optou-se também por utilizar o modelo de Redução de Ruído de expressão gênica para gerar dados de alteração de expressões genéticas com menor ruído e, com isso, analisar o resultado obtido.

4.1.3 Utilizando o modelo de Redução de Ruido de expressão gênica em variações genéticas mais precisas

O último experimento visa aplicar o modelo de Redução de Ruído de expressão gênica em expressões gênicas mais precisas, com maior número de pontos decimais.

4.2 Base de dados

O L1000 é uma plataforma que mede a expressão de 1.000 genes de referência em resposta a diferentes perturbações celulares e químicas. O dataset LINCS1000 (SUBRAMANIAN et al., 2017) é uma base de dados gerada por meio dessa plataforma que contém assinaturas de expressões genéticas de milhares de perturbações através de diferentes

condições. Sua função é de explorar estudar diferentes comportamentos em perturbações de células e moléculas. O dataset do LINCS1000 é dividido em 5 níveis:

1. Expressão gênica bruta obtida diretamente dos experimentos, contendo 978 genes de referência.
2. Aplicação de técnicas de normalização e remoção de outliers para melhorar a qualidade dos dados.
3. Dados completamente normalizados e ajustados, prontos para análise. Esse é o nível mais utilizado para modelos de aprendizado de máquina, pois mantém a fidelidade dos experimentos enquanto remove ruídos indesejados.
4. Expressões agregadas por amostras biológicas replicadas, consolidando medições de múltiplas execuções experimentais.
5. Dados altamente processados e inferidos, incluindo a extrapolação da expressão de 11.350 genes adicionais a partir dos 978 genes de referência usando modelos computacionais.

Na área de criação de modelos de aprendizado de máquina, o uso mais comum do nível 3, visto que apresenta os resultados normalizados, e que o uso de nível 5 se mostrou ter muito ruído e tornar a inferência do modelo mais ineficiente (PHAM et al., 2021).

Nos conjuntos de dados abaixo, todos utilizam 978 marcações de pontos de expressão gênica.

4.2.1 Dataset utilizado no algoritmo TransGEM

No trabalho proposto o TransGEM, é apresentado um dataset com dados do L1000 chamado de subLINCS. Este dataset apresenta 6929 moléculas únicas sendo afetadas em 14 tipos de célula e em 6950 diferentes substâncias. Todas as marcações genéticas foram feitas com uma dose de $10\mu M$ com duração de 24 horas. A representação utilizada no subLINCS foi a SELFIES.

4.2.2 Dataset utilizado no algoritmo FAME

O dataset utilizado pelo algoritmo FAME é também uma amostra do LINCS1000 que foi usado em Méndez-Lucio et al. (2020), o qual apresenta 31.821 expressões genéticas do nível 5 induzidas por 19.768 componentes dos tipos celulares MCF7 e VCAP com doses de 5 e $10\mu M$ depois de 24 horas de exposição. O dataset foi dividido entre treino, validação e teste em uma taxa de 80:10:10, respectivamente. A representação de moléculas utilizadas foi a SMILES.

Para a criação do GED, utilizou-se expressões gênicas do nível 4.

4.2.3 Dataset Utilizado neste estudo

O dataset utilizado nos experimentos foi o mesmo dataset utilizado no FAME, porém com algumas modificações de pré-processamento, a fim de torná-lo mais semelhante com os experimentos feitos com o algoritmo TransGEM.

O pré-processamento consistiu nas seguintes etapas:

1. Retirar moléculas repetidas. Esta etapa se dá pelo fato de uma mesma molécula na mesma perturbação poderia resultar em expressões genéticas diferentes. Desse modo, optou-se por retirar esses casos do estudo e manter somente as moléculas únicas no dataset para cada tipo celular.

2. Alterar a representação molecular de SMILES para SELFIES. Em (KRENN et al., 2020) mostra-se que a representação do SELFIES apresenta melhores resultados que o SMILES, além de gerar apenas moléculas válidas e melhorar a diversidade interna do conjunto gerado.

3. Selecionar moléculas para teste. As moléculas utilizadas para a geração foi ao todo 14 moléculas, 7 do tipo celular MCF7 e 7 do VCAP, todas com dose de $10\mu M$. Para o resultado de geração de novos componentes, escolheu-se a perturbação de dois componentes nos tipos celulares:

   a) Testosterona: um hormônio essencial para o desenvolvimento de características sexuais primárias e secundárias, principalmente para os animais do sexo masculino. Estudos recentes sugerem que ela influencia interações sociais ao estimular a busca e manutenção de status. Pesquisas com administração controlada de testosterona em humanos mostraram seu impacto em processos emocionais, como vigilância a ameaças, confiança e inferência emocional (EISENEGGER; HAUSHOFER; FEHR, 2011).

   b) Dexametasona: um componente corticosteroide usado no tratamento de diversas doenças, entre as quais reumatismo, várias doenças da peles, alergias graves, asma, doença pulmonar obstrutiva crônica, crupe, edema cerebral. Sua relevância se dá devido ao componente ter sido muito utilizado e estudado contra o COVID-19, depois de mostrar resultados de redução de mortalidade em pacientes com a doença que fizeram o tratamento (AHMED; HASSAN, 2020).

4. Divisão de dados de treinamento e validação. Para isso, utilizou-se uma proporção de 11% de dados de treino e teste para MCF7 e VCAP. A divisão foi feita de modo com que os compostos de perturbação não se repetissem no treino nem no teste.

Após o pre-processamento, dois datasets foram gerados: o dataset com a expressão gênica de 978 pontos de genes, e o dataset do resultado obtido após a aplicação do GED, com um vetor de genes transformados representados por 64 pontos. Ambos os datasets apresentam os pontos de expressão gênica com 6 pontos decimais. Além disso, o dataset original varia seu valor entre -10 e 10, enquanto o dataset após aplicado o GED varia entre -0.45 e 0.53.

Tabela 4 – Características dos Datasets.

Dataset	**# Moléculas**	**# Células**	**# Substâncias**	**# Treino**	**# Validação**	**# Teste**
TransGEM	6929	14	6950	25364	2000	14
FAME	19768	2	19222	25658	3291	2872
FAME adaptado	18590	2	18159	21939	2462	14

4.3 Experimentos

Os experimentos realizados neste trabalho foram três:

1. Estudar a influência dos pontos de precisão na geração de moléculas, com base em expressões gênicas com um, três e seis pontos decimais;
2. Aplicar o GED nos pontos da variação da expressão gênica nos algoritmos de entrada do TransGEM.
3. Aplicar o GED nos pontos da variação da expressão gênica nos algoritmos de entrada do TransGEM com base em expressões gênicas com três e seis pontos decimais;

4.4 Hiperparâmetros

O modelo contará, como dimensões de entrada, os seguintes valores para o conjunto de tipo celular e expressão gênica, sendo n_{IM} o valor inteiro máximo do conjunto de dados, e n_{PD} o número de pontos decimais utilizado. A soma do valor 2 representanta o embedding de One-Hot encodding dos 2 tipos celulares utilizados, e o 1 representando o sinal – positivo ou negativo – da expressão gênica.

- *Values*: $2 + 1$
- *One-Hot*: $2 + 1 + \lceil n_{IM} \rceil + 9 \times n_{PD}$
- *Binary*: $2 + 1 + \lceil \log_2 n_{IM} \rceil + 4 \times n_{DP}$
- *Tenfold-Binary*: $2 + 1 + \lceil \log_2 10^{\lceil n_{IM} \rceil \times 10^{n_{DP}}} \rceil$

O resto dos hiperparâmetros foi mantido o mesmo do artigo original de Liu et al. (2024), que foram obtidos por meio de uma busca em grade : o número de neurônios na camada oculta após o embedding da expressão gênica é 64; para o número de entradas de moléculas, o tamanho é fixado em 52. O número de camadas do decodificador Transformer e de atenção multi-head são definidos, respectivamente, como 6 e 8. A dimensão da camada feed-forward é definida como 512. O número de épocas de treinamento, o tamanho do batch e a taxa de aprendizado são definidos como 200, 4 e 0,0001, respectivamente.

4.5 Ferramentas

Todos os experimentos foram realizados utilizando a linguagem de programação Python, que é amplamente usada para aplicações de aprendizado de máquina. A versão Python utilizada foi a 3.8. As bibliotecas utilizadas podem ser vistas dentro do arquivo de *environment.yaml*, no *GitHub* (CASAROTTO, 2025).

O algoritmo TranGEM foi adaptado do código original, disponível em <https://github.com/hzauzqy/TransGEM>. Os dados originais do algoritmo FAME com e sem a aplicação do modelo GED também foram obtidas do código original do autor, disponível em <https://github.com/pth1993/FAME>.

4.6 Medidas de Avaliação

As medidas de avaliação são aplicadas diretamente nas moléculas geradas pelo algoritmo, e são as seguintes:

1. Número de moléculas únicas geradas (por padrão, o algoritmo gera 3000 moléculas; entretanto, o número de moléculas únicas é menor).

2. Validade: Se refere à proporção de moléculas válidas geradas. Ela é definida como:

$$\text{Validade} = \frac{N_{\text{válidas}}}{N_{\text{geradas}}}, \tag{4.12}$$

   onde $N_{\text{válidas}}$ representa o número de moléculas geradas que são estruturalmente corretas, e N_{total} é o número total de moléculas geradas.

