Chemical Language Models for Molecular Design

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

HENGWEI CHEN

aus Shanxi, China

Bonn 2025

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

Gutachter/Betreuer:	Prof. Dr. rer. nat. Jürgen Bajorath
Gutachter:	PD Dr. rer. nat. Martin Vogt
Tag der Promotion:	31. Juli 2025
Erscheinungsjahr:	2025

The research towards this thesis was carried out at the Department of Life Science Informatics and Data Science at the b-it Institute of the University of Bonn under the supervision of Prof. Dr. Jürgen Bajorath.

Abstract

In drug discovery, chemical language models (CLMs) inspired by natural language processing (NLP) provide innovative solutions for molecular design. CLMs learn the vocabulary, syntax, and conditional probabilities of molecular representations, enabling various sequence-to-sequence mappings. Leveraging neural language architectures, particularly transformers with multi-head self-attention and parallel processing, CLMs effectively handle diverse sequence types, enabling efficient training and molecular translation tasks. Their versatility in machine translation and property conditioning opens new opportunities for generative molecular design. This dissertation investigates the development and application of chemical and biochemical language models (LMs) for various medicinal chemistry and drug design challenges, including activity cliff (AC) prediction, highly potent compound design, analogue series extension, and active compound generation from protein sequences. The first project focused on conditional transformers (DeepAC) for predicting ACs and designing new AC compounds. During pre-training, the models learned source-to-target compound mappings from diverse activity classes, conditioned on potency differences caused by structural modifications. Fine-tuning enabled accurate generation of target compounds satisfying potency constraints bridging between predictive modeling and compound design. The subsequent study generalized predictions beyond ACs to design highly potent compounds from weakly potent templates across unseen activity classes. Further, the next study incorporated meta-learning, enabling effective generative design even in low-data regimes. Building on these predictive capabilities, the second project developed the DeepAS models for iterative analogue series (AS) extension in lead optimization. The initial DeepAS model predicted substituents for AS arranged by ascending potency, successfully reproducing AS across various targets from which the terminal (most potent) analogue was removed. DeepAS 2.0 expanded this approach to multi-site AS extension using a BERT-based architecture, while DeepAS 3.0 integrated structure–activity relationship matrix (SARM) formalism, enabling core modifications in AS with multiple substitution sites. The final project extended CLMs into the biochemical domain by developing a dual-component LM combining a pre-trained protein language model

(PLM) with a conditional CLM. This model learned mappings from protein sequence embeddings conditioned on potency to active compounds; it consistently reproduced known compounds with varying potency across various activity classes not encountered during training. Additionally, the biochemical LM generated structurally diverse candidate compounds departing from both fine-tuning and test compounds. Taken together, this thesis highlights the promising capability of CLMs to address previously challenging or unfeasible prediction scenarios in molecular design, providing new opportunities for advancing in medicinal chemistry and drug discovery.

Acknowledgments

First and foremost, I would like to express my heartfelt gratitude to my supervisor, Prof. Dr. Jürgen Bajorath, for giving me the opportunity to pursue this scientific journey and for his invaluable guidance and support throughout my doctoral studies. I am also sincerely thankful to PD Dr. Martin Vogt for kindly serving as the co-referee and for his thoughtful evaluation of my dissertation. My appreciation also extends to Prof. Dr. Martin Hofmann-Apitius and Prof. Dr. Matthias Wüst for graciously agreeing to serve as members of my PhD committee.

I am deeply grateful to all my current colleagues at the Life Science Informatics group, including Dr. Elena Xerxa, Dr. Andrea Mastropietro, Dr. Tiago Janela, Alec Lamens, Jannik P. Roth, Sanjana Srinivasan, Selina Koch, and Antonia Mera, for creating a collaborative, warm, and inspiring research environment. Your kindness and support have made my time here both productive and truly enjoyable. In particular, I would like to thank PD Dr. Martin Vogt once again for his scientific advice, constructive suggestions, and many fruitful discussions that have significantly contributed to my research.

To my former colleagues at B-IT—Dr. Huabin Hu, Dr. Javed Iqbal, Dr. Atsushi Yoshimori, Dr. Kosuke Takeuchi, Dr. Shunsuke Tamura, Christian Feldmann, Sabrina Mendonça, and Lisa Piazza—thank you for your kind support, generous help, and the warm encouragement that accompanied me throughout this journey.

Finally, I owe my deepest gratitude to my family and friends for their unwavering encouragement and personal support throughout this journey. A very special thanks goes to Rebekah, whose unconditional love and constant support helped me persevere through every challenging moment.

Contents

List of abbreviations

1D, 2D, 3D	One, two, three-dimensional
AAE	Adversarial autoencoder
AC	Activity cliff
ADME	Absorption, distribution, metabolism, excretion
AE	Autoencoder
AS	Analogue series
BERT	Bidirectional encoder representations from transformers
BPE	Byte-pair encoding
CADD	Computer-aided drug design
CCR	Compound-core relationship
CLM	Chemical language model
CNN	Convolutional neural network
cryo-EM	Cryo-electron microscopy
CVAE	Conditional variational autoencoder
DGM	Deep generative model
DL	Deep learning
DNN	Deep neural network
ECFP	Extended connectivity fingerprint
ECIF	Extended connectivity interaction features
EDM	Equivariant diffusion model
ENBED	Ensemble nucleotide byte-level encoder-decoder
FEP	Free energy perturbation
GAN	Generative adversarial network
GATNet	Graph attention network
GCN	Graph convolutional network
GIN	Graph isomorphism network
GLM	Genomic language model
GNN	Graph neural network

GPT	Generative pre-trained transformer
GRU	Gated recurrent unit
HTS	High-throughput screening
InChI	International chemical identifier
LBDD	Ligand-based drug design
LM	Language model
LSTM	Long short-term memory
MACCS	Molecular access system
MAML	Model-agnostic meta-learning
MCS	Maximum common substructure
MD	Molecular dynamics
MMP	Matched molecular pair
MMPA	Matched molecular pair analysis
MMS	Matching molecular series
MPNN	Message-passing neural network
NLP	Natural language processing
NN	Neural network
PLEC	Protein-ligand extended connectivity
PLM	Protein language model
QED	Quantitative estimate of drug-likeness
QSAR	Quantitative structure–activity relationship
RECAP	Retrosynthetic combinatorial analysis procedure
RF	Random forest
RL	Reinforcement learning
RNN	Recurrent neural network
SA	Synthetic accessibility
SAR	Structure-activity relationship
SARM	Structure–activity relationship matrix
SBDD	Structure-based drug design
SELFIES	SELF-referencing embedded strings
SMILES	Simplified molecular input line-entry system
SMILES-PE	SMILES pair encoding

SVM	Support vector machine
T5	Text-to-text transfer transformer
VAE	Variational autoencoder

Chapter 1

Introduction

1.1 Drug discovery

Small molecule drug discovery is a complex, interdisciplinary process aimed at identifying, optimizing, and developing new pharmaceutical products. It typically unfolds across six distinct stages, each presenting unique scientific challenges: 1) target discovery, 2) hit discovery, 3) hit-to-lead generation, 4) lead optimization, 5) in vivo activity, absorption, distribution, metabolism, excretion (ADME) and toxicology optimization in animal models, and 6) human clinical trials.[1] The process begins with target discovery, where new biological targets relevant to a specific disease are identified. A critical aspect of this stage is understanding the target's role in the disease's underlying biological mechanisms.[2] Once a viable target is established, hit discovery follows, aiming to identify small-molecule compounds that modulate the target's activity. Traditionally, this is achieved through high-throughput screening (HTS), where hundreds of thousands of compounds are tested against the target protein to identify potential modulators.[3] The subsequent hit-to-lead and lead optimization stages focus on refining the physicochemical and biological properties of identified compounds. Structure-Activity Relationship (SAR) studies guide this optimization, enhancing potency, selectivity, and other key pharmacological properties. In parallel, ADME and toxicology assessments are conducted, initially in vitro and later in vivo, to evaluate the compounds' pharmacokinetic profiles and ensure their safety for clinical trials. Each stage in this pipeline is scientifically demanding, time-consuming, and costly. Contrary to Moore's law, which predicts exponential growth in computing power, Eroom's law observes a steady decline in pharmaceutical productivity, with the number of FDA-approved drugs per billion US dollars invested halving approximately every nine years since 1950.[4] The estimated cost of developing a new drug now reaches up to $3 billion,[5] with the entire process typically spanning 10 to 15 years, and clinical trials alone consuming nearly a decade.[6] The overall success rate from initial

in vitro screening to market approval is estimated to be below 0.01%.[4] Historically, only about 1,900 compounds have received FDA approval,[7,8] despite at least 119 million synthesized and researched molecules.[9] Notably, a significant proportion of marketed pharmaceuticals resulted from serendipitous discoveries.[10] The pharmaceutically relevant chemical space is vast, estimated to contain between ~ 10^{30} and 10^{60} possible molecules,[11] yet less than 10^{9} have been synthesized and explored. This highlights the immense challenge of molecular discovery, which requires multi-property optimization in a practically infinite discrete search space. Traditionally, this process has been driven by medicinal chemists' empirical knowledge, synthetic intuition, and experience. While invaluable, these approaches are inherently biased, ad hoc, and non-exhaustive, underscoring the need for innovative strategies to navigate this vast chemical landscape.

1.1.1 Traditional Computer-Aided Drug Discovery

To standardize the traditionally subjective process of drug discovery, computational approaches have become indispensable. Various computer-aided drug design (CADD) methods have been developed and widely adopted, encompassing two primary strategies: structure-based drug design (SBDD) and ligand-based drug design (LBDD), each with distinct requirements. SBDD relies on three-dimensional (3D) structural information, such as X-ray crystallography or cryo-electron microscopy (cryo-EM) structures of proteins and protein-ligand complexes, to identify binding sites and optimize ligand binding affinity. For example, docking calculations employ scoring functions to evaluate receptor-ligand interactions and estimate binding energies.[12–14] Additionally, molecular dynamics (MD) simulations use highly parameterized force fields to calculate atomic interactions and system energies, providing dynamic ligand binding poses and enabling detailed contact analysis for drug design.[15] Free energy perturbation (FEP) methods have also been introduced to predict the relative potencies of congeneric compounds in structure-based settings.[16,17] At the ligand level, LBDD focuses on non-linear quantitative structure–activity relationship (QSAR) analysis,[18] building predictive models that correlate molecular structure with biological activity or physicochemical properties by leveraging machine learning (ML) techniques.

1.1.2 Machine Learning in Drug Discovery

The integration of ML has significantly expanded the scope of QSAR modeling, enabling the capture of non-linear relationships for molecular activity and property predictions.[18] ML models such as support vector machines (SVMs)[19] and random forests (RFs)[20] are commonly applied to binary classification tasks, predicting whether a compound is active or inactive against a target of interest.[21] These models are trained on datasets comprising known active compounds alongside randomly selected inactive compounds. Deep learning (DL) has further advanced data-driven modeling by analyzing large and diverse datasets, extracting complex non-linear patterns for compound potency predictions.[22] Various neural network (NN) architectures have been employed, including convolutional NN (CNN),[23] recurrent NN (RNN),[24] graph convolutional network (GCN),[25] and message-passing NN (MPNN).[26] However, DL models require extensive training data to learn internal parameters effectively, posing challenges in early-phase drug discovery where data availability is limited.[27] Assessing ML/DL models for quantitative compound potency prediction in benchmarking settings also presents challenges. In particular, benchmark predictions from different ML/DL models are often separated from randomized predictions by only small error margins, making it difficult to unambiguously evaluate the relative model performance.[28] Due to data scarcity and inherent evaluation limitations, no universally accepted criteria currently exist for prioritizing ML approaches in quantitative compound potency predictions.[28] In addition to qualitative activity classification, semi-quantitative approaches can be attempted by deep generative models (DGMs), which aim to generate molecules with desired properties such as highly potency, thus providing a complementary approach to traditional potency prediction.[22]

1.2 Molecular Representations

Representing molecular data concisely and unambiguously while capturing all relevant structural and chemical characteristics is crucial for applying ML/DL in drug discovery. Effective molecular representations should be interpretable by both humans and machines while providing sufficient information for computational processing. To meet these requirements, various molecular representation formats have been

developed over the years, with fingerprints, molecular graphs, and Simplified Molecular Input Line-Entry System (SMILES) being among the most commonly used (Figure 1).

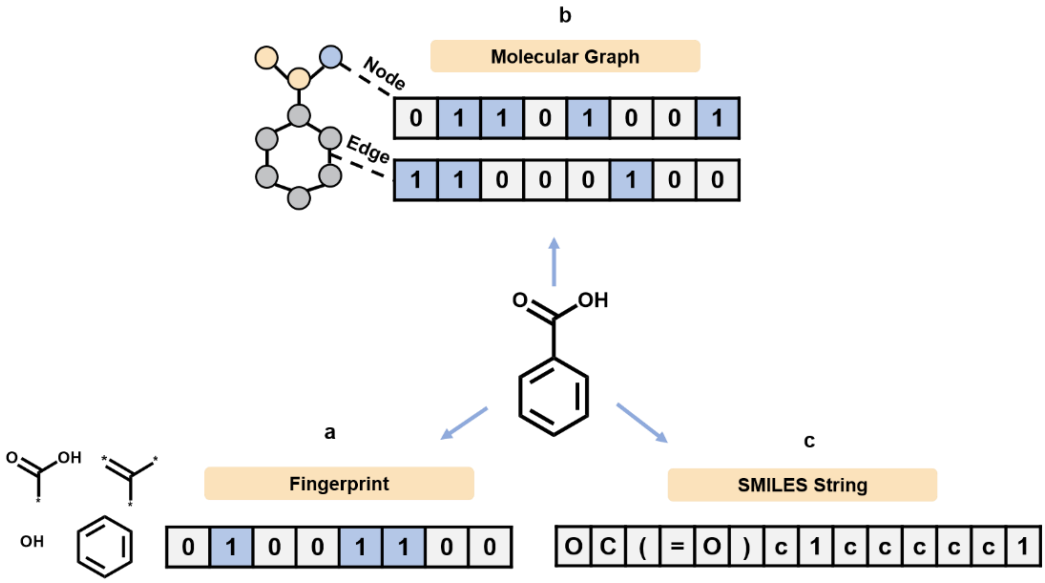


Figure 1: Molecular representations. The most commonly used include (a) Fingerprints, which encode local substructures as bits, (b) Molecular graph, in which nodes correspond to atoms and edges represent chemical bonds, and (c) SMILES strings, which use specific characters to encode atoms, bonds, branches, aromaticity, rings, and stereochemistry of molecules.

1.2.1 Fingerprints

Molecular fingerprints encode the presence (or absence) of substructures within molecules, typically in a sparse vector format. They generally fall into two categories: (1) substructure-key fingerprints based on matching molecular substructures of an expert-defined set and (2) topology-based fingerprints that use algorithmic enumeration and hashing of molecular substructures. In substructure key-based fingerprints, molecules are encoded using a predefined dictionary of structural features, where each bit position corresponds to a specific substructure. One commonly used example is the Molecular ACCess System (MACCS) fingerprint, which includes 166 predefined structural keys.[29] In contrast, topology-based fingerprints use a hashing function to encode structural features. A prominent example is the Extended Connectivity

Fingerprint (ECFP),[30] which is based on the Morgan algorithm.[31] ECFPs encode circular atom environments up to a specified radius, typically using a fixed-length bit vector (e.g., 1024 or 2048 bits). Beyond two-dimensional (2D) fingerprints, 3D molecular representations incorporate spatial properties such as molecular conformation and topology.[32,33] Examples include protein-ligand interaction fingerprints like the Protein-Ligand Extended Connectivity Fingerprint (PLEC)[34] and Extended Connectivity Interaction Features (ECIF)[35] which capture 3D binding interactions. A key limitation of fingerprint-based representations is their inconvertibility, meaning the complete molecular structure cannot be directly reconstructed from a fingerprint.[36] Despite this, fingerprints remain widely used in ML-based classification tasks, such as distinguishing active from inactive compounds for a given target.

1.2.2 Molecular Graph

Molecules can be represented as undirected graphs, where nodes correspond to atoms and edges represent chemical bonds. Each node is labeled with an atomic identity, while edges indicate bond valence. Formally, a molecular graph is defined as $G = (V, E)$, where nodes $v_i \in V$ represent atoms, and edges $(v_i, v_j) \in E$ define the bonds between atoms v_i and v_j. The adjacency matrix of a molecular graph encodes structural information by listing atomic numbers along the main diagonal and indicating connectivity between atoms through bond type values (e.g., single, double, triple, or aromatic bonds).[37,38] Molecular graph representations have been widely adopted in combination with CNN or graph neural networks (GNNs) for predictive modeling. However, they require significant memory due to the large amount of information necessary to encode a single molecule. To address this limitation, one-dimensional (1D) sequence-based representations such as SMILES have been developed, providing a more compact and human-readable alternative.[39]

1.2.3 SMILES

Sequence-based representations use linear strings to encode molecular structures, offering simplicity in processing and storage. The most widely used 1D formats are the International Chemical Identifier (InChI) and SMILES. InChI, developed by IUPAC, encodes molecular structures hierarchically, capturing detailed chemical information such as charge and stereochemistry.[40] However, InChI strings tend to be lengthy and complex, particularly for large molecules. To improve searchability and retrieval, a hashed version called InChIKey was introduced. Despite its structural comprehensiveness, InChI's intricate syntax and valency/branching constraints make it less practical for use in LMs.[41] In contrast, SMILES provides a more intuitive, compact string-based representation, using specific characters to denote atoms, bonds, branches, aromaticity, rings, and stereochemistry.[42] This character-level encoding facilitates efficient tokenization, making SMILES the preferred input format for molecular LMs. However, a key challenge with SMILES is its non-uniqueness—multiple valid SMILES strings can represent the same molecule. This variability can be addressed through canonicalization, which standardizes SMILES strings, or by leveraging multiple representations as a data augmentation strategy, enhancing molecular property prediction[43–45] and molecular generation.[46,47] In generative modeling, SMILES strings are typically converted into one-hot encodings, enabling models to learn categorical distributions. However, a common issue is the generation of invalid SMILES strings, often caused by mismatched ring closure symbols or bond valence violations. To mitigate this, DeepSMILES was introduced, modifying SMILES syntax to eliminate unbalanced parentheses.[48] SELF-referencIng Embedded Strings (SELFIES) further addressed validity concerns by enforcing valence-bond constraints through predefined derivation rules.[49] Unlike SMILES, SELFIES inherently guarantee 100% validity during generation by ensuring proper branch lengths and ring closures. Nevertheless, SELFIES strings can sometimes be too short to represent meaningful molecular structures. Despite these alternatives, canonicalized SMILES remains the most widely used format in generative molecular models due to its compatibility with language modeling and sequence generation.

1.3 Deep Generative Models for Molecular Design

DGMs have attracted increasing attention in molecular design due to their ability to learn implicit chemical knowledge from data by identifying structural patterns—such as valency rules, reactive groups, molecular conformations—to generate molecules with desired properties. Unlike rules-based or enumeration methods that rely on predefined chemical rules, DGMs operate more autonomously, reducing the likelihood of producing non-synthesizable molecules with unstable groups. DGMs are primarily categorized based on the molecular representation they employ, typically SMILES strings or molecular graphs. Correspondingly, these models can be classified into sequence-based models and graph-based models. DGMs not only generate molecules with targeted properties but also explore broader chemical space by leveraging biased learning methods that guide generation towards molecules meeting specific conditions or exhibiting analogous structures and chemical properties.[50–53] Additionally, DGMs effectively map large areas of chemical space by learning chemical rules that facilitate molecular structure reconstruction from encoded representations.[54,55] Their cost-effectiveness and time efficiency further boost their application in modern molecular design.

1.3.1 Variational Autoencoders

Variational autoencoders (VAEs) extend the classical autoencoders (AEs) framework, which consists of an encoder that transforms input data into a lower-dimensional latent vector and a decoder that reconstructs the original input from this latent representation,[56] as illustrated in Figure 2. While AEs focus on accurate input reconstruction, VAEs introduce regularization by modeling the latent space as probability distributions rather than discrete points, enhancing the model's generalization capacity. The first VAE-based generative model for molecular de novo design was introduced in 2018.[57] Recent advancements have aimed at achieving disentangled representations in VAEs, where each latent variable encodes a distinct molecular property. In molecular generation, disentangled VAEs such as ChemVAE,[57] GrammarVAE,[58] and SD-VAE[59] allow fine-tuning of specific molecular properties by adjusting the corresponding latent variables. Conditional VAEs (CVAEs) extend the VAEs framework by incorporating molecular properties into the encoding process, enabling the generation of

drug-like molecules with user-defined properties and allowing property control without compromising overall molecular structure.[60] Additionally, Adversarial autoencoders (AAEs) offer an alternative to VAEs by introducing adversarial training, where the encoder maps inputs to a latent space while a discriminator attempts to distinguish encoded points from samples drawn from a predefined distribution.[61] Notable AAE-based models have shown promise in generating chemically diverse molecules with tailored properties.[62–65]

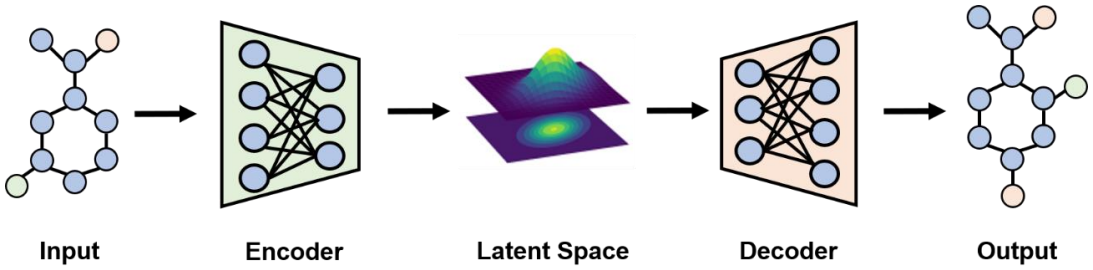


Figure 2: Variational autoencoder. Shown is a schematic representation of the VAE architecture, comprising input, encoder, latent space, decoder, and output modules. During training, input molecules are encoded into a continuous latent space. For generation, random points are sampled from this space and decoded to generate novel molecular structures.

1.3.2 Generative Adversarial Networks

Generative adversarial networks (GANs)[66] adopt a fundamentally different approach from VAEs by not relying on an explicit probability density function. Instead, GANs employ an adversarial training framework consisting of two competing NNs: a generator and a discriminator (Figure 3). The generator aims to produce molecules that closely resemble real ones, while the discriminator attempts to distinguish between synthetic and real molecules. This adversarial process continues until the generator produces molecules that the discriminator can no longer reliably differentiate from real data. Early applications of GANs in molecular generation include models such as ORGAN[67] and ORGANIC,[68] which demonstrated the ability to generate a diverse array of new structures, some featuring entirely novel scaffolds.

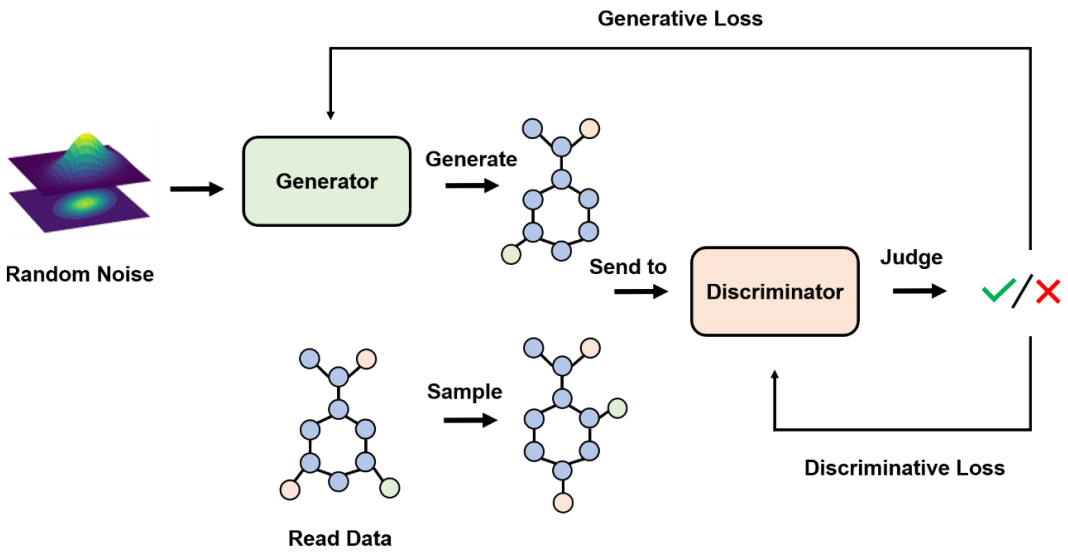


Figure 3: Generative adversarial network. Schematic of the GAN framework comprising a generator and a discriminator. The generator learns to produce realistic molecular structures, while the discriminator distinguishes real from generated data. Through adversarial training, the generator improves its ability to generate convincing molecules.

1.3.3 Graph Neural Networks

GNNs extend CNNs to process graph-structured data, where nodes represent atoms and edges represent bonds.[69] GNNs operate through pairwise message passing, enabling nodes to update their representations by exchanging information with neighboring nodes[70] (Figure 4). This trainable architecture makes GNNs particularly well-suited for generating novel molecular graphs from databases of existing structures by encoding chemically relevant bonds and atoms directly as edges and nodes within a mathematical graph. Various GNN architectures have been applied in molecular design, including Graph Isomorphism Networks (GINs),[71] Graph SAGE,[72] Graph Attention Networks (GATNets),[73] and Graph Convolutional Networks (GCNs).[74] These architectures have been applied for prediction tasks such as protein–protein interactions, protein–drug interactions, drug–disease interactions, and drug repurposing.

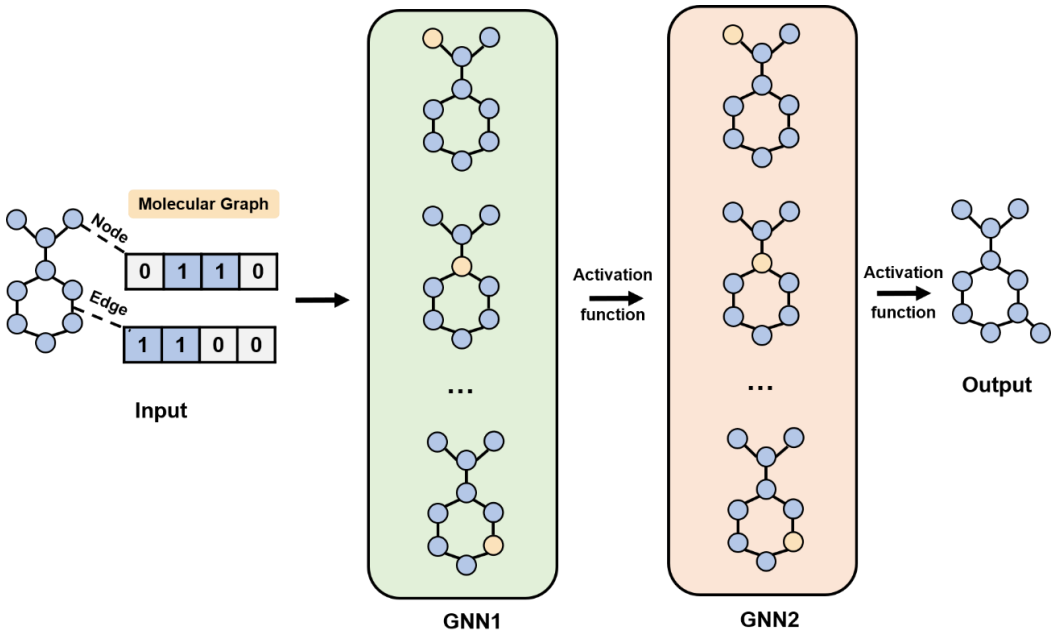


Figure 4: Graph neural network. Schematic of the GNN, where a molecular graph consists of atoms as nodes and chemical bonds as edges. The GNN operates through two key modules: message passing and node updates. During message passing, each node aggregates information from its neighbors based on the graph structure and node features, which is then processed through an activation function to update the node representation. This process is repeated across layers to capture higher-order dependencies, with the updated node representations used for molecule generation.

1.3.4 Flow-Based Models

Flow-based models explicitly define data densities through invertible transformations,[75] as illustrated in Figure 5. Normalizing flows transform complex data densities into simpler distributions via a series of differentiable functions, enabling the application of techniques such as Gaussian mixture modeling and log-likelihood maximization, which are especially useful in classification tasks. Compared to GANs and VAEs, flow-based models offer advantages such as eliminating output noise and enhancing training stability.[76] In molecular generation, flow-based models construct molecular graphs by creating adjacency and feature matrices.[77] Autoregressive versions of these models further improve validity and quality control by generating molecular graphs step-by-step.[78,79] Additionally, strategies such as gradient ascent on a property

predictor[77] or reinforcement learning (RL) have been employed to guide molecule generation, ensuring that generated structures align with desired properties.[78,79] Hybrid approaches, like Graph Flow-VAE, combine VAE encoders with flow-based decoders to harness the strengths of both frameworks, enhancing molecular generation capabilities.[80]

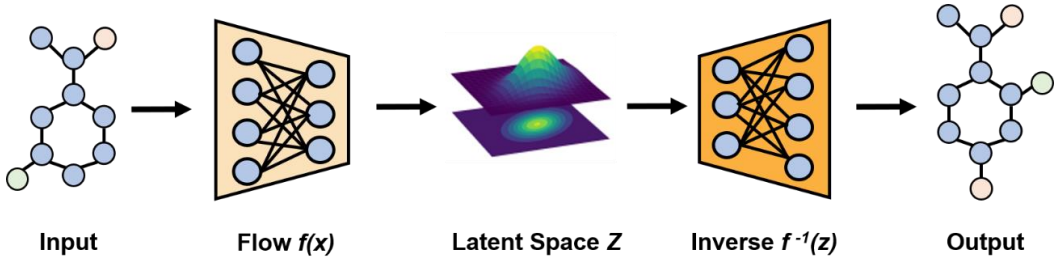


Figure 5: Flow-based models. Schematic of the flow-based generative model, consisting of a series of invertible transformations that map molecular data from the original space to a latent space with a Gaussian distribution. During training, the model optimizes these transformations to maximize data likelihood. For generation, samples are drawn from the Gaussian latent space and transformed back through the learned inverse mappings to generate novel molecular structures.

1.3.5 Diffusion-Based Models

Diffusion-based models have gained considerable attention in molecular generation.[81,82] Unlike flow-based models, diffusion processes do not require invertible transformations. These models operate in two distinct phases: during the forward process, stochastic noise is iteratively added to molecular data over a Markov chain, progressively transforming the data into a Gaussian distribution (Figure 6). Notably, this forward pass involves no trainable parameters. In the reverse process, a deep NN is trained to gradually denoise samples from the Gaussian distribution, reconstructing molecules with desired characteristics. One prominent example is the equivariant diffusion model (EDM),[81] which simultaneously operates on categorical atom types and continuous atom coordinates, enabling the generation of 3D molecular structures while ensuring equivariance to Euclidean transformations. Despite their

promising results, diffusion-based models present challenges such as high computational demands and prolonged training and sampling times.

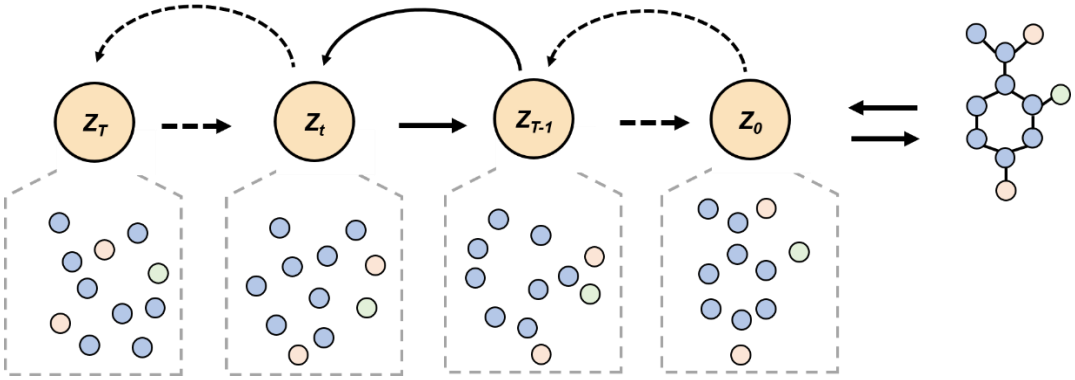


Figure 6: Diffusion-based models. Schematic of the diffusion-based generative model, comprising two main components: the forward process (curved arrows), where noise is progressively added to the data, transforming it into a Gaussian distribution, and the reverse process (straight arrows), where data is reconstructed by iteratively removing noise through learned denoising steps.

1.3.6 Recurrent Neural Networks

Recurrent Neural Networks (RNN), originally introduced by Hopfield over 40 years ago,[83] are a class of NN designed to process sequential data (Figure 7). In molecular generation, RNNs are employed to handle 1D molecular representations, such as SMILES strings. An RNN processes molecular sequences token by token, using hidden states to retain contextual information across sequence steps. These hidden states are updated recurrently, allowing the network to capture sequential dependencies and generate chemically valid structures. However, RNNs encounter difficulties when learning long-range dependencies due to the vanishing or exploding gradient problem during backpropagation, especially in lengthy sequences. To address these challenges, variants such as Long Short-Term Memory (LSTM) units and Gated Recurrent Units (GRUs) were developed. These gated architectures incorporate trainable gating mechanisms that regulate information flow, effectively mitigating gradient-related issues and enabling the learning of long-term dependencies at the cost of increased model complexity. [84,85]

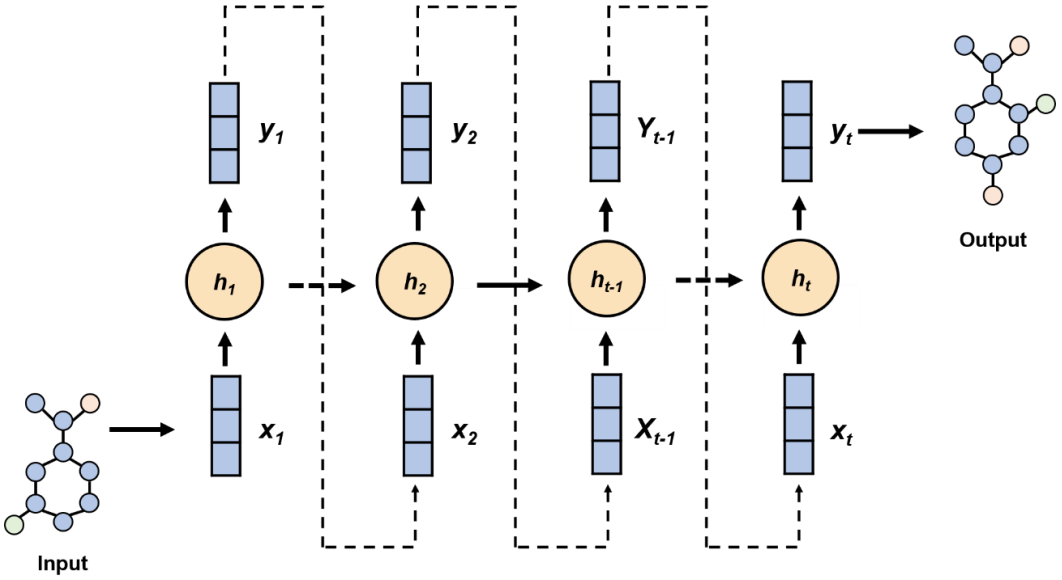


Figure 7: Recurrent neural network. Shown is a schematic diagram of the RNN model. The RNN generation process is autoregressive: at each step, the model generates a probability distribution over all possible tokens based on inputs from the previous and current steps. A token is then sampled from this distribution as the output for the current step, which is used to predict the next token in the sequence.

1.3.7 Transformer

Transformers, introduced by Vaswani et al. in 2017,[86] have surpassed RNNs in processing sequential data due to their ability to capture long-range dependencies through self-attention mechanisms. The basic transformer architecture, as illustrated in Figure 8, consists of multiple encoder-decoder neural modules equipped with attention mechanisms. Within this architecture, the encoder module comprises a stack of sub-layers, including a multi-head self-attention sub-layer and a fully connected feed-forward network sub-layer. The encoder reads an input sequence and compresses it into a context vector in its final hidden state, which then serves as the input for the decoder. The decoder, consisting of a feed-forward sub-layer and two multi-head attention sub-layers, reinterprets the context vector to generate an output sequence token by token. During training, both the encoder and decoder leverage the attention mechanism to comprehensively learn from the feature space. Unlike RNNs, Transformers process input sequences in parallel, enhancing efficiency and scalability.[87] This

advantage has led to the development of various transformer variants, such as Bidirectional Encoder Representations from Transformers (BERT),[88] Generative Pre-trained Transformer (GPT),[89] and Text-to-Text Transfer Transformer (T5),[90] each tailored to different encoder-decoder architectures. Transformers commonly follow a pre-training and fine-tuning paradigm. In this framework, models are pre-trained on massive datasets to learn general language representations, which can then be fine-tuned for specific downstream tasks. This scalability has enabled the creation of large pre-trained models, such as pre-trained BERT and GPT, further enhancing performance across diverse NLP applications. BERT, introduced by Google in 2018, employs an encoder-only Transformer architecture to achieve bidirectional context understanding,[88] as illustrated in Figure 8. It consists of an embedding layer, multiple transformer encoder layers, and a task-specific output layer. In the embedding layer, input word tokens are embedded into a continuous vector space, and a pre-defined positional encoding vector is added to each embedding vector. In the encoder layers, each token exchanges information with all others through the self-attention mechanism. The final layer typically includes a fully connected dense layer, which further processes the encoder's output to address specific tasks such as text classification or next-sentence prediction. GPT, developed by OpenAI, adopts a decoder-only architecture comprising positional encoding, a masked multi-head self-attention module, a pointwise feed-forward network unit, and normalization operations,[89] as illustrated in Figure 8. Unlike BERT, which uses a bidirectional approach by considering both left and right contexts, GPT employs a unidirectional approach, predicting the next word based solely on preceding words. This left-to-right method makes GPT particularly effective for natural language generation and creative writing. T5, introduced by Google in 2019, takes a unified approach to various NLP tasks, including machine translation. Rather than treating translation as a sequence-to-sequence task, T5 frames all NLP tasks as text-to-text tasks, where both inputs and outputs are treated as text strings.[90] In translation, for instance, the source language text serves as the input, while the target language text serves as the output. This unified framework simplifies the translation process by enabling consistent handling of diverse language mapping pairs. Pre-trained on a large corpus of text and fine-tuned on translation-specific data, T5 has achieved promising results in machine translation tasks.