3. Unicidade: É a taxa de moléculas não repetidas dividido pela taxa de moléculas repetidas. Ela é definida como:

$$\text{Unicidade} = \frac{N_{\text{únicas}}}{N_{\text{válidas}}}, \tag{4.13}$$

   onde $N_{\text{únicas}}$ é o número de moléculas distintas geradas, e $N_{\text{válidas}}$ é o número total de moléculas válidas.

4. Novidade: É a proporção de moléculas geradas que não estavam no conjunto de dados de treinamento e nem validação; que foram moléculas geradas fora do espaço pré-conhecido pelo modelo. Ela é definida como:

$$\text{Novidade} = \frac{1}{N_{\text{únicas}}} \sum_{i=1}^{N_{\text{únicas}}} \nVdash(m_i \notin D_{\text{treino}} \cup D_{\text{validação}}), \tag{4.14}$$

onde $\nVdash(m_i \notin D_{\text{treino}} \cup D_{\text{validação}})$ é uma função indicadora que assume valor 1 se a molécula m_i não estiver no conjunto de dados de treinamento (D_{treino}) ou validação ($D_{\text{validação}}$), e 0 caso contrário.

5. Diversidade Interna (InDiv): É a medida de diversidade interna de um grupo de moléculas, obtido pela seguinte fórmula:

$$\text{IntDiv}(M) = 1 - \sqrt{\frac{1}{|M|^2} \sum_{m_1, m_2 \in M} \text{smi}\ (m_1, m_2)}, \tag{4.15}$$

na qual M indica um grupo de moléculas, m_1 e m_2 um subconjunto delas, e *sim* representa alguma métrica de similaridade. Neste estudo, a similaridade foi calculada por pela Similaridade de Tanimoto (BAJUSZ; RÁCZ; HÉBERGER, 2015).

6. QED médio: Estimativa quantitativa de similaridade de drogas (*Quantitative Estimation of Drug-likenes*), que é uma medida de avaliação do quão próxima uma droga é da outra, e é dada por esta fórmula:

$$QED = \exp\left(\frac{1}{n} \sum_{i=1}^{n} w_i \ln d_i\right), \tag{4.16}$$

na qual d_i representa a *função de desejabilidade* para a propriedade i da molécula, que está contido no intervalo entre 0 e 1, w_i é o peso atribuído a cada propriedade, e n é o número total de propriedades consideradas. A métrica final contem a média do QED para todas as moléculas geradas.

5 Análise e discussão dos resultados

Esta seção apresenta os resultados obtidos nos experimentos. Os valores em negrito das tabelas mostram os melhores valores obtidos.

5.1 Resultados gerais obtidos

O resultado mais expressivo obtido foi que, para todos os experimentos realizados, a métrica de novidade atingiu 100%, ou seja, todas as moléculas geradas, em todos os casos, não haviam sido apresentadas nem na etapa de treinamento e nem na de validação.

Além disso, todas as moléculas geradas em todos os experimentos apresentaram 100% de validade. Isso ocorreu pois a representação de moléculas utilizada foi o SELFIES, o qual tem por característica sempre gerar moléculas válidas.

Por estes motivos, as métrica de validade e novidade foram omitidas nas tabelas e nas discussões particulares.

5.2 Uso de variações genéticas mais precisas

A Tabela 5 exibe os resultados comparativos obtidos para o uso de 1, 3 e 6 pontos decimais de precisão de expressões genéticas para diferentes embeddings de expressão gênica. A última coluna apresenta a médias das 3 métrica coletadas.

Tabela 5 – Tabela comparativa por embedding do uso de diferente pontos decimais na expressão gênica.

Embedding (PD)	# Moléculas	Unicidade	Div. Interna	QED Médio	Média Métricas
Values (1)	190.00±177.70	0.06±0.06	0.79±0.06	0.74±0.08	0.53
Values (3)	1530.36±915.00	**0.51±0.30**	0.82±0.02	**0.80±0.02**	**0.71**
Values (6)	74.21±148.02	0.02±0.05	**0.89±0.04**	0.52±0.09	0.48
One-Hot (1)	512.07±354.31	0.17±0.12	**0.78±0.02**	**0.80±0.04**	0.58
One-Hot (3)	3000.00±00	**1.00±0.00**	0.35±0.05	0.42±0.03	0.59
One-Hot (6)	1083.13±608.13	0.36±0.20	0.71±0.17	0.73±0.07	**0.60**
Binary (1)	914.29±1077.74	0.30±0.36	0.39±0.30	0.41±0.13	0.37
Binary (3)	1711.57±352.37	**0.57±0.12**	**0.75±0.08**	0.71±0.01	**0.68**
Binary (6)	1591.71±526.48	0.53±0.18	0.68±0.06	**0.81±0.05**	0.67
Tenfold-Binary (1)	2081.57±485.79	**0.69±0.16**	0.38±0.16	0.51±0.18	0.53
Tenfold-Binary (3)	1683.36±142.84	0.56±0.05	**0.72±0.02**	**0.73±0.01**	**0.67**
Tenfold-Binary (6)	49.21±41.73	0.02±0.01	0.71±0.13	0.45±0.11	0.39

As Tabelas 6, 7 e 8 apresentam os resultados comparativos de cada embedding para o uso de 1, 3 e 6 pontos decimais de precisão de expressão gênica, respectivamente.

Tabela 6 – Métrica da média do grupo de moléculas geradas para 1 ponto decimal.

Embeddng	**# Moléculas**	**Unicidade**	**Div. Interna**	**QED Médio**
Values	190.00±177.70	0.06±0.06	**0.79±0.06**	0.74±0.08
One-Hot	512.07±354.31	0.17±0.12	0.78±0.02	**0.80±0.04**
Binary	914.29±1077.74	0.30±0.36	0.39±0.30	0.41±0.13
Tenfold-Binary	2081.57±485.79	**0.69±0.16**	0.38±0.16	0.51±0.18
Média	924.48±523.88	0.31±0.18	0.59±0.13	0.62±0.11

Tabela 7 – Métrica da média do grupo de moléculas geradas para 3 pontos decimais.

Embeddng	**# Moléculas**	**Unicidade**	**Div. Interna**	**QED Médio**
Values	1530.36±915.00	0.51±0.30	**0.82±0.02**	**0.80±0.02**
One-Hot	3000.00±00	**1.00±0.00**	0.35±0.05	0.42±0.03
Binary	1711.57±352.37	0.57±0.12	0.75±0.08	0.71±0.01
Tenfold-Binary	1683.36±142.84	0.56±0.05	0.72±0.02	0.73±0.01
Média	1981.82±352.05	0.66±0.12	0.66±0.04	0.67±0.02

Tabela 8 – Métrica da média do grupo de moléculas geradas para 6 pontos decimais.

Embeddng	**# Moléculas**	**Unicidade**	**Div. Interna**	**QED Médio**
Values	74.21±148.02	0.02±0.05	**0.89±0.04**	0.52±0.09
One-Hot	1083.13±608.13	0.36±0.20	0.71±0.17	0.73±0.07
Binary	1591.71±526.48	**0.53±0.18**	0.68±0.06	**0.81±0.05**
Tenfold-Binary	49.21±41.73	0.02±0.01	0.71±0.13	0.45±0.11
Média	699.06±331.09	0.23±0.11	0.75±0.10	0.63±0.08

5.2.1 Análise de Ponto Decimal

De forma geral, é possível ver que o uso de três pontos decimais na expressão gênica foi o que mais gerou altos valores de unicidade, isto é, de moléculas únicas criadas. Todos

tiveram uma taxa de unicidade maior que 50%, com destaque para o One-Hot encodding, que conseguiu uma taxa de 100% de moléculas únicas em todas as 14 moléculas de teste.

Quando analisada a Diversidade Interna, é possível ver que o uso de 6 pontos decimais na expressão gênica foi o que mais gerou grupo de moléculas diversos uns dos outros. Além disso, o aumento de casas decimais gerou um aumento nesta métrica (com exceção para o embedding *One-Hot*).

Quando comparamos o QED médio por grupo de molécula, é possível ver que o valor médio foi muito diferente para cada tipo de embedding utilizado.

5.2.2 Análise de Embedding

É interessante notar que cada embedding apresentou respostas diferentes ao uso de diferentes pontos decimais na expressão gênica. O embedding *Values* apresentou um desempenho médio melhor para o uso de 3 pontos decimais. Além disso, foi o experimento com maior valor de média das métricas de unicidade, divisão interna e QED médio.

O embedding *One-Hot* encodding apresentou resultados melhores de diversidade interna e QED médio com o uso de 1 ponto decimal; além disso, o uso de três pontos decimais gerou resultados destoantes do uso desse embedding para 1 e 6 pontos decimais, apresentado elevado aumento na métrica de Unicidade, mas grande queda na diversidade interna e no QED médio. Entretanto, a média simples das três médias mostra grande similaridade entre o uso dos 3 pontos decimais.

O embedding *Binary* encodding apresentou resultados bons principalmente quando utilizando três pontos decimais. Mesmo assim, o uso de 6 pontos decimais apresentou um resultado muito similar ao uso de 3 quando obtida a média simples das três medidas. Seu pior resultado foi com a utilização de somente 1 ponto decimal.

O *Tenfold-Binary* também mostrou resultados ótimos quando utilizados 3 pontos decimais, sendo inclusive superior à média simples das métrica apresentadas.

5.3 Uso do modelo de Redução de Ruido de expressão gênica

A Tabela 9 exibe os resultados comparados de moléculas geradas utilizando o modelo de Redução de Ruido de expressão gênica com 1 ponto decimal. A última coluna apresenta a médias das 3 métrica coletadas.

5.3.1 Análise de Embeddings

A aplicação do GED trouxe resultados diferente para os modelos que utilizaram o algoritmo do *Values* e o do *One* Hot para os que usaram o *Binary* e o *Tenfold-Binary*.

Tabela 9 – Comparação de geração de moléculas com e sem a aplicação do GED, incluindo a média das métricas.

Embedding	GED	# Moléculas	Unicidade	Div. Interna	QED Médio	Média
Values	Não	190.00±177.70	0.06±0.06	**0.79±0.06**	**0.74±0.08**	0.53
	Sim	893.43±667.32	**0.30±0.22**	0.64±0.21	0.66±0.14	0.53
	Ganho (%)	-	+400.0	-19.0	-10.8	0.0
One-Hot	Não	512.07±354.31	**0.17±0.12**	**0.78±0.02**	**0.80±0.04**	**0.58**
	Sim	279.50±66.02	0.09±0.02	0.71±0.03	0.69±0.03	0.50
	Ganho (%)	-	-47.1%	-9.0%	-13.8%	-13.8
Binary	Não	914.29±1077.74	**0.30±0.36**	0.39±0.30	0.41±0.13	0.37
	Sim	543.14±260.30	0.18±0.09	**0.62±0.07**	**0.72±0.08**	**0.51**
	Ganho (%)	-	-40	+59.0	+75.6	+37.8
Tenfold-Binary	Não	2081.57±485.79	**0.69±0.16**	0.38±0.16	0.51±0.18	0.53
	Sim	1350.57±1307.75	0.45±0.44	**0.62±0.19**	**0.63±0.09**	**0.57**
	Ganho (%)	-	-34.8	+63.2	+23.5	+7.5

No primeiro grupo, o resultado de diversidade interna dos grupos e o QED médio caiu para 14.9% para *Values* e 11.4% para *One-Hot* na média das métricas. Já para o número de moléculas únicas geradas, somente o embedding de *Values* apresentou um ganho, o qual foi um ganho significativo de 400%. O embedding de *One-Hot* apresentou grande queda no valor de unicidade.

No segundo grupo, o uso do GED trouxe grande aumento na diversidade interna das molécula (61.1% de aumento médio em ambos); embora o aumento tenha sido acompanhado de uma grande queda na unidade de 37.4% na média. Em relação ao QED médio, o valor de ambos os embeddings tiveram um aumento na métrica, com destaque para *Binary* que obteve um ganho de 75.6%.

Com isso, é possível ver que o GED apresentou impacto positivo para a diversidade interna e o QED médio dos embeddings *Binary* e *Tenfold-Binary*.

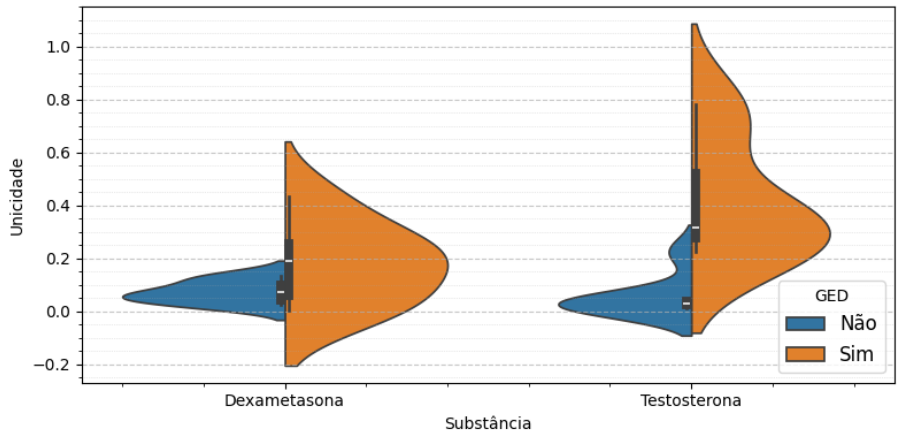
5.3.2 Análise de Moléculas por Componente Químico

Esta subseção apresenta a análise da distribuição das métricas para cada molécula utilizada pelas substâncias da testosterona e da dexametasona nos dados de teste. Cada substância teve 7 moléculas diferentes utilizadas para originarem novas moléculas.

5.3.2.1 Values

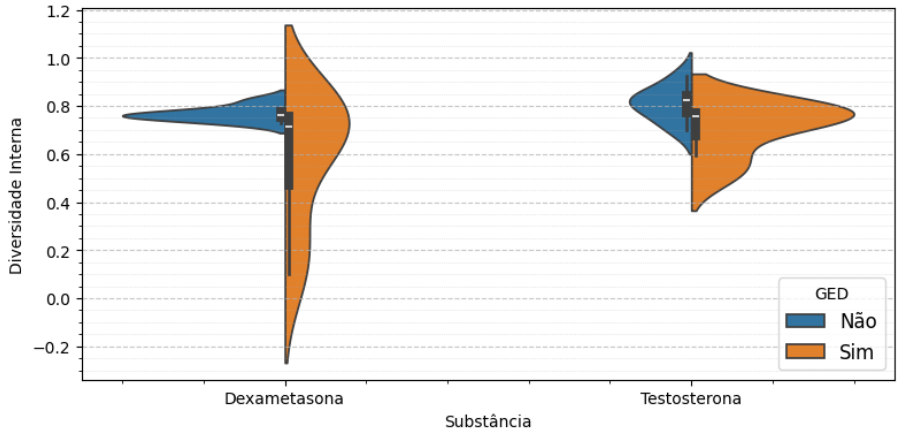
As Figuras 7, 8 e 9 apresentam a distribuição de cada substância química gerada.

Figura 7 – Distribuição de unicidade por substâncias do Embeding *Values* para 1 ponto decimal



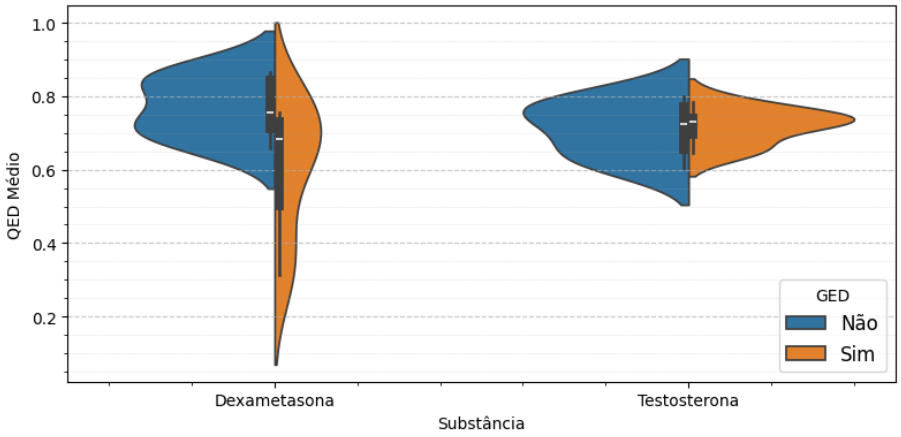
Fonte: Próprio autor.

Figura 8 – Distribuição de Diversidade Interna por substâncias do Embeding *Values* para 1 ponto decimal



Fonte: Próprio autor.

Figura 9 – Distribuição de QED médio por substâncias do Embeding *Values* para 1 ponto decimal



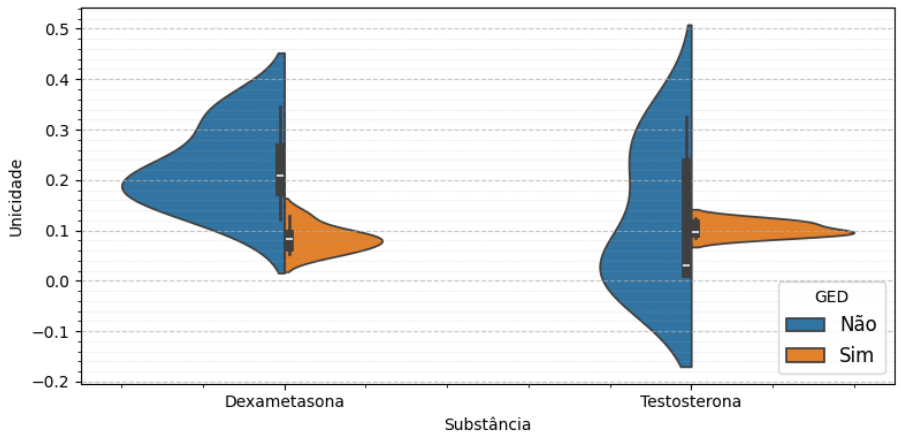
Fonte: Próprio autor.

É possível perceber que a substância da dexametasona foi no geral negativamente impactada pelo modelo, pois a distribuição de QDE médio dela foi a menor. Quando considerada a testosterona, o impacto foi positivo, pois mesmo com uma grande quantidade de moléculas única criadas, a diversidade interna e o QED médio deles se mantiveram próximos do uso sem o modelo.