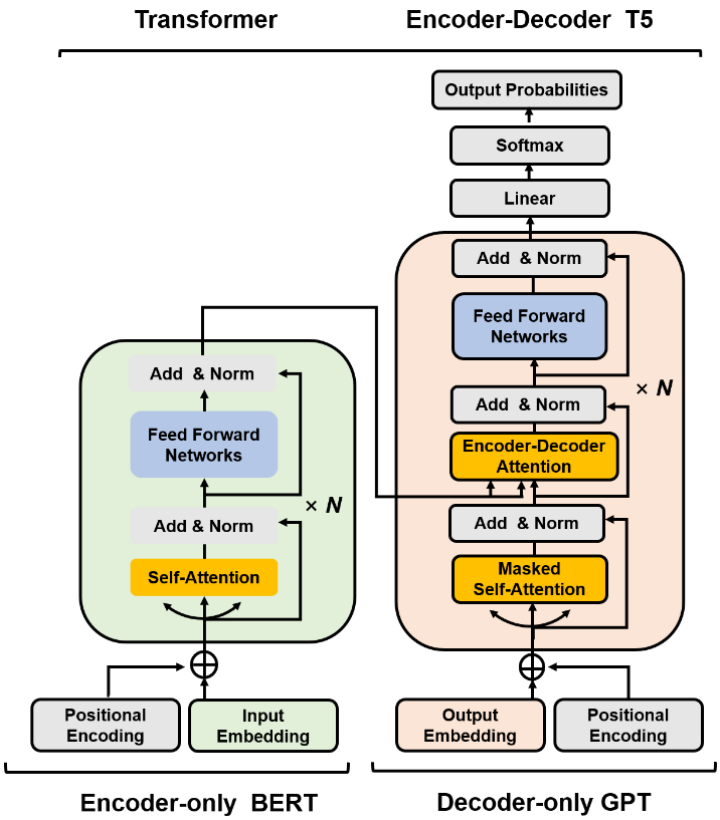


Figure 8: Transformer model. Schematic of the basic transformer model, comprising multiple encoder-decoder modules with attention mechanisms. The encoder consists of layers with multi-head self-attention and a feed-forward network, processing the input sequence into a context vector passed to the decoder. The decoder, with a feed-forward layer and two multi-head attention layers, generates the output sequence token by token. Transformer variants include encoder-only models (e.g., BERT), decoder-only models (e.g., GPT), and encoder-decoder models (e.g., T5).

1.4 Language Models in Drug Discovery

Transformer-based LMs are designed to translate sequences of characters, making them highly versatile for various machine translation tasks.[91] Their adaptability has led to widespread adoption in life sciences and other scientific fields.[92,93] These models, often employing encoder-decoder architectures with attention mechanisms, exemplify the power of DL in enabling novel applications. A key strength lies in their capacity to learn mappings between diverse types of sequential or textual data representations,

opening opportunities for generative modeling across multiple biological modalities.[93,94,95] For example, sequence translation tasks may involve mapping between compounds, proteins, or chemical reactions, offering potential applications in drug discovery. Depending on the type of biochemical sequence, LMs can model small molecules, macromolecular proteins, genomic sequences, and even their combinations.[93] Various tokenization schemes are implemented accordingly to handle these different modalities.[93]

1.4.1 Chemical Language Models

In the context of chemical compounds, LMs designed to translate molecular sequences are known as CLMs.[96,97] These models learn the chemical vocabulary and syntax used to represent molecules while capturing conditional probabilities of character occurrence based on preceding characters in a sequence.[95] CLMs are typically pre-trained on large molecular datasets to grasp fundamental chemical patterns and subsequently fine-tuned on smaller specific datasets to focus on compounds with desired properties, such as activity against target proteins. Several CLM architectures have been developed for different purposes. Encoder-only models, such as MolBERT,[98] SMILES-BERT,[99] and chemBERTa,[100] excel at understanding molecular representations, enhancing tasks like molecular property prediction. Decoder-only models, such as MolGPT[101] and cMolGPT,[102] as well as encoder-decoder architectures like Chemformer,[103] ChemReactNet,[104] and X-MOL,[105] have been applied to molecular generation and chemical reaction prediction. A crucial step in CLM development is tokenizing input data, which may include chemical structures, molecular properties, or other features. For compound SMILES strings, a basic approach is character-level tokenization, where each character is treated as a separate token. However, this method has limitations, as chemically meaningful information about single atoms may span multiple characters, leading to ambiguity. To address this, SMILES are typically tokenized at the atom level using regular expressions[106] or by incorporating positional and connectivity information to distinguish identical atoms in different molecular contexts.[107] For example, compounds can be encoded as canonical SMILES strings with atoms represented as single-character tokens (e.g., "C" or "N"), two-character tokens (e.g., "Cl" or "Br"), or tokens enclosed in brackets (e.g., "[nH]" or "[O-]"). Additionally,

substructure-level tokenization methods, such as SMILES Pair Encoding (SMILES-PE),[108] iteratively merge frequently occurring token pairs to build a more compact vocabulary, drawing inspiration from byte-pair encoding. In conditional CLMs, molecular property values must also be transformed into input tokens. Various tokenization strategies have been proposed, including binning[109,110] and numerical tokenization.[111] For instance, in a binning-based approach, the globally observed potency range of [4.00, 12.52] pKi units was divided into 852 bins, each with a constant width of 0.01. This fine granularity captures the limits of experimental potency annotations, encoding each bin as a single token and assigning potency values accordingly.

1.4.2 Protein Language Models

Much like words form sentences, protein sequences — strings of 20 amino acids that make up the protein "vocabulary" — determine the structure and function of proteins. This ordering of amino acids is crucial, as it influences how proteins fold and interact within biological systems. Inspired by NLP principles, PLMs embed long protein sequences as sentences of characters, where one or more residues form words.[112,113] The resulting sequence embeddings implicitly capture structural and functional characteristics of proteins, making them valuable for diverse applications.[114] Early PLMs primarily used BERT-like encoder-only architectures and denoising autoencoding training objectives. These models were pre-trained on large, unlabeled datasets of protein sequences to encode protein sequences or structures into fixed-length vector representations, capturing structural and functional characteristics for downstream tasks. Prominent pre-trained protein sequence encoders include ESM-1b,[112] ProteinBERT,[115] and ProtTrans,[116] which have been applied to tasks such as secondary structure prediction, contact prediction, remote homology detection, and the prediction of post-translational modifications and biophysical properties. Decoder-only PLMs have also been developed, playing a predominant role in protein generation and design. Notable models include ProGen[117] and ProtGPT2.[118] Additionally, T5-based encoder-decoder PLMs, such as ProtT5,[119] facilitate translation between protein sequences and structures. For PLM modeling, protein sequences are encoded as standard uppercase residue symbols and tokenized using space delimiters. The token

vocabulary includes 21 entries: the 20 canonical amino acids and a special 'X' token denoting for rare amino acids.

1.4.3 Genomic Language Models

Genomic language models (GLMs), a class of LMs trained on DNA and RNA sequences, enable the interpretation of genomes and the analysis of interactions between DNA/RNA elements at multiple biological scales. Key applications of GLMs include functional constraint prediction and sequence design.[120] In encoder-only architectures, notable models include DNABERT,[121] iEnhancer-BERT,[122] and scBERT,[123] these models employ a masked training mechanism where portions of gene sequences are masked, prompting the model to predict and complete them, thereby learning inherent patterns within gene sequences. Decoder-only models have also gained attention due to their generative capacity. For example, GenSLMs[124] leverage genome-scale LMs comprising multiple layers of attention-based decoders to elucidate the evolutionary dynamics of SARS-CoV-2, effectively capturing the evolutionary landscape of SARS-CoV-2 genomes. Encoder-decoder models in genomics, such as the Ensemble Nucleotide Byte-level Encoder-Decoder (ENBED)[125] and MegaDNA,[126] represent significant advancements in bioinformatics. These models combine the strengths of both encoder and decoder to analyze and interpret complex genomic sequences. The encoder compresses the input genomic data into a meaningful representation, capturing essential features and patterns, while the decoder generates or reconstructs sequences and performs other bioinformatics tasks. For genomic sequences, single-nucleotide tokenization—using a dictionary of four nucleotides (for DNA: "A," "C," "G," and "T"; for RNA: "A," "C," "G," and "U")—simplifies model interpretation and enhances its ability to handle genomic variations. Additionally, *k*-mer and byte-pair encoding (BPE) tokenization[127] create artificially defined nucleotide pair vocabularies, reducing input sequence length and enabling models to handle longer contexts.

1.5 Molecular Mappings and Transformations

One of the central challenges in drug discovery is identifying molecules that achieve an optimal balance of multiple properties. This challenge can be framed as a machine translation problem, where a source compound (input molecule) is translated into a target compound (output molecule) with improved properties. Using SMILES-based molecular representations, CLMs can be trained to predict optimized molecular structures.[109,110] Constructing diverse and well-defined source-to-target compound mappings is essential for CLM modeling, as these mappings capture structural transformations and their associated property changes.[94,95] ASs represent a foundation for studying molecular transformations and enable systematic analysis of structural modifications that impact biological activity or other molecular properties. For instance, in compound optimization, ASs are essential for assessing SAR progression.[128] Over the years, various computational methods in cheminformatics have been developed to analyze structural modifications, organize large ASs, and systematically monitor SAR progression.[129]

1.5.1 Matched Molecular Pairs

The Matched Molecular Pair (MMP) formalism provides a structural framework for defining molecular similarity. In most approaches, an MMP is defined as a pair of compounds differing by a small structural change at a single site (Figure 9).[130] The limited nature of these structural differences makes MMP analysis (MMPA) highly interpretable compared to many other similarity-based methods.[131] MMPA enables systematic analysis of chemical modifications, allowing researchers to quantify the average effect of a given transformation.[132,133] Identification of MMPs can be approached in three distinct ways, depending on algorithmic constraints and practical considerations. The first approach explicitly defines MMPs based on a set of predefined chemical transformations, specifying how one compound is converted into another.[130] A variation of this method uses predefined substructures instead of full transformations, simplifying the problem to a substructure search. While computationally efficient, this method is constrained by predefined rules. The second approach involves computing the maximum common substructure (MCS) of two molecules, ensuring that

the difference between them is confined to a single substructure modification.[134,135] Unlike the predefined transformation method, MCS does not rely on predefined rules and can discover novel transformations. However, MCS computations require pairwise molecular comparisons, making this approach computationally demanding. To mitigate this, pre-filtering strategies can reduce the number of necessary comparisons. The third approach systematically fragments molecules into core structures, classifying a pair as an MMP if both molecules can be reduced to the same core scaffold.[136] This rule-based fragmentation is computationally efficient, particularly for large datasets, and does not depend on predefined transformations. One commonly used method for this process is the retrosynthetic combinatorial analysis procedure (RECAP) algorithm, which applies 11 predefined bond cleavage rules to generate RECAP-MMPs.[137,138] Following MMP extraction, molecules sharing a common core scaffold can be grouped into matching molecular series (MMS). An MMS is defined as a set of two or more compounds that share a common molecular core but differ at a single substitution site, offering a structured approach for analyzing SAR trends.

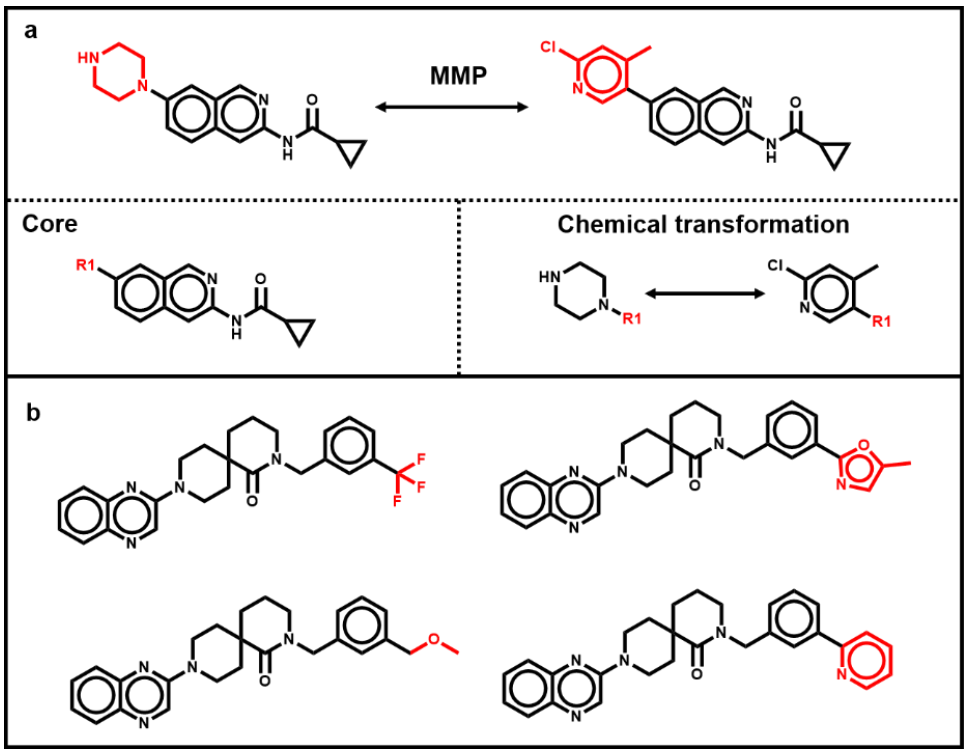


Figure 9: Matched molecular pair. Shown are exemplary analogues forming a pair (a) or a series (b). For the MMP relationship (a), the core structure and chemical transformation are provided. The structural modifications (exchanged substituents) are highlighted in red.

1.5.2 Systematic Identification of Analogue Series

The compound-core relationship (CCR) approach is designed to identify structural analogues that share a common core structure with modifications at multiple substitution sites.[139] In this method, systematic fragmentation of molecules at one or more positions generates a connected core structure with corresponding R-group substituents (Figure 10). Unlike the original Hussain and Rea fragmentation method,[136] which primarily focuses on single-site modifications, the CCR approach extends the analysis to multi-site variations, allowing for a more comprehensive exploration of analogue series. To ensure chemical feasibility, the CCR method incorporates retrosynthetic fragmentation rules, ensuring that the generated analogue series reflect realistic molecular transformations.[139] Additionally, the concept of the hydrogen-substituted core structure was introduced, where all substituent positions of the core scaffold are replaced by hydrogen atoms. By grouping fragmentations that share the same hydrogen-substituted core, analogue series with substituents at different sites naturally emerge. This strategy facilitates the identification of scaffolds with multiple potential modification sites, even when only a limited number of non-hydrogen substitutions are present in the dataset.

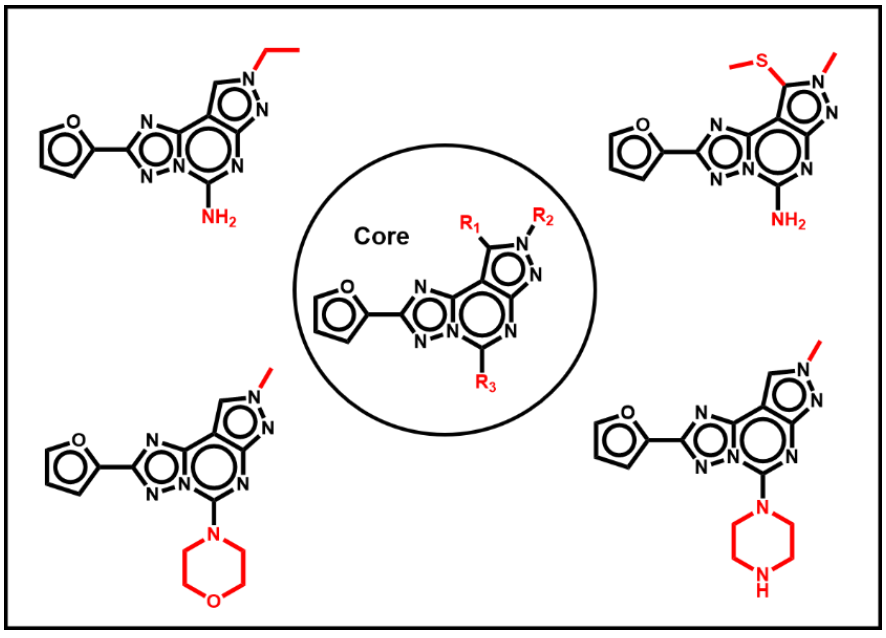


Figure 10: Analogue series. An AS of four analogue compounds generated by CCR algorithm is shown. The Markush structure representing the AS is displayed in the center. The structural modifications (exchanged substituents) are highlighted in red.

1.5.3 Structure-Activity Relationship Matrix

The SARM methodology was developed to systematically identify and organize structurally related ASs from large compound datasets.[140,141] SARM identifies structurally analogous core scaffolds and organizes them into a matrix-like format, akin to R-group tables,[140] as illustrated in Figure 11. Each SARM captures a collection of analogue series with structurally related cores, enabling the systematic extraction of SAR from compound datasets.[140] Depending on the structural diversity present, a dataset typically yields multiple SARMs. SARM generation follows a two-step fragmentation process adapted from MMPA.[136] First, compounds are fragmented at exocyclic single bonds, producing "keys" (core structures) and "values" (substituents), which are stored in an index table (Figure 11). In the second step, the core scaffolds undergo further fragmentation, identifying subsets of cores that differ only by a single chemical change. This process results in a second index table (Figure 11). Each subset of analogous core structures, along with the compounds containing each core, forms an individual SARM. The matrix structure follows a well-defined organization: each row represents an AS where all molecules share the same core scaffold, while each column contains compounds from different ASs that share the same substituent. Each cell in the matrix corresponds to a unique compound, which could be either an existing molecule or a virtual analogue (i.e., an unexplored combination of core and substituent). This matrix-based representation allows for intuitive SAR visualization, especially when potency values (or other molecular properties) are used to color-code matrix cells. By applying potency-based coloring, SARMs effectively illustrate structure-activity trends across compound datasets.[142] Recent advances have integrated SARM with CCR, enabling a more systematic approach to identifying ASs that incorporate both scaffold modifications and multi-site substitutions.[143] This integration further enhances the exploration of molecular transformations, providing deeper insights into optimizing lead compounds.

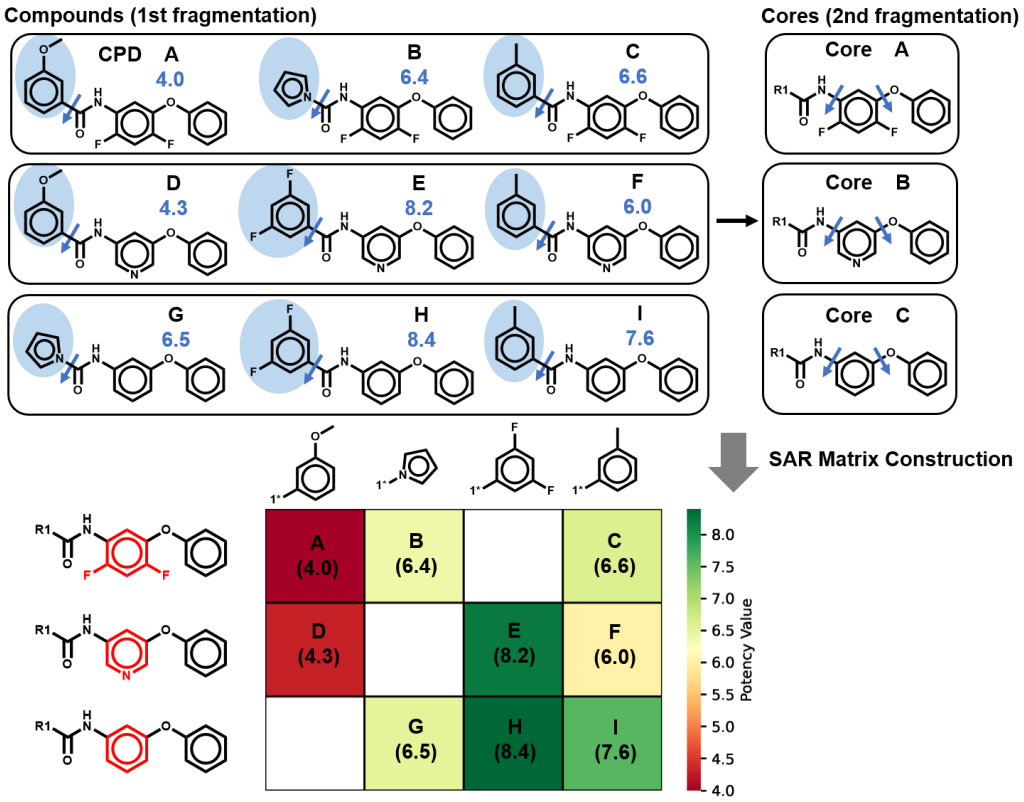


Figure 11: Structure-activity relationship matrix. SARM construction is illustrated using a model dataset of nine compounds (CPD A–I), with pIC_{50} values shown in blue. Substituents distinguishing analogues are highlighted on a light blue background. SARM generation follows a dual-step fragmentation scheme, identifying analogue series with structurally related cores. Substructures distinguishing cores are shown in red. Each SARM cell represents a unique compound (A–I), while empty cells represent virtual analogues—unexplored combinations of core and substituent.

1.6 Evaluation Metrics

An objective and fair evaluation of DGMs is essential for molecular generative design. The final step in generating molecules using these models involves assessing the quality and relevance of the generated molecules through well-defined evaluation metrics. These metrics can be broadly categorized into four main types. The first category assesses the overall performance of the model by considering all generated molecules as a whole. Key metrics include validity, uniqueness, and novelty, typically expressed as percentages.[91] Validity measures the proportion of generated molecules

that adhere to fundamental chemical rules, reflecting the model's understanding of chemical syntax and grammar. Uniqueness indicates the percentage of distinct molecules within the generated set, while novelty quantifies the fraction of generated molecules absent from the training dataset, thereby indicating the model's ability to learn the underlying data distribution and produce original structures. Higher novelty values suggest reduced model overfitting.[91] The second category focuses on evaluating the properties of individual molecules, such as drug-likeness and synthetic accessibility. The quantitative estimate of drug-likeness (QED)[144] quantifies drug-likeness on a scale from 0 to 1 by integrating several molecular descriptors, including molecular weight, logP, topological polar surface area, hydrogen bond donors and acceptors, aromatic rings, rotatable bonds, and the presence of undesirable chemical functionalities. Similarly, the synthetic sccessibility (SA) score[145] estimates molecular synthetic feasibility, ranging from 1 (easy to synthesize) to 10 (very difficult to synthesize), based on fragment contributions and structural complexity. Benchmarking tools like GuacaMol[146] and MOSES[147] are frequently employed to standardize performance evaluation. The third category evaluates the bioactivity or physicochemical properties of generated molecules. Docking-based scoring functions and ML-based prediction models are commonly applied for this purpose. However, these approaches introduce additional uncertainty, as hypothetical models guide the generative process. Finally, assessing the generative model's generalization ability is crucial, as it reflects the model's predictive performance on novel tasks. Proper evaluation of generalizability requires rigorous data splitting and selection to prevent data leakage and biased outcomes.[148] Time-based splits, simulating real-world scenarios, are preferred in industry settings.[149] However, absence of temporal information in public datasets limits their use in academic research. To overcome this challenge, models can be tested on target-based activity classes excluded during training. Additionally, training and evaluation can be conducted on structurally distinct compound subsets, generated through comprehensive analogue series identification and splitting.[150,151] The most stringent proof-of-concept involves verifying the model's capacity to reproduce known compounds with desirable bioactivity or physicochemical properties not encountered during training.[152]

1.7 Thesis outline

This dissertation investigates the development and application of transformer-based CLMs to address various challenges in medicinal chemistry and molecular design. The thesis is structured into eight chapters, with Chapters 2 to 7 presenting six original publications that constitute the core of this work.

- *Chapter 2* focuses on developing and evaluating a CLM for AC prediction. To achieve this, a conditional transformer is adapted for constructing a CLM. Structural analogue pairs from diverse activity classes, conditioned on potency differences, are generated for model pre-training. Subsequently, the pre-trained models are fine-tuned to predict ACs and benchmarked against other machine learning methods.
- *Chapter 3* extends the application of the CLM to predict highly potent compounds from weakly potent template molecules. A conditional CLM is implemented to facilitate compound design conditioning on large potency differences. Additionally, a novel compound test system is devised to rigorously assess model performance.
- *Chapter 4* explores the prediction of potent compounds in low-data regimes. To address this challenge, meta learning is incorporated into the conditional transformer, enhancing the model's ability to generalize from limited data. The performance of this meta learning-based CLM is systematically compared to other basic CLM in the presence of varying amounts of fine-tuning data.
- *Chapter 5* investigates the extension of AS in lead optimization, focusing on series with multiple substitution sites. A specialized coding and tokenization scheme is designed to represent evolving AS, and a transformer variant is implemented to predict new potent analogues. The model's performance is evaluated and compared to other generative models.
- *Chapter 6* advances the analogue design approach by targeting potent compounds with both core structure and substituent modifications at multiple sites. To support this task, a novel compound decomposition protocol is devised to accommodate analogue series with complex substitution patterns. Furthermore, a new coding and tokenization scheme is developed to represent

core structure-substituent combinations. Multiple model variants are derived, investigated, and compared.

- *Chapter 7* extends the generative design framework to predict active compounds with desired potency from target protein sequences. A dual-component conditional LM is designed to learn from multimodal data, comprising a PLM that generates target sequence embeddings and a conditional CLM that predicts new active compounds conditioning on desired potency. The performance of this dual-component model is rigorously assessed and benchmarked against control models.
- Finally, *Chapter 8* summarizes the main findings of this dissertation, highlighting the novel molecular design concepts introduced by CLMs. Special attention is given to the off-the-beaten-path applications in molecular design that have not been previously explored.

Chapter 2

DeepAC–Conditional Transformer-Based Chemical Language Model for the Prediction of Activity Cliffs Formed by Bioactive Compounds

The following chapter summarizes the research published as

Chen, H.; Vogt, M.; Bajorath, J. DeepAC—Conditional transformer-based chemical language model for the prediction of activity cliffs formed by bioactive compounds. *Digital Discov.* **2022**, *1*, 898–909. DOI:10.1039/D2DD00077F

Author contributions: Hengwei Chen: Methodology, Data, Code, Investigation, Analysis, Writing - review and editing. **Martin Vogt**: Methodology, Investigation, Analysis, Writing - review and editing. **Jürgen Bajorath**: Conceptualization, Methodology, Analysis, Writing - original draft, Writing – review and editing.

2.1 Summary

In drug discovery, CLMs inspired by NLP offer innovative solutions for molecular design. CLMs learn the vocabulary, syntax, and conditional probabilities of molecular representations, enabling sequence-to-sequence mappings. Their versatility in machine translation and property conditioning paves the way for exploring new molecular design concepts. ACs are formed by pairs of structurally similar or analogous active small molecules with large differences in potency. In medicinal chemistry, ACs are of high interest as they often reveal SAR determinants for compound optimization. Systematic identification of ACs across activity classes has provided the basis for computational AC predictions, with initial attempts reported a decade ago.[153, 154] Although advanced deep neural networks (DNNs) have recently been applied to predict ACs from molecular fingerprints, images, or graphs via representation learning, AC predictions present three main challenges. First, the underlying SARs are highly discontinuous; second, datasets of ACs and non-ACs are imbalanced; and third, predictions must be made at the compound pair level rather than for individual compounds, as is typical in compound classification or molecular property prediction. Therefore, alternative computational methods are required to revolutionize AC predictions. In this chapter, we investigate the application of CLMs for predictive modeling of ACs. An encoding strategy is devised to predict target compounds from source compounds and associated potency differences. Seq2Seq and conditional transformer models are pre-trained on pairs of structural analogues with varying potency differences and compared. The pre-trained transformer is then fine-tuned on ACs and non-ACs from different activity classes and evaluated against other machine learning reference models.

For CLM modeling, a systematic search of bioactive compounds with high-confidence activity data in ChEMBL identified 357,343 transformation size-restricted MMPs originating from a total of 600 activity classes. Each MMP was represented as a triple:

(*Source compound*, *Potency difference*) → (*Target compound*).

In each triple, the source and target molecules were represented as canonical SMILES strings, which were tokenized to construct a chemical vocabulary containing all

possible chemical tokens. Potency differences captured by MMPs were tokenized by binning, with each bin encoded by a single token, assigning each potency difference to the token of the corresponding bin. The CLMs, adapted from Seq2Seq and conditional transformer architectures, were pre-trained on mappings from a large general dataset of MMP-triples, learning MMP-associated potency differences caused by given chemical transformations. Given a new (Source compound, Potency difference) test instance, trained models generated a set of target candidate compounds meeting the potency difference condition. The ability of Seq2Seq and conditional transformer models to reproduce target compounds for test sets was evaluated using the reproducibility measure. As a result, the pre-trained transformer outperformed the Seq2Seq model, achieving a reproducibility of 81.8%. The pre-trained transformer was then fine-tuned for AC prediction. By definition, an MMP-Cliff consists of two MMP-forming compounds exhibiting a potency difference of at least two orders of magnitude (100-fold; i.e., $\Delta pKi \geq 2.0$), while MMP-nonCliffs are restricted to a maximal potency difference of one order of magnitude (10-fold; $\Delta pKi \leq 1$). The pre-trained transformer was fine-tuned on MMP-Cliffs and MMP-nonCliffs extracted from four large activity classes excluded from pre-training. For fine-tuning, 5%, 25%, and 50% of MMP-Cliffs and MMP-nonCliffs from each class were randomly selected. The resulting models were tested on the remaining 50% of MMP-Cliffs and MMP-nonCliffs. For performance comparison, reference classification models using different fingerprint-based ML methods, including SVM, RF, and XGBoost, were developed. The results showed that, compared to these reference methods, the conditional transformer (DeepAC) was less effective in predicting non-ACs but outperformed reference methods in predicting ACs, especially when training data was limited. This study demonstrates that CLMs learn structural relationships and associated potency differences, enabling the reproduction of ACs. Compared to earlier studies that used classification models to predict ACs, a unique feature of DeepAC is its ability to extend AC predictions by producing new AC compounds. This integrates predictive and generative modeling in the context of AC analysis and AC-based compound design. In the next chapter, compound predictions conditioned on large potency differences are generalized beyond ACs by adjusting the training protocol to predict highly potent compounds from weakly potent input templates.

Chapter 3

Designing Highly Potent Compounds using a Chemical Language Models

The following chapter summarizes the research published as

Chen, H.; Bajorath, J. Designing highly potent compounds using a chemical language model. *Sci. Rep.* **2023**, *13*, 7412. DOI:10.1038/s41598-023-34683-x

Author contributions: Hengwei Chen: Methodology, Code, Formal analysis, Investigation, Writing - original draft preparation, Writing - review and editing. **Jürgen Bajorath**: Conceptualization, Methodology, Formal analysis, Writing - original draft preparation, Writing - review and editing.

3.1 Summary

As shown in the previous chapter, the DeepAC model, a conditional transformer-based CLM, predicted ACs with an accuracy at least comparable to top-performing machine learning models while further extending such predictions through the generative design of new AC compounds. This capability, along with the observed prediction characteristics, renders DeepAC attractive for practical applications aimed at revolutionizing compound potency prediction in generative design. Compound potency prediction is a major task in drug design, for which a variety of computational methods have been developed or adapted. Mainstays include QSAR analysis to design increasingly potent analogues of active compounds and methods for ligand- or structure-based virtual screening to identify new hits. For conventional potency prediction, the assessment and comparison of different methods typically rely on standard benchmark settings. While such benchmark calculations are necessary, they are insufficient to fully evaluate the potential of potency prediction methods for practical applications. In addition to exploring the applicability domain of standard QSAR modeling, we aimed to design structurally diverse compounds beyond analogues. This prediction task could not be adequately addressed using conventional ML models, necessitating the development of a different methodological framework. Therefore, in this chapter, we adapted a conditional transformer architecture previously employed for AC predictions (*Chapter 2*), which demonstrated that compound generation could be conditioned on potency differences. Given that ACs encode large potency differences, we reasoned that this methodology could be adapted and further extended for the design of highly potent compounds. Accordingly, we devised and implemented a CLM for the prediction of highly potent compounds, using weakly potent compounds as input. To rigorously assess model performance, a compound pair-based test system was generated that covered all possible prediction outcomes, enabling a well-defined and comprehensive evaluation.

For CLM modeling, bioactive compounds with high-confidence activity data were assembled from ChEMBL and grouped into 496 target-based activity classes. A systematic search for ASs with single or multiple substitution sites (up to five) was conducted using the CCR method, yielding a total of 881,990 pairs of structural analogues (termed All_CCR pairs). From these pairs, All_CCR triples (CpdA, CpdB,

PotB-PotA) were generated by recording the potency difference for each pair. In this setup, CpdA represented the source compound, concatenated with the potency difference (PotB-PotA), while CpdB represented the target compound. Each All_CCR pair produced two triples, ensuring that each compound served as both source and target. A conditional transformer-based CLM, previously employed for AC predictions (*Chapter 2*), was adapted and pre-trained on the general set of All_CCR triples. To evaluate the model, 10 individual activity classes excluded from pre-training were reserved for fine-tuning and testing. For each class, All_CCR pairs were extracted and divided into CCR pairs with potency differences of less than 100-fold, and AC-CCR pairs capturing potency differences of at least 100-fold. The pre-trained CLM was fine-tuned on activity class-dependent AC-CCR pairs and tested on structurally distinct CCR pairs with no core structure overlap between fine-tuning and test sets. Additionally, a compound pair-based test system was created to cover all possible prediction outcomes, providing a rigorous assessment of model performance. The analysis confirmed the model's remarkable ability to reproduce known potent compounds not encountered during training, achieving unexpectedly high success rates. Predictions included both analogues of weakly potent source compounds and structurally distinct compounds. Across activity classes, median potency increases were close to or exceeded 100-fold, with multiple predictions surpassing 1000-fold potency improvements, demonstrating the model's high performance. Furthermore, the CLM generated numerous novel compounds for the activity classes, absent from both the fine-tuning and test sets. The next chapter further advances this molecular generative design approach, enabling the prediction of highly potent compounds in low-data regimes by implementing a CLM variant incorporating meta-learning.

Chapter 4

Meta-Learning for Transformer-Based Prediction of Potent Compounds

The following chapter summarizes the research published as

Chen, H.; Bajorath, J. Meta-Learning for Transformer-Based Prediction of Potent Compounds. *Sci. Rep.* **2023**, *13*, 16145. DOI:10.1038/s41598-023-43046-5

Author contributions: Hengwei Chen: Methodology, Code, Formal analysis, Investigation, Writing - original draft preparation, Writing - review and editing. **Jürgen Bajorath**: Conceptualization, Methodology, Formal analysis, Writing - original draft preparation, Writing - review and editing.

4.1 Summary

In the previous chapters, transformer-based CLMs were introduced to explore novel molecular generative design concepts, circumventing the limitations of conventional benchmark settings for potency predictions. The CLMs demonstrated the ability to predict ACs with high accuracy, while extending such predictions through the generative design of new AC compounds. More importantly, these models generated structurally diverse, highly potent compounds by conditioning on large potency differences from weakly potent input templates. However, compound activity and potency predictions inherently depend on the availability of high-quality data for model training. In early-phase drug discovery, such data are often sparse, limiting generative design. Therefore, specialized learning strategies are required to address sparsely distributed data in this context. In this chapter, a previously developed transformer architecture designed for predicting potent compounds was adapted as a base model for deriving meta-learning models. The potential of meta-learning was assessed for predicting highly potent compounds across different activity classes with varying amounts of training data.

To further advance CLMs for low-data compound design, we adapted the pre-training dataset described in *Chapter 3*, which comprises 881,990 All_CCR pairs generated from bioactive compounds with high-confidence activity data across 496 target-based activity classes from ChEMBL. These All_CCR pairs were subsequently categorized into CCR pairs (<100-fold potency difference) and AC-CCR pairs (≥100-fold difference). The Meta-CLM architecture consisted of two modules: the base CLM model, previously derived for generating highly potent target compounds from weakly potent source compounds conditioned on potency differences, and a meta-learning module. The model-agnostic meta-learning (MAML) framework was adopted to optimize the model for an activity class-specific prediction task distribution. For deriving the meta-learning module, a subset of 176 activity classes was selected, each containing at least 300 All_CCR pairs, resulting in 491,688 qualifying All_CCR triples. Each activity class was treated as a separate training task, where All_CCR triples were randomly split into a support set (80%) and a query set (20%). During meta-training, the model f_θ was first updated to a task-specific model f_θ' using its support set. The corresponding query set was then used to determine the prediction

loss of model f_θ for that task. This process was repeated across all prediction tasks (activity classes), and the model parameters were further adjusted by minimizing the sum of the prediction losses over all activity classes. The objective was to learn parameter settings that could be rapidly adapted to new prediction tasks with minimal fine-tuning. For model validation, the trained Meta-CLM was fine-tuned on 10 activity classes excluded from pre-training, and its ability to reproduce known potent candidate compounds was evaluated in the presence of varying amounts of fine-tuning data. The performance of the Meta-CLM was compared to that of the reference CLM. All models successfully reproduced known potent target compounds; however, the meta-learning approach significantly increased the number of reproduced compounds across all activity classes, particularly when fine-tuning data were limited. Moreover, the meta-learning models produced target compounds with higher overall potency and larger potency differences between templates and targets compared to reference CLMs. Additionally, the generative models designed for predicting potent compounds yielded large numbers of novel structures. In summary, a CLM variant incorporating meta-learning was successfully implemented to enable the generative design of highly potent compounds in low-data regimes. The next chapter further explores CLM applications for lead optimization in medicinal chemistry by iteratively extending analogue series with new potent compounds.

Chapter 5

Extension of Multi-site Analogue Series with Potent Compounds using a Bidirectional Transformer-Based Chemical Language Model

The following chapter summarizes the research published as

Chen, H.; Yoshimori, A.; Bajorath, J. Extension of Multi-site Analogue Series with Potent Compounds using a Bidirectional Transformer-based Chemical Language Model. *RSC Med. Chem.* **2024**, *15*, 2527– 2537. DOI:10.1039/D4MD00423J

Author contributions: Hengwei Chen: Conceptualization, Data curation, Methodology, Formal analysis, Writing – original draft, writing – reviewing and editing; **Atsushi Yoshimori**: Methodology, Writing – reviewing & editing; **Jürgen Bajorath**: Conceptualization, Methodology, Supervision, Writing – original draft, writing – reviewing and editing.

5.1 Summary

As demonstrated in the previous chapters, CLMs effectively handle diverse SMILES-to-SMILES mappings, enabling efficient training and molecular translation tasks through the use of transformer architecture. Their versatility in machine translation and property conditioning has demonstrated promising results for generative molecular design, ranging from AC prediction to the generation of highly potent compounds from weakly potent compounds, and even compound design in low-data regimes. However, generating potent compounds for evolving AS remains a key challenge in medicinal chemistry. In the practice of medicinal chemistry, a crucial question is which analogue(s) to synthesize next to further improve compound potency and other molecular properties relevant for drug development. This optimization process continues to rely heavily on chemical knowledge and experience. Previously, an RNN-based CLM (termed DeepAS) was devised to extend evolving AS with new potent analogues, leveraging the SAR transfer principle. This principle stems from findings that AS with activity against different targets often contain corresponding analogues with comparable potency progression. However, a principal limitation of this approach was that predictions were confined to AS with single substitution sites. Considering that multiple substitution sites are common in medicinal chemistry, an advanced computational framework is required to address the increased complexity of the prediction task. To this end, a transformer-based CLM variant was developed to enable direct comparison with the RNN-based DeepAS for AS extension with single substitution sites. Additionally, a novel AS encoding strategy was devised, facilitating the prediction of R-group combinations for extending AS with multiple substitution sites.