5.3.2.2 One-Hot

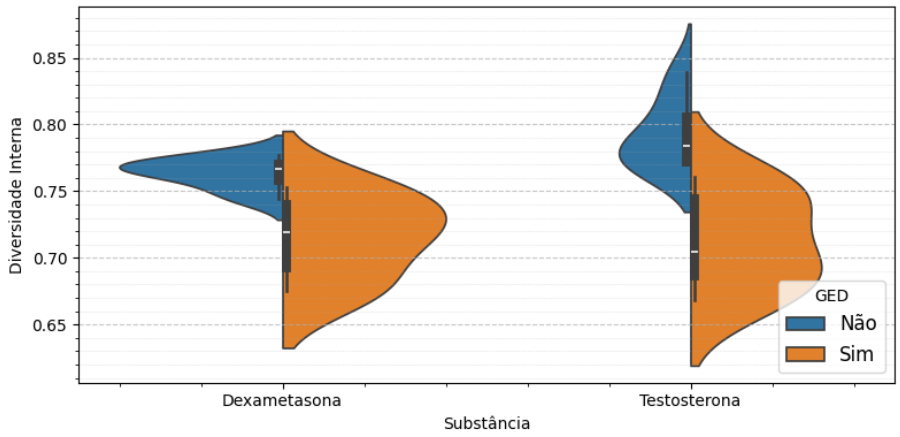
As Figuras 10, 11 e 12 apresentam a distribuição de cada substância química gerada.

Figura 10 – Distribuição de unicidade por substâncias do Embeding *One-Hot* para 1 ponto decimal



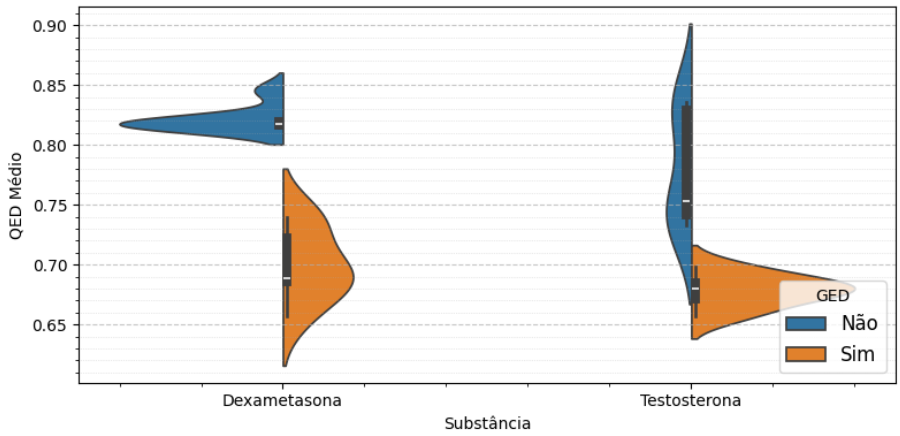
Fonte: Próprio autor.

Figura 11 – Distribuição de Diversidade Interna por substâncias do Embeding *One-Hot* para 1 ponto decimal



Fonte: Próprio autor.

Figura 12 – Distribuição de QED médio por substâncias do Embeding *One-Hot* para 1 ponto decimal



Fonte: Próprio autor.

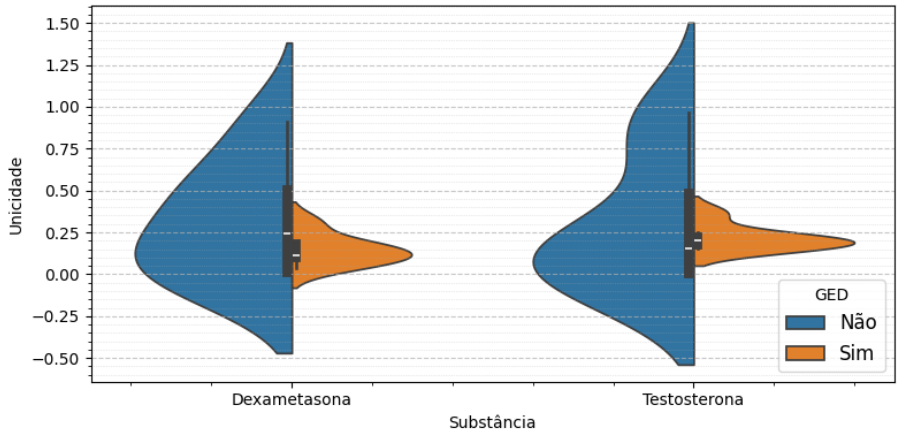
Quando analisada a Unicidade, a principal informação observada é de que, para ambas as substâncias, as moléculas geradas com o modelo GED tiveram uma maior concentração do que as que foram geradas sem o modelo; além disso, a média de valores foi maior para a testosterona, e menor para a dexametasona

Quando analisada a diversidade interna, também para ambas as substâncias, o modelo gerou resultados mais diversos, atingindo valores menores, mesmo com um número de moléculas já reduzido. O mesmo ocorre para a métrica de QED médio, onde o valor é menor. Para moléculas afetadas pela substância dexametasona, a concentração ficou mais dispersa; já para as geradas com base na afetação da testosterona, o modelo gerou resultados mais concentrados.

5.3.2.3 Binary

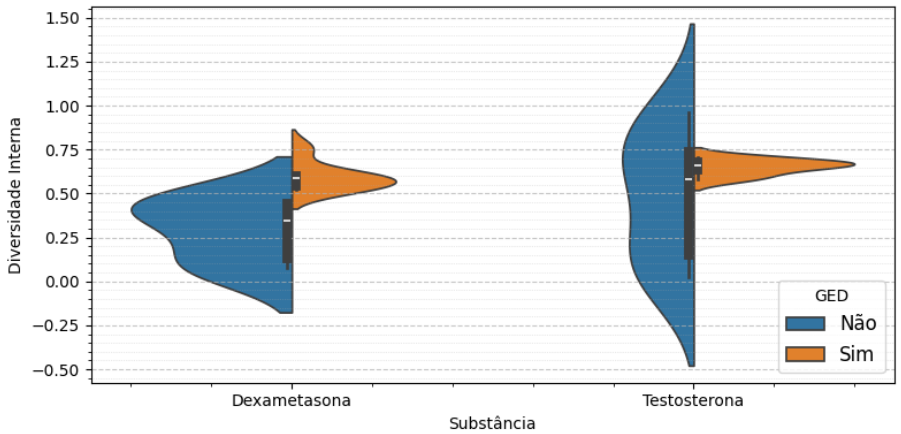
As Figuras 13, 14 e 15 apresentam a distribuição de cada substância química gerada.

Figura 13 – Distribuição de unicidade por substâncias do Embeding *Binary* para 1 ponto decimal



Fonte: Próprio autor.

Figura 14 – Distribuição de Diversidade Interna por substâncias do Embeding *Binary* para 1 ponto decimal

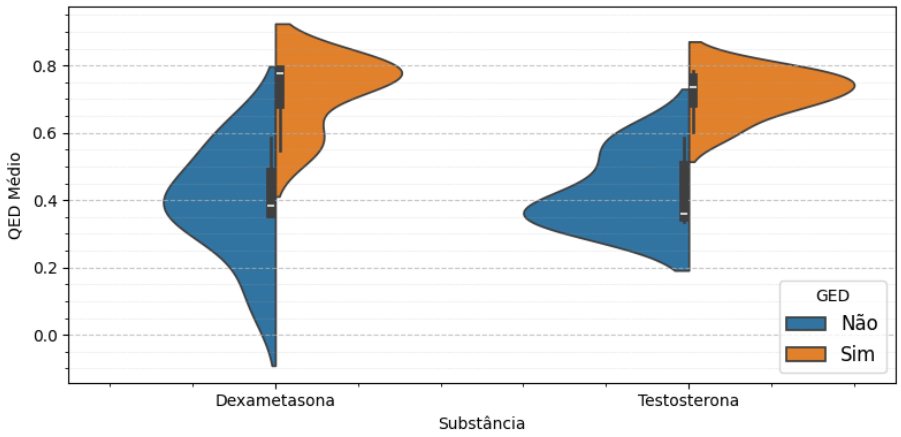


Fonte: Próprio autor.

Assim como para o embedding de *One-Hot*, neste também houve uma maior concentração do grupo de moléculas geradas, os quais ficaram com uma média inferior a 25%, muito similar à média sem o modelo GED.

Quando analisada a diversidade interna, o modelo também foi capaz de aumenta a concentração. Isso significa que, para cada substância, as 7 moléculas usadas no conjunto de teste tiveram performances bem parecidas.