For model derivation, 104,627 MMS from 2,195 target-based activity classes were extracted from ChEMBL. Each MMS was converted into potency-ordered R-group sequences following an increasing potency gradient. These sequences were then encoded as sentences in which each R-group was represented as an individual token, with each sentence containing a minimum of two R-group tokens. The length of each sentence was set to 35 tokens, and the total number of label tokens amounted to 3,855, encompassing 3,852 unique R-groups plus three special tokens. BERT was chosen as the CLM for AS extension (termed DeepAS 2.0) due to its bidirectional characteristics,

which are well-suited for next-sentence (R-group) prediction. Both DeepAS 2.0 and the original RNN-based DeepAS were trained on 84,259 MMS covering nearly 2,200 targets and tested on 9,363 distinct MMS, enabling direct comparison for AS extension with single substitution sites. For model evaluation, the final R-group token was omitted from each test AS (not encountered during training) and predicted as the label based on derived conditional probabilities and corresponding log-likelihood scores. The model's ability to accurately predict final R-groups within the top-5 of all 3,855 R-group tokens served as the primary criterion for validation. As a result, DeepAS 2.0 further improved performance over DeepAS in systematic R-group predictions for MMS. The models' ability to extend AS in an activity class-specific manner was further investigated for MMS from 10 activity classes excluded from training. DeepAS 2.0 outperformed DeepAS in eight of 10 classes, and fine-tuning consistently increased the performance of DeepAS 2.0 across all activity classes. To explore the ability of DeepAS 2.0 for multi-site AS extension, a total of 16,538 AS with one to five substitution sites from 864 target-based activity classes was obtained. A new AS encoding scheme and R-group data structure were devised to represent multi-site AS as sequences, where R-groups of each analogue were concatenated into a combined R-group (combination) token. The vocabulary of possible labels consisted of 36,647 concatenated R-group (combination) tokens extracted from multi-site AS, supplemented by special tokens. Data augmentation involved transforming each AS into multiple sentences, expanding each training instance into sentences capturing an increasing number of R-group tokens. For deriving DeepAS 2.0 to extend AS with multiple substitution sites, termed multi-site DeepAS 2.0 (MS-DeepAS 2.0), a dataset comprising 10,863 AS with one to five substitution sites from 854 activity classes was used for training, while 2,716 AS were reserved for testing. Additionally, five activity classes with multi-site AS were excluded from training for fine-tuning. Despite the inherent challenges of predicting R-group combinations of potent compounds, MS-DeepAS 2.0 successfully predicted potent analogues with varying R-group combinations for multi-site AS with activity against many different targets. In summary, DeepAS 2.0 demonstrated enhanced performance for single-site AS extension, while MS-DeepAS 2.0 successfully extended multi-site AS with new encoding scheme. In the next chapter, the AS design strategy is extended to incorporate core structure and substituent modifications at multiple sites, further advancing AS extension for lead optimization.

Chapter 6

Combining a Chemical Language Model and the Structure−Activity Relationship Matrix Formalism for Generative Design of Potent Compounds with Core Structure and Substituent Modifications

The following chapter summarizes the research published as

Chen, H.; Bajorath, J. Combining a Chemical Language Model and the Structure−Activity Relationship Matrix Formalism for Generative Design of Potent Compounds with Core Structure and Substituent Modifications. *J. Chem. Inf. Model.* **2024**, *64*, 8784-8795. DOI: 10.1021/acs.jcim.4c01781

Author contributions: Hengwei Chen: Conceptualization, Data curation, Methodology, Formal analysis, Writing – original draft, writing – reviewing and editing; **Jürgen Bajorath**: Conceptualization, Methodology, Supervision, Writing – original draft, writing – reviewing and editing.

6.1 Summary

In Chapter 5, the BERT-based CLM, termed DeepAS 2.0, was developed to improve the systematic extension of AS with single substitution sites, surpassing the performance of the RNN-based DeepAS. The bidirectional characteristics of BERT facilitated next-sentence (R-group) prediction, enhancing activity class-specific MMS extension. More importantly, DeepAS 2.0 was further advanced with a new AS encoding scheme and R-group combination structure, enabling the extension of AS with multiple substitution sites. The resulting MS-DeepAS 2.0 accurately prioritized R-group combinations of potent analogues across various multi-site AS. Building on the success of extending single- and multi-site AS using CLMs, this chapter aims to expand the methodology by incorporating core structure modifications alongside substituent changes, thus moving beyond AS with invariant cores (scaffolds). Modifying core structures is crucial for compound optimization, such as introducing new substitution sites and/or heteroatoms at specific positions, but it remains computationally challenging. Traditional approaches often rely on scaffold hopping, underscoring the need for a new methodology to extend AS through combined core and substituent modifications. To address this, the SARM formalism and data structure were introduced to extract AS with structurally related cores. A new structural decomposition approach was devised to capture AS with multiple substitution sites and structurally related cores. DeepAS 3.0 was developed with a novel encoding scheme to represent core structure-substituent combinations, facilitating the extension of AS through combined core and multiple substituent site modifications. Various model variants were derived and compared to assess performance.

A total of 19,556 SARMs from 2,895 target-based activity classes were generated from ChEMBL using the newly designed SARM-CCR approach, which systematically organized subsets of multi-site AS with related cores. For each SARM, multi-site AS with structurally related cores were combined, and analogues were arranged in increasing potency order, yielding a consensus series. This consensus series served as model input, representing potency-ordered analogue sequences. Each analogue in the consensus series was encoded as a compound token consisting of its core-substituent combination, utilizing substructure-based tokenization. Following the encoding protocol established in *Chapter 5*, R-groups at different substitution sites were

represented as unique R-group combinations, ensuring consistent application across AS with varying substitution sites. Each SARM-based consensus series was encoded as a sentence, with each analogue represented by an individual compound token. Sentence length was standardized to 35 tokens, and the total number of label tokens amounted to 95,910, covering all possible analogues extracted from qualifying SARMs. DeepAS 3.0 adapted the BERT CLM in *chapter 5*, initially trained on a global dataset of 17,140 SARMs, comprising 132,310 analogue sequences from 2,885 activity classes. For model evaluation, the final analogue token was removed from each test sequence (unseen during training) and predicted based on the model's conditional probabilities and log-likelihood scores. The primary validation criterion was the model's ability to correctly predict the final analogue within the top-ranked tokens. As a control, 20 label tokens were randomly selected for each test series to assess the probability of the correct final analogue's presence. Performance analysis was conducted across four subsets containing varying numbers of substitution sites. Additionally, individual models were developed for each subset to explore their relative predictive abilities. Fine-tuning of DeepAS 3.0 was performed using analogue sequences from 10 activity classes excluded from pre-training. The pre-trained global model was also fine-tuned and tested using these subsets as a control. The predictive performance of the global general model and the subset-based models was comparable, demonstrating the bidirectional transformer's ability to learn the chemical space of compound series with extensive structural variations. Fine-tuning on AS from activity classes excluded from pre-training yielded promising predictions, with a confined but consistent performance increase for the subset-based models over the global fine-tuned model. Test calculations with both general and fine-tuned models generated a wealth of candidate compounds, confirming the models' ability to introduce diverse core structure and substituent modifications, thereby chemically diversifying input series. In summary, DeepAS 3.0 further advances the AS extension by incorporating core structure modifications in AS with multiple substitution sites. In the next chapter, a new CLM methodology is introduced to revisit sequence-based drug design, aiming to predict active compounds from protein sequence embeddings with potency conditioning.

Chapter 7

Generative Design of Compounds with Desired Potency from Target Protein Sequences using a Multimodal Biochemical Language Model

The following chapter summarizes the research published as

Chen, H.; Bajorath, J. Generative design of compounds with desidered potency from target protein sequences using a multimodal biochemical language model. *J. Cheminf.* **2024**, *16*, 55. DOI:10.1186/s13321-024-00852-x

Author contributions: Hengwei Chen: Conceptualization, Data curation, Methodology, Formal analysis, Writing – original draft, writing – reviewing and editing; **Jürgen Bajorath**: Conceptualization, Methodology, Supervision, Writing – original draft, writing – reviewing and editing.

7.1 Summary

In previous chapters, various transformer-based CLMs were introduced for molecular SMILES-to-SMILES translation tasks and specific applications in molecular design. These include predicting ACs (*Chapter 2*), generating highly potent compounds (*Chapters 3* and *4*), and extending AS with increasingly potent compounds (*Chapters 5* and *6*). While these approaches focused on learning compound-to-compound mappings for predicting new active compounds, efforts have also been made to establish direct links between biological targets and chemical entities, such as through protein sequence-to-compound modeling, thus revitalizing the concept of sequence-based compound design. However, designing active compounds directly from protein sequences remains challenging, as typically only a small subset of residues is involved in ligand binding, and high overall sequence similarity is often necessary to infer comparable binding characteristics between targets. Consequently, such design efforts are controversial and difficult to pursue with standard ML methods. However, the advent of PLMs offers a fresh perspective on this challenging task. PLMs embed long protein sequences as sentences of characters, where one or more residues form words. The resulting sequence embeddings are shown to implicitly capture structural and functional characteristics of proteins, making them attractive for various applications. In this chapter, a dual-component LM was designed to learn from multimodal data. It combined a PLM component for generating target sequence embeddings with a conditional CLM for predicting new active compounds with desired potency. The biochemical LM was trained to map combined protein sequence and compound potency value embeddings to corresponding compounds. It was then fine-tuned on individual activity classes not encountered during model derivation and evaluated on compound test sets that were structurally distinct from training sets.

Compounds with high-confidence activity data were selected from ChEMBL, represented as canonical SMILES strings, and divided into target-based activity classes. Only targets with a maximal sequence length of 4000 residues were considered. For each qualifying target, the protein sequence was extracted in FASTA format from UniProt. This resulted in 1575 activity classes, comprising a total of 87,839 unique compounds. A new multimodal conditional compound generator combining two LM components was devised, termed the biochemical LM. It consisted

of a pre-trained PLM, ProtT5XLUniref50 (adapted from ProtTrans), generating fixed-size target sequence embeddings of 1024 dimensions, and a conditional CLM predicting new active compounds with desired potency. The biochemical LM was trained to map target sequence embeddings conditioned on potency values to active compounds. Pre-training was conducted on a general set of 212,004 target-compound pairs from 1565 activity classes. As a control, an unconditional model with the same architecture but without potency conditioning was also derived. Subsequently, the model was fine-tuned on 10 different activity classes not included in model derivation. Fine-tuning and evaluation were carried out on structurally distinct compound subsets generated through comprehensive AS identification and AS-based compound splitting. The most rigorous proof-of-concept criterion for the approach was the model's ability to exactly reproduce known active compounds not encountered during training. Consequently, the biochemical LM consistently reproduced varying numbers of known active compounds across all test activity classes. Compared to the control model, the conditional model consistently reproduced larger numbers of known compounds as well as more potent compounds, revealing a clear positive effect of potency value conditioning on prediction success. Furthermore, for most activity classes, the potency distribution of correctly reproduced compounds closely matched the potency distribution of all test compounds, consistent with reproducing compounds at different potency levels. Subsequent molecular similarity analysis demonstrated the capacity of biochemical LM to generate structurally diverse candidate compounds distinct from both fine-tuning and test compounds, indicating its generalization potential. Taken together, generative compound design conditioned on potency value from target sequence embeddings using the dual-component biochemical LM yielded promising results, highlighting its potential for sequence-based compound design.

Chapter 8

Conclusion

One of the major applications of DL in computational drug discovery is generative molecular design, which focuses on creating highly diverse or focused chemical libraries and design new compounds with desired properties. These models typically operate on molecular graphs or textual formats such as SMILES strings. By learning the underlying probabilistic distribution of molecular representations in the training data, generative models can produce new molecular structures. Among the generative approaches, CLMs have gained increasing popularity, particularly for molecular sequence-to-sequence translation tasks across various applications in drug discovery. CLMs are especially attractive due to their versatility in relating different types of sequence representations, enabling exploration of previously challenging applications. While various DGMs have been widely applied to sequential data processing, transformer-based CLMs are beginning to dominate the field. This shift is largely due to the self-attention mechanism of transformers, which reduces errors in generated representations and improves computational efficiency through parallel processing. Moreover, transformers offer flexibility in conditioning generation on molecular properties or other constraints, further enhancing their applicability in molecular design. In this context, different chemical and biochemical LMs are investigated and derived for specific applications in medicinal chemistry and molecular design. In the first study (*Chapter 2*), we investigated CLMs for AC prediction. To this end, an encoding strategy was devised to predict target compounds from source compounds and associated potency differences. Seq2Seq and transformer models were pre-trained on pairs of structural analogues with varying potency differences, representing true SARs. Comparative analysis revealed the superior performance of the transformer architecture in reproducing test compound pairs. Building on this, the pre-trained transformer was fine-tuned using both ACs and non-ACs across different activity classes. Compared to reference methods, DeepAC achieved the highest prediction accuracy and exhibited unique predictive behavior. While its performance on non-ACs was limited, DeepAC outperformed baseline methods in identifying ACs, particularly

in data-scarce settings. A key strength of DeepAC is its dual capability: in addition to predicting ACs, it can generate novel AC compounds. This integration of predictive and generative modeling enables both AC analysis and AC-based compound design. Furthermore, the approach can be generalized beyond AC prediction by conditioning on large potency differences, enabling the model to generate highly potent compounds from weakly potent input templates. In *Chapter 3*, we adapted the CLM previously used for AC predictions (*Chapter 2*) to predict highly potent compounds from weakly potent ones. To achieve this, the CLM was pre-trained on a general set of CCR triples, which comprised ASs with single or multiple substitution sites. The pre-trained model was then fine-tuned on pairs of source and target compounds with associated potency differences from 10 activity classes excluded from pre-training, enabling an evaluation of its ability to predict structurally diverse compounds with substantial potency increases relative to input molecules. To rigorously assess model performance, a compound pair-based test system was generated, covering all possible prediction outcomes. Our analysis confirmed that the model reproduced known potent compounds not encountered during training at high rates, including both analogues of source compounds and structurally distinct compounds. Median potency gains across activity classes approached or exceeded 100-fold, with several predictions demonstrating over 1000-fold increases, indicating high model performance. Additionally, the CLM generated a large number of novel compounds that were not part of the fine-tuning or test sets. However, the accuracy of compound potency predictions inherently depends on the availability of high-quality training data. As is often the case in early-phase drug discovery, potency measurements for specific targets are generally sparse, posing a challenge for generative design. Addressing this issue requires specialized learning strategies capable of handling sparsely distributed data, which is explored in the subsequent chapter. In the third study (*Chapter 4*), the CLM was further advanced to enable the generative design of highly potent compounds in low-data regimes. Building upon the transformer architecture investigated in *Chapter 3*, a specialized meta-learning module was incorporated into the pre-trained transformer, resulting in a meta-learning model. Meta-CLMs were derived for different activity classes, and their performance in designing potent compounds was compared to reference CLMs. For model validation, the primary criterion was the ability to reproduce known potent target compounds. All models successfully reproduced known target candidates; however, the Meta-CLMs significantly increased the number

of correctly predicted target compounds across all activity classes, particularly as the number of fine-tuning samples decreased. This outcome aligned with expectations for successful meta-learning and highlighted its advantages in low-data scenarios. Furthermore, the meta-learning models generated target compounds with overall higher potency than basic CLMs and achieved larger potency differences between template and target compounds. Beyond these results, generating potent compounds for evolving AS remains a key challenge in medicinal chemistry and compound optimization. Therefore, in the next study (*Chapter 5*), we investigated the extension of AS with potent compounds using CLMs. Building on the original RNN-based DeepAS model, which predicted R-groups for potent analogues with single substitution sites, we developed DeepAS 2.0—a BERT-based CLM that improved performance in systematic MMS extension. Additional fine-tuning confirmed the activity class sensitivity of DeepAS 2.0. More importantly, a framework for handling AS with multiple substitution sites was explored by introducing a new AS encoding scheme and R-group data structure in DeepAS 2.0. Despite the inherent challenges of predicting R-group combinations for potent compounds, the resulting MS-DeepAS 2.0 accurately prioritized R-group combinations of potent analogues across diverse multi-site AS with activity against various targets. Building upon the successful extension of single- and multi-site AS with potent compounds using CLMs, we aimed to further extend the method to enable the combined modification of core structures and substituents, thereby moving beyond AS with invariant scaffolds. In *Chapter 6*, the SARM formalism was introduced and further advanced with a new compound decomposition protocol SARM-CCR designed to cover structurally related AS with multiple substitution sites. Additionally, a new CLM coding and tokenization scheme was developed to represent core structure–substituent combinations effectively. Building upon the BERT architecture, a global general CLM was derived alongside four other models tailored to subsets of AS with varying numbers of substitution sites. Both the global and subset-based models accurately predicted terminal analogues of test series at high ranks. Fine-tuning the models on AS from unseen activity classes yielded promising predictions, with the subset-based models consistently showing a slight performance advantage over the global fine-tuned model. Test calculations using both general and fine-tuned models generated a wealth of candidate compounds, confirming the models' ability to introduce diverse core structure and substituent modifications, thereby further chemically diversifying input series. Different

transformer-based CLMs have been introduced for various molecular string-to-string translation tasks. Beyond learning compound-to-compound mappings for predicting new active or highly potent compounds, protein sequence-to-compound modeling presents an opportunity to establish direct links between biological targets and chemical entities, revitalizing the concept of sequence-based compound design. In *Chapter 7*, we advanced this concept by developing a dual-component biochemical LM for multimodal learning, aiming to predict new active compounds with desired potency from protein sequence embeddings. The model integrated two components: a pre-trained PLM to generate target sequence embeddings, and a conditional transformer that operated on operating on the output of the PLM. The biochemical LM was initially pre-trained to map target sequence embeddings, conditioned on potency values, to active compounds, then individually fine-tuned on 10 different activity classes not included in model derivation. Fine-tuning and evaluation were carried out on structurally distinct compound subsets generated through AS-based compound splitting. The model consistently reproduced known active compounds not encountered during training across all activity classes. Notably, it outperformed an unconditional version of the model by reproducing a greater number of known compounds, highlighting the beneficial effect of conditioning on potency values. In most cases, the potency distribution of correctly reproduced compounds closely matched the potency distribution of all test compounds, reflecting the model's ability to capture compounds across different potency levels. Subsequent molecular similarity analysis showed that the biochemical LM could also generate structurally diverse candidate compounds, distinct from those in the fine-tuning and test sets. Collectively, these findings demonstrate that the biochemical LM enables generative design of active compounds with desired potency from target sequences, offering a novel and effective strategy for sequence-based compound design.

In conclusion, this dissertation explores the development and application of chemical and biochemical LMs to tackle various challenges in medicinal chemistry and drug design. Inspired by NLP, transformers were adapted to learn the chemical vocabulary, syntax, and conditional probabilities of molecular SMILES representations, enabling a range of molecular translation tasks. By learning the mapping pairs of structural analogues with varying potency differences, DeepAC accurately predicted ACs and further extended such predictions through generative

design of new AC compounds. Additionally, by modifying the training protocol to condition on large potency differences, compound predictions were generalized beyond ACs enabling the generation of highly potent compounds from weakly potent input templates. To address the challenges of low-data regimes, a CLM variant incorporating meta-learning was implemented for the generative design of highly potent compounds from limited training data. Furthermore, a BERT-based CLM was developed for lead optimization in medicinal chemistry. This model iteratively extended AS with potent compounds by introducing substituent replacements at multiple sites (DeepAS 2.0) and core structure modifications (DeepAS 3.0). Finally, compound-to-compound mapping was advanced to protein sequence-to-compound mapping through a dual-component biochemical LM, facilitating the generative design of active compounds with desired potency directly from target protein sequences. These studies demonstrate the potential of CLMs to address previously challenging or unfeasible prediction scenarios in molecular design, offering new opportunities for advancements in medicinal chemistry and drug discovery.

Bibliography

[1] Hughes, J. P.; Rees, S.; Kalindjian, S. B.; Philpott, K. L. Principles of Early Drug Discovery. *Br J Pharmacol* **2011**, *162*, 1239–1249.

[2] Tabana, Y.; Babu, D.; Fahlman, R.; Siraki, A. G.; Barakat, K. Target Identification of Small Molecules: An Overview of the Current Applications in Drug Discovery. *BMC Biotechnol* **2023**, *23*, 44.

[3] Bleicher, K. H.; Böhm, H.-J.; Müller, K.; Alanine, A. I. Hit and Lead Generation: Beyond High-Throughput Screening. *Nat Rev Drug Discov* **2003**, *2*, 369–378.

[4] Scannell, J. W.; Blanckley, A.; Boldon, H.; Warrington, B. Diagnosing the Decline in Pharmaceutical R&D Efficiency. *Nat Rev Drug Discov* **2012**, *11*, 191–200.

[5] DiMasi, J. A.; Grabowski, H. G.; Hansen, R. W. Innovation in the Pharmaceutical Industry: New Estimates of R&D Costs. *J Health Econ* **2016**, *47*, 20–33.

[6] Brown, D. G.; Wobst, H. J.; Kapoor, A.; Kenna, L. A.; Southall, N. Clinical Development Times for Innovative Drugs. *Nat Rev Drug Discov* **2022**, *21*, 793–794.

[7] Kinch, M. S.; Haynesworth, A.; Kinch, S. L.; Hoyer, D. An Overview of FDA-Approved New Molecular Entities: 1827-2013. *Drug Discov Today* **2014**, *19*, 1033–1039.

[8] de la Torre, B. G.; Albericio, F. The Pharmaceutical Industry in 2024: An Analysis of the FDA Drug Approvals from the Perspective of Molecules. *Molecules* **2025**, *30*, 482.

[9] Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem 2025 Update. *Nucleic Acids Res* **2025**, *53*, D1516–D1525.

[10] Hargrave-Thomas, E.; Yu, B.; Reynisson, J. Serendipity in Anticancer Drug Discovery. *World J Clin Oncol* **2012**, *3*, 1–6.

[11] Polishchuk, P. G.; Madzhidov, T. I.; Varnek, A. Estimation of the Size of Drug-like Chemical Space Based on GDB-17 Data. *J Comput Aided Mol Des* **2013**, *27*, 675–679.

[12] Pagadala, N. S.; Syed, K.; Tuszynski, J. Software for Molecular Docking: A Review. *Biophys Rev* **2017**, *9*, 91–102.

[13] Liu, J.; Wang, R. Classification of Current Scoring Functions. *J Chem Inf Model* **2015**, *55*, 475–482.

[14] Guedes, I. A.; Pereira, F. S. S.; Dardenne, L. E. Empirical Scoring Functions for Structure-Based Virtual Screening: Applications, Critical Aspects, and Challenges. *Front Pharmacol* **2018**, *9*, 1089.

[15] Filipe, H. A. L.; Loura, L. M. S. Molecular Dynamics Simulations: Advances and Applications. *Molecules* **2022**, *27*, 2105.

[16] Mobley, D. L.; Gilson, M. K. Predicting Binding Free Energies: Frontiers and Benchmarks. *Annu Rev Biophys* **2017**, *46*, 531–558.

[17] Williams-Noonan, B. J.; Yuriev, E.; Chalmers, D. K. Free Energy Methods in Drug Design: Prospects of "Alchemical Perturbation" in Medicinal Chemistry. *J Med Chem* **2018**, *61*, 638–649.

[18] Vamathevan, J.; Clark, D.; Czodrowski, P.; Dunham, I.; Ferran, E.; Lee, G.; Li, B.; Madabhushi, A.; Shah, P.; Spitzer, M.; Zhao, S. Applications of Machine Learning in Drug Discovery and Development. *Nat Rev Drug Discov* **2019**, *18*, 463–477.

[19] Cortes, C.; Vapnik, V. Support-Vector Networks. *Machine Learning* **1995**, *20*, 273–297.

[20] Breiman, L. Random Forests. *Machine Learning* **2001**, *45*, 5–32.

[21] Rodríguez-Pérez, R.; Bajorath, J. Evolution of Support Vector Machine and Regression Modeling in Chemoinformatics and Drug Discovery. *J Comput Aided Mol Des* **2022**, *36*, 355–362.

[22] Walters, W. P.; Barzilay, R. Applications of Deep Learning in Molecule Generation and Molecular Property Prediction. *Acc Chem Res* **2021**, *54*, 263–270.

[23] Huo, X.; Xu, J.; Xu, M.; Chen, H. An Improved 3D Quantitative Structure-Activity Relationships (QSAR) of Molecules with CNN-Based Partial Least Squares Model. *Artif Intell Life Sci* **2023**, *3*, 100065.

[24] Li, Y.; Xu, Y.; Yu, Y. CRNNTL: Convolutional Recurrent Neural Network and Transfer Learning for QSAR Modeling in Organic Drug and Material Discovery. *Molecules* **2021**, *26*, 7257.

[25] Wang, F.; Lei, X.; Liao, B.; Wu, F.-X. Predicting Drug-Drug Interactions by Graph Convolutional Network with Multi-Kernel. *Brief Bioinform* **2022**, *23*, bbab511.

[26] Tang, M.; Li, B.; Chen, H. Application of Message Passing Neural Networks for Molecular Property Prediction. *Curr Opin Struct Biol* **2023**, *81*, 102616.

[27] Pasupa, K.; Sunhem, W. A Comparison between Shallow and Deep Architecture Classifiers on Small Dataset. 8th International Conference on Information Technology and Electrical Engineering. 2016; pp 1–6.

[28] Janela, T.; Bajorath, J. Simple Nearest-Neighbour Analysis Meets the Accuracy of Compound Potency Predictions Using Complex Machine Learning Models. *Nat Mach Intell* **2022**, *4*, 1246–1255.

[29] Durant, J. L.; Leland, B. A.; Henry, D. R.; Nourse, J. G. Reoptimization of MDL Keys for Use in Drug Discovery. *J Chem Inf Comput Sci* **2002**, *42*, 1273–1280.

[30] Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J Chem Inf Model* **2010**, *50*, 742–754.

[31] Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *J Chem Doc* **1965**, *5*, 107–113.

[32] Lim, J.; Ryu, S.; Park, K.; Choe, Y. J.; Ham, J.; Kim, W. Y. Predicting Drug-Target Interaction Using a Novel Graph Neural Network with 3D Structure-Embedded Graph Representation. *J Chem Inf Model* **2019**, *59*, 3981–3988.

[33] Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D. R. Protein-Ligand Scoring with Convolutional Neural Networks. *J Chem Inf Model* **2017**, *57*, 942–957.

[34] Wójcikowski, M.; Kukiełka, M.; Stepniewska-Dziubinska, M. M.; Siedlecki, P. Development of a Protein-Ligand Extended Connectivity (PLEC) Fingerprint and Its Application for Binding Affinity Predictions. *Bioinformatics* **2019**, *35*, 1334–1341.

[35] Sánchez-Cruz, N.; Medina-Franco, J. L.; Mestres, J.; Barril, X. Extended Connectivity Interaction Features: Improving Binding Affinity Prediction through Chemical Description. *Bioinformatics* **2021**, *37*, 1376–1382.

[36] Bian, Y.; Wang, J.; Jun, J. J.; Xie, X.-Q. Deep Convolutional Generative Adversarial Network (dcGAN) Models for Screening and Design of Small Molecules Targeting Cannabinoid Receptors. *Mol Pharm* **2019**, *16*, 4451–4460.

[37] Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. Proceedings of the 34th International Conference on Machine Learning. 2017; pp 1263–1272.

[38] Raghunathan, S.; Priyakumar, U. D. Molecular Representations for Machine Learning Applications in Chemistry. *International Journal of Quantum Chemistry* **2022**, *122*, e26870.

[39] David, L.; Thakkar, A.; Mercado, R.; Engkvist, O. Molecular Representations in AI-Driven Drug Discovery: A Review and Practical Guide. *J Cheminform* **2020**, *12*, 56.

[40] Heller, S. R.; McNaught, A.; Pletnev, I.; Stein, S.; Tchekhovskoi, D. InChI, the IUPAC International Chemical Identifier. *J Cheminform* **2015**, *7*, 23.

[41] Handsel, J.; Matthews, B.; Knight, N. J.; Coles, S. J. Translating the InChI: Adapting Neural Machine Translation to Predict IUPAC Names from a Chemical Identifier. *J Cheminform* **2021**, *13*, 79.

[42] Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J Chem Inf Comput Sci* **1988**, *28*, 31–36.

[43] Bjerrum, E. J. SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. 2017, arXiv:1703.07076. arXiv.org e-Print archive. https://arxiv.org/abs/1703.07076

[44] Li, X.; Fourches, D. Inductive Transfer Learning for Molecular Activity Prediction: Next-Gen QSAR Models with MolPMoFiT. *J Cheminform* **2020**, *12*, 27.

[45] Tetko, I. V.; Karpov, P.; Bruno, E.; Kimber, T. B.; Godin, G. Augmentation Is What You Need! Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions: 28th International Conference on Artificial Neural Networks. 2019; pp 831–835.

[46] Arús-Pous, J.; Johansson, S. V.; Prykhodko, O.; Bjerrum, E. J.; Tyrchan, C.; Reymond, J.-L.; Chen, H.; Engkvist, O. Randomized SMILES Strings Improve the Quality of Molecular Generative Models. *J Cheminform* **2019**, *11*, 71.

[47] van Deursen, R.; Ertl, P.; Tetko, I. V.; Godin, G. GEN: Highly Efficient SMILES Explorer Using Autodidactic Generative Examination Networks. *J Cheminform* **2020**, *12*, 22.

[48] O'Boyle, N.; Dalke, A. DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures. 2018, ChemRxiv. https://doi.org/10.26434/chemrxiv.7097960.v1.

[49] Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A. Self-Referencing Embedded Strings (SELFIES): A 100% Robust Molecular String Representation. *Mach. Learn.: Sci. Technol.* **2020**, *1*, 045024.

[50] Segler, M. H. S.; Kogej, T.; Tyrchan, C.; Waller, M. P. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Cent Sci* **2018**, *4*, 120–131.

[51] Goel, M.; Raghunathan, S.; Laghuvarapu, S.; Priyakumar, U. D. MoleGuLAR: Molecule Generation Using Reinforcement Learning with Alternating Rewards. *J Chem Inf Model* **2021**, *61*, 5815–5826.

[52] Pereira, T.; Abbasi, M.; Ribeiro, B.; Arrais, J. P. Diversity Oriented Deep Reinforcement Learning for Targeted Molecule Generation. *J Cheminform* **2021**, *13*, 21.

[53] Queiroz, L. P.; Rebello, C. M.; Costa, E. A.; Santana, V. V.; Rodrigues, B. C. L.; Rodrigues, A. E.; Ribeiro, A. M.; Nogueira, I. B. R. Transfer Learning Approach to Develop Natural Molecules with Specific Flavor Requirements. *Ind Eng Chem Res* **2023**, *62*, 9062–9076.

[54] Arús-Pous, J.; Blaschke, T.; Ulander, S.; Reymond, J.-L.; Chen, H.; Engkvist, O. Exploring the GDB-13 Chemical Space Using Deep Generative Models. *J Cheminform* **2019**, *11*, 20.

[55] Li, X.; Xu, Y.; Yao, H.; Lin, K. Chemical Space Exploration Based on Recurrent Neural Networks: Applications in Discovering Kinase Inhibitors. *J Cheminform* **2020**, *12*, 42.

[56] Kingma, D. P.; Welling, M. Auto-Encoding Variational Bayes. 2013, arXiv:1312.6114. arXiv.org e-Print archive. https://arxiv.org/abs/1312.6114

[57] Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.;

Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent Sci* **2018**, *4*, 268–276.

[58] Kusner, M. J.; Paige, B.; Hernández-Lobato, J. M. Grammar Variational Autoencoder. ICML'17: Proceedings of the 34th International Conference on Machine Learning. 2017; pp 1945–1954.

[59] Dai, H.; Tian, Y.; Dai, B.; Skiena, S.; Song, L. Syntax-Directed Variational Autoencoder for Structured Data. 2018, arXiv:1802.08786. arXiv.org e-Print archive. https://arxiv.org/abs/1802.08786.

[60] Lim, J.; Ryu, S.; Kim, J. W.; Kim, W. Y. Molecular Generative Model Based on Conditional Variational Autoencoder for de Novo Molecular Design. *J Cheminform* **2018**, *10*, 31.

[61] Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; Frey, B. Adversarial Autoencoders. 2015, arXiv:1511.05644. arXiv.org e-Print archive. https://arxiv.org/abs/1511.05644.

[62] Kadurin, A.; Nikolenko, S.; Khrabrov, K.; Aliper, A.; Zhavoronkov, A. druGAN: An Advanced Generative Adversarial Autoencoder Model for de Novo Generation of New Molecules with Desired Molecular Properties in Silico. *Mol Pharm* **2017**, *14*, 3098–3104.

[63] Kadurin, A.; Aliper, A.; Kazennov, A.; Mamoshina, P.; Vanhaelen, Q.; Khrabrov, K.; Zhavoronkov, A. The Cornucopia of Meaningful Leads: Applying Deep Adversarial Autoencoders for New Molecule Development in Oncology. *Oncotarget* **2017**, *8*, 10883–10890.

[64] Polykovskiy, D.; Zhebrak, A.; Vetrov, D.; Ivanenkov, Y.; Aladinskiy, V.; Mamoshina, P.; Bozdaganyan, M.; Aliper, A.; Zhavoronkov, A.; Kadurin, A. Entangled Conditional Adversarial Autoencoder for de Novo Drug Discovery. *Mol Pharm* **2018**, *15*, 4398–4405.

[65] Blaschke, T.; Olivecrona, M.; Engkvist, O.; Bajorath, J.; Chen, H. Application of Generative Autoencoder in De Novo Molecular Design. *Mol Inform* **2018**, *37*, 1700123.

[66] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Commun. ACM* **2020**, *63*, 139–144.

[67] Guimaraes, G. L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P. L. C.; Aspuru-Guzik, A. Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models. 2017, arXiv:1705.10843. arXiv.org e-Print archive. https://arxiv.org/abs/1705.10843.

[68] Sanchez-Lengeling, B.; Outeiral, C.; Guimaraes, G. L.; Aspuru-Guzik, A. Optimizing Distributions over Molecular Space. An Objective-Reinforced Generative Adversarial

Network for Inverse-Design Chemistry (ORGANIC). 2017, ChemRxiv. . https://doi.org/10.26434/chemrxiv.5309668.v3.

[69] Lavecchia, A. Machine-Learning Approaches in Drug Discovery: Methods and Applications. *Drug Discov Today* **2015**, *20*, 318–331.

[70] Xiong, J.; Xiong, Z.; Chen, K.; Jiang, H.; Zheng, M. Graph Neural Networks for Automated de Novo Drug Design. *Drug Discov Today* **2021**, *26*, 1382–1393.

[71] Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How Powerful Are Graph Neural Networks? 2018, arXiv:1810.00826. arXiv.org e-Print archive. https://arxiv.org/abs/1810.00826.

[72] Hamilton, W. L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs. NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017; pp 1025–1035.

[73] Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. 2017, arXiv:1710.10903. arXiv.org e-Print archive. https://arxiv.org/abs/1710.10903.

[74] Kipf, T. N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. 2016, arXiv:1609.02907. arXiv.org e-Print archive. https://arxiv.org/abs/1710.10903.

[75] Kobyzev, I.; Prince, S. J.D.; Brubaker, M. A. Normalizing Flows: An Introduction and Review of Current Methods. 2019, arXiv:1908.09257. arXiv.org e-Print archive. https://arxiv.org/abs/1908.09257.

[76] Rezende, D. J.; Mohamed, S. Variational Inference with Normalizing Flows. 2015, arXiv:1505.05770. arXiv.org e-Print archive. https://arxiv.org/abs/1505.05770.

[77] Zhang, C.; Wang, F. MoFlow: An Invertible Flow Model for Generating Molecular Graphs KDD '20: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020; pp 617–626.

[78] Shi, C.; Xu, M.; Zhu, Z.; Zhang, W.; Zhang, M.; Tang, J. GraphAF: A Flow-Based Autoregressive Model for Molecular Graph Generation. 2020, arXiv:2001.09382. arXiv.org e-Print archive. https://arxiv.org/abs/2001.09382.

[79] Luo, Y.; Yan, K.; Ji, S. GraphDF: A Discrete Flow Model for Molecular Graph Generation. Proceedings of the 38th International Conference on Machine Learning. 2021; pp 7192–7203.

[80] Ma, C.; Zhang, X. GF-VAE: A Flow-based Variational Autoencoder for Molecule Generation. CIKM '21: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021; pp 1181–1190.

[81] Hoogeboom, E.; Satorras, V. G.; Vignac, C.; Welling, M. Equivariant Diffusion for Molecule Generation in 3D. Proceedings of the 39th International Conference on Machine Learning. 2022; pp 8867–8887.

[82] Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020; pp 6840–6851.

[83] Hopfield, J. J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proc Natl Acad Sci USA* **1982**, *79*, 2554–2558.

[84] Cho, K.; van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. 2014; pp 103–111.

[85] Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **1997**, *9*, 1735–1780.

[86] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł. ukasz; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst* **2017**, *30*, 6000–6010.

[87] Chen, Y.; Wang, Z.; Zeng, X.; Li, Y.; Li, P.; Ye, X.; Sakurai, T. Molecular Language Models: RNNs or Transformer? *Brief Funct Genomics* **2023**, *22*, 392–400.

[88] Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2019; pp 4171–4186.

[89] Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[90] Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P. J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *The Journal of Machine Learning Research* **2020**, *21*, 1–67.

[91] Pang, C.; Qiao, J.; Zeng, X.; Zou, Q.; Wei, L. Deep Generative Models in De Novo Drug Molecule Generation. *J Chem Inf Model* **2024**, *64*, 2174–2194.

[92] Lavecchia, A. Advancing Drug Discovery with Deep Attention Neural Networks. *Drug Discov Today* **2024**, *29*, 104067.

[93] *Scientific Large Language Models: A Survey on Biological & Chemical Domains | ACM Computing Surveys*. https://dl.acm.org/doi/10.1145/3715318 accessed 2025-03-28.

[94] Yoshimori, A.; Chen, H.; Bajorath, J. Chemical Language Models for Applications in Medicinal Chemistry. *Future Med Chem* **2023**, *15*, 119–121.

[95] Bajorath, J. Chemical Language Models for Molecular Design. *Mol Inform* **2024**, *43*, e202300288.

[96] Grisoni, F. Chemical Language Models for de Novo Drug Design: Challenges and Opportunities. *Curr Opin Struct Biol* **2023**, *79*, 102527.

[97] Flores-Hernandez, H.; Martinez-Ledesma, E. A Systematic Review of Deep Learning Chemical Language Models in Recent Era. *J Cheminform* **2024**, *16*, 129.

[98] Li, J.; Jiang, X. Mol-BERT: An Effective Molecular Representation with BERT for Molecular Property Prediction. *Wireless Communications and Mobile Computing* **2021**, *2021*, 7181815.

[99] Wang, S.; Guo, Y.; Wang, Y.; Sun, H.; Huang, J. SMILES-BERT: Large Scale Unsupervised Pre-Training for Molecular Property Prediction. BCB '19: Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. 2019; pp 429–436.

[100] Chithrananda, S.; Grand, G.; Ramsundar, B. ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction. 2020, arXiv:2010.09885. arXiv.org e-Print archive. https://arxiv.org/abs/2010.09885.

[101] Bagal, V.; Aggarwal, R.; Vinod, P. K.; Priyakumar, U. D. MolGPT: Molecular Generation Using a Transformer-Decoder Model. *J Chem Inf Model* **2022**, *62*, 2064–2076.