Figura 15 – Distribuição de QED médio por substâncias do Embeding *Binary* para 1 ponto decimal



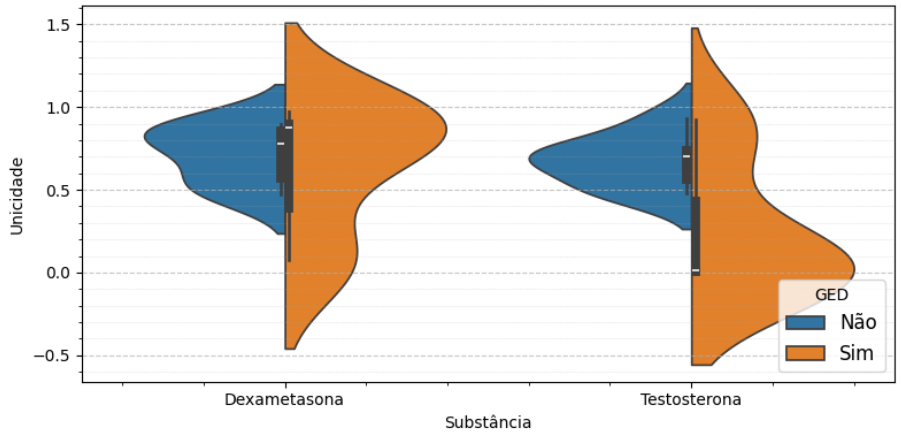
Fonte: Próprio autor.

Com esse embedding, o modelo foi capaz de trazer uma taxa de QED mais alta com a aplicação do modelo, obtendo um valor médio maior do que 75%, para ambas as substâncias.

5.3.2.4 Tenfold-Binary

As Figuras 16, 17 e 18 apresentam a distribuição de cada substância química gerada.

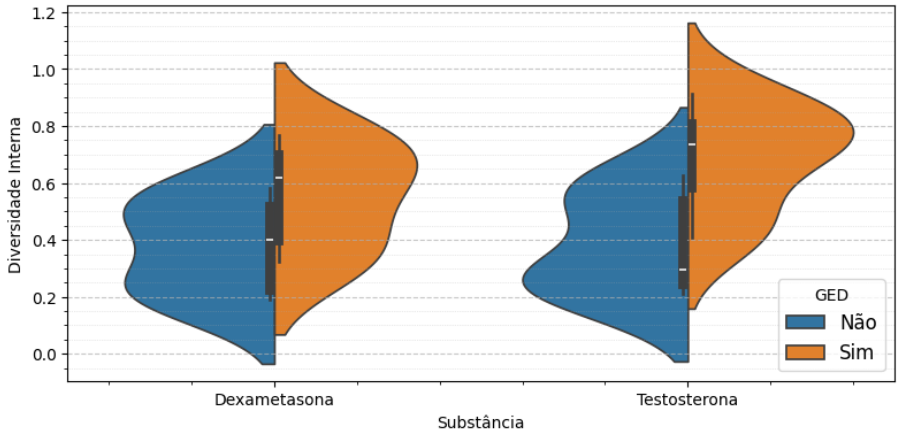
Figura 16 – Distribuição de unicidade por substâncias do Embeding *Tenfold-Binary* para 1 ponto decimal



Fonte: Próprio autor.

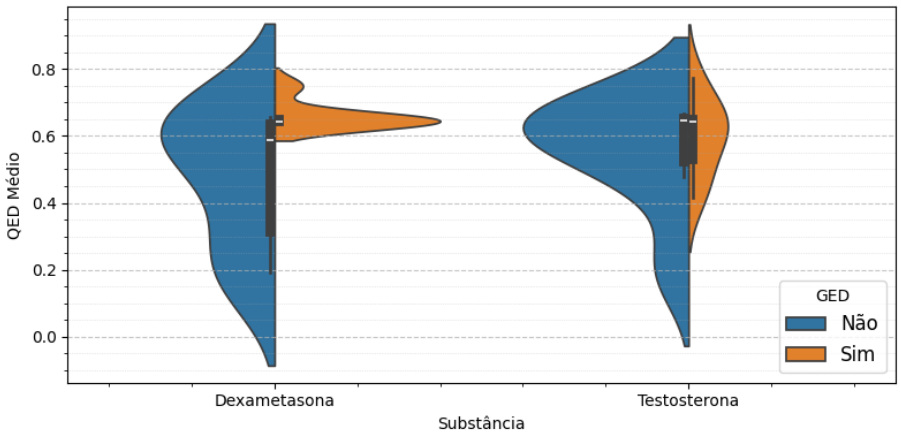
Quando analisada a unicidade das moléculas geradas, é possível ver que para a substância da dexametasona, o perfil de grupos gerados tanto com e sem o modelo foi muito parecido; entretanto, ao analisar a testosterona, é possível ver uma drástica queda no padrão observado, com a média dos valores observados diminuindo drasticamente.

Figura 17 – Distribuição de Diversidade Interna por substâncias do Embeding *Tenfold-Binary* para 1 ponto decimal



Fonte: Próprio autor.

Figura 18 – Distribuição de QED médio por substâncias do Embeding *Tenfold-Binary* para 1 ponto decimal



Fonte: Próprio autor.

Para a Diversidade Interna, a distribuição se manteve a mesma, mas com um aumento significativo das médias (o aumento médio das substâncias da média delas foi de 63.2%).

Olhando o QED médio, é possível perceber que, para a substância da dexametasona, o uso do modelo diminuiu consideravelmente o desvio padrão dos dados; já para a testosterona, os resultados foram semelhantes com e sem o modelo.

5.3.3 Análise por Componente Químico

Este subseção tem por objetivo analisar o desempenho do gerador nos melhores casos para cada embedding e com e sem a aplicação do modelo GED. O resultado máximo

obtido é o que se encontra dentro do conjunto das 7 moléculas utilizadas para treino para cada uma das substâncias.

A Tabela 10 apresenta os resultados máximos obtidos para cada um dos parâmetros analisados com e sem a aplicação do GED por componente químico.

Tabela 10 – Tabela com os valores com e sem a aplicação do modelo GED para diferentes embeddings e a diferença percentual, com expressões genéticas de 1 ponto decimal

Embedding	**Com GED**	**Unicidade**		**Div. Interna**		**QED Médio**	
		Test.	**Dexa.**	**Test.**	**Dexa.**	**Test.**	**Dexa.**
Value	Não	0.22	0.13	**0.92**	**0.83**	**0.80**	**0.87**
	Sim	**0.78**	**0.43**	0.79	0.77	0.78	0.76
	Ganho (%)	254.55	230.77	-14.13	-7.22	-2.50	-12.64
One-Hot	Não	**0.33**	**0.35**	**0.84**	**0.78**	**0.84**	**0.85**
	Sim	0.12	0.13	0.76	0.75	0.70	0.74
	Ganho (%)	-63.64	-62.86	-9.52	-3.85	-16.67	-12.94
Binary	Não	**0.96**	**0.91**	**0.96**	0.46	0.59	0.59
	Sim	0.37	0.31	0.70	**0.76**	**0.78**	**0.79**
	Ganho (%)	-61.46	-65.93	-27.08	65.22	32.20	33.90
Tenfold-Binary	Não	**0.93**	0.90	0.63	0.58	0.67	0.66
	Sim	0.92	**0.98**	**0.91**	**0.77**	**0.77**	**0.75**
	Ganho (%)	-1.08	8.89	44.44	32.76	15.38	13.64

É possível perceber que houve um acompanhamento de tendência muito parecido com as substâncias testosterona e dexametasona, exceto em dois casos (Diversidade Interna do embedding *One-Hot*, e Unicidade do embedding *Tenfold-Binary*).

Novamente, o modelo GED performou diferentemente para cada tipo de embedding escolhido. O embedding *Values* apresentou muita vantagem na criação de mais moléculas únicas; entretanto, teve uma queda média de 8.3% em Diversidade Interna e 9.9% em QED médio. O embedding *One-Hot* foi o único que não apresentou nenhuma melhora com a aplicação do modelo.

O embedding do *Tenfold-Binary* foi o único que apresentou melhora na diversidade interna das moléculas gerada com a aplicação do modelo, apresentando ganho médio de 38.6%. O QED médio apresentou uma pequena melhora média de 14.5% por substâncias. A unicidade apresentou mínimo decréscimo (-1.1%) para a substância da testosterona, e uma pequena melhora (8.9%) para a substâncias da dexametasona.

5.4 Uso do modelo de Redução de Ruido de expressão gênica para variações genéticas mais precisas

O último experimento relaciona os dois primeiros estudos, e busca explorar o impacto do uso do modelo de Redução de Ruído de expressão gênica em variações genéticas mais precisas. A comparação dos resultados obtidos nos experimentos da seção 5.2.1 foram utilizados nessa seção como comparação à aplicação do GED.

Vale ressaltar que a novidade e a validade também foram de 100% para todos os casos a seguir apresentados.

5.4.1 Uso do modelo de Redução de Ruido de expressão gênica com três pontos decimais de precisão

A Tabela 11 apresenta o uso do modelo GED nos diferentes embeddings para expressões genéticas mais precisas, com três pontos de precisão.

Tabela 11 – Comparação em diferentes embeddings do uso do modelo de Redução de Ruído da expressão gênica com 3 pontos decimais

Embedding	GED	# Moléculas	Unicidade	Div. Interna	QED Médio	Média
Values	Não	1530.36±915.00	**0.51±0.30**	**0.82±0.02**	**0.80±0.02**	**0.71**
	Sim	1428.14±1107.82	0.48±0.37	0.45±0.27	0.73±0.02	0.55
	Ganho (%)	-	-5.9	-45.1	-8.8	-22.5
One-Hot	Não	3000.00±00	**1.00±0.00**	0.35±0.05	0.42±0.03	**0.59**
	Sim	1472.07±638.09	0.49±0.21	**0.55±0.18**	**0.61±0.08**	0.55
	Ganho (%)	-	-51.0	57.1	45.2	-6.8
Binary	Não	1711.57±352.37	**0.57±0.12**	**0.75±0.08**	**0.71±0.01**	**0.68**
	Sim	458.93±120.67	0.15±0.04	0.64±0.04	0.70±0.02	0.50
	Ganho (%)	-	-73.7	-14.7	-1.4	-26.5
Tenfold-Binary	Não	1683.36±142.84	**0.56±0.05**	0.72±0.02	0.73±0.01	0.67
	Sim	1227.50±311.58	0.41±0.10	**0.82±0.03**	**0.77±0.03**	0.67
	Ganho (%)	-	-26.8	13.9	5.5	0.0

A aplicação do modelo de forma geral não melhorou o desempenho do gerador. Para os embeddings *Values* e *Binary*, a utilização do modelo não trouxe melhora em nenhuma das métricas analisadas. Já para os embeddings *One-Hot* e *Tenfold-Binary*, o GED trouxe melhora para a métrica de Diversidade Interna e QED Médio, principalmente para o embedding *One-Hot*, que obteve aumento de 57.1% de Diversidade Interna e 45.2% de QED Médio; entretanto, para os dois casos, houve uma perda significativa de Unicidade.

A Tabela 12 apresenta os resultados máximos obtidos para cada grupo de componentes químicos para o uso do gerador com e sem o modelo GED, para expressões genéticas com 3 pontos decimais.

Tabela 12 – Tabela com os valores para "Com GED"e "Sem GED"para diferentes embeddings e a diferença percentual, com expressões genéticas de 3 pontos decimais

Embedding	**Com GED**	**Unicidade**		**Div. Interna**		**QED Médio**	
		Test.	**Dexa.**	**Test.**	**Dexa.**	**Test.**	**Dexa.**
Value	Não	0.87	**0.88**	**0.84**	**0.85**	**0.84**	**0.84**
	Sim	**1.00**	0.83	0.72	0.71	0.77	0.74
	Ganho (%)	14.94	-5.68	-14.29	-16.47	-8.33	-11.90
One-Hot	Não	**1.00**	**1.00**	0.40	0.39	0.45	0.47
	Sim	0.78	0.95	**0.60**	**0.83**	**0.64**	**0.82**
	Ganho (%)	-22.00	-5.00	50.00	112.82	42.22	74.47
Binary	Não	**0.81**	**0.76**	**0.82**	**0.84**	**0.75**	0.72
	Sim	0.21	0.18	0.68	0.69	0.74	**0.74**
	Ganho (%)	-74.07	-76.32	-17.07	-17.86	-1.33	2.78
Tenfold-Binary	Não	**0.65**	0.56	0.74	0.75	0.75	0.73
	Sim	0.49	**0.65**	**0.89**	**0.87**	**0.80**	**0.81**
	Ganho (%)	-24.62	16.07	20.27	16.00	6.67	10.96

Assim como no resultado geral, para os embeddings *Value* e *Binary*, o grupo de moléculas submetido ao gerador sem o GED apresentou melhores resultados dos que os que foram submetidos, chegando a ter uma perda de até 76.3% na métrica de Unicidade.

Para os embeddings *One-Hot* e *Tenfold-Binary*, o grupo de moléculas com o modelo GED foi capaz de gerar métricas mais altas. Os ganhos obtidos para o embedding *One-Hot* foram os que mais se destacaram, tendo aumento de 112.8% e 74.5% para as métricas de diversidade interna e QED médio para a dexametasona, respectivamente.

5.4.2 Uso do modelo de Redução de Ruido de expressão gênica com seis pontos decimais de precisão

A Tabela 13 apresenta o uso do modelo GED nos diferentes embeddings para expressões genéticas mais precisas, com seis pontos de precisão.

A aplicação do modelo GED apresentou melhores resultados para o embedding *Tenfold-Binary*, em que a unicidade teve um incrível aumento de 2300.0% e o QED médio

Tabela 13 – Comparação em diferentes embeddings do uso do modelo de Redução de Ruído da expressão gênica com 6 pontos decimais, incluindo a média dos valores.

Embedding	GED	# Moléculas	Unicidade	Div. Interna	QED Médio	Média
Values	Não	74.21±148.02	0.02±0.05	**0.89±0.04**	0.52±0.09	0.36
	Sim	1751.50±964.51	**0.58±0.32**	0.34±0.21	**0.55±0.11**	0.75
	Ganho (%)	-	2800.0	-61.8	5.8	108.3
One-Hot	Não	1083.13±608.13	**0.36±0.20**	0.71±0.17	**0.73±0.07**	0.66
	Sim	190.79±115.16	0.06±0.04	**0.88±0.02**	0.68±0.06	0.42
	Ganho (%)	-	-83.3	23.9	-6.8	-36.4
Binary	Não	1591.71±526.48	0.53±0.18	0.68±0.06	**0.81±0.05**	0.75
	Sim	2075.57±724.71	**0.69±0.24**	**0.73±0.05**	0.65±0.02	0.77
	Ganho (%)	-	30.2	7.4	-19.8	2.7
Tenfold-Binary	Não	49.21±41.73	0.02±0.01	**0.71±0.13**	0.45±0.11	0.30
	Sim	1447.14±282.72	**0.48±0.09**	0.69±0.06	**0.73±0.02**	0.60
	Ganho (%)	-	2300.0	-2.8	62.2	100.0

de 62.2%, acompanhado somente de uma pequena (-2.8%) baixa no valor de diversidade interna.

O embedding *Values* apresentou incrível ganho de unicidade (2800.0%); mas com o aumento do número de moléculas únicas geradas, o valor de diversidade interna entre elas apresentou queda de 61.8%. O valor de QED médio foi beneficiado com o modelo, tendo um aumento de 5.8%.

O embedding do *One-Hot* foi o único que apresentou queda de Unicidade, tendo uma perca de 83.3% relativo ao gerador sem o modelo GED. O valor de diversidade interna dessas novas moléculas aumentou, como esperado, e teve ganho de 23.9%. O QED Médio sofreu queda de 6.8%.

O embedding *Binary* foi o único que apresentou aumento tanto em unicidade (30.2%) quanto em diversidade interna (7.4%), mas apresentou uma queda de 19.8% QED Médio.

De forma geral, este experimento obteve resultados competitivos com o uso de somente uma casa decimal sem o model do GED, inclusive obtendo resultado superior para o embedding *Binary* e *Tenfold-Binary* para as 3 métricas apresentadas.

A Tabela 14 apresenta os resultados máximos obtidos para cada grupo de componentes químicos para o uso do gerador com e sem o modelo GED, para expressões genéticas com 6 pontos decimais.

Tabela 14 – Tabela com os valores para "Com GED"e "Sem GED"para diferentes embeddings e a diferença percentual, com expressões genéticas de 6 pontos decimais

Embedding	Com GED	Unicidade		Div. Interna		QED Médio	
		Test.	**Dexa.**	**Test.**	**Dexa.**	**Test.**	**Dexa.**
Value	Não	0.01	0.15	**0.92**	**0.92**	0.52	**0.73**
	Sim	**0.99**	**0.92**	0.31	0.86	**0.67**	0.70
	Ganho (%)	9800.00	513.33	-66.96	-6.52	28.85	-4.11
One-Hot	Não	**0.59**	**0.70**	**0.92**	0.88	**0.81**	0.75
	Sim	0.08	0.18	0.90	**0.90**	0.74	**0.77**
	Ganho (%)	-86.44	-74.29	-2.17	2.27	-8.64	3.12
Binary	Não	0.69	0.78	0.69	0.78	**0.90**	**0.85**
	Sim	**0.90**	**0.87**	**0.78**	**0.82**	0.71	0.67
	Ganho (%)	30.43	11.54	13.04	5.13	-21.11	-21.18
Tenfold-Binary	Não	0.05	0.04	**0.84**	**0.84**	0.54	0.67
	Sim	**0.64**	**0.54**	0.77	0.76	**0.77**	**0.75**
	Ganho (%)	1180.00	1250.00	-8.33	-9.52	42.59	11.94

Analisando os melhores casos, É possível ver que houve grande sinergia entre os ganhos de cada componente químico, com exceção do QED médio.

Quando analisado o embedding *Values*, o ganho de unicidade da testosterona é de 9800%, um valor muito alto. Tal valor é acompanhado de uma aumento de 28.9% de QED Médio deste componente, e com uma taxa de Diversidade Interna menor, de 31% (queda de 67.0%). Para a dexametasona, o aumento de unicidade interna também foi alto, de 513.3%; entretanto, a diversidade interna e o QED médio sofreram pequenas quedas em seus valores (6.5% e 4.1%).

No embedding *One-Hot*, a apliação do modelo GED não trouxe benefícios. a unicidade das moléculas criadas tiveram uma alta taxa de queda de 80.4% na média. O restante das métricas tiveram pequenas variações que não forma significativas.

O embedding *Binary* apresentou ganho com o modelo GED em ambas as substâncias para as métricas de unicidade (21.0% na média) e diversidade interna (9.1% na média). Entretanto, o valor de QED médio abaixou para 21.2% na média.