[102] Wang, Y.; Zhao, H.; Sciabola, S.; Wang, W. cMolGPT: A Conditional Generative Pre-Trained Transformer for Target-Specific De Novo Molecular Generation. *Molecules* **2023**, *28*, 4430.

[103] Irwin, R.; Dimitriadis, S.; He, J.; Bjerrum, E. J. Chemformer: A Pre-trained Transformer for Computational Chemistry. *Mach. Learn.: Sci. Technol.* **2022**, *3*, 015022.

[104] Tetko, I. V.; Karpov, P.; Van Deursen, R.; Godin, G. State-of-the-Art Augmented NLP Transformer Models for Direct and Single-Step Retrosynthesis. *Nat Commun* **2020**, *11*, 5575.

[105] Xue, D.; Zhang, H.; Chen, X.; Xiao, D.; Gong, Y.; Chuai, G.; Sun, Y.; Tian, H.; Wu, H.; Li, Y.; Liu, Q. X-MOL: Large-Scale Pre-Training for Molecular Understanding and Diverse Molecular Analysis. *Sci Bull (Beijing)* **2022**, *67*, 899–902.

[106] Schwaller, P.; Gaudin, T.; Lányi, D.; Bekas, C.; Laino, T. "Found in Translation": Predicting Outcomes of Complex Organic Chemistry Reactions Using Neural Sequence-to-Sequence Models. *Chem Sci* **2018**, *9*, 6091–6098.

[107] Ucak, U. V.; Ashyrmamatov, I.; Lee, J. Improving the Quality of Chemical Language Model Outcomes with Atom-in-SMILES Tokenization. *J Cheminform* **2023**, *15*, 55.

[108] Li, X.; Fourches, D. SMILES Pair Encoding: A Data-Driven Substructure Tokenization Algorithm for Deep Learning. *J Chem Inf Model* **2021**, *61*, 1560–1569.

[109] He, J.; You, H.; Sandström, E.; Nittinger, E.; Bjerrum, E. J.; Tyrchan, C.; Czechtizky, W.; Engkvist, O. Molecular Optimization by Capturing Chemist's Intuition Using Deep Neural Networks. *J Cheminform* **2021**, *13*, 26.

[110] He, J.; Nittinger, E.; Tyrchan, C.; Czechtizky, W.; Patronov, A.; Bjerrum, E. J.; Engkvist, O. Transformer-Based Molecular Optimization beyond Matched Molecular Pairs. *J Cheminform* **2022**, *14*, 18.

[111] Born, J.; Manica, M. Regression Transformer Enables Concurrent Sequence Regression and Generation for Molecular Language Modelling. *Nat Mach Intell* **2023**, *5*, 432–444.

[112] Rives, A.; Meier, J.; Sercu, T.; Goyal, S.; Lin, Z.; Liu, J.; Guo, D.; Ott, M.; Zitnick, C. L.; Ma, J.; Fergus, R. Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences. *Proc Natl Acad Sci USA* **2021**, *118*, e2016239118.

[113] Bepler, T.; Berger, B. Learning the Protein Language: Evolution, Structure, and Function. *Cell Syst* **2021**, *12*, 654-669.e3.

[114] Singh, R.; Sledzieski, S.; Bryson, B.; Cowen, L.; Berger, B. Contrastive Learning in Protein Language Space Predicts Interactions between Drugs and Protein Targets. *Proc Natl Acad Sci USA* **2023**, *120*, e2220778120.

[115] Brandes, N.; Ofer, D.; Peleg, Y.; Rappoport, N.; Linial, M. ProteinBERT: A Universal Deep-Learning Model of Protein Sequence and Function. *Bioinformatics* **2022**, *38*, 2102–2110.

[116] Elnaggar, A.; Heinzinger, M.; Dallago, C.; Rehawi, G.; Wang, Y.; Jones, L.; Gibbs, T.; Feher, T.; Angerer, C.; Steinegger, M.; Bhowmik, D.; Rost, B. ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. *IEEE Trans Pattern Anal Mach Intell* **2022**, *44*, 7112–7127.

[117] Madani, A.; McCann, B.; Naik, N.; Keskar, N. S.; Anand, N.; Eguchi, R. R.; Huang, P.-S.; Socher, R. ProGen: Language Modeling for Protein Generation. 2020, arXiv:2004.03497. arXiv.org e-Print archive. https://arxiv.org/abs/2004.03497.

[118] Ferruz, N.; Schmidt, S.; Höcker, B. ProtGPT2 Is a Deep Unsupervised Language Model for Protein Design. *Nat Commun* **2022**, *13*, 4348.

[119] Heinzinger, M.; Weissenow, K.; Sanchez, J. G.; Henkel, A.; Mirdita, M.; Steinegger, M.; Rost, B. Bilingual Language Model for Protein Sequence and Structure. *NAR Genomics and Bioinformatics* **2024**, *6*, lqae150.

[120] Benegas, G.; Ye, C.; Albors, C.; Li, J. C.; Song, Y. S. Genomic Language Models: Opportunities and Challenges. *Trends Genet* **2025**, *41*, 286–302.

[121] Ji, Y.; Zhou, Z.; Liu, H.; Davuluri, R. V. DNABERT: Pre-Trained Bidirectional Encoder Representations from Transformers Model for DNA-Language in Genome. *Bioinformatics* **2021**, *37*, 2112–2120.

[122] Luo, H.; Chen, C.; Shan, W.; Ding, P.; Luo, L. iEnhancer-BERT: A Novel Transfer Learning Architecture Based on DNA-Language Model for Identifying Enhancers and

Their Strength. *Intelligent Computing Theories and Application* **2022**, *13394*, pp 153–165.

[123] Yang, F.; Wang, W.; Wang, F.; Fang, Y.; Tang, D.; Huang, J.; Lu, H.; Yao, J. scBERT as a Large-Scale Pretrained Deep Language Model for Cell Type Annotation of Single-Cell RNA-Seq Data. *Nat Mach Intell* **2022**, *4*, 852–866.

[124] Zvyagin, M.; Brace, A.; Hippe, K.; Deng, Y.; Zhang, B.; Bohorquez, C. O.; Clyde, A.; Kale, B.; Perez-Rivera, D.; Ma, H.; Mann, C. M.; Irvin, M.; Ozgulbas, D. G.; Vassilieva, N.; Pauloski, J. G.; Ward, L.; Hayot-Sasson, V.; Emani, M.; Foreman, S.; Xie, Z.; Lin, D.; Shukla, M.; Nie, W.; Romero, J.; Dallago, C.; Vahdat, A.; Xiao, C.; Gibbs, T.; Foster, I.; Davis, J. J.; Papka, M. E.; Brettin, T.; Stevens, R.; Anandkumar, A.; Vishwanath, V.; Ramanathan, A. GenSLMs: Genome-Scale Language Models Reveal SARS-CoV-2 Evolutionary Dynamics. *The International Journal of High Performance Computing Applications* **2023**, *37*, 683–705.

[125] Malusare, A.; Kothandaraman, H.; Tamboli, D.; Lanman, N. A.; Aggarwal, V. Understanding the Natural Language of DNA Using Encoder–Decoder Foundation Models with Byte-Level Precision. *Bioinformatics Advances* **2024**, *4*, vbae117.

[126] Shao, B.; Yan, J. A Long-Context Language Model for Deciphering and Generating Bacteriophage Genomes. *Nat Commun* **2024**, *15*, 9392.

[127] Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. Proceedings of the 54th Annual Meeting of the Association for Computational. 2016; pp 1715–1725.

[128] Stumpfe, D.; Dimova, D.; Bajorath, J. Computational Method for the Systematic Identification of Analog Series and Key Compounds Representing Series and Their Biological Activity Profiles. *J Med Chem* **2016**, *59*, 7667–7676.

[129] Yoshimori, A.; Bajorath, J. Computational Analysis, Alignment and Extension of Analogue Series from Medicinal Chemistry. *Future Sci OA* **2022**, *8*, FSO804.

[130] Leach, A. G.; Jones, H. D.; Cosgrove, D. A.; Kenny, P. W.; Ruston, L.; MacFaul, P.; Wood, J. M.; Colclough, N.; Law, B. Matched Molecular Pairs as a Guide in the Optimization of Pharmaceutical Properties; a Study of Aqueous Solubility, Plasma Protein Binding and Oral Exposure. *J Med Chem* **2006**, *49*, 6672–6682.

[131] Wassermann, A. M.; Dimova, D.; Iyer, P.; Bajorath, J. Advances in Computational Medicinal Chemistry: Matched Molecular Pair Analysis. *Drug Develop Res* **2012**, *73*, 518–527.

[132] Kramer, C.; Fuchs, J. E.; Whitebread, S.; Gedeck, P.; Liedl, K. R. Matched Molecular Pair Analysis: Significance and the Impact of Experimental Uncertainty. *J Med Chem* **2014**, *57*, 3786–3802.

[133] Tyrchan, C.; Evertsson, E. Matched Molecular Pair Analysis in Short: Algorithms, Applications and Limitations. *Comput Struct Biotechnol J* **2017**, *15*, 86–90.

[134] Sheridan, R. P.; Hunt, P.; Culberson, J. C. Molecular Transformations as a Way of Finding and Exploiting Consistent Local QSAR. *J Chem Inf Model* **2006**, *46*, 180–192.

[135] Warner, D. J.; Griffen, E. J.; St-Gallay, S. A. WizePairZ: A Novel Algorithm to Identify, Encode, and Exploit Matched Molecular Pairs with Unspecified Cores in Medicinal Chemistry. *J Chem Inf Model* **2010**, *50*, 1350–1357.

[136] Hussain, J.; Rea, C. Computationally Efficient Algorithm to Identify Matched Molecular Pairs (MMPs) in Large Data Sets. *J Chem Inf Model* **2010**, *50*, 339–348.

[137] Lewell, X. Q.; Judd, D. B.; Watson, S. P.; Hann, M. M. RECAP--Retrosynthetic Combinatorial Analysis Procedure: A Powerful New Technique for Identifying Privileged Molecular Fragments with Useful Applications in Combinatorial Chemistry. *J Chem Inf Comput Sci* **1998**, *38*, 511–522.

[138] De La Vega De León, A.; Bajorath, J. Matched Molecular Pairs Derived by Retrosynthetic Fragmentation. *Med Chem Commun* **2014**, *5*, 64–67.

[139] Naveja, J. J.; Vogt, M.; Stumpfe, D.; Medina-Franco, J. L.; Bajorath, J. Systematic Extraction of Analogue Series from Large Compound Collections Using a New Computational Compound-Core Relationship Method. *ACS Omega* **2019**, *4*, 1027–1032.

[140] Wassermann, A. M.; Haebel, P.; Weskamp, N.; Bajorath, J. SAR Matrices: Automated Extraction of Information-Rich SAR Tables from Large Compound Data Sets. *J Chem Inf Model* **2012**, *52*, 1769–1776.

[141] Gupta-Ostermann, D.; Bajorath, J. The 'SAR Matrix' Method and Its Extensions for Applications in Medicinal Chemistry and Chemogenomics. *F1000Res* **2014**, *3*, 113.

[142] Yoshimori, A.; Tanoue, T.; Bajorath, J. Integrating the Structure–Activity Relationship Matrix Method with Molecular Grid Maps and Activity Landscape Models for Medicinal Chemistry Applications. *ACS Omega* **2019**, *4*, 7061–7069.

[143] Chen, H.; Bajorath, J. Combining a Chemical Language Model and the Structure–Activity Relationship Matrix Formalism for Generative Design of Potent Compounds with Core Structure and Substituent Modifications. *J Chem Inf Model* **2024**, *64*, 8784–8795.

[144] Bickerton, G. R.; Paolini, G. V.; Besnard, J.; Muresan, S.; Hopkins, A. L. Quantifying the Chemical Beauty of Drugs. *Nat Chem* **2012**, *4*, 90–98.

[145] Ertl, P.; Schuffenhauer, A. Estimation of Synthetic Accessibility Score of Drug-like Molecules Based on Molecular Complexity and Fragment Contributions. *J Cheminform* **2009**, *1*, 8.

[146] Brown, N.; Fiscato, M.; Segler, M. H. S.; Vaucher, A. C. GuacaMol: Benchmarking Models for de Novo Molecular Design. *J Chem Inf Model* **2019**, *59*, 1096–1108.

[147] Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; Kadurin, A.; Johansson, S.; Chen, H.; Nikolenko, S.; Aspuru-Guzik, A.; Zhavoronkov, A. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Front Pharmacol* **2020**, *11*, 565644.

[148] Bender, A.; Schneider, N.; Segler, M.; Patrick Walters, W.; Engkvist, O.; Rodrigues, T. Evaluation Guidelines for Machine Learning Tools in the Chemical Sciences. *Nat Rev Chem* **2022**, *6*, 428–442.

[149] Gangwal, A.; Lavecchia, A. Unleashing the Power of Generative AI in Drug Discovery. *Drug Discov Today* **2024**, *29*, 103992.

[150] Chen, H.; Bajorath, J. Designing Highly Potent Compounds Using a Chemical Language Model. *Sci Rep* **2023**, *13*, 7412.

[151] Chen, H.; Bajorath, J. Meta-Learning for Transformer-Based Prediction of Potent Compounds. *Sci Rep* **2023**, *13*, 16145.

[152] Chen, H.; Bajorath, J. Generative Design of Compounds with Desired Potency from Target Protein Sequences Using a Multimodal Biochemical Language Model. *J Cheminform* **2024**, *16*, 55.

[153] Guha, R. Exploring uncharted territories: predicting activity cliffs in structure-activity landscapes. *J Chem Inf Model* **2012**, *52*, 2181–2191.

[154] Heikamp, K.; Hu, X.; Yan, A.; Bajorath, J. Prediction of activity cliffs using support vector machines. *J Chem Inf Model* **2012**, *52*, 2354–2365.

Appendix A

DeepAC–Conditional Transformer-Based Chemical Language Model for the Prediction of Activity Cliffs Formed by Bioactive Compounds

Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Digital Discovery*, 2022, **1**, 898

DeepAC – conditional transformer-based chemical language model for the prediction of activity cliffs formed by bioactive compounds

Hengwei Chen, Martin Vogt and Jürgen Bajorath *

Activity cliffs (ACs) are formed by pairs of structurally similar or analogous active small molecules with large differences in potency. In medicinal chemistry, ACs are of high interest because they often reveal structure–activity relationship (SAR) determinants for compound optimization. In molecular machine learning, ACs provide test cases for predictive modeling of discontinuous (non-linear) SARs at the level of compound pairs. Recently, deep neural networks have been used to predict ACs from molecular images or graphs *via* representation learning. Herein, we report the development and evaluation of chemical language models for AC prediction. It is shown that chemical language models learn structural relationships and associated potency differences to reproduce ACs. A conditional transformer termed DeepAC is introduced that accurately predicts ACs on the basis of small amounts of training data compared to other machine learning methods. DeepAC bridges between predictive modeling and compound design and should thus be of interest for practical applications.

Received 19th July 2022
Accepted 28th October 2022

DOI: 10.1039/d2dd00077f

rsc.li/digitaldiscovery

1 Introduction

In medicinal chemistry, compound optimization relies on the exploration of structure–activity relationships (SARs). Therefore, series of structural analogues are generated to probe substitution sites in specifically active compounds with different functional groups and improve potency and other lead optimization-relevant molecular properties. For lead optimization, the activity cliff (AC) concept plays an important role. ACs are defined as pairs or groups of structurally similar compounds or structural analogues that are active against a given target and have large differences in potency.[1–3] As such, ACs represent strongly discontinuous SARs because small chemical modifications lead to large biological effects. In medicinal chemistry, SAR discontinuity captured by ACs helps to identify substituents that are involved in critically important ligand–target interactions. In compound activity prediction, the presence of SAR discontinuity prevents the derivation of quantitative SAR (QSAR) models relying on continuous SAR progression and requires non-linear machine learning models.[1,2]

For a non-ambiguous and systematic assessment of ACs, similarity and potency difference criteria must be clearly defined.[2,3] Originally, molecular fingerprints (that is, bit string representations of chemical structure) have been used as molecular representations to calculate the Tanimoto coefficient,[4] a whole-molecule similarity metric, for identifying similar compounds forming ACs.[2] Alternatively, substructure-based similarity measures have been adapted for defining ACs, which have become increasingly popular in medicinal chemistry, because they are often chemically more intuitive than calculated whole-molecule similarity.[3] For example, a widely used substructure-based similarity criterion for AC analysis is the formation of a matched molecular pair (MMP), which is defined as a pair of compounds that are only distinguished by a chemical modification at a single site.[5] Thus, MMPs can be used to represent pairs of structural analogues, which explains their popularity in medicinal chemistry. Moreover, MMPs can also be efficiently identified algorithmically.[5] Although statistically significant potency differences for ACs can be determined for individual compound activity classes,[6] for the systematic assessment of ACs and computational modeling, a potency difference threshold of at least two orders of magnitude (100-fold) has mostly been applied.[2,3]

While medicinal chemistry campaigns encounter ACs on a case-by-case basis, systematic compound database analysis has identified ACs across different compound activity classes, providing a wealth of SAR information.[2,7] Here, computational and medicinal chemistry meet. With rapidly increasing numbers of publicly available bioactive compounds, AC populations have also grown over time.[3] However, the rate at which ACs are formed across different activity classes has essentially remained constant. Only ~5% of pairs of structural analogues sharing the same activity form ACs across different activity classes.[3,7] Thus, as expected for compounds representing the

Department of Life Science Informatics and Data Science, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Friedrich-Hirzebruch-Allee 5/6, D-53115 Bonn, Germany. E-mail: bajorath@bit.uni-bonn.de; Fax: +49-228-7369-100; Tel: +49-228-7369-100

pinnacle of SAR discontinuity, structural analogues rarely form ACs.

Systematic identification of ACs across activity classes has also provided the basis for computational predictions of ACs. For machine learning, AC predictions generally present a challenge, for three reasons. First, as discussed, the underlying SARs that need to be accounted for are highly discontinuous; second, data sets of ACs and non-ACs are unbalanced; third, predictions need to be made at the level of compound pairs, rather than individual compounds, which is usually the case in compound classification or molecular property prediction. Initial attempts to predict ACs were reported a decade ago.[8,9] ACs were first accurately predicted using support vector machine (SVM) modeling on the basis of special kernel functions enabling compound pair predictions.[9] These findings have also catalyzed further AC predictions using SVR variants[10–12] and other methods,[13–18] as discussed below. Recently, various deep neural network architectures have been used to predict ACs from images[14,15] and molecular graphs using representation learning[16] or derive regression models for potency prediction of AC compounds.[17,18]

In this work, we further extend this methodological spectrum by introducing chemical language models for combined AC prediction and generative compound design. Compared to earlier studies predicting ACs using classification models, the approach presented herein was designed to extend AC predictions with the capacity to produce new AC compounds, thus integrating predictive and generative modeling in the context of AC analysis and AC-based compound design.

2 Methods

2.1 Compounds and activity data

Bioactive compounds with high-confidence activity data were assembled from ChEMBL (version 26).[19] The following selection criteria were applied. Only compounds involved in direct interactions with human targets at the highest assay confidence level (assay confidence score 9) were selected and only numerically specified equilibrium constants (K_i values) were accepted as potency measurements. Equilibrium constants were recorded as (negative logarithmic) pK_i values. Multiple measurements for the same compound were averaged, provided all values fell within the same order of magnitude; if not, the compound was disregarded. Hence, in a given class, all compounds were active against a specific target. Compounds were represented using molecular-input line-entry system (SMILES) strings.[20]

2.2 Matched molecular pairs

From activity classes, all possible MMPs were generated by systematically fragmenting individual exocyclic single bonds in compounds and sampling core structures and substituents in index tables.[5] For substituents, size restrictions were applied to limit MMP formation to structural analogues typical for medicinal chemistry. Accordingly, a substituent was permitted to contain at most 13 non-hydrogen atoms and the core structure was required to be at least twice as large as a substituent. In addition, for MMP compounds, the maximum difference in non-hydrogen atoms between the substituents was set to eight, yielding transformation size-restricted MMPs.[21] The systematic search identified 357 343 transformation size-restricted MMPs originating from a total of 600 activity classes.

2.3 Data set for model derivation

From the MMPs, a large general data set for model training was assembled by combining 338 748 MMPs from 596 activity classes. The majority of MMPs captured only minor differences in potency. Importantly, model pre-training, as specified below, did not require the inclusion of explicit target information because during this phase, the model must learn MMP-associated potency differences caused by given chemical transformations. Each MMP represented a true SAR, which was of critical relevance in this context, while target information was not required for pre-training. By contrast, subsequent fine-tuning then focused the model on target-specific activity classes for AC prediction and compound design.

MMPs comprising the general data set were represented as triples:

$$(\text{Compound}_A, \text{Compound}_B, \text{Potency}_B - \text{Potency}_A).$$

Compound_A represented the source compound that was concatenated with the potency difference ($\text{Potency}_B - \text{Potency}_A$) while Compound_B represented the target compound. Each MMP yielded two triples, in which each MMP compound was used once as the source and target compound, respectively. The source and target compounds were then used as the input and associated output for model training, respectively. Furthermore, for MMP-triples, data ambiguities could arise if an MMP was associated with multiple potency values for different targets or if a given source compound and potency difference was associated with multiple target compounds from different activity classes. Such MMPs were eliminated. Finally, for the general data set, a total of 338 748 qualifying MMP-triples were obtained.

For modeling, MMP-triples were randomly divided into training (80%), validation (10%), and test (10%) sets. Source and target compounds from MMP-triples displayed nearly indistinguishable potency value distributions.

For the initial evaluation of chemical language models, three different test (sub)set versions were designed:

(i) Test-general: complete test set of 33 875 MMP-triples excluded from model training.

(ii) Test-core: subset of 2576 test set MMP-triples with core structures not present in training compounds.

(iii) Test-sub: subset of 14 193 MMP-triples with substituents (R-groups) not contained in training compounds.

For the generation of the training subsets, compounds were decomposed into core structures and substituents *via* MMP fragmentation.[5]

2.4 Activity cliffs

For ACs, the MMP-Cliff definition was applied.[21] Accordingly, a transformation size-restricted MMP from a given activity class represented an AC if the two MMP-forming compounds had a potency difference of at least two orders of magnitude (100-fold; *i.e.*, $\Delta pK_i \geq 2.0$). MMP-Cliffs were distinguished from "MMP-nonCliffs", that is, pairs of structural analogues not representing an AC. To avoid potency boundary effects in AC prediction, compounds forming an MMP-nonCliff were restricted to a maximal potency difference of one order of magnitude (10-fold; $\Delta pK_i \leq 1$). Hence, MMPs capturing potency differences between 10- and 100-fold were not considered for AC prediction.

MMP-Cliffs and MMP-nonCliffs were extracted from four large activity classes including inhibitors of thrombin (ChEMBL ID 204) and tyrosine kinase Abl (1862) as well as antagonists of the Mu opioid receptor (233) and corticotropin releasing factor receptor 1 (1800). For MMP-Cliffs and MMP-nonCliffs, triples were ordered such that Compound_A had lower potency than (or equal potency to) Compound_B. These activity classes were excluded from the general data set and their MMP-Cliffs and MMP-nonCliffs thus formed an external/independent test set for AC prediction (Table 1).

2.5 Deep chemical language models

Chemical language models for AC prediction were designed to learn the following mapping from MMP-triples:

(Source compound, Potency difference) → (Target compound).

Then, given a new (Source compound, Potency difference) test instance, trained models were supposed to generate a set of target candidate compounds meeting the potency difference condition.

Sequence-to-sequence (Seq2Seq) models represent an encoder–decoder architecture to convert an input sequence (such as a character string) into an output sequence.[22] These models can be adapted for a variety of applications, especially for neural machine translation.[22] The encoder reads an input sequence and compresses it into a context vector as its last hidden state. The context vector serves as the input for the decoder network component that interprets the vector to predict an output sequence. Because long input sequences often present challenges for generating context vectors,[23] an attention mechanism[24] was introduced that utilizes hidden states from each time step of the encoder. As a further advance, a transformer neural network architecture was introduced that only relies on the attention mechanism.[25] The transformer architecture comprises multiple encoder–decoder modules (Fig. 1). An encoder module consists of a stack of encoding layers composed of two sub-layers including a multi-head self-attention sub-layer and a fully connected feed-forward network (FFN) sub-layer. Multi-head attention has multiple, single attention functions acting in parallel such that different positions in the input sequence can be processed simultaneously. The attention mechanism is based upon the following function:

$$\text{Attention}(Q,\ K,\ V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

The input for the attention layer is received in the form of three parameters including query (Q), keys (K), and values (V). In addition, a scaling factor d_k (equal to the size of weight matrices) prevents calculations of excessive dot products.[25] More details concerning the attention function are provided in the original literature of the transformer model.[25] The FFN sub-layer employs rectified linear unit (ReLU) activation.[26] The multi-head self-attention and FFN sub-layers are then linked *via* layer normalization[27] and a residual skip-connection.[28] Each decoder layer contains three sub-layers including an FFN sub-layer and two multi-head attention sub-layers. The first attention sub-layer was controlled by a mask function.

In this work, all source and target molecules were represented as canonical SMILES strings generated using RDKit[29] and further tokenized to construct a chemical vocabulary containing all the possible chemical tokens. The start and end of a sequence were represented by two special "start" and "end" tokens, respectively. For AC prediction, models must be guided towards the generation of compounds meeting potency difference constraints. Therefore, potency differences captured by MMPs were tokenized by binning.[23] The potency difference, ranging from −8.02 to 9.53, was partitioned into 1755 bins of width 0.01 that were also added to the chemical vocabulary. Each bin was encoded by a single token and each potency difference was assigned to the token of the corresponding bin (Fig. 1), *e.g.*, a potency difference of 2.134 was encoded as 'pK_i_change_(2.13, 2.14)'. Accordingly, the tokenization preserved the quantitative relationship between bins. The SMILES representation of a source compound combined with its potency difference token then represented the input sequence for the transformer encoder and was converted into

Table 1 Compound activity classes for activity cliff prediction

Target name	ChEMBL ID	Total MMPs	MMP- Cliffs	MMP-nonCliffs
Thrombin	204	4249	438	2976
Mu opioid receptor	233	5875	329	4319
Tyrosine kinase Abl	1862	5403	564	3093
Corticotropin releasing factor receptor 1	1800	3068	317	1889

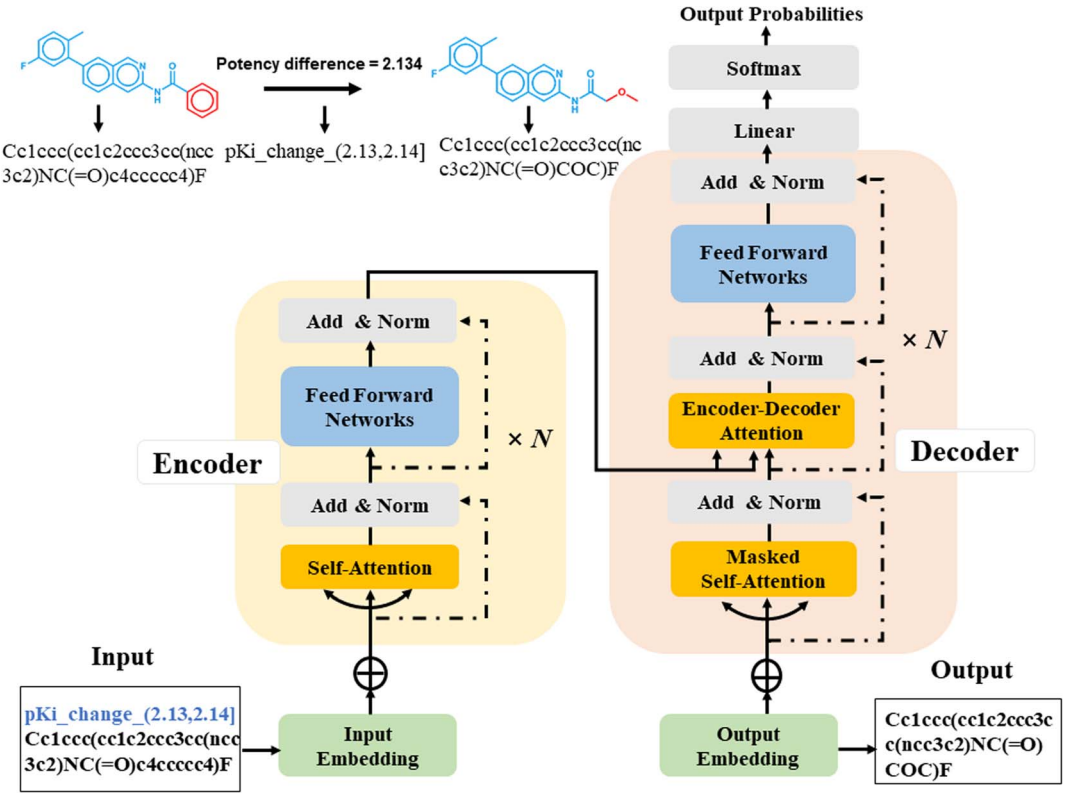


Fig. 1 Architecture of a transformer encoder–decoder with attention mechanism.

a latent representation. Based on this representation, the transformer decoder iteratively generated output SMILES sequences until the end token was obtained. During training, the transformer model minimized the cross-entropy loss between the ground-truth target and output sequence.

2.6 Model derivation and selection

Seq2Seq and transformer models were implemented using Pytorch.[30] The Adam optimizer with learning rate 0.0001 and a batch size of 64 was applied. For transformer models, default hyperparameter settings were used,[25] except for the input and output encoding dimension, which was reduced from 512 to 256, and label smoothing, which was set to 0. On the basis of the general training set, models were derived over 200 epochs. A checkpoint was saved at each epoch and for the validation set, minimal loss was determined for selecting the final model. For the test set, generated candidate compounds were canonicalized using RDkit and compared to the target compounds.

2.7 Reference methods for activity prediction

For AC prediction, the chemical language models were compared to models of different machine learning methods including support vector machine (SVM),[31] random forest (RF),[32] and extreme gradient boosting (XGboost)[33] that were generated using scikit-learn.[34] As a molecular representation, the extended connectivity fingerprint with bond diameter of 4

Table 2 Hyperparameter settings for optimization of different models

Model	Hyperparameters	Value space for optimization
SVM	Kernel function	'Linear', 'sigmoid', 'poly', 'rbf', 'tanimoto'
	C	1, 10, 100, 1000, 10 000
	Gamma	10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}
RF	Max_depth	3, 4, 5, 6, 7, 8, 9, 10
	Max_features	32, 64, 128, 256, 512, 1024
	n_estimators	1, 2, 4, 8, 16, 32, 64, 100, 200
XGboost	Max_depth	3, 4, 5, 6, 7, 8, 9, 10
	n_estimators	1, 2, 4, 8, 16, 32, 64, 100, 200
	Learning_rate	0.0001, 0.001, 0.01, 0.1, 0.2, 0.3
	Subsample	0.5, 0.6, 0.7, 0.8, 0.9, 1
	Min_child_weight	0, 1, 2, 3, 4, 5

(ECFP4) was used.[35] For the common core of an MMP and the two substituents defining the chemical transformation, fingerprint vectors were generated. For use of the MMP kernel,[9] these vectors were concatenated to yield a single vector[9] as input for deriving SVM, RF, and XGboost models. Hyperparameters of all models were optimized using the Hyperopt[36] package with five-fold cross-validation, as reported in Table 2.

2.8 Evaluation metrics

A reproducibility criterion was introduced to measure the ability of a chemical language model to reproduce a target compound for a given source compound and potency difference. An MMP-triple met this criterion if it was reproduced when generating a pre-defined number of target candidate compounds. In our calculations, up to 50 distinct molecules were generated for each source compound to determine the reproducibility of a target compound, defined as:

$$\text{Reproducibility} = \frac{\text{MMP}_{\text{repro}}}{\text{MMP}_{\text{test}}} \quad (2)$$

MMP_{test} and $\text{MMP}_{\text{repro}}$ denote the number of MMP-triples that were tested and reproduced by a model, respectively. Notably, this definition of reproducibility directly corresponds to the recall of labeled instances for classification models.

AC predictions were also evaluated by determining the true positive rate (TPR), true negative rate (TNR), and balanced accuracy (BA),[37] defined as:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$\text{TNR} = \frac{\text{TN}}{\text{TP} + \text{FN}} \quad (4)$$

$$\text{BA} = \frac{\text{TPR} + \text{TNR}}{2} \quad (5)$$

TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives respectively.

3 Results and discussion

3.1 Study concept

The basic idea underlying the use of chemical language models for AC prediction was learning the following mapping based on textual/string representations:

(Source compound, Potency difference) → (Target compound).

Then, given a new (Source compound, Potency difference) test instance, the pre-trained models should generate target compounds with appropriate potency. For deriving pairs of source and target compounds, the MMP formalism was applied. For AC prediction, pre-trained models were subjected to fine-tuning on MMP-Cliffs and MMP-nonCliffs from given activity classes, corresponding to the derivation of other supervised machine learning models.

3.2 Pre-trained chemical language models

Initially, the ability of Seq2Seq and transformer models to reproduce target compounds for test (sub)sets was evaluated by calculating the reproducibility measure. The results are summarized in Table 3. Therefore, for each test set triple, the source compound/potency difference concatenation was used as input and 50 target candidate compounds were sampled. Notably, the sampling procedure is an integral part of chemical language models in order to generate new candidate compounds, hence setting these models apart from standard class label prediction/classification approaches.

For the entire test set, the Seq2Seq and transformer model achieved reproducibility of 0.719 and 0.818, respectively. Hence, the models were able to regenerate more than 70% and 80% of the target compounds from MMP-triples not used for training, respectively. However, reproducibility was consistently higher for the transformer and all training set versions than for the Seq2Seq model (Table 3). Hence, preference for AC prediction was given to the transformer. The test-general reproducibility of more than 80% was considered high. Attempting to further increase this reproducibility might compromise the ability of the model to generate novel compounds by strongly focusing on chemical space encountered during training. As expected, the test-core reproducibility was generally lowest because in this case, the core structures of MMPs were not available during training (limiting reproducibility much more than in the case of test-sub, *i.e.*, evaluating novel substituents).

3.3 Fine-tuning for activity cliff prediction

The transformer was first applied to reproduce MMP-Cliffs and MMP-nonCliffs from the four activity classes excluded from pre-training. Therefore, for each MMP-triple, the source compound/potency difference concatenation was used as input for generating target compounds. As expected for activity classes not encountered during model derivation, reproducibility of MMP-Cliffs and MMP-nonCliffs was low, reaching maximally 5% for MMP-Cliffs and ~19% for MMP-nonCliffs (Table 4).

Therefore, a transfer learning approach was applied by fine-tuning the pre-trained transformer on these activity classes. For fine-tuning, 5%, 25%, and 50% of MMP-Cliffs and MMP-nonCliffs of each class were randomly selected. The resulting models were then tested on the remaining 50% of the MMP-Cliffs and MMP-nonCliffs.

Only 5% of the training data were required for fine-tuning to achieve reproducibility rates of 70% to greater than 80% for

Table 3 Reproducibility of target compounds by chemical language models

	Test-general	Test-core	Test-sub
Seq2Seq	0.719	0.370	0.759
Transformer	0.818	0.528	0.850

Table 4 Reproducibility of MMP-Cliffs and MMP-nonCliffs by pre-trained DeepAC

	Activity classes			
Reproducibility	ChEMBL204	1862	233	1800
MMP-Cliffs	0.050	0.007	0.049	0.006
MMP-nonCliffs	0.185	0.081	0.188	0.035

MMP-Cliffs from the different activity classes (Fig. 2A, solid lines). For MMP-nonCliffs, 25% of the training data were required to achieve reproducibility between 60% and 80% for the different classes (Fig. 2B, solid lines). For practical applications, these findings were encouraging because for any given target, there were many more MMP-nonCliffs available than MMP-Cliffs.

Furthermore, to directly test whether high reproducibility achieved through fine-tuning only depended on learning structural relationships encoded by MMPs or if potency differences were also learned, a prerequisite for meaningful AC prediction, control calculations with inverted potency differences were carried out. Therefore, for all MMP-Cliffs, potency differences were set to $\Delta pK_i = 0.1$ and for all MMP-nonCliffs, potency differences were set to $\Delta pK_i = 2.0$. Using these hypothetical (SAR-nonsensical) data as test instances, reproducibility rates were determined again. In this case, reproducibility rates remained well below 50% for both MMP-Cliffs (Fig. 2A, dashed lines) and MMP-nonCliffs (Fig. 2B, dashes lines) and further decreased with increasing amounts of training data used for fine-tuning. These findings conclusively showed that the conditional transformer associated structural relationships with corresponding potency differences, thereby learning to reproduce and differentiate between MMP-Cliffs and MMP-nonCliffs.

In the following, the conditional transformer for AC prediction is referred to as DeepAC.

We also evaluated the capability of the model to reconstruct both MMP-Cliffs and MMP-nonCliffs originating from the same source compound. For each activity class, we compiled a set of source compounds from the original test data. Then, models were fine-tuned with varying amounts of data and applied to reproduce MMP-Cliff and MMP-nonCliff target compounds from the same source compound. As shown in Fig. 3, DeepAC reproduced more than 80% of the target compounds using 5%, 25%, or 50% of fine-tuning data, depending on the activity class.

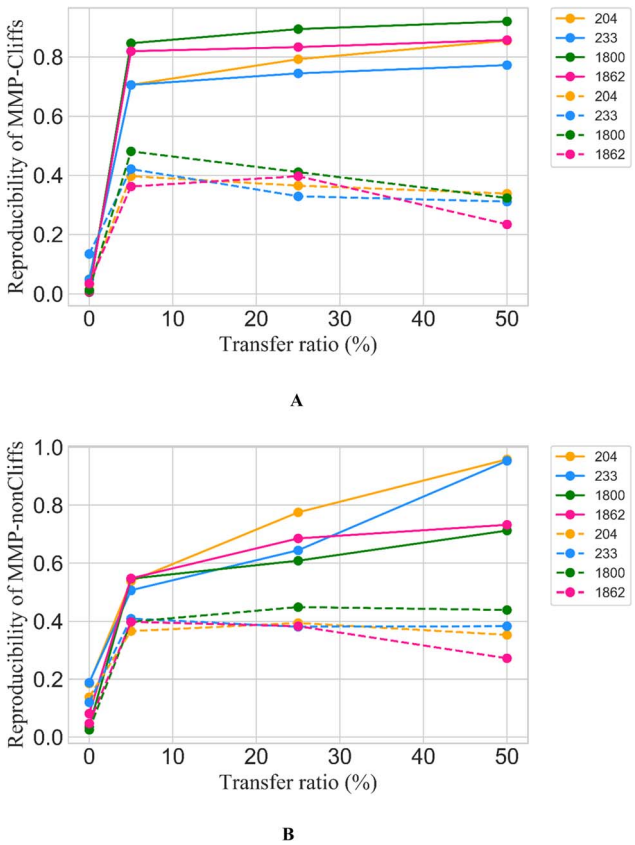


Fig. 2 Reproducibility of MMP-Cliffs and MMP-nonCliffs after fine-tuning. For (A) MMP-Cliffs and (B) MMP-nonCliffs from different activity classes (identified by ChEMBL target IDs according to Table 1), reproducibility is reported as a function of transfer ratio accounting for the percentage of training data used for fine tuning. Solid lines represent results for true MMP-Cliffs and MMP-nonCliffs and dashed lines for control data obtained by inverting potency differences for MMP-Cliffs and MMP-nonCliffs.