O embedding *Tenfold-Binary* apresentou um alto ganho de unicidade, atingindo um aumento médio de 1219.5% para ambas as substâncias. Mesmo com esse grande aumento, o valor de diversidade interna das novas moléculas geradas não foi tão afetado quanto se era esperado, tendo uma queda somente de 8.9% na média das duas. Além disso, a métrica

de QED Médio sofreu grande aumento, atingindo 42.5% para a testosterona e 11.9% para a dexametasona.

6 Conclusão

Neste trabalho, foi apresentado um estudo explorando a qualidade e representação dos dados de expressão gênica em diferentes embeddings, com o intúito de estudar seu impacto no processo de criação de moléculas. Para estes estudo, utilizou-se o TransGEM (LIU et al., 2024), um modelo de geração de moléculas com base em alterações de expressões gênicas em diferentes tipagens celulares. O estudo abordo o impacto do uso de pontos decimais de expressão genética mais precisos, e também o impacto do uso de um redutor de ruído na expressão gênica no modelo.

6.1 Principais descobertas

O resultado mais interessante do estudo foi de que, para todas as moléculas geradas, o valor de validade e novidade atingiram 100% de resultado. Isso demonstra que variações na expressão genética não impactam a natureza da molécula gerada, e também não é responsável por gerar moléculas novas dentro do conjunto de dados.

Sobre o primeiro experimento, de forma geral pode-se observar que o uso de três casas decimais de expressão genética obteve os melhores resultados, principalmente relacionado ao QED médio e à métrica de unicidade. Além disso, o uso de 3 casas decimais gerou uma menor variação dos resultados obtidos para as 7 moléculas base de cada tipagem celular, mostrando maior consistência dos resultados.

Sobre o segundo experimento, de forma geral a aplicação do modelo GED fez com que o valor de unicidade do modelo diminuísse consideravelmente, mas com um aumento de diversidade interna maior que a queda, e acompanhado por um aumento de QED médio. Além disso, o padrão observado para as 2 substâncias foi muito próximo um do outro, indicando que o modelo foi capaz de gerar estes mesmos resultados, mesmo para substâncias diferentes.

Sobre o terceiro experimento, para expressões gênicas de três pontos decimais foi de desencontro com o resultado do primeiro experimento, e de forma geral, não gerou um impacto positivo no modelo. Já o uso de seis pontos decimais de precisão foi capaz de gerar resultados positivos, inclusive sendo superior em dois casos em relação ao modelo de uma casa decimal sem tratamento de ruído.

Por fim, este trabalho também tem como contribuição a discussão do tópico sobre geração de moléculas no Brasil.

6.2 Limitações

Uma limitação deste trabalho foi de não ter utilizado o mesmo conjunto de dados original do algoritmo TransGEM, o subLINCS. Isto ocorreu por não haver a disponibilidade pública dos dados originais utilizados com um valor de expressão gênica mais preciso.

Outra limitação do estudo foi a falta de variedade de tipagens celulares. Seria possível incluir um dataset mais robusto com outros tipos celulares e analisar se o padrão observado seria repetido para eles. Além disso, um estudo com somente uma tipagem celular no conjunto de teste – como feito no artigo original – também poderia trazeres resultados interessantes para se discutir.

Por fim, o estudo também poderia ter sido expandido para uma análise dos resultados utilizando diferentes hiperparâmetros, visto que o conjunto de dados utilizado é diferente dos do artigo original e que, portanto, os melhores hiperparâmetros também poderiam ser diferentes.

6.3 Trabalhos Futuros

Uma possível evolução deste estudo seria a criação de um dataset com maior variedade de tipagens celulares. Isso possibilitaria uma melhor comparação com os resultados obtidos pelo modelo do TransGEM.

Outra possível evolução seria o uso de outros modelos de redução de ruído diferentes do GED. Há uma grande variedade de pesquisas relacionadas a esta área, como apresentado na seção 3, e algum outro estudo poderia ser explorado.

Por fim, novos embeddings diferentes dos quatro apresentados poderiam ser explorados, a fim de observar melhora nas métricas deração de moléculas, e também de analisar sua eficiência.

Referências

AGOSTINELLI, A. et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023. Citado na página 30.

AHMED, M. H.; HASSAN, A. Dexamethasone for the treatment of coronavirus disease (covid-19): a review. *SN comprehensive clinical medicine*, Springer, v. 2, n. 12, p. 2637–2646, 2020. Citado na página 49.

BAJUSZ, D.; RÁCZ, A.; HÉBERGER, K. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of cheminformatics*, Springer, v. 7, p. 1–13, 2015. Citado na página 52.

BANNERJEE, G. et al. Artificial intelligence in agriculture: A literature survey. *international Journal of Scientific Research in computer Science applications and Management Studies*, v. 7, n. 3, p. 1–6, 2018. Citado na página 21.

BHUKYA, R. Encoding gene expression using deep autoencoders for expression inference. *Int. Arab J. Inf. Technol.*, v. 18, n. 5, p. 625–633, 2021. Citado na página 44.

BORN, J. et al. Paccmannrl: De novo generation of hit-like anticancer molecules from transcriptomic data via reinforcement learning. *Iscience*, Elsevier, v. 24, n. 4, 2021. Citado na página 41.

BRIGANTI, G.; MOINE, O. L. Artificial intelligence in medicine: today and tomorrow. *Frontiers in medicine*, Frontiers, v. 7, p. 509744, 2020. Citado na página 21.

BROWN, T. et al. Language models are few-shot learners. *Advances in neural information processing systems*, v. 33, p. 1877–1901, 2020. Citado 2 vezes nas páginas 31 e 32.

CASAROTTO, P. H. *TCC - Repositório do GitHub*. 2025. Acessado em: 10 fev. 2025. Disponível em: <https://github.com/PedroHCasarotto/TCC>. Citado 2 vezes nas páginas 45 e 51.

CHAN, H. S. et al. Advancing drug discovery via artificial intelligence. *Trends in pharmacological sciences*, Elsevier, v. 40, n. 8, p. 592–604, 2019. Citado na página 21.

CHEN, M. et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. Citado na página 30.

CHENTHAMARAKSHAN, V. et al. Cogmol: Target-specific and selective drug design for covid-19 using deep generative models. *Advances in Neural Information Processing Systems*, v. 33, p. 4320–4332, 2020. Citado 2 vezes nas páginas 26 e 42.

CLARK, D. E.; WESTHEAD, D. R. Evolutionary algorithms in computer-aided molecular design. *Journal of Computer-Aided Molecular Design*, Springer, v. 10, p. 337–358, 1996. Citado na página 26.

DAS, D. et al. Gex2sgen: designing drug-like molecules from desired gene expression signatures. *Journal of Chemical Information and Modeling*, ACS Publications, v. 63, n. 7, p. 1882–1893, 2023. Citado na página 42.

DESAI, K.; JOHNSON, J. Virtex: Learning visual representations from textual annotations. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2021. p. 11162–11173. Citado na página 31.

DEVI, R. V.; SATHYA, S. S.; COUMAR, M. S. Evolutionary algorithms for de novo drug design–a survey. *Applied Soft Computing*, Elsevier, v. 27, p. 543–552, 2015. Citado 2 vezes nas páginas 25 e 26.

DEVLIN, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. Citado 2 vezes nas páginas 31 e 33.

ECK, D.; SCHMIDHUBER, J. Finding temporal structure in music: Blues improvisation with lstm recurrent networks. In: IEEE. *Proceedings of the 12th IEEE workshop on neural networks for signal processing*. [S.l.], 2002. p. 747–756. Citado na página 30.

EISENEGGER, C.; HAUSHOFER, J.; FEHR, E. The role of testosterone in social interaction. *Trends in cognitive sciences*, Elsevier, v. 15, n. 6, p. 263–271, 2011. Citado na página 49.

EKINS, S. The next era: deep learning in pharmaceutical research. *Pharmaceutical research*, Springer, v. 33, n. 11, p. 2594–2603, 2016. Citado na página 21.

FEUERRIEGEL, S. et al. Generative ai. *Business & Information Systems Engineering*, Springer, v. 66, n. 1, p. 111–126, 2024. Citado na página 30.

GAO, C. et al. Dockingga: enhancing targeted molecule generation using transformer neural network and genetic algorithm with docking simulation. *Briefings in Functional Genomics*, Oxford University Press, p. elae011, 2024. Citado na página 42.

GAVISH, M.; DONOHO, D. L. The optimal hard threshold for singular values is 4/\sqrt {3} .*IEEE Transactions on Information Theory*, $IEEE, v.\ 60, n.\ 8, p.5040 - -5053, 2014. Citadonapágina44$.

GOLBERG, D. E. Genetic algorithms in search, optimization, and machine learning. *Addion wesley*, v. 1989, n. 102, p. 36, 1989. Citado na página 26.

GOMES, B.; ASHLEY, E. A. Artificial intelligence in molecular medicine. *New England Journal of Medicine*, Mass Medical Soc, v. 388, n. 26, p. 2456–2465, 2023. Citado na página 21.

GOODFELLOW, I. et al. Generative adversarial nets. *Advances in neural information processing systems*, v. 27, 2014. Citado na página 30.

GROVER, A.; LESKOVEC, J. node2vec: Scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2016. p. 855–864. Citado na página 31.