3.4 Performance comparison of unconditional and conditional DeepAC

We also compared model performance of conditional DeepAC and unconditional DeepAC generated by randomly shuffling potency differences of MMPs during fine-tuning. Accordingly, for each activity class, potency differences were randomly shuffled for the three training set sizes (5, 25, and 50%) for the fine-tuning MMPs; then the pre-trained transformer was fine-tuned using these artificial MMPs. As shown in Fig. 4A, for MMP-Cliffs, the reproducibility of conditional DeepAC was significantly higher than of unconditional DeepAC. However, for the reproducibility of MMP-nonCliffs, conditional DeepAC only yielded slight improvement than unconditional DeepAC

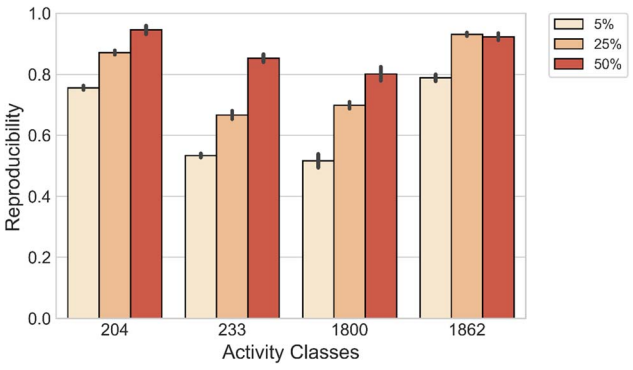


Fig. 3 Reproducibility of MMP-Cliffs and MMP-nonCliffs originating from the same source compound.

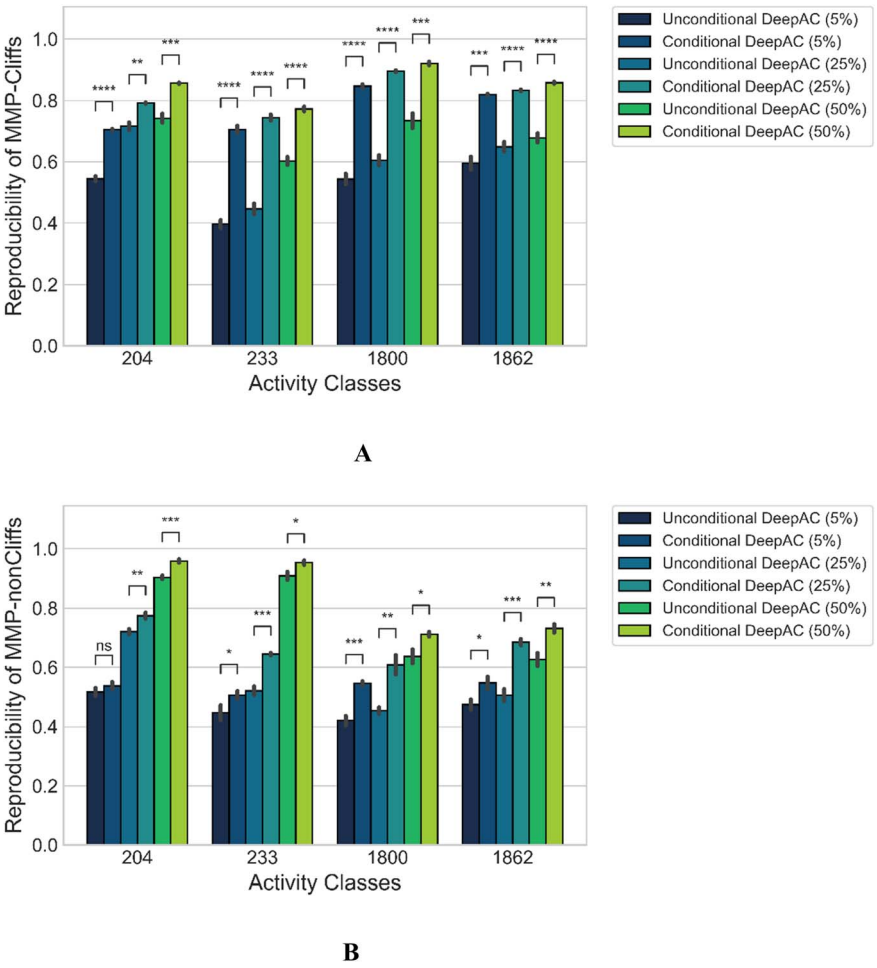


Fig. 4 Performance of conditional *vs.* unconditional DeepAC. Reproducibility is reported for (A) MMP-Cliffs and (B) MMP-nonCliffs. Mean and standard deviations (error bars) are provided for each activity class. Independent-samples *t* tests were conducted: $0.05 < p \leq 1.00$ (ns), $0.01 < p \leq 0.05$ (*), $0.001 < p \leq 0.01$ (**), $0.0001 < p \leq 0.001$ (***), $p \leq 0.0001$ (****).

(Fig. 4B). This was principally expected because potency differences of most MMP-nonCliffs remained similar (less than one order of magnitude). These findings further demonstrated that potency difference of ACs played a critical role for model derivation.

3.5 Alternative fine-tuning

As an additional control, fine-tuning was carried out using MMP-nonCliffs ($\Delta p K_i < 1.0$) and MMPs with $1.0 \leq \Delta p K_i < 2.0$ that were initially excluded from the analysis to prevent potential bias due to boundary effects. Then, the reproducibility of MMP-Cliffs of the fine-tuned models was determined and compared to regular fine-tuning. Fig. 5A shows that fine-tuning only with MMP-nonCliffs yielded reproducibility of 0.306–0.576 for the activity classes, reflecting a baseline learning effect of MMPs and associated potency differences, even if these were only small. However, fine-tuning with MMPs ($1.0 \leq \Delta p K_i < 2.0$), significantly increased the reproducibility of MMP-Cliffs to 0.620 for thrombin inhibitors, 0.607 for Mu opioid receptor ligands, 0.726 for corticotropin releasing factor receptor 1 ligands and 0.716 for tyrosine kinase Abl inhibitors. Fine-tuning using increasing proportions of MMP-Cliffs further increased reproducibility. Taken together, these findings clearly demonstrated the influence of MMP-associated potency differences for AC predictions. Furthermore, consistent with these observations, Fig. 5B shows that fine-tuning with MMP-nonCliffs, led to very high reproducibility of MMP-nonCliffs, which was substantially reduced when fine-tuning was carried out with MMPs capturing larger potency differences.

3.6 Global performance comparison

The performance of DeepAC in activity prediction was compared to other machine learning methods including SVM, RF, and XGboost. First, the reproducibility/recall of MMP-Cliffs and MMP-nonCliffs from the four activity classes was compared for unbalanced training and test sets according to Table 1. For

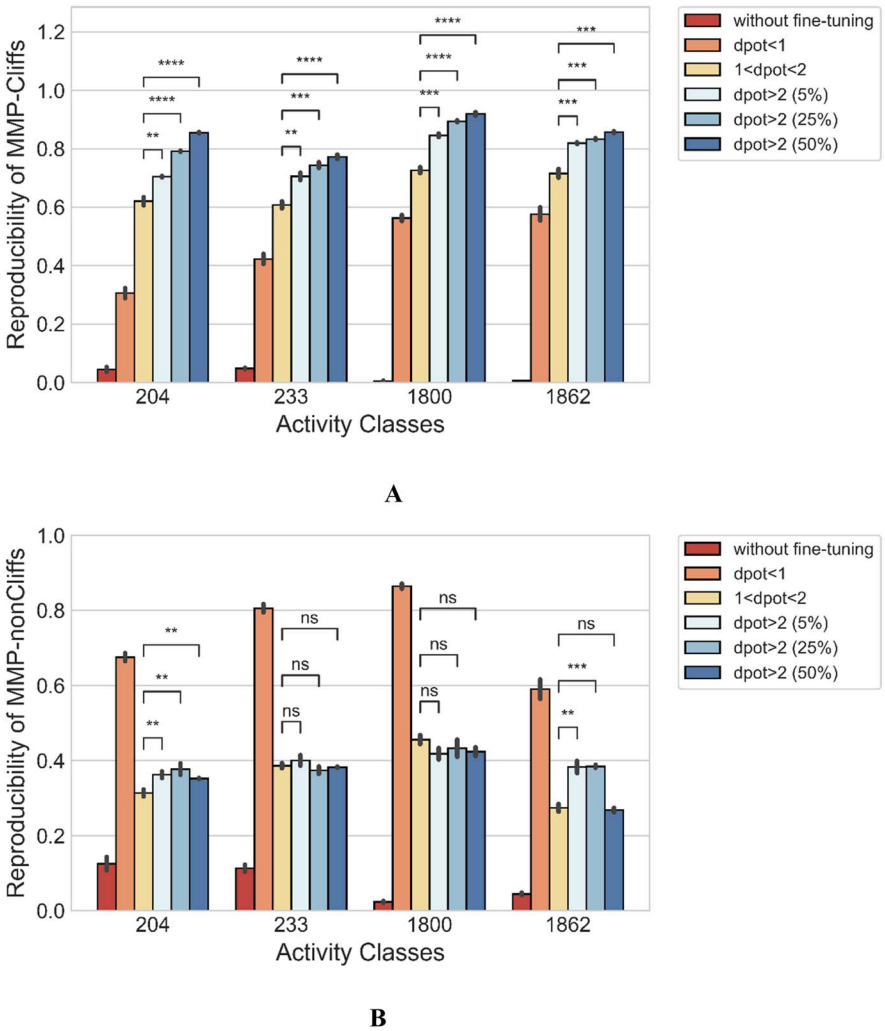


Fig. 5 Model performance comparison after alternative fine-tuning with different types of MMPs. Reproducibility of (A) MMP-Cliffs and (B) MMP-nonCliffs is reported. Mean and standard deviations (error bars) are provided for each activity class. Independent-samples *t* tests were conducted: $0.05 < p \leq 1.00$ (ns), $0.01 < p \leq 0.05$ (*), $0.001 < p \leq 0.01$ (**), $0.0001 < p \leq 0.001$ (***), $p \leq 0.0001$ (****).

AC prediction, unbalanced sets were deliberately used to account for the fact that ACs are rare compared to other pairs of structural analogues with minor potency differences, thus providing a realistic prediction scenario.

The predictions using different methods were generally stable, yielding only low standard deviations over independent trials (Fig. 6). Using 5% of training data for fine-tuning or model derivation, the recall (TPR) of MMP-Cliffs was consistently higher for DeepAC than the reference methods, which failed on two activity classes (Fig. 6). For increasing amounts of training data, recall performance of the reference methods further increased and SVM reached the 80% or 90% recall level of DeepAC in two cases when 50% of available data were used for training (Fig. 6).

For MMP-nonCliffs, representing the majority class for the predictions, a different picture was obtained. Here, the recall of reference methods for increasing amounts of training data was mostly greater than 90% and significantly higher than of DeepAC (Fig. 7). For DeepAC, recall/reproducibility increased with increasing amounts of training data and reached highest performance very similar to the reference methods for two activity classes when 50% training data were used.

Calculation of BA for the prediction of MMP-Cliffs and MMP-nonCliffs gave similar results for all methods (Fig. 8). The level of 80% BA was generally reached for 25% or 50% training data. For largest training sets, all methods were comparably accurate for two activity classes, SVM reached highest accuracy for one class, and DeepAC for another (Fig. 8). Compared to the other methods, DeepAC produced higher TPR and lower TNR values,

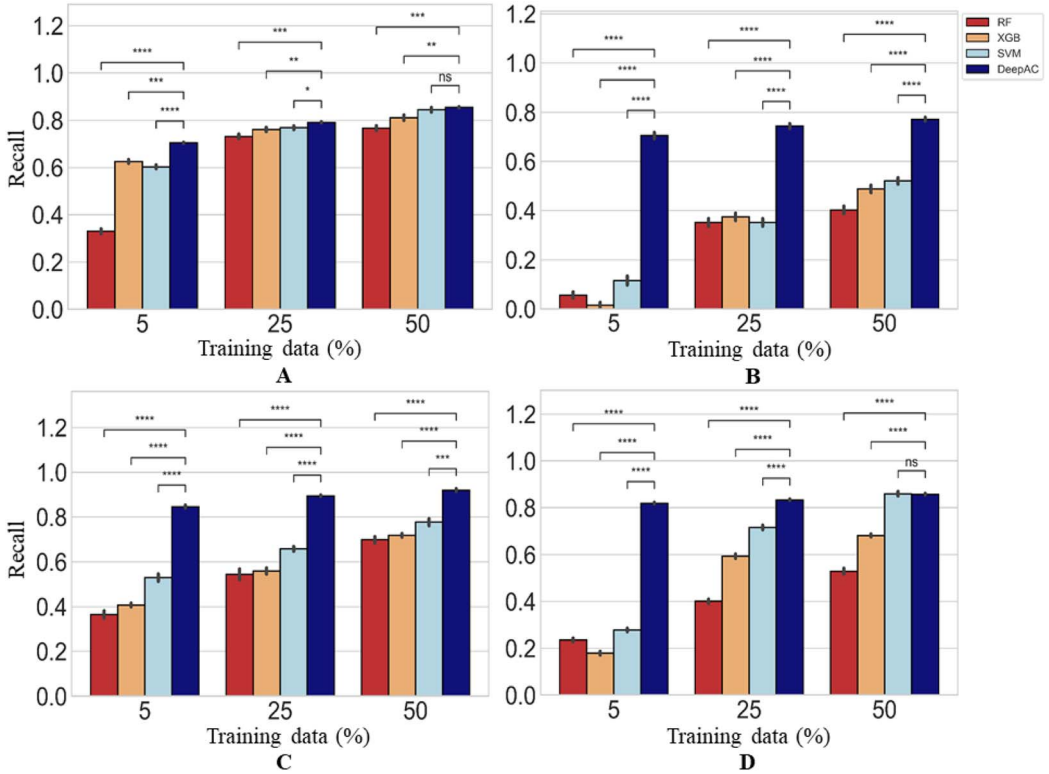


Fig. 6 Recall of MMP-Cliffs. For four different methods, recall/reproducibility of MMP-Cliffs is reported for (A) thrombin inhibitors, (B) Mu opioid receptor ligands, (C) corticotropin releasing factor receptor 1 ligands, and (D) tyrosine kinase Abl inhibitors. Average recall over five independent trials is reported for increasing amounts of training data randomly selected from the complete data set (error bars indicate standard deviations). Statistical tests are shown according to Fig. 4.

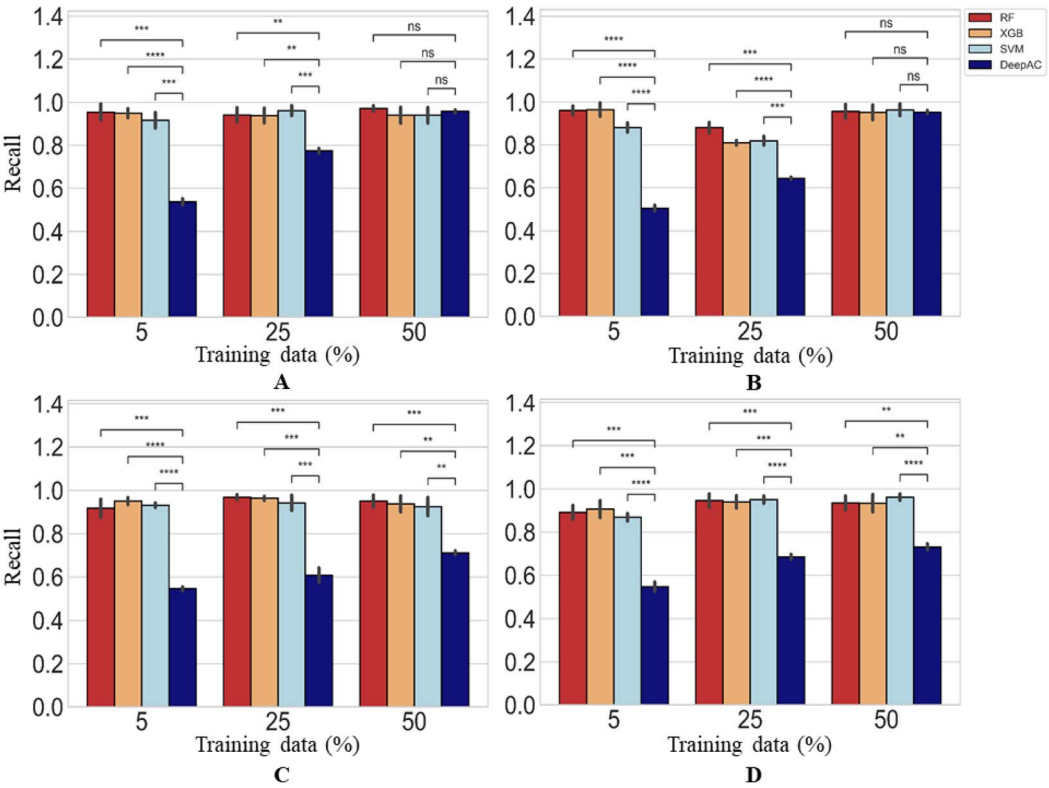


Fig. 7 Reproducibility of MMP-nonCliffs. In (A)–(D), reproducibility of MMP-nonCliffs is reported using four different methods. Statistical tests are shown according to Fig. 4.

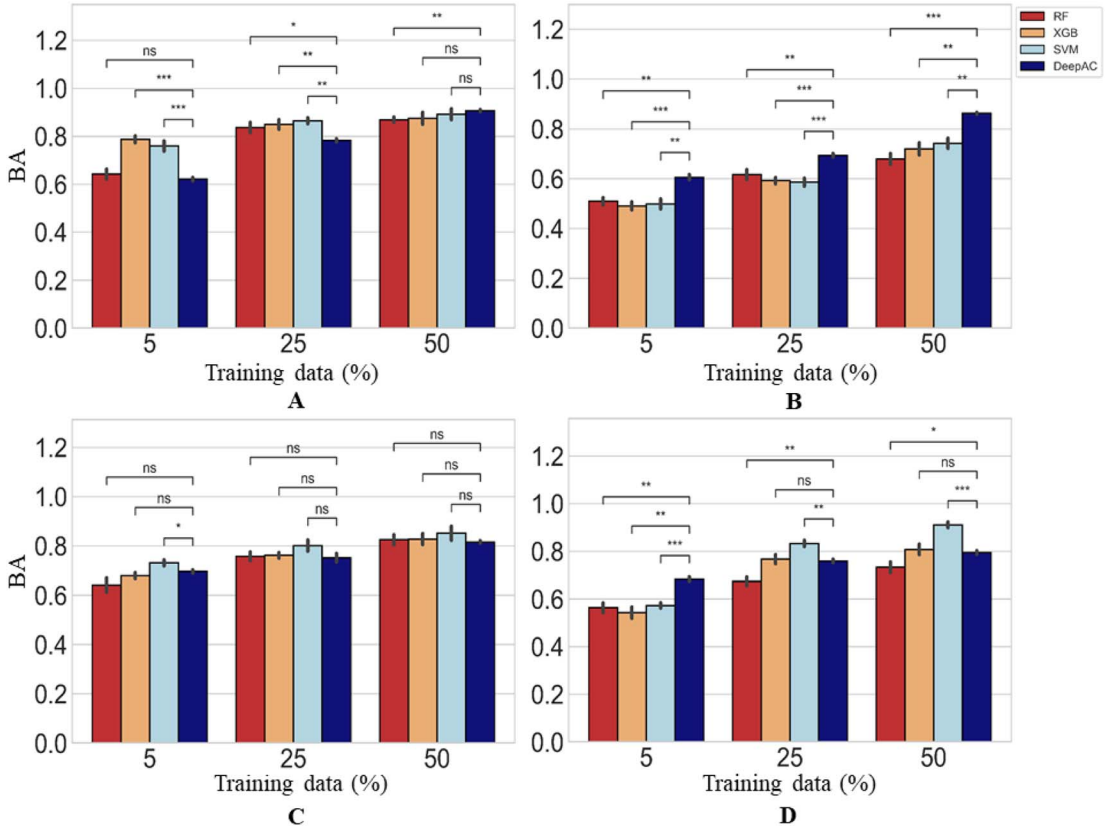


Fig. 8 Prediction accuracy. Reported are mean BA values and standard deviation (error bars) for the prediction of MMP-Cliffs and MMP-nonCliffs. In (A)–(D), results are reported using four different methods and statistical tests according to Fig. 4.

resulting in overall comparable BA. Clearly, a major strength of DeepAC was the ability to accurately predict MMP-Cliffs on the basis of small training data sets.

3.7 Activity cliff predictions in context

As discussed above, AC predictions have been reported previously in independent studies, which are summarized (and ordered chronologically) in Table 5. In 2012, AC predictions with SVM and newly designed MMP kernels yielded high accuracy,[9] which was also achieved in several subsequent studies using modified SVM approaches (Table 5). In addition, in our current study, we have investigated decision tree methods for AC predictions using molecular representations adapted from SVM, which yielded comparably high accuracy. Hence, although AC predictions are principally challenging, for reasons discussed above, different machine learning methods have produced high-quality models for different compound data sets. Accordingly, there would be little incentive to investigate increasingly complex models for AC predictions. Nonetheless, recent studies have investigated deep learning approaches for AC predictions, with different specific aims. These investigations included the use of convolutional neural networks for predicting ACs from image data[14,15] and the use of graph neural networks for AC representation learning.[16] While these studies provided proof-of-concept for the utility of novel methodologies for AC predictions, improvements in prediction accuracy compared to SVM in earlier studies have been marginal at best. The first eight studies in Table 5 report classification models of varying complexity for AC prediction. While most of these studies applied the MMP-Cliff formalism, their system set-ups, calculation conditions, and test cases differed such that prediction accuracies can only be globally compared and put into perspective including our current study. Furthermore, the last two studies[17,18] in Table 5 report regression models for potency prediction of individual AC compounds that are distinct from the others, precluding comparison of the results (these studies also used different AC definitions). However, they are included for completeness.

With DeepAC, we have introduced the use of conditional chemical language models for AC prediction. Given that most studies in Table 5 reported F1 (ref. 38) and Matthews' correlation coefficient (MCC)[39] scores for evaluating prediction accuracies, we also calculated these scores for the DeepAC predictions reported herein. With F1 of 0.50–0.78 and MCC of 0.43–0.75, DeepAC also yielded state-of-the-art prediction accuracy (and higher accuracy than recent AC predictions using

Table 5 Activity cliff predictions[a]

Study	AC criteria, similarity/potency difference	Prediction task	Methods	Prediction accuracy
Heikamp *et al.*[9]	MMP/100-fold	ACs for 9 activity classes	Fingerprint-based SVM with MMP kernels	F1: 0.70–0.99
Husby *et al.*[13]	Binding mode similarity (80%)/100-fold	3D-ACs for 9 activity classes	Docking/VLS	AUC: 0.75–0.97
Horvath *et al.*[10]	MMP/100-fold	ACs for 7 activity classes	CGR and descriptor recombination-based SVM/SVR	F1: 0.61–0.92
Tamura *et al.*[12]	MMP/100-fold	ACs for 9 activity classes	Fingerprint-based SVM with Tanimoto kernel	MCC: ∼0.20–0.80
Iqbal *et al.*[14]	MMP/100-fold	ACs from MMP images and R-groups (5 activity classes)	Image-based CNN with transfer learning	F1: 0.28–0.76 MCC: 0.24–0.73
Iqbal *et al.*[15]	MMP/100-fold	ACs from MMP images (3 activity classes)	Image-based CNN	F1: 0.36–0.85 AUC: 0.92–0.97 MCC: 0.39–0.83
Tamura *et al.*[11]	MMP/100-fold	ACs for 2 activity classes	Fingerprint-based SVM with MMP kernel	AUC: 0.46–0.69 MCC: 0.69–0.89
Park *et al.*[16]	MMP/100-fold	ACs for 3 activity classes	GCN	F1: 0.34–0.49 AUC: 0.91–0.94 MCC: 0.40–0.49
Jiménez-Luna *et al.*[17]	MCS/10-fold	—	RF/DNN/GRAPHNET/GCN/MPNN/GAT	RMSE: 0.698–1.029
Tilborg *et al.*[18]	Scaffold SMILES similarity (90%)/10-fold	ACs for 30 activity classes	KNN/RF/GBM/SVM/MPNN/GAT/GCN/AFP/LSTM/CNN/Transformer	RMSE: 0.62–1.60

[a] Abbreviations: SVM/R (support vector machine/regression); F1 (mean F1 score); AUC (area under the ROC curve); MCC (Matthews' correlation coefficient); 3D-ACs (three-dimensional activity cliffs); VLS (virtual ligand screening); CGR (condensed graphs of reaction); CNN (convolutional neural network); MCS (maximum common substructure); RF (random forest); DNN (deep neural network); GCN (graph convolutional network); MPNN (message passing neural network); GAT (graph attention network); RMSE (root mean square error); KNN (K-nearest neighbor); GBM (gradient boosting machine); AFP (attentive fingerprint); LSTM (long short-term memory network).

graph neural networks[16]). However, DeepAC is principally distinguished from other AC predictions approaches by its ability to generate new compounds meeting AC criteria, which partly motivated its development.

4 Conclusion

In this work, we have investigated chemical language models for predictive modeling of ACs, a topical issue in both chemical informatics and medicinal chemistry, with high potential for practical applications. ACs are rich in SAR information and represent focal points of compound optimization efforts. For chemical language models, an encoding strategy was devised to predict target compounds from source compounds and associated potency differences. Seq2Seq and transformer models were pre-trained on pairs of structural analogues with varying potency differences representing true SARs and compared, revealing superior performance of the transformer architecture in reproducing test compound pairs. The pre-trained transformer was then fine-tuned on ACs and non-ACs from different activity classes. It was conclusively shown that the transformer learned structural relationships in combination with associated potency differences and thus accounted for SARs. Compared to reference methods, the conditional transformer (DeepAC) reached state-of-the-art prediction accuracy but displayed different prediction characteristics. DeepAC was less effective in predicting non-ACs, but predicted ACs with higher accuracy than reference methods, especially on the basis of small training data sets. A unique feature of DeepAC is its ability to generate novel candidate compounds. This ability and the observed prediction characteristics render DeepAC attractive for practical applications aiming to generate new highly-potent AC compounds, which will be investigated in future studies.

Data availability

All calculations were carried out using publicly available programs, computational tools, and compound data. Python scripts used for implementing chemical language models and curated activity classes used for AC predictions are freely available *via* the following link: **https://doi.org/10.5281/zenodo.7153115**

Author contributions

All authors contributed to designing and conducting the study, analyzing the results, and preparing the manuscript.

Conflicts of interest

There are no conflicts of interest to declare.

Acknowledgements

H. C. is supported by the China Scholarship Council.

References

1 G. M. Maggiora, *J. Chem. Inf. Model.*, 2006, **46**, 1535.
2 D. Stumpfe, Y. Hu, D. Dimova and J. Bajorath, *J. Med. Chem.*, 2014, **57**, 18–28.
3 D. Stumpfe, H. Hu and J. Bajorath, *ACS Omega*, 2019, **4**, 14360–14368.
4 D. R. Flower, *J. Chem. Inf. Comput. Sci.*, 1998, **38**, 379–386.
5 J. Hussain and C. Rea, *J. Chem. Inf. Model.*, 2010, **50**, 339–348.
6 H. Hu, D. Stumpfe and J. Bajorath, *Future Med. Chem.*, 2019, **11**, 379–394.
7 D. Stumpfe and J. Bajorath, *Future Med. Chem.*, 2015, **7**, 1565–1579.
8 R. Guha, *J. Chem. Inf. Model.*, 2012, **2**, 2181–2191.
9 K. Heikamp, X. Hu, A. Yan and J. Bajorath, *J. Chem. Inf. Model.*, 2012, **52**, 2354–2365.
10 D. Horvath, G. Marcou, A. Varnek, S. Kayastha, A. de la Vega de León and J. Bajorath, *J. Chem. Inf. Model.*, 2016, **56**, 1631–1640.
11 S. Tamura, S. Jasial, T. Miyao and K. Funatsu, *Molecules*, 2021, **26**, 4916.
12 S. Tamura, T. Miyao and K. Funatsu, *Mol. Inf.*, 2020, **39**, 2000103.
13 J. Husby, G. Bottegoni, I. Kufareva, R. Abagyan and A. Cavalli, *J. Chem. Inf. Model.*, 2015, **55**, 1062–1076.
14 J. Iqbal, M. Vogt and J. Bajorath, *Artif. Intell. Life Sci.*, 2021, **1**, 100022.
15 J. Iqbal, M. Vogt and J. Bajorath, *J. Comput.-Aided Mol. Des.*, 2021, **35**, 1157–1164.
16 J. Park, G. Sung, S. Lee, S. Kang and C. Park, *J. Chem. Inf. Model.*, 2022, **62**, 2341–2351.
17 J. Jiménez-Luna, M. Skalic and N. Weskamp, *J. Chem. Inf. Model.*, 2022, **62**, 274–283.
18 D. van Tilborg, A. Alenicheva and F. Grisoni, *Exposing the limitations of molecular machine learning with activity cliffs*, ChemRxiv preprint, 2022.
19 A. Gaulton, A. Hersey, M. Nowotka, A. P. Bento, J. Chambers, D. Mendez, P. Mutowo, F. Atkinson, L. J. Bellis, E. Cibrián-Uhalte, M. Davies, N. Dedman, A. Karlsson, M. P. Magarinos, J. P. Overington, G. Papadatos, I. Smit and A. R. Leach, *Nucleic Acids Res.*, 2017, **45**, D945–D954.
20 D. Weininger, *J. Chem. Inf. Comput. Sci.*, 1988, **28**, 31–36.
21 X. Hu, Y. Hu, M. Vogt, D. Stumpfe and J. Bajorath, *J. Chem. Inf. Model.*, 2012, **52**, 1138–1145.
22 I. Sutskever, O. Vinyals and Q. V. Le, *Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
23 J. He, H. You, E. Sandström, E. Nittinger, E. J. Bjerrum, C. Tyrchan, W. Czechtizky and O. Engkvist, *J. Cheminf.*, 2021, **13**, 1–7.
24 M.-T. Luong, H. Pham and C. D. Manning, *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1412–1421.
25 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, *Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
26 V. Nair and G. E. Hinton, *ICML*, 2010, pp. 807–814.
27 J. Ba, J. R. Kiros and G. E. Hinton, arXiv preprint arXiv:1607.06450, 2016.
28 K. He, X. Zhang, S. Ren and J. Sun, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 770–778.
29 G. Landrum, *RDkit: Open-source cheminformatics*, 2006.
30 A. Aszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai and S. Chintala, *Adv. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 8026–8037.
31 V. N. Vapnik, *The nature of statistical learning theory*, Springer, New York, 2000.
32 L. Breiman, *Mach. Learn.*, 2001, **45**, 5–32.
33 T. Chen and C. Guestrin, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, p. 785794.
34 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay and G. Louppe, *J. Mach. Learn. Res.*, 2011, **12**, 2825–2830.
35 D. Rogers and M. Hahn, *J. Chem. Inf. Model.*, 2010, **50**, 742–754.
36 J. Bergstra, B. Komer, C. Eliasmith, D. Yamins and D. Cox, *Comput. Sci. Discovery*, 2015, **8**, 014008.
37 K. H. Brodersen, C. S. Ong, K. E. Stephan and J. M. Buhmann, *Proceedings of the 20th International Conference on Pattern Recognition (ICPR)*, 2010, pp. 3121–3124.
38 C. J. Van Rijsbergen, *Information retrieval*, Butterworth-Heinemann, Oxford, 1979.
39 B. W. Matthews, *Biochim. Biophys. Acta, Protein Struct.*, 1975, **405**, 442–451.

Appendix B

Designing Highly Potent Compounds using a Chemical Language Models

scientific reports

Check for updates

OPEN

Designing highly potent compounds using a chemical language model

Hengwei Chen & Jürgen Bajorath✉

Compound potency prediction is a major task in medicinal chemistry and drug design. Inspired by the concept of activity cliffs (which encode large differences in potency between similar active compounds), we have devised a new methodology for predicting potent compounds from weakly potent input molecules. Therefore, a chemical language model was implemented consisting of a conditional transformer architecture for compound design guided by observed potency differences. The model was evaluated using a newly generated compound test system enabling a rigorous assessment of its performance. It was shown to predict known potent compounds from different activity classes not encountered during training. Moreover, the model was capable of creating highly potent compounds that were structurally distinct from input molecules. It also produced many novel candidate compounds not included in test sets. Taken together, the findings confirmed the ability of the new methodology to generate structurally diverse highly potent compounds.

Compound design is one of the major tasks for computational approaches in medicinal chemistry. The primary aim is the generation of compounds with desired properties, first and foremost, compounds with activity against individual pharmaceutical targets and high potency. For compound design and potency predictions, a variety of computational methods have been developed or adapted. Mainstays include quantitative structure–activity relationship (QSAR) analysis[1] for the design of increasingly potent analogues of active compounds and methods for ligand- or structure-based virtual screening[2,3] to identify new hits. Ligand- and structure-based methods have different requirements. For example, for docking calculations[4], a variety of scoring functions have been developed to evaluate the quality and strength of receptor-ligand interactions and estimate binding energies[5,6]. For the structure-based prediction of relative potencies of congeneric compounds, free energy perturbation methods have been introduced[7,8]. At the ligand level, machine learning (ML) methods are widely used for hit identification and non-linear QSAR modeling[9]. For potency prediction, support vector regression (SVR)[10] has become a standard ML approach. Furthermore, for both computational compound screening and potency prediction, deep neural network (DNN) architectures are also increasingly investigated[11–13]. Recently, a methodological framework was developed for evaluating the performance of deep generative models and a recurrent neural network (RNN) was used to explore predictions based on sparse training data[14]. However, the analysis mainly focused on physicochemical properties. For potency prediction, the assessment and comparison of different methods typically relies on the use of standard benchmark settings. Such benchmark calculations are required but not sufficient to evaluate potency prediction methods and their potential for practical applications. Moreover, such calculations should be considered with caution. Notably, in benchmark settings, nearest neighbor analysis and mean or median value regression often meet the accuracy of increasingly complex ML methods[15]. The high performance of these simple reference methods is supported by potency value distributions in commonly used compound data sets[15]. In addition, narrow error margins separating ML-based and randomized potency value predictions limit conclusions that can be drawn from conventional benchmarking[15]. Such findings call for alternatives to conventional benchmarking such as focusing predictions on the most potent data set compounds, consistent with the final goal of compound optimization efforts.

While potency predictions are mostly carried out for individual compounds, they can also be applied to assess potency differences in compound pairs such as activity cliffs (ACs), which are formed by structurally similar (analogous) active compounds with large differences in potency[16]. In principle, ACs can be predicted by explicitly calculating potency differences between compounds in pairs or by distinguishing between ACs and other pairs of analogues using classification methods, which implicitly accounts for potency differences of varying magnitude.

Department of Life Science Informatics and Data Science, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Friedrich-Hirzebruch-Allee 5/6, 53115 Bonn, Germany. ✉email: bajorath@bit.uni-bonn.de

Previously, we have reported a deep learning approach for the prediction of ACs that further extended other ML classification methods by its ability to not only predict ACs, but also generate new AC compounds[17]. Since ACs encode large potency differences, we have reasoned that this methodology might be adapted and further extended for the design of highly potent compounds. Therefore, in this work, we have devised and implemented a chemical language model (CLM) for the prediction of highly potent compounds from weakly potent ones used as input. These predictions do not depend on conventional benchmark settings and are thus not affected by their intrinsic limitations.

Methods

Compounds, activity data, and analogue series. From ChEMBL (release 29)[18], bioactive compounds with high-confidence activity data were assembled. Only compounds with reported direct interactions (assay relationship type: "D") with human targets at the highest assay confidence level (assay confidence score 9) were considered. As potency measurements, only numerically specified equilibrium constants (K_i values) were accepted and recorded as (negative logarithmic) pK_i values. If multiple measurements were available for the same compound, the geometric mean was calculated as the final potency annotation, provided all values fell within the same order of magnitude; otherwise, the compound was disregarded. Qualifying compounds were organized into target-based activity classes. A total of 496 activity classes were obtained.

For each activity class, a systematic search for analogue series (ASs) was conducted using the compound-core relationship (CCR) method[19], which uses a modified matched molecular pair (MMP) fragmentation procedure[20] based on retrosynthetic rules[21] to systematically identify ASs with single or multiple (maximally five) substitution sites. The core structure of an AS was required to consist of at least twice the number of non-hydrogen atoms of the combined substituents[19].

Ultimately, 10 classes comprising ligands of different G protein coupled receptors were extracted as test cases for compound predictions that each contained more than 900 compounds and more than 100 analogue series. Table 1 summarizes the targets and composition of these activity classes (first four columns from the left) and Fig. 1 shows exemplary ASs with single or multiple substitution sites.

ChEMBL ID	Target name	Compounds	ASs	CCR pairs	AC-CCR pairs
218	Cannabinoid CB1 receptor	1118	250	8889	585
226	Adenosine A1 receptor	1924	318	18,623	1207
233	Mu opioid receptor	1216	169	10,430	1110
234	Dopamine D3 receptor	1529	213	21,008	755
237	Kappa opioid receptor	940	129	19,277	2897
251	Adenosine A2a receptor	1825	312	16,084	870
256	Adenosine A3 receptor	2033	434	42,621	6219
3371	Serotonin 6 receptor	1535	201	36,735	2485
4792	Orexin receptor 2	1133	131	12,368	1271
5113	Orexin receptor 1	1086	155	23,169	817

Table 1. Activity classes.

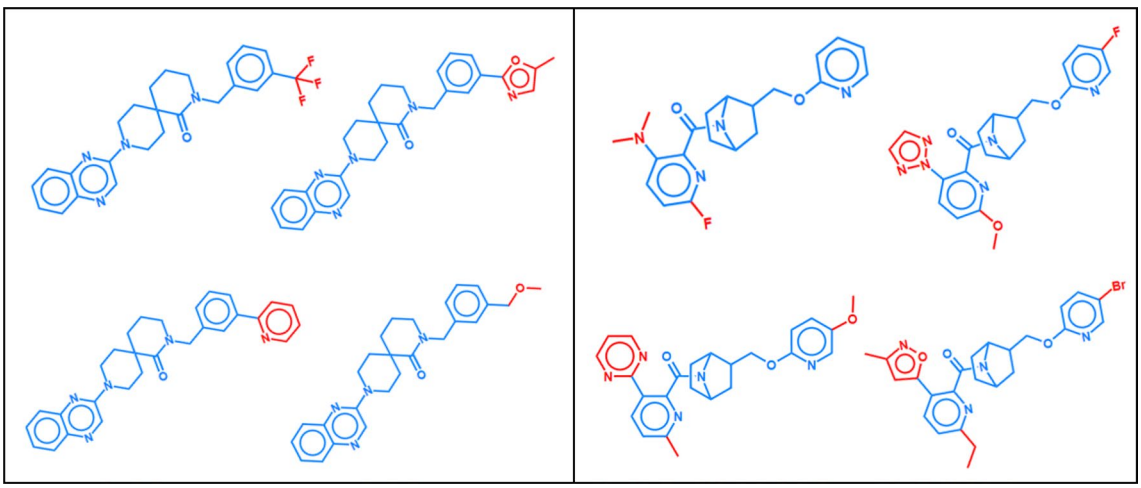


Figure 1. Exemplary analogue series. Shown are small ASs with single (left) or multiple substitution sites (right). Core structures are colored blue and substituents red.

For each of 10 activity classes, the number of compounds, ASs, CCR pairs, and AC-CCR pairs are provided. In addition, for each class, the ChEMBL target ID, target name, and abbreviation are given. AS, CCR, and AC stand for analogue series, compound-core relationship, and activity cliff, respectively.

From each of the activity classes, all possible pairs of analogues (termed *All_CCR* pairs) were extracted, as illustrated in Fig. 2 that shows All_CCR pairs for two different ASs. The 496 activity classes yielded a total of 881,990 All_CCR pairs.

Tokenization. For use by a CLM, compounds and potency differences must be tokenized. All compounds were represented as molecular-input line-entry system (SMILES) strings[22] generated using RDKit[23] and tokenized using a single chemical character with the exception of two-character tokens (i.e., "Cl" and "Br") and tokens in brackets (e.g. "[nH]" and "[O-]"). For the conditional transformer, potency differences must also be transformed into input tokens. For tokenization of value ranges, different approaches have been introduced including binning[17,24,25] and, more recently, numerical tokenization[26]. Since human readability of token sequences supported by numerical approaches played no role for our analysis and encoding of drug discovery-relevant compound potency ranges via binning has yielded accurate predictions previously[17], we continued to use binned tokens herein. Accordingly, potency differences between source and target compounds, ranging from −6.62 to 6.52 pK_i units, were partitioned into 1314 binned tokens of a constant width of 0.01. This granularity (resolution) defines the limits of experimental potency measurements and was thus most appropriate for our analysis. Each bin was encoded by a single token and each potency difference was assigned to the token of the corresponding bin[17].

Tokenization of compound SMILES strings and potency ranges yielded the chemical vocabulary for our model. In addition, the two special tokens "start" and "end" were added to the vocabulary indicating the start and end point of a sequence, respectively.

Generative chemical language model. *Architecture.* For compound design, a CLM with the transformer architecture previously reported for the DeepAC approach for AC prediction[17] was used. The transformer architecture consisted of multiple encoder-decoder neural modules with attention mechanism[27]. In the model, a stack of encoding sub-layers including a multi-head self-attention sub-layer and a fully connected feed-forward network sub-layer constituted the encoder module. The encoder read an input sequence and compressed it into a context vector in its final hidden state. The context vector served as the input for the decoder block that interpreted the vector to predict an output sequence. Subsequently, the decoder module, which was composed of a feed-forward sub-layer and two multi-head attention sub-layers, re-converted the encodings into a sequence of tokens (one token at a time). Both encoder and decoder utilized the attention mechanism during training to comprehensively learn from feature space.

During pre-training, the model was supposed to learn mappings of source to target compounds based on potency differences resulting from changes in substituent(s) (termed chemical transformations):

$$(\textit{Source compound},\ \textit{Potency difference}) \rightarrow (\textit{Target compound}).$$

Then, given a new *(Source compound, Potency difference)* test instance, the model was applied to generate a set of candidate target compounds meeting the potency difference constraints, that is, having higher potency than the source compound (according to the given potency difference).

During pre-training, distinguishing between different activity classes was not required because at this stage, the model should learn the syntax of textual molecular representations and, in addition, a variety of analogue

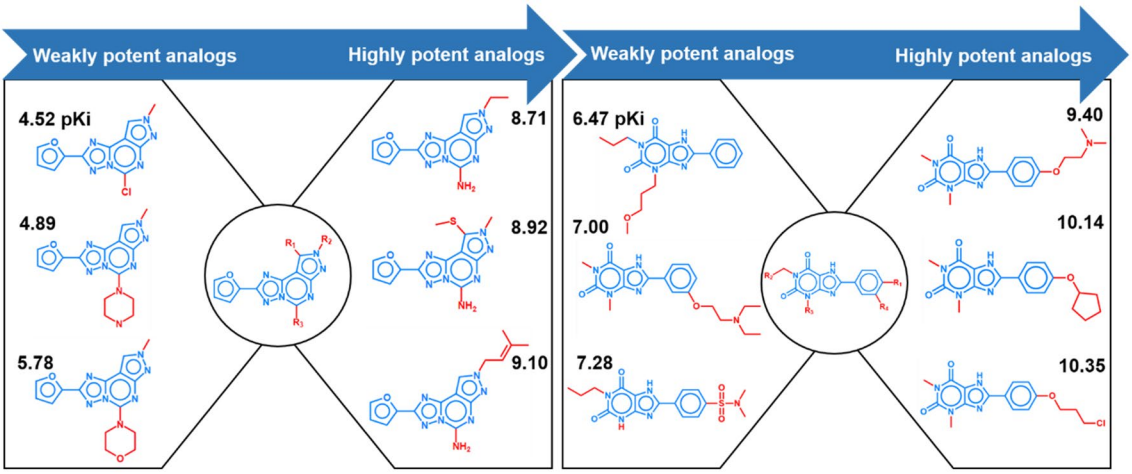


Figure 2. Analogue pairs. For each of two exemplary ASs, three representative All_CCR pairs are shown (top, middle, and bottom; increasing potency from the left to the right). The Markush structure representing each AS is displayed in the center. Core structures are colored blue and substituents red. For each compound, its pK_i value is reported.

pair-associated potency differences caused by chemical transformations. By contrast, during fine-tuning, activity class (target) information was required to focus the model on specific compound series or classes, as further discussed below.

Model derivation. The transformer model was implemented using Pytorch[28]. Default hyperparameter settings were used together with a batch size of 64, learning rate of 0.0001, and encoding dimension of 256. The models were derived over 200 epochs on the basis of the general training set. During training, the transformer model minimized the cross-entropy loss between the ground-truth and output sequence. A checkpoint was saved at each epoch and for a validation set, minimal loss was determined for selecting the final model.

Model pre-training. A general data set for model pre-training was derived from the 881,990 All_CCR pairs of the 496 activity classes. From All_CCR pairs, All_CCR triples *(Cpd_A, Cpd_B, Pot_B-Pot_A)* were generated by recording the potency difference for an All_CCR pair. Here, Cpd_A represented the *source compound* that was concatenated with the potency difference (Pot_B-Pot_A) and Cpd_B represented the *target compound*. For each All_CCR pair, two triples were obtained such that each All_CCR compound was used once as the source and target compound. To avoid data ambiguities, All_CCR pairs were eliminated if (1) a given source compound and potency difference was associated with multiple target compounds from different activity classes or (2) multiple potency values from different classes were available for a pair. On the basis of these criteria, a curated general data set of 522,331 qualifying All_CCR triples was obtained and used for pre-training.

For each triple, the SMILES representation of the source compound concatenated with the binned token of the associated potency difference served as the input sequence for the encoder that was converted into a latent representation. Based on this representation, the decoder iteratively generated output SMILES sequences until the end token was detected.

Model fine-tuning. For model fine-tuning and evaluation, the 10 activity classes in Table 1 were used. For fine-tuning, All_CCR pairs were extracted from each of the 10 activity classes and divided into subsets of so-called *CCR* pairs with a less than 100-fold potency difference and *AC-CCR* pairs capturing an at least 100-fold difference in potency. Accordingly, AC-CCR pairs represented analogue pairs forming ACs. Depending on the activity class, 8889–42,621 CCR pairs and 585–6219 AC-CCR pairs were obtained (Table 1, last two columns on the right). AC-CCR triples were ordered such that Cpd_B was highly and Cpd_A weakly potent.

The pre-trained model was then separately fine-tuned and tested for each activity class. Therefore, AC-CCR pairs from each class were randomly divided into 80% fine-tuning and 20% test instances such that there was no overlap in core structures between these sets. Thus, the fine-tuning set exclusively consisted of AC-CCR pairs and was selected to train the model on activity class dependent analogue pairs with large potency differences. CCR pairs sharing core structures with the fine-tuning set were omitted from further consideration. The remaining CCR pairs were added to the test set. Hence, the fine-tuning and test sets were structurally distinct. Model evaluation is detailed below.

Results

Study concept. Our study had three primary goals. First, we aimed to devise a novel approach specifically for predicting highly potent compounds from weakly potent input molecules. Thus, rather than striving for prediction of potency values across large ranges, as is conventionally attempted using SVR or other machine learning methods, the primary focus was on potent compounds, in line with the practical relevance of potency predictions. Second, we aimed to generate a structural spectrum of output compounds, ranging from analogues of input molecules to structurally distinct compounds, thereby increasing medicinal chemistry novelty of predicted candidates. Third, it was intended to evaluate the methodology in a way that was not affected by limitations of conventional benchmarking of potency predictions, as discussed above, and enabled a non-ambiguous assessment of the ability to predict potent compounds. To meet the first two goals, which were central to our study, we implemented a CLM consisting of a chemical transformer architecture conditioned on compound potency differences. To meet the third goal, we designed a new compound test system.

Compound pair-based test system. For model evaluation, a compound pair-based test system was generated using the test set. By design, the fine-tuning and test sets were structurally distinct. Furthermore, in contrast to the fine-tuning set, the test set contained analogue pairs capturing small or large differences in potency (i.e., CCR and AC-CCR pairs, respectively). Table 2 summarizes the composition of the test set.

For each activity class, the test set contained varying numbers of CCR pairs and AC-CCR pairs yielding varying numbers of unique CCR and AC-CCR compounds. In the following, *SC* and *TC* are used as abbreviations for source (input) and target compound, respectively. For the evaluation of the fine-tuned CLM, test set compounds were divided into instances with maximally 1 μmol potency (corresponding to a pK_i value of 6), which served as SCs, and candidate compounds with higher than 1 μmol potency ($pK_i > 6$), which served as *known candidate compounds* (*KCCs*) for comparison with newly generated TCs.

In addition, the model generated varying numbers of novel (hypothetical) TCs. For each activity class, smaller numbers of SCs than KCCs were available. With the exception of activity class 251 (3838 KCCs), the test set contained 366–824 KCCs for the activity classes (Table 2), with on average 576 KCCs per class. Each CCR-SC ($pK_i \leq 6$) and AC-CCR-SC ($pK_i \leq 6$) was once used as an input compound for the model and in each case, 50 TCs were sampled, canonicalized, and compared to KCCs to search for exact matches, that is, fully reproduced compounds with known potency. Because the model generated novel TCs, probabilities for re-generating known TCs could not be derived in a meaningful way. Consequently, the main measure for establishing proof-of-principle

ChEMBL ID	CCR pairs	Unique CCR CPDs	AC-CCR pairs	Unique AC-CCR CPDs	Overlapping CPDs	Unique CCR + AC-CCR CPDs	SCs (pki ≤ 6)	KCCs (pki > 6)
218	2198	579	6	12	9	582	129	453
226	5950	1174	144	84	80	1178	359	819
233	2332	590	36	36	33	593	76	517
234	7790	913	50	53	53	913	89	824
237	1032	477	31	24	20	481	115	366
251	4706	5210	85	57	38	5229	1391	3838
256	5012	888	40	44	42	890	250	640
3371	4420	722	42	44	44	722	40	682
4792	1941	615	49	50	48	617	146	471
5113	7543	664	13	15	15	664	256	408

Table 2. Test set. CPD stands for compound, SC for source compound, and KCC for known candidate compound. According to our analysis scheme, target compounds (TCs) produced by the model were compared to KCCs.

for the ability of the model to predict potent compounds was the reproduction of *any KCCs*. For each activity class, compound statistics were derived over three independent sampling trials, as reported below.

Table 3 reports the possible predictions outcomes for the compound pair-based test system.

For each SC, a TC could be a known CCR or AC-CCR compound or a novel (hypothetical) compound representing a TC not contained in the fine-tuning or test set. Taking core structure matches into consideration (that is, a TC either contained the same core structure as a SC or not), a total of 12 formally defined prediction outcomes were possible, including six each for CCR-SCs and AC-CCR-SCs, as identified by indices 1.1.–1.6. and 2.1.–2.6. in Table 3, respectively. Accordingly, a newly generated compound might be a structural analogue of a given SC (having the same core structure) or contain a different core structure. Furthermore, SCs and TCs might be distinguished by single or multiple substituents. On the basis of this classification scheme, CLM predictions were rigorously evaluated focusing on the reproduction of known active compounds, as explained above. This was the most relevant measure of model performance because it enabled the exact determination of potency differences between SCs and TCs and hence the ability of the CLM to predict highly potent compounds. For novel (hypothetical) compounds generated by the model, no assessment was possible (without subsequent experimental evaluation).

Model performance. For the SCs from all activity classes, systematic compound predictions were carried out using the CLM. The model only produced 0.5–2% invalid SMILES (assessed using RDKit) for all activity classes.

With the exception of class 251 (1391 SCs), the test set contained 40–359 SCs for the activity classes, with on average 162 compounds per class (Table 2). The predictions were then assessed on the basis of well-defined pair categories detailed above, as reported in Table 4.

For each activity class and compound pair category indexed according to Table 3 (top row), the number of unique TCs produced by the CLM is reported. With the exception of categories 1.5., 1.6., 2.5., and 2.6., which report novel (hypothetical) candidate compounds not contained in the fine-tuning or test set, the TCs represent KCCs, as defined in the text.

Encouragingly, for all activity classes, the CLM successfully reproduced large numbers of KCCs for all SCs (categories 1.1.–1.4. and 2.1.–2.4., respectively). Frequently, multiple KCCs were obtained for the same SC. Furthermore, depending on the activity class, the model produced varying numbers of TCs with the same or different core structure, thus confirming its ability to generate frequent core structure transformations. In many cases, more structurally unique TCs were generated than analogues of SCs. Moreover, large numbers of hypothetical candidate compounds not contained in the training set were obtained (categories 1.5.–1.6. and 2.5.–2.6., respectively). The reproducibility of the limited numbers of available KCCs representing known ACs (12–84 unique compounds per activity class) was of particular interest (categories 2.1.–2.4.). AC-CCR KCCs were consistently reproduced and for five activity classes, the total count exceeded the number of unique AC-CCR KCCs per class (due to multiple reproductions of individual KCCs). Table 5 reports statistics for reproduction of KCCs.

Reported are statistics for the re-generation of KCCs including the mean number of KCCs over three independent sampling trials and the proportion of reproduced KCCs relative to all available KCCs with standard deviations (±). In addition, the mean number of non-KCCs over three independent trials is provided.

The proportion of exactly reproduced KCCs over independent sampling trials ranged from ~7 to ~37%, depending on the activity class (with generally small standard deviations). For nine, six, and two classes, more than 10, 20, and 30% of all available KCCs were reproduced, respectively. Applying the most rigorous criterion of exact re-generation of known potent compounds as a performance measure (see above), the observed numbers and proportions represented unexpectedly good predictions, which clearly established proof-of-concept for the approach.

For each activity class, ASs were also extracted from newly generated (predicted) compounds. Table 6 reports the number of ASs (multiple compounds having the same core structure) and singletons (compounds with

Index same/different core	Compound pair category
1.1./1.2.	(CCR-SC, CCR-TC)
1.3./1.4.	(CCR-SC, AC-CCR-TC)
1.5./1.6.	(CCR-SC, novel CPD)
2.1./2.2.	(AC-CCR-SC, AC-CCR-TC)
2.3./2.4.	(AC-CCR-SC, CCR-TC)
2.5./2.6.	(AC-CCR-SC, novel CPD)

Table 3. Possible predictions.

ChEMBL ID	1.1.	1.2.	1.3.	1.4.	1.5.	1.6.	2.1.	2.2.	2.3.	2.4.	2.5.	2.6.
218	73	192	2	5	436	3301	3	3	4	11	24	34
226	262	433	4	25	1067	5030	11	11	27	79	129	529
233	217	179	2	14	252	570	6	9	0	10	21	45
234	141	92	3	2	286	705	3	2	6	7	24	13
237	488	250	0	11	181	766	9	26	14	4	4	10
251	2367	1400	235	128	1031	13,523	17	5	36	13	55	199
256	112	66	1	2	657	5336	10	7	0	12	13	359
3371	60	116	0	4	42	1202	7	4	3	8	33	101
4792	224	662	7	42	253	1222	7	6	7	17	17	25
5113	433	349	1	5	304	1638	5	2	11	2	15	24

Table 4. Prediction results.

ChEMBL ID	KCCs	Non-KCCs	Reproduced KCCs (%)
218	103	3445	22.74 ± 1.10
226	211	5139	25.76 ± 0.49
233	143	1005	27.66 ± 1.35
234	92	825	11.17 ± 0.24
237	128	839	34.97 ± 1.37
251	251	4996	6.54 ± 0.29
256	76	5165	11.88 ± 0.63
3371	72	2145	10.56 ± 1.17
4792	172	1084	36.52 ± 1.91
5113	117	1499	28.68 ± 1.72

Table 5. Reproducibility of *known candidate compounds*.

ChEMBL ID	ASs	Singletons	Reproduced cores (%)
218	858	1235	4
226	905	1762	4
233	188	255	12
234	90	245	9
237	175	303	7
251	1304	978	4
256	1414	1386	7
3371	321	1022	4
4792	146	219	18
5113	233	440	9

Table 6. Structural organization of predicted compounds. "Reproduced cores" reports the percentage of the core structures contained in each original activity class that were detected in predicted compounds.

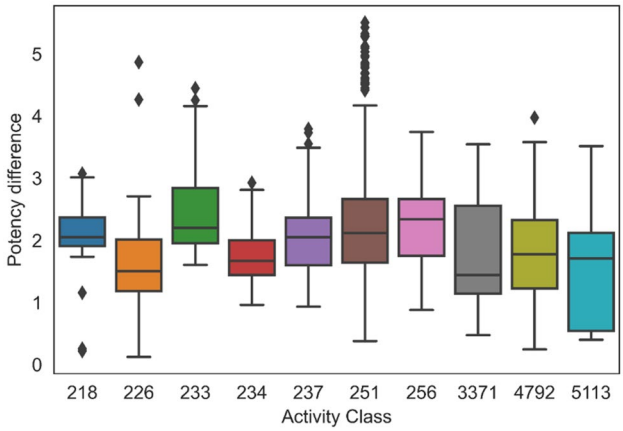


Figure 3. Potency difference distribution. For all activity classes, boxplots report the distributions of logarithmic potency differences between pairs of known source and target compounds involving compounds from ACs. In boxplots, the median value is represented by the horizontal line, and the box defines upper and lower quantile. Upper and lower whiskers represent the maximum and minimum value, respectively. Diamond symbols mark statistical outliers.

unique core structures not belonging to any AS). Depending on the activity class, 90–1414 ASs and 219–1762 singletons were obtained, respectively.

Since each AS and singleton contained a unique core structure (scaffold), the core structure diversity of newly generated compounds was generally high. Between 4 and 18% of the core structures contained in the original activity classes (from ASs and singletons) were reproduced by the model, as also reported in Table 6.

Having confirmed the ability of the CLM to generate structurally analogous and diverse TCs including KCCs, the key question then was whether or not the model would produce TCs that had much higher potency than the corresponding SCs. Figure 3 shows the distributions of potency differences between pairs of known source and target compounds with experimental potency values involving compounds from ACs. For five activity classes, the median potency difference fell between one and two orders of magnitude (10–100-fold) and for the other five classes, the median value exceeded two orders of magnitude (100-fold). Furthermore, for all but one class, multiple compounds with at least 1000-fold higher potency than the corresponding SCs were generated (including highly potent statistical outliers). Thus, these observations unambiguously confirmed the ability of the CLM to generate highly potent compounds from weakly potent (micromolar) input molecules.

Figure 4 shows exemplary pairs of SCs and newly designed compounds (TCs) with different structural relationships. Given our design strategy, all SCs were known compounds with experimentally determined potency. The generated TCs included known potent analogues of SCs (Fig. 4a), structurally distinct known potent compounds (Fig. 4b), and novel (hypothetical) compounds (Fig. 4c). Taken together, these examples illustrate successful CLM predictions.

Conclusion

The underlying idea for the development of the approach reported herein was to predict highly potent compounds from individual weakly potent input molecules. For all practical purposes, this represents an ultimate goal of potency prediction, especially for compound optimization in medicinal chemistry. This prediction task could not be addressed using conventional regression models. In addition, going beyond the applicability domain of standard QSAR modeling, we also aimed to design structurally diverse compounds, in addition to analogues. Therefore, a different methodological framework was required and we adapted a conditional transformer architecture previously used for AC predictions. These predictions established that compound generation could be conditioned on potency differences. However, since AC predictions were also confined to structurally analogous compounds, it remained unclear whether or not potency difference conditioning was transferable to the design of structurally diverse compounds with high potency. The CLM reported herein was fine-tuned on pairs of SCs and TCs with associated potency differences and we then examined its ability to predict structurally diverse compounds with large increases in potency relative to input molecules. Therefore, a compound pair-based test system was generated that covered all possible prediction outcomes and enabled a well-defined and rigorous assessment of model performance. Our analysis confirmed the ability of the model to reproduce known potent compounds not encountered during training at unexpectedly high rates, including both analogues of weakly potent SCs and structurally distinct compounds. With median potency increases close to or above 100-fold across activity classes and multiple predictions with more than 1000-fold increases in compound potency, model performance was generally high. In addition, the CLM also produced large numbers of novel compounds for the activity classes that were not contained in the fine-tuning or test set.

Taken together, our findings indicate that the approach reported herein should have considerable potential for practical applications. In compound optimization, we envision that the CLM will be fine-tuned using sets of

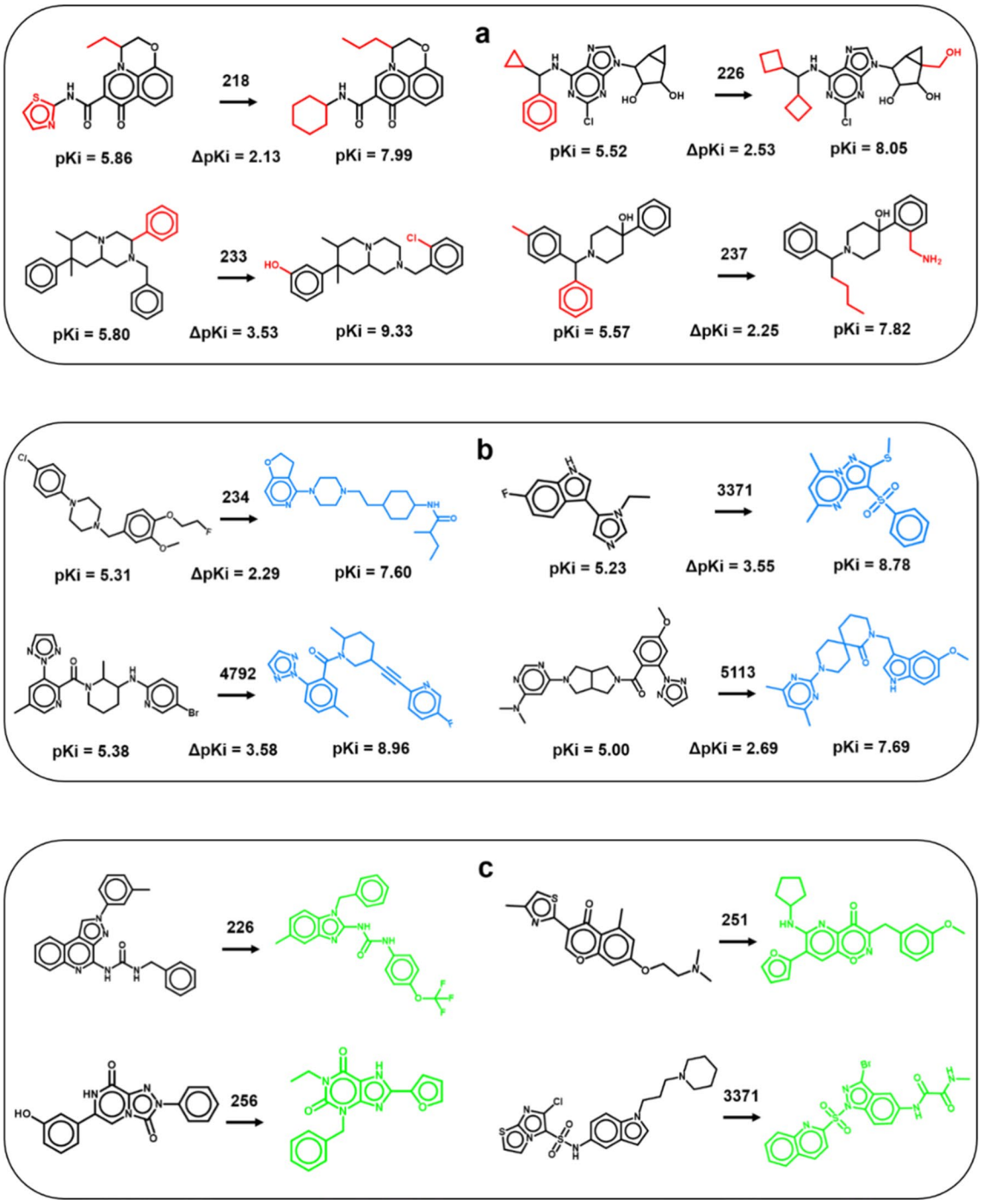


Figure 4. Exemplary predictions. Shown are pairs of corresponding source compounds (left of the arrow) and new compounds generated by the CLM (right) including (**a**) potent known compounds with conserved core structures (black, distinguishing substituents are red), (**b**) potent known compounds with distinct structures (blue), and (**c**) hypothetical compounds (green). For hypothetical compounds, no potency values were available. Numbers on arrows identify activity classes according to Table 1. Potency differences between SCs and KCCs are reported.

active compounds for a target of interest and that the predictions will then focus on input compounds prioritized by medicinal chemistry. For these and other applications, the CLM is made freely available as a part of our study.

Data availability

All calculations were carried out using publicly available programs and compound data. Python scripts used for implementing CLMs and the activity classes used herein are freely available via the following link: https://doi.org/10.5281/zenodo.7744763.

Received: 2 February 2023; Accepted: 5 May 2023
Published online: 07 May 2023

References

1. Lewis, R. A. & Wood, D. Modern 2D QSAR for drug discovery. *WIREs Comput. Mol. Sci.* **4**, 505–522 (2014).
2. Geppert, H., Vogt, M. & Bajorath, J. Current trends in ligand-based virtual screening: Molecular representations, data mining methods, new application areas, and performance evaluation. *J. Chem. Inf. Model.* **50**, 205–216 (2010).
3. Cheng, T., Li, Q., Zhou, Z., Wang, Y. & Bryant, S. H. Structure-based virtual screening for drug discovery: A problem-centric review. *AAPS J.* **14**, 133–141 (2012).
4. Pagadala, N. S., Syed, K. & Tuszynski, J. Software for molecular docking: A review. *Biophys. Rev.* **9**, 91–102 (2017).
5. Liu, J. & Wang, R. Classification of current scoring functions. *J. Chem. Inf. Model.* **55**, 475–482 (2015).
6. Guedes, I. A., Pereira, F. S. & Dardenne, L. E. Empirical scoring functions for structure-based virtual screening: Applications, critical aspects, and challenges. *Front. Pharmacol.* **9**, e1089 (2018).
7. Mobley, D. L. & Gilson, M. K. Predicting binding free energies: Frontiers and benchmarks. *Annu. Rev. Biophys.* **46**, 531–558 (2017).
8. Williams-Noonan, B. J., Yuriev, E. & Chalmers, D. K. Free energy methods in drug design: Prospects of "Alchemical perturbation" in medicinal chemistry. *J. Med. Chem.* **61**, 638–649 (2018).
9. Vamathevan, J. *et al.* Applications of machine learning in drug discovery and development. *Nat. Rev. Drug. Discov.* **18**, 463–477 (2019).
10. Smola, A. J. & Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **14**, 199–222 (2004).
11. Hou, F. *et al.* Comparison study on the prediction of multiple molecular properties by various neural networks. *J. Phys. Chem. A* **122**, 9128–9134 (2018).
12. Feinberg, E. N. *et al.* PotentialNet for molecular property prediction. *ACS Cent. Sci.* **4**, 1520–1530 (2018).
13. Walters, W. P. & Barzilay, R. Applications of deep learning in Molecule generation and molecular property prediction. *Acc. Chem. Res.* **54**, 263–270 (2020).
14. Skinnider, M. A., Stacey, R. G., Wishart, D. S. & Foster, L. J. Chemical language models enable navigation in sparsely populated chemical space. *Nat. Mach. Intell.* **3**, 759–770 (2021).
15. Janela, T. & Bajorath, J. Simple nearest-neighbour analysis meets the accuracy of compound potency predictions using complex machine learning models. *Nat. Mach. Intell.* **4**, 1246–1255 (2022).
16. Stumpfe, D., Hu, H. & Bajorath, J. Evolving concept of activity cliffs. *ACS Omega* **4**, 14360–14368 (2019).
17. Chen, H., Vogt, M. & Bajorath, J. DeepAC—Conditional transformer-based chemical language model for the prediction of activity cliffs formed by bioactive compounds. *Digital Discov.* **1**, 898–909 (2022).
18. Bento, A. P. *et al.* The CHEMBL bioactivity database: An update. *Nucleic Acids Res.* **42**, D1083–D1090 (2014).
19. Naveja, J. J., Vogt, M., Stumpfe, D., Medina-Franco, J. L. & Bajorath, J. Systematic extraction of analogue series from large compound collections using a new computational compound–core relationship method. *ACS Omega* **4**, 1027–1032 (2019).
20. Stumpfe, D., Dimova, D. & Bajorath, J. Computational method for the systematic identification of analog series and key compounds representing series and their biological activity profiles. *J. Med. Chem.* **59**, 7667–7676 (2016).
21. Lewell, X. Q., Judd, D. B., Watson, S. P. & Hann, M. M. RECAP - retrosynthetic combinatorial analysis procedure: A powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. *J. Chem. Inf. Comput. Sci.* **38**, 511–522 (1998).
22. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **28**, 31–36 (1988).
23. RDKit: Cheminformatics and Machine Learning Software. http://www.rdkit.org (accessed on 1 July 2021).
24. He, J. *et al.* Molecular optimization by capturing chemist's intuition using Deep Neural Networks. *J. Cheminform.* **13**, 26 (2021).
25. He, J. *et al.* Transformer-based molecular optimization beyond matched Molecular Pairs. *J. Cheminform.* **14**, 18 (2022).
26. Born, J. & Manica, M. Regression transformer enables concurrent sequence regression and generation for molecular language modelling. *Nat. Mach. Intell.* **5**, 432–444 (2023).
27. Vaswani, A. *et al.* Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30**, 6000–6010 (2017).
28. Aszke, A. *et al.* PyTorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32**, 8026–8037 (2019).

Acknowledgements

The authors thank Martin Vogt for many helpful suggestions. H.C. is supported by the China Scholarship Council (CSC).

Author contributions

All authors contributed to designing and conducting the study, analyzing the results, and preparing the manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to J.B.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Appendix C

Meta-Learning for Transformer-Based Prediction of Potent Compounds

scientific reports

Check for updates

OPEN

Meta-learning for transformer-based prediction of potent compounds

Hengwei Chen & Jürgen Bajorath✉

For many machine learning applications in drug discovery, only limited amounts of training data are available. This typically applies to compound design and activity prediction and often restricts machine learning, especially deep learning. For low-data applications, specialized learning strategies can be considered to limit required training data. Among these is meta-learning that attempts to enable learning in low-data regimes by combining outputs of different models and utilizing meta-data from these predictions. However, in drug discovery settings, meta-learning is still in its infancy. In this study, we have explored meta-learning for the prediction of potent compounds via generative design using transformer models. For different activity classes, meta-learning models were derived to predict highly potent compounds from weakly potent templates in the presence of varying amounts of fine-tuning data and compared to other transformers developed for this task. Meta-learning consistently led to statistically significant improvements in model performance, in particular, when fine-tuning data were limited. Moreover, meta-learning models generated target compounds with higher potency and larger potency differences between templates and targets than other transformers, indicating their potential for low-data compound design.

Predicting new active compounds is one of the major tasks in computer-aided drug discovery, for which machine learning approaches have been widely applied over the past two decades[1,2]. In recent years, deep learning has also been increasingly applied for compound activity and property predictions[1,2]. The prediction of compounds exhibiting a desired biological activity (that is, activity against a target of interest) is mostly attempted using machine learning models for binary classification (that is, a compound is predicted to have or not to have a specific activity)[3–5]. For this purpose, models for class label prediction (active versus inactive compounds) are typically derived based on training sets of known specifically active compounds and randomly selected compounds assumed to be inactive. These qualitative activity predictions mostly involve virtual screening of compound databases to identify new hits. In addition to qualitative predictions of biological activity, predicting compounds that are highly potent against a given target also is of interest. Compound potency prediction can be quantitative or semi-quantitative in nature. Quantitative predictions aim to specify numerical potency values using, for example, quantitative structure–activity relationship (QSAR)[6,7] or free energy methods[8,9]. Different from qualitative predictions and virtual screening, quantitative potency predictions are usually carried out for small compound sets or structural analogues from lead series. Furthermore, semi-quantitative approaches aim to predict new potent compounds, that is, compounds having higher potency than known actives. For example, such predictions might focus on activity cliffs[10], which are defined as pairs of structurally similar compounds or structural analogues with large potency differences[10]. Prediction of activity cliffs fall outside the applicability domain of standard QSAR methods[4].

While quantitative potency predictions are widely carried out, they are difficult to evaluate in benchmark settings. It has been observed that benchmark predictions of different machine learning models and randomized predictions are typically only separated by small error margins[11], which makes it difficult to non-ambiguously assess relative method performance[11]. Therefore, we currently prefer semi-quantitative approaches focusing on the prediction of potent compounds (rather than trying to predict compound potency values across wide potency ranges). Semi-quantitative predictions can be attempted by deep generative modeling[2]. For example, transformer models have been derived based on pairs of active structural analogues with varying potency to predict activity cliffs and design potent compounds[12,13]. Therefore, the transformer models were conditioned on observed potency differences. This generative design approach successfully reproduced highly potent compounds

Department of Life Science Informatics and Data Science, B-IT, Lamarr Institute for Machine Learning and Artificial Intelligence, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Friedrich-Hirzebruch-Allee 5/6, 53115 Bonn, Germany. ✉email: bajorath@bit.uni-bonn.de

for different activity classes based on weakly potent input compounds[13]. Transformer models have also been derived for other compound property predictions[14–16] and generative compound design applications[17–19] as well as for the prediction of drug-target interactions[20–22].

Notably, all compound activity and potency predictions depend on available data for learning. Like many other data in early-phase drug discovery, high-quality compound potency measurements for given targets are generally sparse, which limits generative design. Therefore, we are considering machine learning approaches for low-data regimes to enable predictions of potent compounds for targets, for which only little compound data is available. Among learning strategies for sparsely distributed data, active learning[23,24] and transfer learning[25,26] have been investigated for machine learning in drug discovery in various studies[24,26]. Transfer learning attempts to use information obtained from related prediction tasks to streamline model derivation for such tasks, while active learning focuses on the selection of most informative training instances for iterative model building. Meta-learning including few-shot learning represents another low-data approach that is relevant for drug discovery[27–30]. In artificial intelligence, meta-learning is a sub-discipline of machine learning[27]. It aims to combine the output of different machine learning models and/or meta-data from these models such as parameters derived from training instances to generate models for other prediction tasks[27]. Alternatively, the same algorithm might be applied to generate models for individual prediction tasks whose outputs are then used to iteratively update a meta-learning model. Hence, meta-learning can also be regarded as a form of ensemble learning. The general aim of meta-learning is achieving transferability of models to related prediction tasks, including the application of prior model knowledge to limit the number of training instances required for new tasks. Given the use of meta-data for learning, the approach is well-suited for parameter-rich deep learning architectures[28] and -compared to transfer learning- principally applicable to a wider spectrum of predictions tasks. However, in compound design and property prediction, the exploration of meta-learning is still in its early stages. Therefore, we have explored meta-learning in semi-quantitative potency predictions. To this end, we have adapted a transformer architecture designed for the prediction of potent compounds[13] as a base model for deriving meta-learning models and assessed the potential of meta-learning for predicting highly potent compounds for different activity classes and varying amounts of training data.

Methods

Compounds, activity data, and analogue series

Bioactive compounds with high-confidence activity data were collected from ChEMBL (release 29)[31]. Only compounds with direct interactions (assay relationship type: "D") with human targets at the highest assay confidence level (assay confidence score 9) were considered. In addition, potency measurements were restricted to numerically specified equilibrium constants (K_i values), which were recorded as (negative decadic logarithmic) pK_i values. When multiple measurements were available for the same compound, the geometric mean was calculated as the final potency annotation, provided all values fell within the same order of magnitude. If not, the compound was disregarded. Qualifying compounds were organized into target-based activity classes.

In activity classes, analogue series (AS) with one to five substitution sites were identified using the compound-core relationship (CCR) algorithm[32]. The core structure of an AS was required to consist of at least twice the number of non-hydrogen atoms as the combined substituents. For each AS, all possible pairs of analogues were generated, termed All_CCR pairs. For each activity class, ALL_CCR pairs from all AS were pooled. All_CCR pairs were then divided into CCR pairs with a potency difference of less than 100-fold and activity cliff (AC)-CCR pairs with a potency difference of at least 100-fold.

On the basis of the specified data curation criteria and AS distributions, 10 activity classes were assembled that consisted of at least ~ 500 qualifying compounds and ~ 50 AS, as summarized in Table 1. These activity classes included ligands of various G protein coupled receptors and inhibitors of different enzymes. Figure 1 shows exemplary AC_CCR pairs for each class.

ChEMBL ID	Target name	Compounds	AS	CCR pairs	AC-CCR pairs
226	Adenosine A1 receptor	1924	318	18,623	1207
234	Dopamine D3 receptor	1529	213	21,008	755
237	Kappa opioid receptor	940	129	19,277	2897
244	Coagulation factor X	702	92	9718	1288
251	Adenosine A2a receptor	1825	312	16,084	870
259	Melanocortin receptor 4	543	145	25,126	3086
264	Histamine H3 receptor	1235	173	10,812	532
1862	Tyrosine-protein kinase ABL	499	64	15,573	1873
2014	Nociceptin receptor	512	52	11,472	1058
4792	Orexin receptor 2	1133	131	12,368	1271

Table 1. Activity classes. The composition of activity classes is summarized. For each class, the ChEMBL target ID and target name are provided.