HASSANE, D. C. et al. Discovery of agents that eradicate leukemia stem cells using an in silico screen of public gene expression data. *Blood, The Journal of the American Society of Hematology*, American Society of Hematology Washington, DC, v. 111, n. 12, p. 5654–5662, 2008. Citado na página 29.

JEON, J. et al. Denoiseit: denoising gene expression data using rank based isolation trees. *BMC bioinformatics*, Springer, v. 25, n. 1, p. 271, 2024. Citado na página 43.

JEON, M. et al. Transforming l1000 profiles to rna-seq-like profiles with deep learning. *BMC bioinformatics*, Springer, v. 23, n. 1, p. 374, 2022. Citado na página 44.

KINGMA, D. P. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. Citado 2 vezes nas páginas 41 e 42.

KOSAKA, T. et al. Identification of drug candidate against prostate cancer from the aspect of somatic cell reprogramming. *Cancer science*, Wiley Online Library, v. 104, n. 8, p. 1017–1026, 2013. Citado na página 29.

KRENN, M. et al. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, IOP Publishing, v. 1, n. 4, p. 045024, 2020. Citado 2 vezes nas páginas 28 e 49.

KULLBACK, S.; LEIBLER, R. A. On information and sufficiency. *The annals of mathematical statistics*, JSTOR, v. 22, n. 1, p. 79–86, 1951. Citado na página 39.

KUMAR, S. et al. Mega x: molecular evolutionary genetics analysis across computing platforms. *Molecular biology and evolution*, Oxford University Press, v. 35, n. 6, p. 1547–1549, 2018. Citado na página 26.

LAMB, J. The connectivity map: a new tool for biomedical research. *Nature reviews cancer*, Nature Publishing Group UK London, v. 7, n. 1, p. 54–60, 2007. Citado na página 29.

LI, Y. et al. Competition-level code generation with alphacode. *Science*, American Association for the Advancement of Science, v. 378, n. 6624, p. 1092–1097, 2022. Citado na página 30.

LIAO, Z. et al. Sc2mol: a scaffold-based two-step molecule generator with variational autoencoder and transformer. *Bioinformatics*, Oxford University Press, v. 39, n. 1, p. btac814, 2023. Citado na página 41.

LIU, S.; YAO, W. Prediction of lung cancer using gene expression and deep learning with kl divergence gene selection. *BMC bioinformatics*, Springer, v. 23, n. 1, p. 175, 2022. Citado na página 39.

LIU, Y. et al. Transgem: a molecule generation model based on transformer with gene expression data. *Bioinformatics*, Oxford University Press, v. 40, n. 5, p. btae189, 2024. Citado 5 vezes nas páginas 34, 35, 36, 51 e 69.

MAZUZ, E. et al. Molecule generation using transformers and policy gradient reinforcement learning. *Scientific Reports*, Nature Publishing Group UK London, v. 13, n. 1, p. 8799, 2023. Citado na página 42.

MÉNDEZ-LUCIO, O. et al. De novo generation of hit-like molecules from gene expression signatures using artificial intelligence. *Nature communications*, Nature Publishing Group UK London, v. 11, n. 1, p. 10, 2020. Citado 2 vezes nas páginas 41 e 48.

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, v. 26, 2013. Citado na página 31.

MOUCHLIS, V. D. et al. Advances in de novo drug design: from conventional to machine learning methods. *International journal of molecular sciences*, MDPI, v. 22, n. 4, p. 1676, 2021. Citado 2 vezes nas páginas 21 e 25.

MOUCHLIS, V. D. et al. Advances in de novo drug design: from conventional to machine learning methods. *International journal of molecular sciences*, MDPI, v. 22, n. 4, p. 1676, 2021. Citado 3 vezes nas páginas 21, 25 e 26.

MOUTSOPOULOS, I. et al. noisyr: enhancing biological signal in sequencing datasets by characterizing random technical noise. *Nucleic Acids Research*, Oxford University Press, v. 49, n. 14, p. e83–e83, 2021. Citado 2 vezes nas páginas 27 e 43.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. [S.l.: s.n.], 2014. p. 1532–1543. Citado na página 31.

PEREIRA, T. et al. Diversity oriented deep reinforcement learning for targeted molecule generation. *Journal of cheminformatics*, Springer, v. 13, n. 1, p. 21, 2021. Citado na página 21.

PERES, R. et al. *On ChatGPT and beyond: How generative artificial intelligence may affect research, teaching, and practice*. [S.l.]: Elsevier, 2023. 269–275 p. Citado na página 30.

PHAM, T.-H. et al. A deep learning framework for high-throughput mechanism-driven phenotype compound screening and its application to covid-19 drug repurposing. *Nature machine intelligence*, Nature Publishing Group UK London, v. 3, n. 3, p. 247–257, 2021. Citado 4 vezes nas páginas 22, 29, 30 e 48.

PHAM, T.-H.; XIE, L.; ZHANG, P. Fame: fragment-based conditional molecular generation for phenotypic drug discovery. In: SIAM. *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*. [S.l.], 2022. p. 720–728. Citado 2 vezes nas páginas 39 e 47.

PRAVALPHRUEKUL, N. et al. De novo design of molecules with multiaction potential from differential gene expression using variational autoencoder. *Journal of chemical information and modeling*, ACS Publications, v. 63, n. 13, p. 3999–4011, 2023. Citado 2 vezes nas páginas 22 e 43.

RADFORD, A. et al. Learning transferable visual models from natural language supervision. In: PMLR. *International conference on machine learning*. [S.l.], 2021. p. 8748–8763. Citado na página 31.

RADFORD, A. et al. Language models are unsupervised multitask learners. *OpenAI blog*, v. 1, n. 8, p. 9, 2019. Citado na página 33.

ROMBACH, R. et al. High-resolution image synthesis with latent diffusion models. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2022. p. 10684–10695. Citado na página 30.

SALKOVIC, E. et al. Outsingle: a novel method of detecting and injecting outliers in rna-seq count data using the optimal hard threshold for singular values. *Bioinformatics*, Oxford University Press, v. 39, n. 4, p. btad142, 2023. Citado na página 44.

SCHNEIDER, P.; SCHNEIDER, G. De novo design at the edge of chaos: Miniperspective. *Journal of medicinal chemistry*, ACS Publications, v. 59, n. 9, p. 4077–4086, 2016. Citado na página 25.

SHA, Y.; PHAN, J. H.; WANG, M. D. Effect of low-expression gene filtering on detection of differentially expressed genes in rna-seq data. In: IEEE. *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. [S.l.], 2015. p. 6461–6464. Citado na página 27.

SHAYAKHMETOV, R. et al. Molecular generation for desired transcriptome changes with adversarial autoencoders. *Frontiers in Pharmacology*, Frontiers Media SA, v. 11, p. 269, 2020. Citado na página 41.

SRIVASTAVA, N.; MANSIMOV, E.; SALAKHUDINOV, R. Unsupervised learning of video representations using lstms. In: PMLR. *International conference on machine learning*. [S.l.], 2015. p. 843–852. Citado na página 30.

STEGMAIER, K. et al. Gene expression–based high-throughput screening (ge-hts) and application to leukemia differentiation. *Nature genetics*, Nature Publishing Group US New York, v. 36, n. 3, p. 257–263, 2004. Citado na página 29.

STERLING, T.; IRWIN, J. J. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, ACS Publications, v. 55, n. 11, p. 2324–2337, 2015. Citado na página 30.

SUBRAMANIAN, A. et al. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, Elsevier, v. 171, n. 6, p. 1437–1452, 2017. Citado 4 vezes nas páginas 22, 26, 27 e 47.

SWINNEY, D. C.; LEE, J. A. Recent advances in phenotypic drug discovery. *F1000Research*, Faculty of 1000 Ltd, v. 9, 2020. Citado na página 21.

TANG, B. et al. Ai-aided design of novel targeted covalent inhibitors against sars-cov-2. *Biomolecules*, MDPI, v. 12, n. 6, p. 746, 2022. Citado na página 26.

TURIAN, J.; RATINOV, L.; BENGIO, Y. Word representations: a simple and general method for semi-supervised learning. In: *Proceedings of the 48th annual meeting of the association for computational linguistics*. [S.l.: s.n.], 2010. p. 384–394. Citado na página 31.

VASWANI, A. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. Citado 4 vezes nas páginas 31, 32, 33 e 42.

WEI, G. et al. Gene expression-based chemical genomics identifies rapamycin as a modulator of mcl1 and glucocorticoid resistance. *Cancer cell*, Elsevier, v. 10, n. 4, p. 331–342, 2006. Citado na página 29.

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. Citado 2 vezes nas páginas 28 e 42.

WIGH, D. S.; GOODMAN, J. M.; LAPKIN, A. A. A review of molecular representation in the age of machine learning. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, Wiley Online Library, v. 12, n. 5, p. e1603, 2022. Citado na página 28.

WONG, S. S.; LUO, W.; CHAN, K. C. Evomd: an algorithm for evolutionary molecular design. *IEEE/ACM transactions on computational biology and bioinformatics*, IEEE, v. 8, n. 4, p. 987–1003, 2010. Citado na página 26.

XIE, R. et al. A deep auto-encoder model for gene expression prediction. *BMC genomics*, Springer, v. 18, p. 39–49, 2017. Citado na página 44.