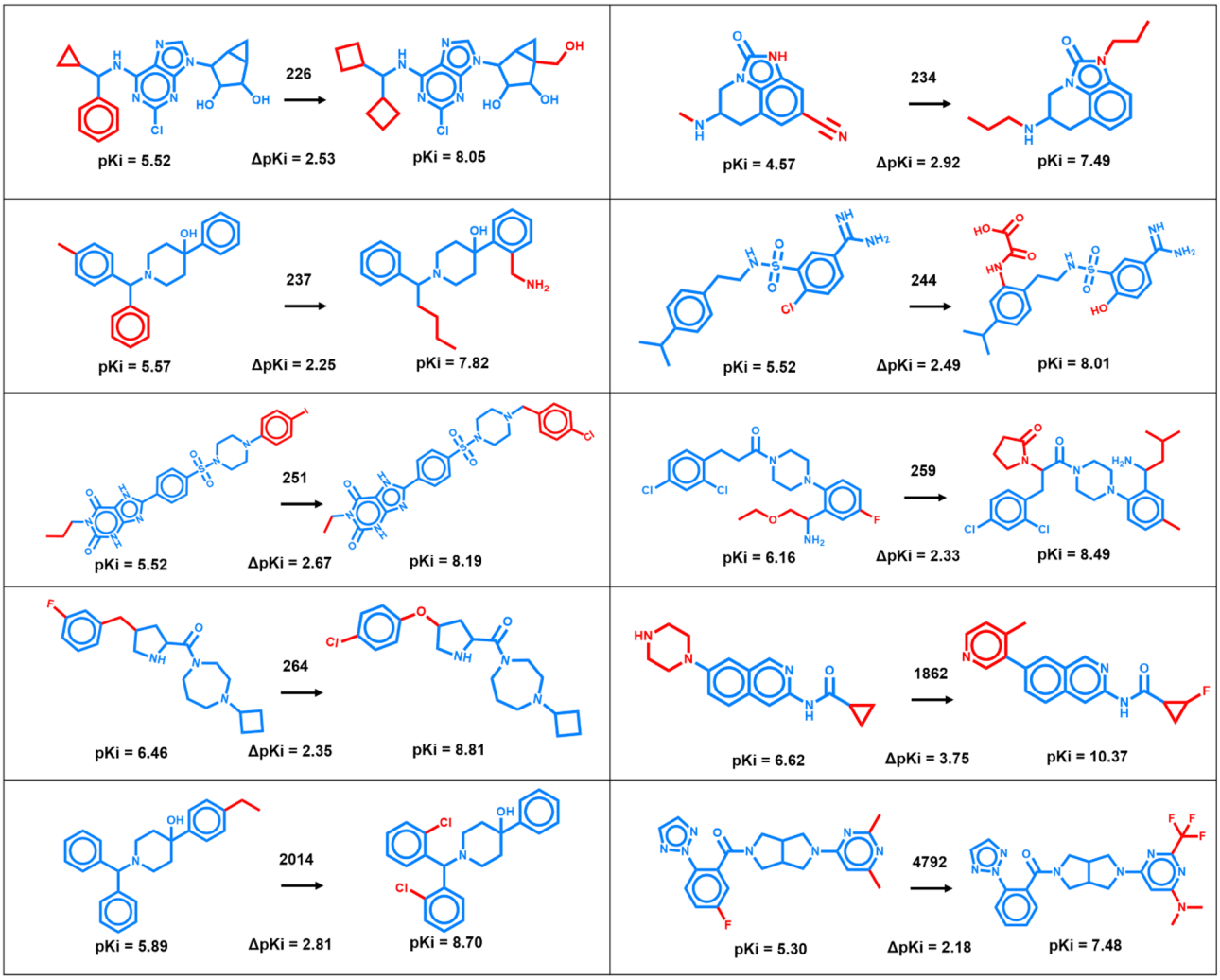


Figure 1. Analogue pairs representing activity cliffs. For each activity class, exemplary AC_CCR pairs are shown and their potency differences are reported. Numbers on arrows identify activity classes according to Table 1. Core structures and substituents are colored blue and red, respectively.

Meta-learning approach

The basic premise of meta-learning, as investigated herein, is parameterizing a model on a series of training tasks by combining and updating parameter settings across individual tasks. This process aims to improve the ability of the model to adapt to new prediction tasks through the use of meta-data.

For designing the meta-learning module of Meta-CLM, we adopted the model-agnostic meta-learning (MAML) framework[28] for an activity class-specific prediction task distribution $p(T)$. Given its model-agnostic nature, the only assumption underlying the MAML approach is that a given model is parameterized using a parameter vector θ. Accordingly, a meta-learning model is considered as a function f_θ with parameter vector θ. The model aims to learn parameter settings θ_{meta} that are derived for individual training tasks and updated across different tasks such that they can be effectively adjusted to new prediction tasks. Therefore, for each of a series of prediction tasks, training data are randomly divided into a support set and a query set Accordingly, when the meta-learning module is applied to a new prediction task T_i such as an activity class the current parameter vector θ_{meta} is updated for task T_i with activity class-specific parameters θ_i obtained by gradient descent optimization minimizing training errors.

During meta-training, as summarized in Fig. 2, the model f_θ is first updated to a task-specific model f_θ' using its support set. Then, the corresponding query set is used to determine the prediction loss of model f_θ' for this task. The procedure is repeated for all prediction tasks (activity classes). Finally, model parameters are further adjusted for testing by minimizing the sum of the prediction loss over all activity classes. Model derivation based on the support sets and evaluation based on query sets are implemented as inner and outer loops, respectively. For meta-testing, the trained meta-learning module is fine-tuned on a specific activity class, for which parameters are adjusted, as also illustrated in Fig. 2. For each class, an individual fine-tuned model is generated.

The meta-learning process aims to capture prior training information through initial parameter vector adjustments, followed by updates through monitoring of the joint loss across all training tasks[29]. Capturing prior training knowledge should enable the model to more effectively adapt to new prediction tasks based on

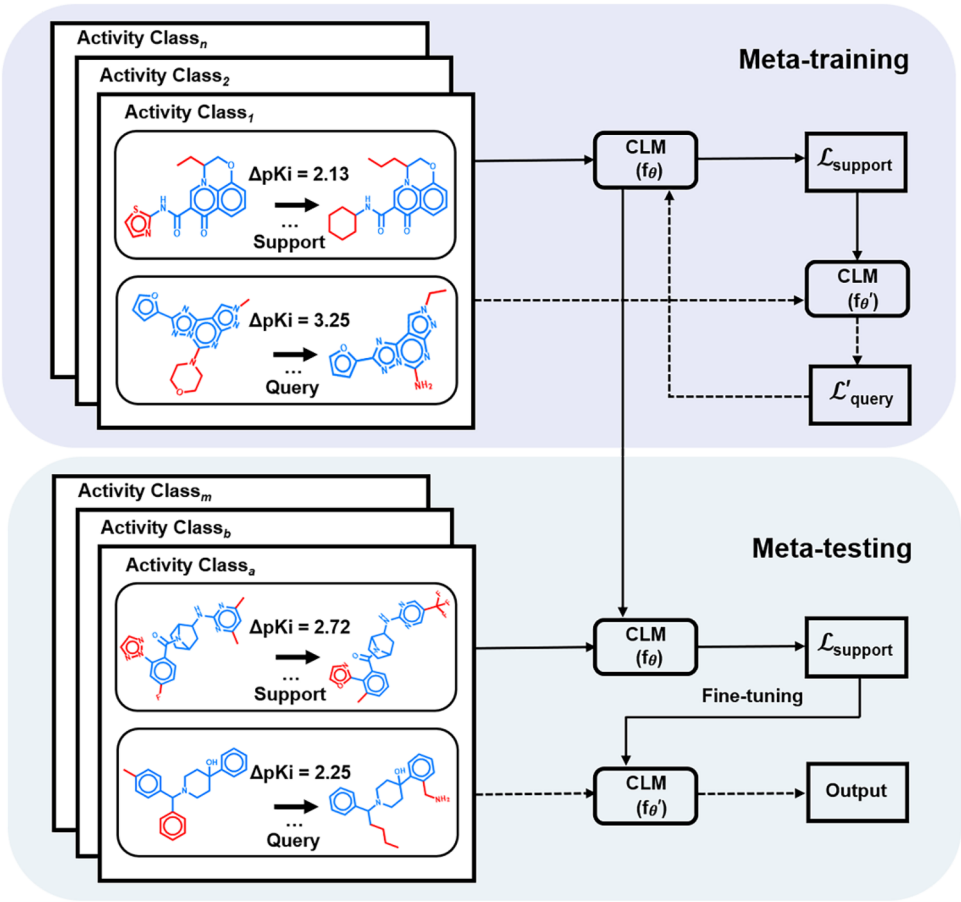


Figure 2. Meta-learning. The illustration summarizes training, fine-tuning, and testing of the meta-learning module of Meta-CLM using exemplary AC-CCR pairs. For each activity class, the support set is used for the initial parameterization of the model (θ). The support loss $\mathscr{L}_{support}$ is calculated for updating model parameters (θ′). Then, the query set is used to calculate the prediction loss $\mathscr{L}_{query}$ for this task. The process is repeated for all training classes, followed by summation of $\mathscr{L}_{query}$ over all tasks to further adjust the parameter settings. The trained module then enters the fine-tuning and testing phase. Solid and dashed lines indicate inner and outer loops, respectively, for meta-training and -testing including fine-tuning.

advanced parameter settings available for initialization and shorter optimization paths with reduced training data requirments[33,34].

This algorithmic approach differs from conventional multi-task learning where a single model is trained on multiple tasks, aiming to share representations and knowledge between these tasks to collectively improve the basis for learning. Hence, the primary goal of multi-task learning is to improve predictive performance for all tasks by leveraging commonalities between them. Accordingly, model weights are updated based on a combination of the losses from all tasks in a single optimization step. Shared representations for multiple tasks support the model's ability to simultaneously learn features common to these tasks.

Transformer models

Base model

For meta-learning, the transformer architecture derived previously for the prediction of highly potent compounds based on weakly potent templates was adopted[13]. Figure 3 illustrates the architecture of the base CLM. The transformer consisted of multiple encoder-decoder modules with attention mechanism[35] and was designed for translating string-based representations of chemical structure. Accordingly, the transformer can be perceived as a chemical language model (CLM). The base model (referred to as CLM in the following) was devised to predict compounds with higher potency for given input compounds[13]. An encoder module consisted of encoding sub-layers including a multi-head self-attention sub-layer and a fully connected feed-forward network sub-layer. The encoder compressed an input sequence into a context vector in its final hidden state, providing the input for the decoder module composed of a feed-forward sub-layer and two multi-head attention sub-layers. The decoder transformed the context vector into a sequence of tokens. Both the encoder and decoder modules utilized the attention mechanism during training to effectively learn from the underlying feature space.

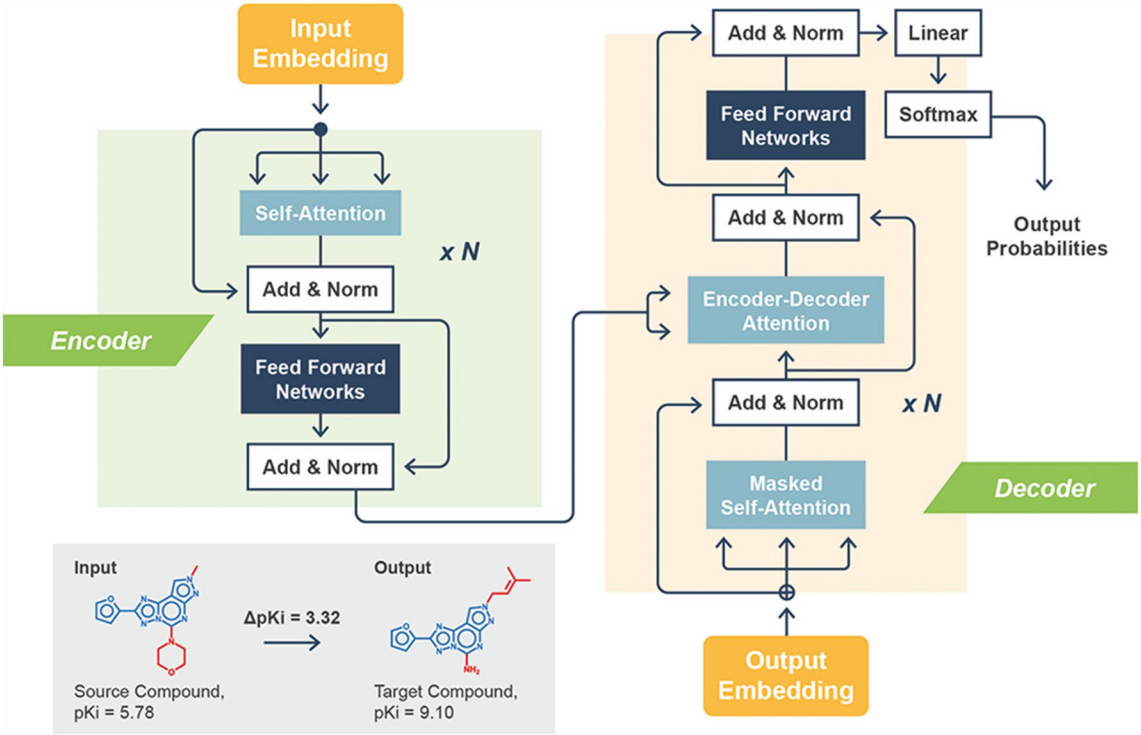


Figure 3. Base CLM. The architecture of the base CLM for designing potent compounds is schematically illustrated (the representation was adapted from ref. 13).

During training, the CLM was challenged to learn mappings of template/source compounds (SCs) to target compounds (TCs) conditioned on potency differences (ΔPot) resulting from replacements of substituent(s):

$$(SC, \Delta Pot) \rightarrow (TC).$$

Hence, training focused on structural analogues with specific potency differences. Then, given a new *(SC, ΔPot)* test instance, the model generated a set of structurally related TCs with putatively higher potency than SCs.

For transformer modeling, compounds and potency differences must be tokenized. Accordingly, compounds were represented as molecular-input line-entry system (SMILES) strings[36] generated using RDKit[37]. Tokenization was facilitated by representing atoms with single-character tokens (e.g., "C"or "N"), two-character tokens (e.g., "Cl" or "Br"), or tokens enclosed in brackets (e.g. "[nH]" or "[O-]"). Potency differences were subjected to binning tokenization[12,13,38,39] by dividing the global range of potency differences (-6.62 to 6.52 pK_i units) into 1314 bins with a constant width of 0.01. Each bin was encoded by a single token and each potency difference was assigned to the corresponding token[12,13]. In addition, two special "start" and "end" tokens were defined as the start and end points of a sequence, respectively.

The model was pre-trained using a large set of 881,990 All_CCR pairs originating from 496 public activity classes[13]. For pre-training, All_CCR triples *(Cpd_A, Cpd_B, Pot_B-Pot_A)* were generated in in which Cpd_A and Cpd_B represented the SC and TC, respectively, and (Pot_B-Pot_A) their potency difference.

CLM was implemented using Pytorch[40]. Default hyperparameter settings were used for the transformer architecture together with a batch size of 64, learning rate of 0.001, and encoding dimension of 256. During training, the transformer model minimized the cross-entropy loss between the ground-truth and output sequence. The Adam optimizer was used[41]. The model was trained for a maximum of 1000 epochs. At each epoch, a checkpoint was saved, and the final model was selected based on the minimal loss.

The base model achieved a reproducibility of 0.857 for the entire test set (corresponding to 10% of pre-training set). Hence, the base CLM model regenerated ~ 86% of the target compounds from CCR-triples not used for training.

Model for meta-learning

The CLM variant for meta-learning was also implemented using Pytorch following the protocol described above. The meta-learning model, designated Meta-CLM, consisted of two modules including the base model for generating mappings of SCs to TCs conditioned on potency differences and the meta-learning module (the design of which is detailed below). For derivation of the metal-learning module, a subset of 176 of the 496 activity classes was selected for which at least 300 All-CCR pairs per class were available, amounting to a total of 491,688 qualifying All_CCR triples. For meta-learning, each activity class was considered a separate training task (see below). Therefore, All_CCR triples from each class were randomly split into support set (80%) and query set (20%). The Adam optimizer was used for gradient descent optimization during meta-learning.

Model fine-tuning

For fine-tuning and comparative evaluation of CLM and Meta-CLM, the 10 activity classes in Table 1 were used. Fine-tuning was separately carried out using AC-CCR pairs from each class. The AC-CCR pairs from each class were randomly divided into fine-tuning (80%) and test instances (20%). In each case, it was confirmed that the fine-tuning and test pairs had no core structure overlap (otherwise, a new partition was generated). For fine-tuning, AC_CCR pairs were exclusively used. AC_CCR triples were ordered such that TC was the highly potent compound. To assess the ability of CLM and Meta-CLM to learn in low-data regimes, model variants were derived based on 10%, 25%, 50% and 100% of the training data. To adapt to differently sized training sets, the pre-trained model was fine-tuned with a smaller learning rate of 0.0001. With a maximum of 200 training epochs, the final fine-tuned model was selected based on minimal cross-entropy loss.

Model evaluation

For each activity class, CCR pairs sharing core structures with the fine-tuning set were excluded, then the final test set was generated by adding the remaining CCR pairs to test AC-CCR pairs. Test set CCR and AC-CCR pairs yielded class-dependent numbers of unique CCR and AC-CCR test compounds. To evaluate the performance of each fine-tuned CLM and corresponding Meta-CLM, test compounds were divided into two categories: SCs with a maximum potency of 1 μM (corresponding to a pK_i value of 6) and TCs with a potency greater than 1 μMol ($pK_i > 6$). These test TCs were termed known target compounds (KTCs), which represented highly potent test compounds. Table 2 reports the test composition for each activity class. Depending on the activity class, 139 to 3838 KTCs were available.

For each test set SC, 50 hypothetical TCs were sampled and compared to available KTCs. The ability of a model to reproduce KTCs was considered as the key criterion for model validation.

Results

Reproducibility of known target compounds

We first analyzed the ability of Meta-CLM to reproduce KTCs in comparison to CLM. The results are reported in Table 3. For all activity classes, Meta-CLM and CLM correctly reproduced multiple KTCs over all fine-tuning conditions, thus providing non-ambiguous proof for the models' ability to predict potent compounds. From correctly predicted SC-KTC pairs, unique KTCs were extracted (a given KTC can occur in multiple pairs). The number of correctly predicted SC-KTC pairs and unique KTCs varied depending on the activity class. Importantly, Meta-CLM consistently predicted more SC-KTC pairs and unique KTCs than CLM across all activity classes, without an exception. For Meta-CLM, the number of SC-KTC pairs varied from 71 to 5102 pairs when utilizing 100% of the training samples and the number of unique KTCs varied from 27 to 287, corresponding to a reproducibility ratio of ~ 7% to ~ 45% of available KTCs per class. For comparison, CLM, the base model, generated from 53 to 4385 SC-KTC pairs, with 23 to 241 unique KTCs and a corresponding reproducibility ratio of ~ 5% to ~ 36% per class. Moreover, for decreasing numbers of fine-tuning samples, Meta-CLM consistently reproduced more KTCs than CLM. For complete fine-tuning sets, Meta-CLM and CLM reached mean reproducibility rates of ~ 21% and ~ 14%, respectively. For only 10% of the fine-tuning samples, Meta-CLM reached a mean reproducibility rate of ~ 15% compared to only ~ 7% for CLM. Thus, Meta-CLM learned more effectively from sparse data than CLM, consistent with the aims of meta-learning.

Figure 4 illustrates the differences in KTC reproducibility rates between Meta-CLM and CLM. Independent-samples t-tests were carried out to assess the statistical significance of the observed differences. For complete fine-tuning sets, increases in reproducibility detected for Meta-CLM were statistically significant for three of 10 activity classes. However, for fine-tuning sets of deceasing size, 25 of 30 increases across all activity classes were statistically significant, thus providing further evidence for the ability of Meta-CLM to more effectively learn from sparse data. For most classes, there was a sharp decline in CLM reproducibility rates when 25% or 10% of the fine-tuning samples were used.

ChEMBL ID	CCR Pairs	Unique CCR CPDs	AC-CCR Pairs	Unique AC-CCR CPDs	Overlapping CPDs	SCs ($pk_i <= 6$)	KTCs ($pk_i > 6$)
226	5950	1174	144	84	80	359	819
234	7790	913	50	53	53	89	824
237	1032	477	31	24	20	115	366
244	1949	308	287	118	88	90	248
251	4706	5210	85	57	38	1391	3838
259	702	169	59	69	33	66	139
264	4756	840	72	81	58	33	830
1862	4554	175	82	51	51	27	148
2014	1388	256	80	62	29	23	266
4792	1941	615	49	50	48	146	471

Table 2. Test sets. For each activity class (ChEMBL IDs are used according to Table 1), the composition of the test set is reported. CPD stands for compound.

ChEMBL ID	Ratio	SC-KTC Pairs		Unique KTCs		Reproducibility (%)	
		Meta-CLM	CLM	Meta-CLM	CLM	Meta-CLM	CLM
226	10	799	379	223	118	27.2	14.4
	25	965	510	263	167	32.1	20.4
	50	1041	614	268	183	32.7	22.3
	100	1193	735	287	216	35.0	26.4
234	10	174	75	50	19	6.1	2.3
	25	268	130	68	36	8.3	4.4
	50	343	197	87	58	10.6	7.0
	100	398	239	101	71	12.3	8.6
237	10	397	325	90	52	24.6	14.2
	25	449	366	101	66	27.6	18.0
	50	433	362	103	81	28.1	22.1
	100	480	429	118	102	32.2	27.9
244	10	109	62	26	11	10.5	4.4
	25	111	66	31	17	12.5	6.9
	50	160	98	39	28	15.7	11.3
	100	193	129	45	36	18.2	14.5
251	10	3930	3288	233	138	6.1	3.6
	25	4685	3959	249	172	6.5	4.5
	50	4856	4153	245	201	6.4	5.2
	100	5102	4385	264	241	6.9	6.3
259	10	51	40	14	5	10.1	3.6
	25	73	60	24	13	17.3	9.4
	50	98	88	30	22	21.6	15.8
	100	129	116	33	30	23.7	21.6
264	10	16	11	14	6	1.7	0.7
	25	33	19	28	17	3.3	2.1
	50	54	33	42	31	5.1	3.7
	100	71	53	57	40	6.9	4.8
1862	10	65	29	25	6	16.9	4.0
	25	96	48	28	14	18.9	9.5
	50	93	56	32	23	21.6	15.5
	100	147	94	33	30	22.3	20.3
2014	10	85	53	20	9	7.5	3.4
	25	102	71	25	12	9.4	4.5
	50	113	84	22	16	8.3	6.0
	100	131	99	27	23	10.2	8.7
4792	10	849	622	176	106	37.4	22.5
	25	976	746	179	129	38.0	27.4
	50	1085	828	199	151	42.3	32.1
	100	1262	969	212	170	45.0	36.1

Table 3. Reproducibility of compound pairs and known target compounds.

We also note that both models produced large numbers of novel candidate compounds for SCs. For complete fine-tuning sets, Meta-CLM and CLM generated on average 2375 and 2818 new candidate compounds per activity class (ranging from 119 to 9952 and 234 to 10,779 candidates, respectively). While these new compounds cannot be considered for model validation, they provide large pools of candidates for practical applications in the search for potent compounds.

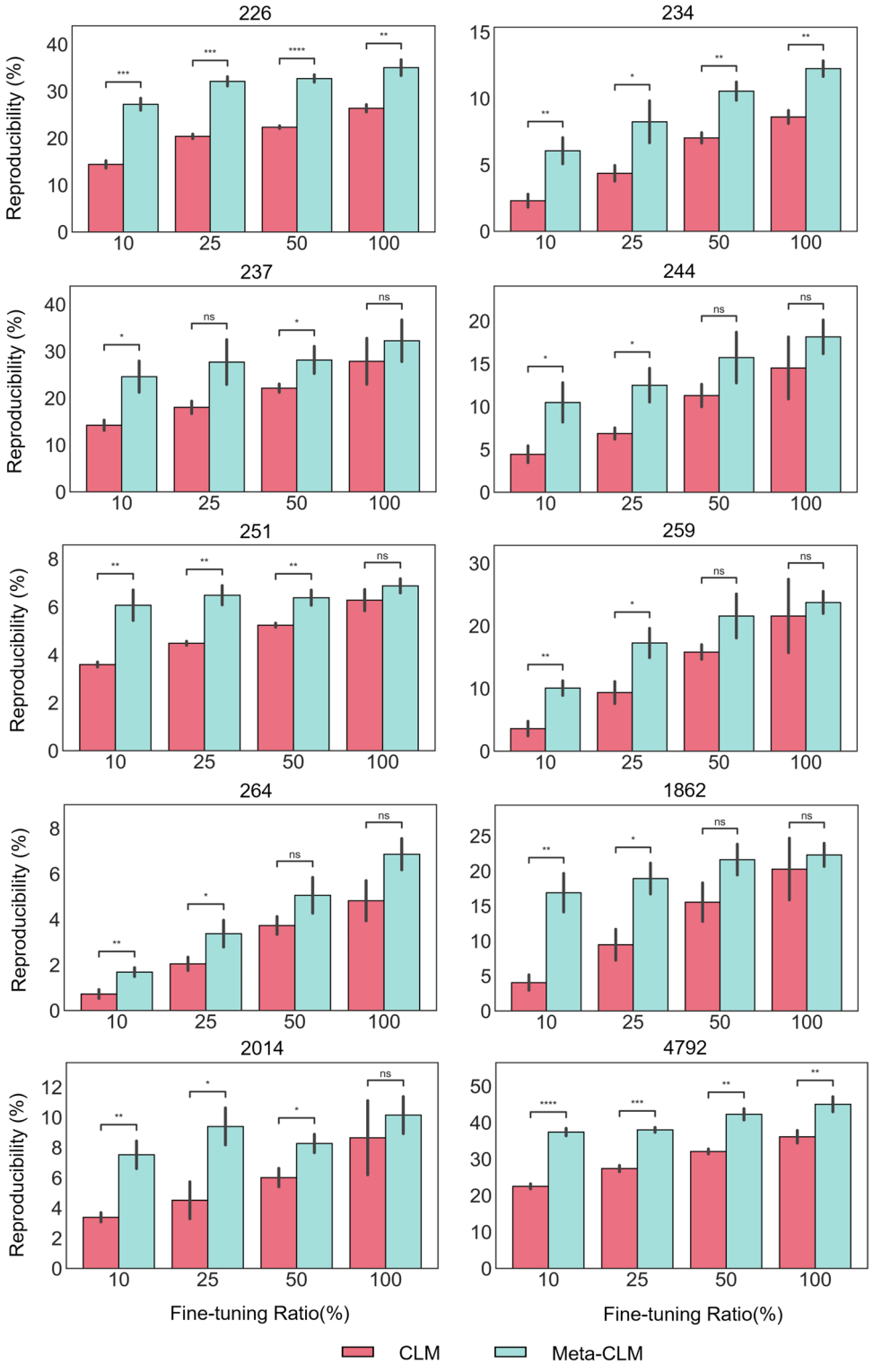


Figure 4. Reproducibility of known target compounds. For each activity class, the proportion of correctly reproduced KTCs is reported for Meta-CLM and CLM over varying percentages of fine-tuning samples. Mean and standard deviations (error bars) are provided. To assess the statistical significance of observed differences between reproducibility rates, independent-samples t tests were conducted: $0.05 < p \leq 1.00$ (ns), $0.01 < p \leq 0.05$ (*), $0.001 < p \leq 0.01$ (**), $0.0001 < p \leq 0.001$ (***), $p \leq 0.0001$ (****). Stars denote increasing levels of statistical significance and "ns" stands for "not significant".

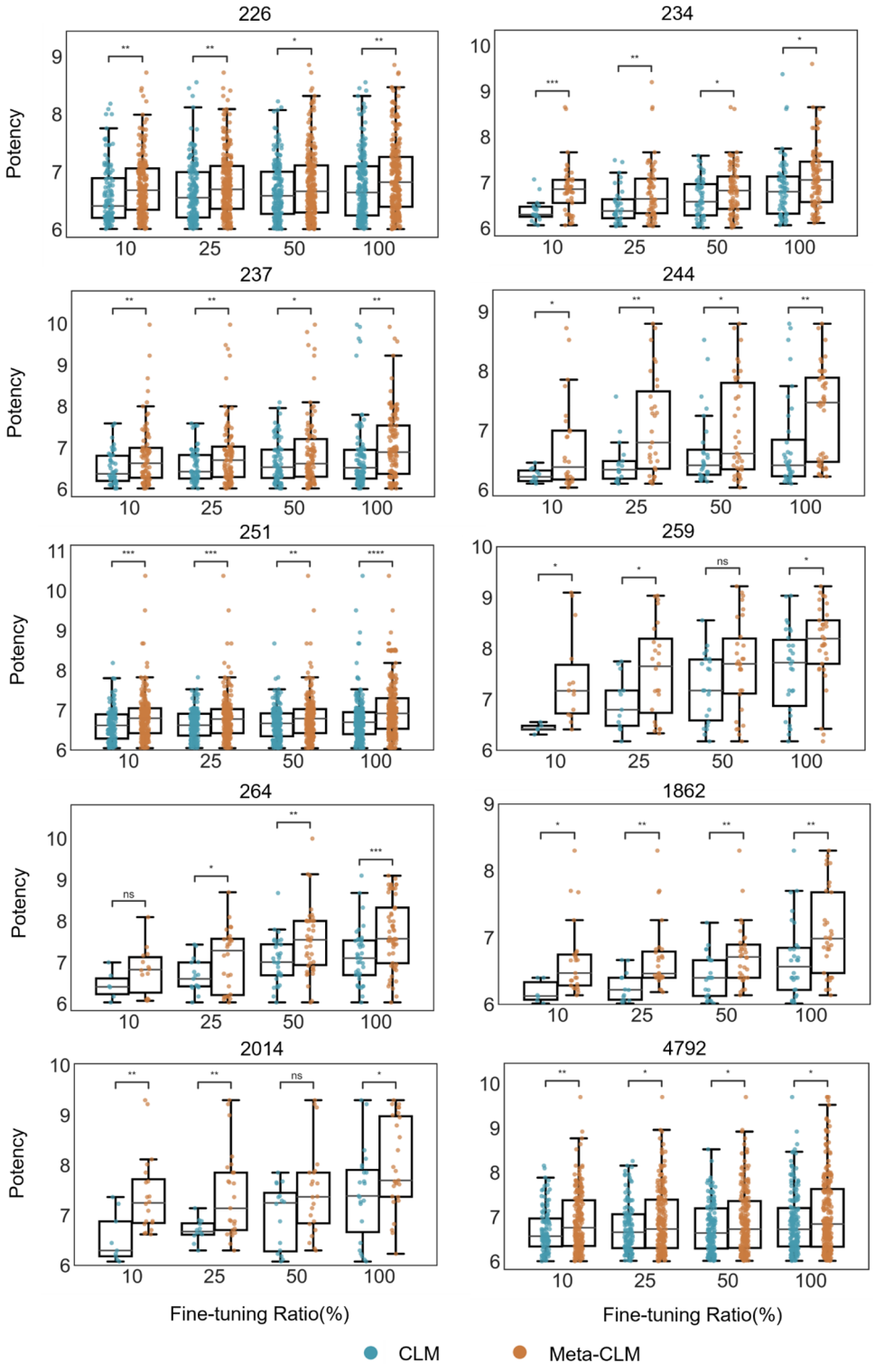


Figure 5. Potency value distribution of reproduced known target compounds. For all activity classes, boxplots report the distributions of logarithmic potency values of KTCs correctly reproduced by Met-CLM and CLM over varying numbers of fine-tuning samples. To assess the statistical significance of differences between potency value distributions, independent-samples t tests were conducted: $0.05 < p \leq 1.00$ (ns), $0.01 < p \leq 0.05$ (*), $0.001 < p \leq 0.01$ (**), $0.0001 < p \leq 0.001$ (***), $p \leq 0.0001$ (****).

Compound potency

In addition to reproducing KTCs, the actual potency level of correctly predicted KTCs and potency differences between SCs and corresponding KTCs represented other highly relevant criteria for model assessment. According to our semi-quantitative design approach, ideally, the models should predict highly potent compounds from given SCs. Therefore, we next analyzed the potency of correctly predicted KTCs and potency differences between Meta-CLM and CLM.

Known target compounds

Figure 5 shows the distributions of logarithmic potency values of KTCs reproduced by Meta-CLM and CLM. Importantly, KTCs generated by Meta-CLM were overall consistently more potent than those generated by CLM across all activity classes and fine-tuning conditions. Thirty-eight of the total of 40 observed differences between the respective potency value distributions were statistically significant. Especially for 25% and 10% of the fine-tuning samples, Meta-CLM generated multiple KTCs with low-nanomolar or even sub-nanomolar potency for each activity class, whereas CLM only generated a few KTCs with potency higher than 10 nM (pKi > 8) for three classes.

Potency differences between source and target compounds

Furthermore, we analyzed potency differences captured by SC-KTC pairs. Following our design strategy, increasingly large potency differences between corresponding SCs and correctly reproduced KTCs were favored. Figure 6 shows the distribution of potency differences between corresponding SCs and KTCs for Meta-CLM and CLM predictions. In the case of Meta-CLM (CLM), four (six) activity classes displayed median potency differences between SCs and corresponding KTCs between one and two orders of magnitude (10- to100-fold) and the remaining six (four) classes displayed median potency differences exceeding two orders of magnitude (> 100-fold) for complete fine-tuning sets. Hence, significant potency differences were generally observed. For half of the activity classes, median potency differences were comparable for all fine-tuning conditions when separately viewed for Meta-CLM and CLM, respectively. However, when Meta-CLM and CLM were compared, potency differences of SC-KTC pairs were consistently larger for Meta-CLM. Again, 38 of 40 observed differences were statistically significant. Overall, many more KTCs with at least 1000-fold higher potency than the corresponding SCs were generated by Meta-CLM compared to CLM. Thus, Meta-CLM predicted KTCs with overall higher potency than CLM and much larger potency differences between SCs and KTCs.

Conclusion

In this work, we have explored meta-learning for the prediction of potent compounds using conditional transformer models. Compound potency predictions are of high interest in drug discovery but high-quality activity data available for machine learning are typically sparse. For these predictions, meta-learning was of particular interest to us because the approach is well-suited for models that are rich in meta-data, yet currently only little explored for drug discovery applications. Therefore, we have adapted a previously investigated transformer architecture to construct a meta-learning model by adding a special meta-learning module to a pre-trained transformer. Then, meta-learning model variants were derived for different activity classes and their performance in the design of potent compounds was compared to reference transformers. For model validation, the ability to reproduce potent KTCs served as the major criterion. All models successfully reproduced KTCs. However, compared to reference models, meta-learning significantly increased the number of correctly predicted KTCs across all activity classes, especially for decreasing numbers of fine-tuning samples. This was an encouraging finding, consistent with expectations for successful meta-learning. Moreover, meta-learning models also produced target compounds with overall higher potency than other transformers and larger potency differences between templates and targets. These improvements were not anticipated but are highly attractive for practical applications. The generative models designed for predicting potent compounds produced large numbers of candidate compounds with novel structures. New candidate compounds predicted by the meta-learning models should represent an attractive resource for prospective applications in searching for potent compounds for targets of interest. Taken together, the results reported herein, provide proof-of-concept for the potential of meta-learning in generative design of potent compounds. Moreover, in light of our findings, we anticipate that meta-learning will also be a promising approach for other compound design applications in low-data regimes.

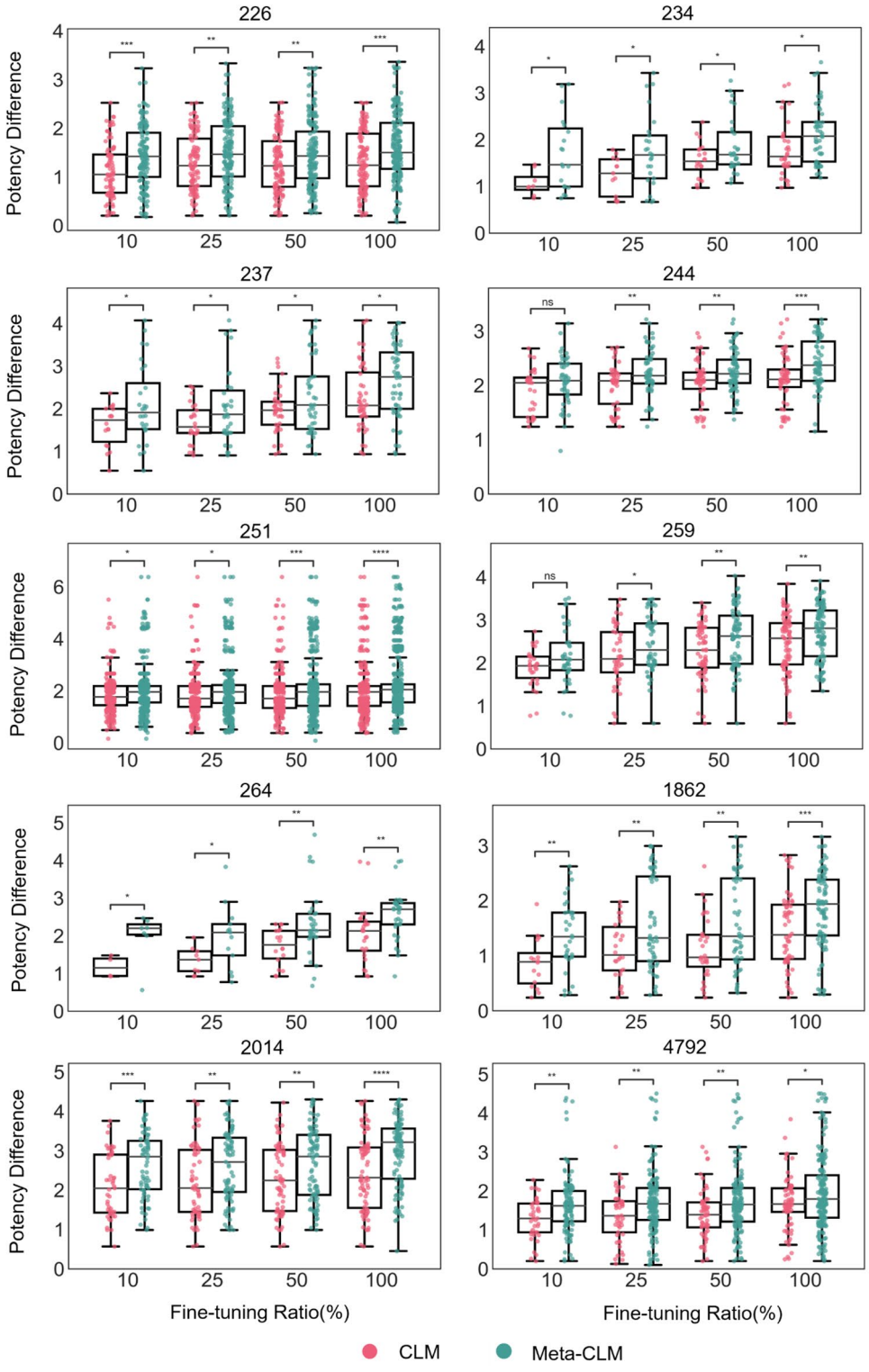


Figure 6. Distribution of potency differences between source and known target compounds. For all activity classes, boxplots report the distributions of logarithmic potency differences for SC-KTC pairs predicted by Meta-CLM and CLM over varying numbers of fine-tuning samples. To assess the statistical significance of differences between the distributions, independent-samples t tests were conducted: $0.05 < p \leq 1.00$ (ns), $0.01 < p \leq 0.05$ (*), $0.001 < p \leq 0.01$ (**), $0.0001 < p \leq 0.001$ (***), $p \leq 0.0001$ (****).

Data availability

Calculations were carried out using publicly available programs and compound data. Python scripts generated for the study and the activity classes used are available via the following link: https://uni-bonn.sciebo.de/s/kfAQZ0mbCGHtr0m.

Received: 22 June 2023; Accepted: 18 September 2023
Published online: 26 September 2023

References

1. Vamathevan, J. *et al.* Applications of machine learning in drug discovery and development. *Nat. Rev. Drug. Discov.* **18**, 463–477 (2019).
2. Walters, W. P. & Barzilay, R. Applications of deep learning in molecule generation and molecular property prediction. *Acc. Chem. Res.* **54**, 263–270 (2020).
3. Lo, Y. C., Rensi, S. E., Torng, W. & Altman, R. B. Machine learning in chemoinformatics and drug discovery. *Drug Discov. Today* **23**, 1538–1546 (2018).
4. Patel, L., Shukla, T., Huang, X., Ussery, D. W. & Wang, S. Machine learning methods in drug discovery. *Molecules* **25**, 5277 (2020).
5. Rodríguez-Pérez, R., Miljković, F. & Bajorath, J. Machine learning in chemoinformatics and medicinal chemistry. *Annu. Rev. Biomed. Data Sci.* **5**, 43–65 (2022).
6. Lewis, R. A. & Wood, D. Modern 2D QSAR for drug discovery. *WIREs Comput. Mol. Sci.* **4**, 505–522 (2014).
7. Muratov, E. N. *et al.* QSAR without borders. *Chem. Soc. Rev.* **49**, 3525–3564 (2020).
8. Mobley, D. L. & Gilson, M. K. Predicting binding free energies: Frontiers and benchmarks. *Annu. Rev. Biophys.* **46**, 531–558 (2017).
9. Williams-Noonan, B. J., Yuriev, E. & Chalmers, D. K. Free energy methods in drug design: Prospects of "Alchemical perturbation" in medicinal chemistry. *J. Med. Chem.* **61**, 638–649 (2018).
10. Stumpfe, D., Hu, H. & Bajorath, J. Evolving concept of activity cliffs. *ACS Omega* **4**, 14360–14368 (2019).
11. Janela, T. & Bajorath, J. Simple nearest-neighbour analysis meets the accuracy of compound potency predictions using complex machine learning models. *Nat. Mach. Intell.* **4**, 1246–1255 (2022).
12. Chen, H., Vogt, M. & Bajorath, J. DeepAC–conditional transformer-based chemical language model for the prediction of activity cliffs formed by bioactive compounds. *Dig. Discov.* **1**, 898–909 (2022).
13. Chen, H. & Bajorath, J. Designing highly potent compounds using a chemical language model. *Sci. Rep.* **13**, 7412 (2023).
14. Chen, D. *et al.* Algebraic graph-assisted bidirectional transformers for molecular property prediction. *Nat. Commun.* **12**, 3521 (2021).
15. Song, Y., Chen, J., Wang, W., Chen, G. & Ma, Z. Double-head transformer neural network for molecular property prediction. *J. Cheminform.* **15**, 27 (2023).
16. Jiang, Y. *et al.* Pharmacophoric-constrained heterogeneous graph transformer model for molecular property prediction. *Commun. Chem.* **6**, 60 (2023).
17. Bagal, V., Aggarwal, R., Vinod, P. K. & Priyakumar, U. D. MolGPT: Molecular generation using a transformer-decoder model. *J. Chem. Inf. Model.* **62**, 2064–2076 (2022).
18. Mazuz, E., Shtar, G., Shapira, B. & Rokach, L. Molecule generation using transformers and policy gradient reinforcement learning. *Sci. Rep.* **13**, 8799 (2023).
19. Wang, Y., Zhao, H., Sciabola, S. & Wang, W. cMolGPT: a conditional generative pre-trained transformer for target-specific de novo molecular generation. *Molecules* **28**, 4430 (2023).
20. Chen, L. *et al.* TransformerCPI: Improving compound–protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments. *Bioinformatics* **36**, 4406–4414 (2020).
21. Huang, K., Xiao, C., Glass, L. M. & Sun, J. MolTrans: molecular interaction transformer for drug–target interaction prediction. *Bioinformatics* **37**, 830–836 (2021).
22. Chen, L. *et al.* Sequence-based drug design as a concept in computational drug design. *Nat. Commun.* **14**, 4217 (2023).
23. Warmuth, M. K. *et al.* Active learning with support vector machines in the drug discovery process. *J. Chem. Inf. Comput. Sci.* **43**, 667–673 (2003).
24. Reker, D. & Schneider, G. Active-learning strategies in computer-assisted drug discovery. *Drug Discov. Today* **20**, 458–465 (2015).
25. Weiss, K., Khoshgoftaar, T. M. & Wang, D. A survey of Transfer Learning. *J. Big Data* **3**, 9 (2016).
26. Cai, C. *et al.* Transfer learning for drug discovery. *J. Med. Chem.* **63**, 8683–8694 (2020).
27. Vilalta, R. & Drissi, Y. A Perspective View and Survey of Meta-Learning. *Artif. Intell. Rev.* **18**, 77–95 (2002).
28. Finn, C., Abbeel, P. & Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of 34th International Conference on Machine Learning* (Eds. Precup, D. & Teh, Y. W.) 1126–1135 (JMLR.org, 2017).
29. Wang, Y., Yao, Q., Kwok, J. T. & Ni, L. M. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.* **53**, 1–34 (2020).
30. Vella, D. & Ebejer, J. P. Few-shot learning for low-data drug discovery. *J. Chem. Inf. Model.* **63**, 27–42 (2023).
31. Bento, A. P. *et al.* The CHEMBL bioactivity database: An update. *Nucleic Acids Res.* **42**, D1083–D1090 (2014).
32. Naveja, J. J., Vogt, M., Stumpfe, D., Medina-Franco, J. L. & Bajorath, J. Systematic extraction of analogue series from large compound collections using a new computational compound–core relationship method. *ACS Omega* **4**, 1027–1032 (2019).
33. Raghu, A., Raghu, M., Bengio, S. & Vinyals, O. Rapid learning or feature reuse? towards understanding the effectiveness of MAML. In *8th International Conference on Learning Representations* (OpenReview.net, 2020).
34. Lv, Q., Chen, G., Yang, Z., Zhong, W. & Chen, C. Y. C. Meta learning with graph attention networks for low-data drug discovery. *IEEE Trans. Neural Netw. Learn. Syst.* **6**, 1–13 (2023).
35. Vaswani, A. *et al.* Attention is all you need. *Adv. Neural Inf. Process. Syst.* **5**, 6000–6010 (2017).
36. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **28**, 31–36 (1988).
37. RDKit: Cheminformatics and Machine Learning Software. http://www.rdkit.org (Accessed on 1 July 2021).
38. He, J. *et al.* Molecular optimization by capturing chemist's intuition using Deep Neural Networks. *J. Cheminform.* **13**, 26 (2021).
39. He, J. *et al.* Transformer-based molecular optimization beyond matched Molecular Pairs. *J. Cheminform.* **14**, 18 (2022).
40. Paszke, A. *et al.* PyTorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32**, 8026–8037 (2019).
41. Kingma, D.P. & Ba, J. Adam: a method for stochastic optimization. In *3th International Conference on Learning Representations* (OpenReview.net, 2015).

Acknowledgements

The authors thank Martin Vogt for many helpful suggestions. H.C. is supported by the China Scholarship Council (CSC).

Author contributions

All authors contributed to designing and conducting the study, analyzing the results, and preparing the manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to J.B.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Appendix D

Extension of Multi-site Analogue Series with Potent Compounds using a Bidirectional Transformer-Based Chemical Language Model

RSC Medicinal Chemistry

ROYAL SOCIETY OF CHEMISTRY

RESEARCH ARTICLE

Cite this: *RSC Med. Chem.*, 2024, **15**, 2527

Extension of multi-site analogue series with potent compounds using a bidirectional transformer-based chemical language model

Hengwei Chen,[a] Atsushi Yoshimori[b] and Jürgen Bajorath *[ac]

Received 10th June 2024,
Accepted 15th June 2024

DOI: 10.1039/d4md00423j

rsc.li/medchem

Generating potent compounds for evolving analogue series (AS) is a key challenge in medicinal chemistry. The versatility of chemical language models (CLMs) makes it possible to formulate this challenge as an off-the-beaten-path prediction task. In this work, we have devised a coding and tokenization scheme for evolving AS with multiple substitution sites (multi-site AS) and implemented a bidirectional transformer to predict new potent analogues for such series. Scientific foundations of this approach are discussed and, as a benchmark, the transformer model is compared to a recurrent neural network (RNN) for the prediction of analogues of AS with single substitution sites. Furthermore, the transformer is shown to successfully predict potent analogues with varying R-group combinations for multi-site AS having activity against many different targets. Prediction of R-group combinations for extending AS with potent compounds represents a novel approach for compound optimization.

Introduction

Generative modeling of compounds has substantially expanded opportunities for molecular design.[1–5] For generative modeling, deep neural networks (DNNs) adapted from natural language processing (NLP) that utilize textual representations of chemical structures[3,4] are particularly versatile. Such models can be derived to learn a variety of sequence-to-sequence mappings for diverse design tasks, giving rise to different types of chemical language models (CLMs).[6–14] Popular DNN architectures from NLP include recurrent neural networks (RNNs), which were first adapted for applications in chemistry,[4–7] and different transformer networks that are increasingly used for molecular design and *de novo* compound generation.[10–15] Different deep generative models including transformers and diffusion models have also been developed in structure-based drug design for the discovery and optimization of new active compounds.[15,16]

In medicinal chemistry, compound optimization efforts result in analogue series (AS), which represent the major source of structure–activity relationship (SAR) information. For evolving AS, the key question in the practice of medicinal chemistry is which analogue(s) to synthesize next to further improve compound potency and/or other molecular properties relevant for drug development. This optimization process continues to be largely driven by chemical knowledge and experience and can be supported by standard computational techniques such as quantitative SAR (QSAR) analysis using linear or non-linear machine learning regression models based on assay data.

Previously, we have devised an RNN-based CLM (termed DeepAS) for extending evolving AS with new potent analogues,[6] representing an off-the-beaten-path prediction task. Conceptually, this approach was based on the notion of the SAR transfer principle.[17,18] SAR transfer refers to findings that AS with activity against different targets frequently contained corresponding analogues with comparable potency progression.[17–19] This observation also reflects the application of similar optimization strategies in medicinal chemistry for optimizing (hydrophobic or hydrophilic) ligand–target interactions. A key aspect of the SAR transfer concept is that corresponding R-group sequences representing ascending compound potency gradients are found in many different AS, regardless of their core structures (scaffolds).[17] Thus, SAR transfer is not dependent on individual scaffolds. Instead, it depends on the transfer of SAR information and progression across AS with different core structures. Therefore, to exploit SAR transfer information, AS are represented as R-group sequences of potency-ordered analogues, without considering the invariant scaffolds of different series. To exploit SAR transfer

[a] *Department of Life Science Informatics and Data Science, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, University of Bonn, Friedrich-Hirzebruch-Allee 5/6, D-53115 Bonn, Germany. E-mail: bajorath@bit.uni-bonn.de; Tel: +49 228 7369 100*
[b] *Institute for Theoretical Medicine, Inc., 26-1 Muraoka-Higashi 2-chome, Fujisawa, Kanagawa 251-0012, Japan*
[c] *Lamarr Institute for Machine Learning and Artificial Intelligence, University of Bonn, Friedrich-Hirzebruch-Allee 5/6, D-53115 Bonn, Germany*

information for generalized analogue design (that is, prediction of new analogues for individual AS, rather than pairs of AS representing individual SAR transfer events), the original DeepAS approach was devised.[6] A basic RNN model was implemented comprising three layers including a long short-term memory (LSTM) layer, a batch normalization, and dense layer. The model was trained on a large number of AS for deriving conditional probabilities for AS extension with compounds carrying a new R-group and displayed promising prediction accuracy in reproducing terminal R-groups in AS for a variety of targets.[6] A principal limitation of the approach was that predictions were confined to AS with single substitution sites. Considering multiple substitution sites, which is common practice in medicinal chemistry, would have substantially increased the complexity of the prediction task and was not attempted in our proof-of-concept study.

Herein, we have further investigated and advanced the approach for extending AS with potent compounds. To these ends, a new transformer model was derived to enable a direct comparison with the RNN CLM for predictions on AS with single substitution sites. Moreover, for the transformer, a novel AS encoding strategy was devised, enabling the prediction of R-group combinations for extending AS having multiple substitution sites. In test calculations, new potent analogues for such AS were successfully predicted across a large target space.

Methodology

Analogue series data

AS with single substitution sites can be identified and represented using the matching molecular series (MMS) formalism.[20,21] MMS are defined as series of compounds that are only distinguished by a chemical modification at a single site[20] and can be identified using a matched molecular pair (MMP) algorithm variant.[21] For the original development of DeepAS,[6] 104 627 MMS from 2195 target-based compound activity classes were extracted from ChEMBL.[22] From these MMS, all R-groups were systematically isolated, yielding 3852 unique R-groups, which represented all R-groups occurring in AS covering the entire target space of active compounds.[6] From this large pool, 10 activity classes each containing at least 700 MMS were randomly selected as test cases for model evaluation and comparison (Table 1). These activity classes included a variety of receptor ligand and enzyme inhibitors. The remaining 2185 activity classes were used to build and evaluate the transformer model and compare it to DeepAS.

Furthermore, a systematic search for AS with multiple substitution sites (multi-site AS) was carried out in activity classes from ChEMBL (release 29) using the compound-core relationship (CCR) algorithm.[23] This method employs a modified and extended matched molecular pair (MMP) fragmentation procedure[24] based on retrosynthetic rules[25] and effectively identifies multi-site AS in compound data sets.[23] Therefore, activity classes with high-confidence activity data were pre-selected from ChEMBL based on the following criteria. Compound activity was required to be determined in direct interaction assays (assay relationship type: "D") with human targets at the highest assay confidence level (assay confidence score 9). Potency measurements were limited to numerically specified equilibrium constants (K_i values) that were recorded at (negative decadic logarithmic) pK_i values. For compounds with multiple measurements, the geometric mean was calculated as the final potency annotation, provided all pK_i values fell within the same order of magnitude; otherwise, the compound was disregarded. Then, in each qualifying activity class, a systematic search for AS with one to five substitution sites was carried out. The core structure of an AS was required to contain at least twice the number of non-hydrogen atoms as the combined substituents.[23] From 864 target-based activity classes containing multi-site AS, a total of 16 538 AS were obtained. Furthermore, from these activity classes, the five classes with largest total number of AS with one to five substitution sites were selected (Table 2). These classes consisted of different G protein coupled receptor (GPCR) ligands and were used for transformer fine-tuning (see below). AS from the remaining activity classes were used to derive and evaluate the transformer model for extending multi-site AS. Fig. 1 shows exemplary ASs with single or multiple substitution sites.

Table 1 Selected activity classes with analogue series containing single substitution sites

UniProt ID	ChEMBL ID	Target name	#AS
P00533	203	Epidermal growth factor receptor erbB1	1692
Q72547	247	Human immunodeficiency virus type 1 reverse transcriptase	1279
Q16539	260	MAP kinase p38 alpha	744
P35968	279	Vascular endothelial growth factor receptor 2	1917
Q13547	325	Histone deacetylase 1	989
O60674	2971	Tyrosine-protein kinase JAK2	763
P11362	3650	Fibroblast growth factor receptor 1	810
P08581	3717	Hepatocyte growth factor receptor	852
P42336	4005	PI3-kinase p110-alpha subunit	866
P56817	4822	Beta-secretase 1	1093

For each activity class, UniProt ID, ChEMBL target ID, target name, and number of AS with single substitution sites are reported.

Table 2 Selected activity classes with AS containing multiple substitution sites

UniProt ID	ChEMBL ID	Target name	#1	#2	#3	#4	#5	#AS
P30542	226	Adenosine A1 receptor	150	105	68	26	31	380
P35462	234	Dopamine D3 receptor	180	79	45	30	27	361
P29274	251	Adenosine A2a receptor	185	101	64	33	29	412
P0DMS8	256	Adenosine A3 receptor	114	87	61	29	29	320
Q9Y5N1	264	Histamine H3 receptor	188	79	41	26	21	355

For each activity class, UniProt ID, ChEMBL target ID, target name, number of AS with one to five substitution sites (#1 to #5), and total number of AS are reported.

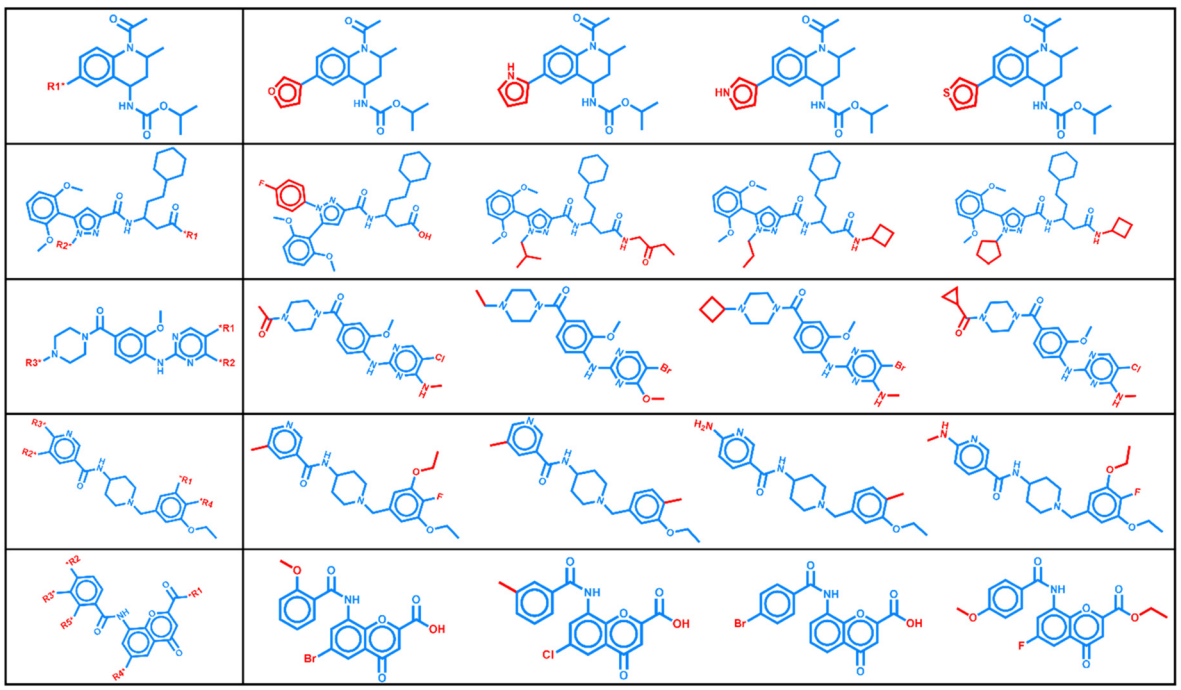


Fig. 1 Exemplary analogue series. Markush structures in the column on the left represent AS with increasing number of substitution sites and each corresponding row shows four exemplary analogues. Substitution sites and non-hydrogen R-groups are highlighted in red.

Analogue design strategy

The approach aimed at the probability-based prediction of R-groups producing potent compounds in evolving AS. Relative potency information was implicitly accounted for by ordering AS according to increasing compound potency. Thus, each AS and its R-group sequence represented an ascending potency gradient. Given that CLM-based probabilities of R-groups depended on the preceding R-group sequence, the design of new R-groups was order-dependent, following the potency gradient. For a given AS, this design strategy corresponded to a search for increasingly potent analogues, thereby bridging between generative modeling and property-based optimization. Leveraging the input data format (sentences), a generative model was supposed to learn conditional probabilities for R-groups based on potency-ordered sequences in which they occurred. A major challenge for the design approach was taking multiple substitution sites in AS into account, corresponding to probability-based prioritization of R-group combinations for compound design.

Representing analogue series as potency-ordered R-group sequences

MMS were converted into potency-ordered R-group sequences following the original DeepAS approach.[6] Each AS was sorted based on increasing potency of their analogues. For CLM derivation, an AS was then encoded as a sentence in which each R-group was represented as an individual R-group token. Three additional special tokens were introduced including the "go" and "stop" tokens, marking the beginning and end of a sentence, respectively, and "none", denoting an empty token. A sentence was required to contain a minimum of two R-group tokens. Each MMS was composed of the sentence and a terminal label, representing the next R-group to be added to the sentence (that is, the prediction for a given input sentence). The length of a sentence was consistently set to 35 tokens and the total number of label tokens amounted to 3855, including 3852 unique R-groups extracted from the qualifying MMS plus the three special tokens.[6]

For AS featuring multiple substitution sites, a novel coding scheme was devised. Initially, these AS were also ordered according to increasing potency and structured in a tabular format where each row contained an analogue represented by the core structure, R-groups at different substitution sites, and its potency annotation (Fig. 2). Subsequently, R-groups of each analogue were concatenated into a combined R-group (combination) token. For CLM derivation, each sample was formatted to contain its sentence with the terminal label at a constant length of 256 tokens. The vocabulary of possible labels consisted of 36 647 concatenated R-group (combination) tokens extracted from multi-site AS, supplemented by special tokens "go", "stop", "none", and "X", the latter representing the absence of a substitution site (Fig. 2). This coding scheme was consistently applicable to AS with varying numbers of substitution sites, as illustrated Fig. 3. Herein, it was applied to cover AS with one to five substitution sites. Notably, the absence of one or more of maximally five substitution sites in an AS, formally defined through the use of the "X" token, was distinct from the presence of an "–H" substituent (label token) at defined substitution sites.

Data augmentation involved transforming each AS into multiple sentences such that each training instance was expanded into sentences capturing an increasing number of R-group tokens (that is, two, three, … all R-group tokens). The final label represented the next R-group to be predicted, with "stop" indicating completion of an AS in iterative R-group predictions.

Transformer variant

Considering the specifics of the design strategy outlined above, a new generative CLM, termed DeepAS 2.0, was devised for the prediction of R-groups producing potent compounds in evolving AS using the bidirectional encoder representations from transformers (BERT) architecture[26] (Fig. 4a). The original transformer architecture comprised multiple encoder–decoder neural modules with attention mechanisms.[27] In this architecture, a stack of encoding layers, each including a multi-head self-attention sub-layer and a fully connected feed-forward network sub-layer, formed the encoder module. The encoder processed an input sequence and compressed it into a context vector in its final hidden state. This context vector served as input for the decoder block, which predicted an output sequence. The decoder module, composed of a feed-forward network sub-layer and two multi-head attention sub-layers, converted the encodings into a sequence of tokens. Both encoder and decoder utilized the attention mechanism during training to effectively learn from the feature space.

Building upon this original architecture, various transformer variants have been developed.[26,28,29] BERT has been successfully used in NLP for learning word vectors based on contextual information, in particular, for text classification and next-sentence prediction.[30–34] Unlike other language models that capture context unidirectionally, BERT was designed as a bidirectional model to analyze sentences in forward and backward direction and predict new words conditioned on all other words in sentences.[26,35] Given that next-sentence prediction was conceptually related to the AS extension task, BERT was chosen as a transformer architecture for R-group prediction.

BERT comprises an embedding layer, multiple transformer encoder layers, and a task-related output layer.[26] In the embedding layer, the input word token is embedded into continuous vector space *via* a matrix and a pre-defined positional encoding vector is added to each embedding

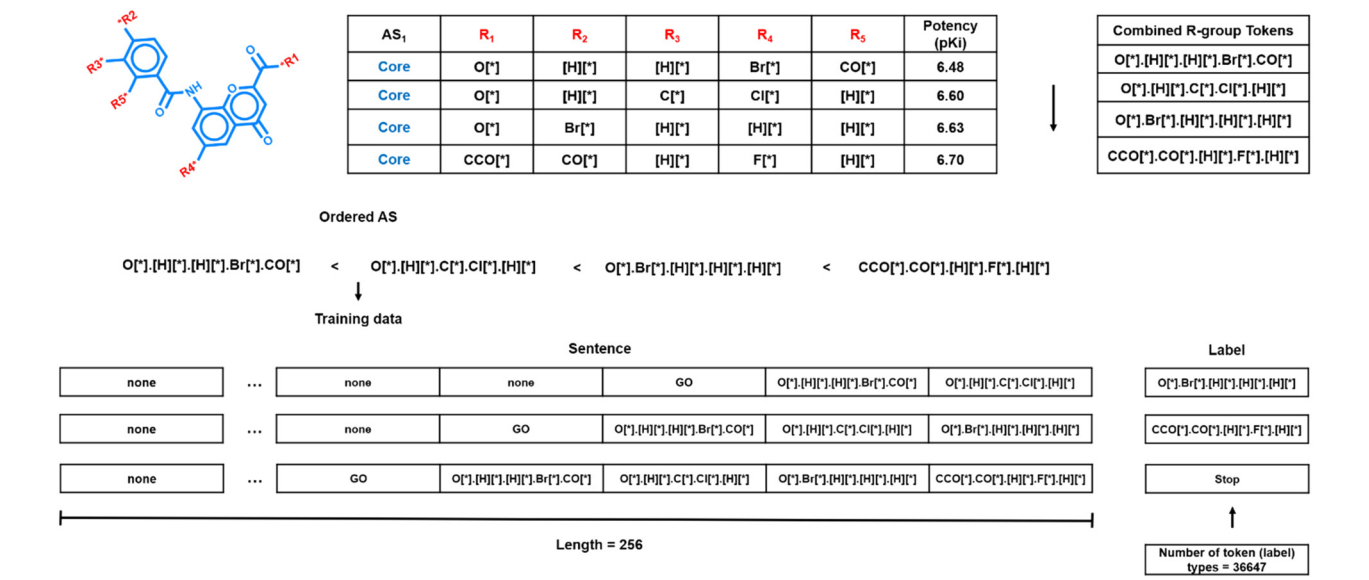


Fig. 2 Analogue series representation. The conversion of an AS into a tabular format and new textual representation for CLM derivation is illustrated.

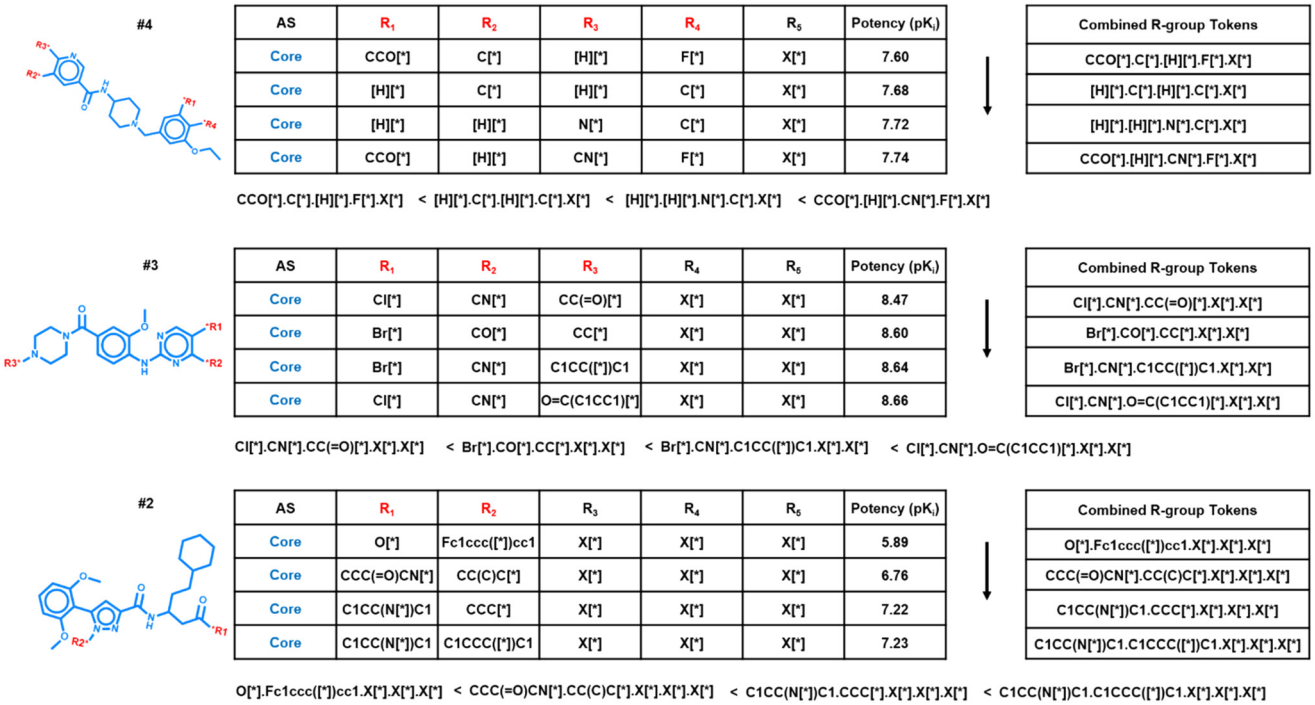


Fig. 3 Encoding of R-group combinations. AS with two, three, or four substitution sites and varying numbers of non-hydrogen R-groups are encoded into potency-ordered R-group sequences using concatenated R-group combination tokens.

vector. In the encoder layer, each word (token) exchanges information with all others through the self-attention mechanism.[26] The last layer typically consists of a fully connected dense layer, which further processes the encoder layer's output and addresses prediction-specific tasks such as text classification or next-sentence prediction.

In DeepAS 2.0, new sentences representing R-group sequences of AS were tokenized using the bidirectional maximum matching (BMM) algorithm.[36] These AS sentences served as the R-group embedding vectors in the input AS sequence, and the segmentation embedding vectors and position embedding vectors were concatenated to the input sequence. The three combined embedding vectors were then submitted to the transformer encoder to learn potency-ordered R-group sequences *via* the self-attention mechanism (Fig. 4b). The self-attention mechanism adjusted the weight of each R-group in the input AS sequence to obtain a global representation vector capturing the context. Subsequently, DeepAS 2.0 predicted labels with probabilities derived from training data *via* the softmax function of the dense layer. The model-based probability was converted into a log-likelihood score using the negative logarithm, resulting in small scores corresponding to high probabilities. For extension of an input AS, all potential R-groups were ranked based on log-likelihood scores (Fig. 4b).

Model derivation and evaluation

DeepAS 2.0 was implemented using PyTorch.[37] The Adam optimizer[38] with a learning rate of 0.0001 and a batch size of 128 was employed. Softmax was utilized as the activation function in the dense layer. Training was conducted on NVIDIA Tesla A40 (48GB) GPUs. Throughout the training process, the cross-entropy loss between the ground truth and the output sequence was minimized. The model was trained for at least 200 epochs and at the end of each epoch, a check point was saved. The final model was selected based on minimal cross-entropy loss.

Initially, DeepAS 2.0 was trained using a data set of 93 622 MMS from 2185 activity classes. As a control, the original DeepAS model was derived using the same data, thus enabling a direct comparison with DeepAS 2.0 for extending AS with single substitution sites. For model evaluation, the final R-group token was omitted from each potency-ordered test AS (not encountered during training) and predicted as the label based on the derived conditional probabilities and corresponding log-likelihood scores. The model's ability to accurately predict final R-groups within top-ranked tokens served as the primary criterion for model validation.

Subsequently, fine-tuning using DeepAS 2.0 was investigated for MMS from the activity classes in Table 1 that were excluded from training. For each activity class, AS were randomly divided into two distinct equally sized subsets without AS overlap that were used for fine-tuning and testing, respectively.

For deriving the DeepAS 2.0 version for extending AS with multiple substitution sites, termed multi-site DeepAS 2.0 (MS-DeepAS 2.0), a data set comprising 10 863 AS with one to five substitution sites from 854 activity classes for training

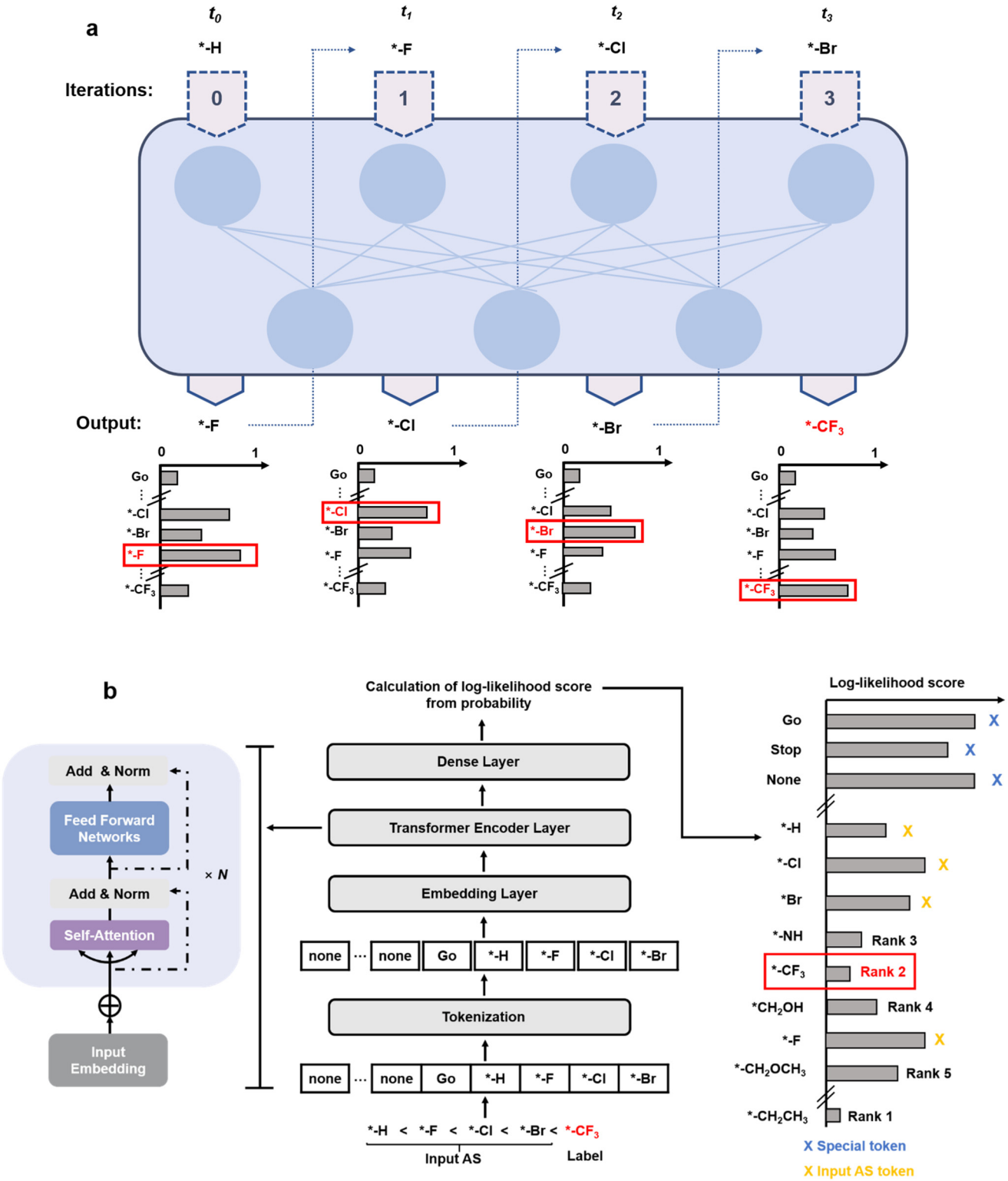


Fig. 4 Transformer model architecture and function. In (a), BERT-based iterative prediction of analogues with new R-groups for an evolving AS is illustrated. In (b), the model architecture is detailed and R-group ranking illustrated (excluding special tokens and R-group tokens from the input sequence).

and 2716 AS for testing were used. Furthermore, five activity classes with multi-site AS excluded from training were used for fine-tuning (Table 2). For each class, AS samples with one to five substitution sites were randomly divided into two distinct equally sized subsets without AS overlap for fine-tuning and testing, respectively.

Table 3 Predictions of general models

Model	Training set	Test set
DeepAS	88.8%	57.4%
DeepAS 2.0	91.0%	69.8%

Reported is the percentage of training and test MMS for which the models correctly predicted the terminal R-group within the top-5 ranked R-group tokens.

Results and discussion

Study concept

Our study had two primary objectives. Firstly, we aimed to develop a new transformer model for generative extension of MMS with potent compounds, including a fine-tuning option, and compare the performance to the original DeepAS RNN. Secondly, we aimed to expand the approach to AS with multiple substitution sites and determine if prediction of R-group combinations for potent compounds might be feasible; a challenging task. To meet these objectives, we implemented a second-generation DeepAS CLM based on the BERT architecture, devised a novel encoding scheme for transforming multi-site AS into potency-ordered R-group combination sequences, and carried out systematic test calculations.

General models for the extension of analogue series with single substitution sites

We first determined the ability of DeepAS 2.0 to predict final R-groups within the top-5 of all 3855 R-group tokens ranked based on log-likelihood scores and compared DeepAS 2.0 to the original DeepAS RNN. Both models were trained on 84 259 MMS (AS with a single substitution site) covering nearly 2200 targets and tested on 9363 distinct MMS. Table 3 reports the results obtained for the general models. While prediction accuracy on the training set was comparably high for both models (at the 90% level), DeepAS 2.0 reached significantly higher performance on the test set than the DeepAS model, with 69.8% *vs.* 57.4% correctly predicted test instances. Thus, predictions across a large target space confirmed the ability of the DeepAS approach to exploit SAR transfer information using alternative CLMs, with further improved performance of the transformer variant compared to the original RNN model.

Activity class-specific extension of analogue series with single substitution sites

We next carried out predictions for the 10 different activity classes in Table 1. Initially, the general DeepAS and DeepAS 2.0 models were used to separately extend test MMS from each class, as reported in Table 4. Both models predicted multiple final R-groups within the top-5 ranked tokens for ~40% to ~60% of test MMS from each activity class, with on average 49.2% for DeepAS and 57.0% for DeepAS 2.0, which further improved the performance of DeepAS for eight of 10 activity classes (Table 4). Furthermore, predictions were repeated with DeepAS 2.0 following class-specific fine-tuning (which was not available in DeepAS). As reported in Table 4, fine-tuning consistently increased the performance of DeepAS 2.0 for all activity classes. In seven of 10 cases, improvements of more than 10% to ~15% were observed, resulting in an average performance of 68.3% for fine-tuned DeepAS 2.0. These findings also indicated that the general DeepAS 2.0 model relying on SAR transfer information remained sensitive to target/activity class-specific SAR features.

Extension of analogue series with multiple substitution sites

At the core of our study has been the expansion of the approach to cover AS with multiple substitution sites and predict R-group combinations improving compound potency. Therefore, given its superior performance compared to the original DeepAS model, a new methodological framework was implemented in DeepAS 2.0 (see Methodology), producing the MS-DeepAS 2.0 version. Increasing the number of substitution sites increased the complexity of the prediction tasks, resulting in 36 647 R-group combinations for AS with one to five substitution sites. Moreover, for AS with increasing numbers of substitution sites, available training data decreased. Thus, both in terms of complexity

Table 4 Activity class-specific predictions

ChEMBL ID	Test MMS	DeepAS	DeepAS 2.0	Fine-tuned DeepAS 2.0
203	846	45.2%	56.4%	70.2%
247	639	48.8%	59.0%	64.0%
260	372	39.8%	46.5%	52.4%
279	959	51.1%	60.3%	73.4%
325	494	67.8%	63.6%	77.9%
2971	382	47.9%	57.3%	68.3%
3650	405	55.3%	53.8%	61.7%
3717	426	43.7%	54.9%	65.7%
4005	433	53.8%	61.0%	76.4%
4822	547	38.4%	57.6%	72.6%

Reported is the percentage of test MMS from 10 activity classes for which the models correctly predicted the terminal R-group within the top-5 ranked R-group tokens. Predictions were carried out with the general DeepAS and DeepAS 2.0 models and fine-tuned DeepAS 2.0.

and training data sparseness, multi-site AS predictions were more challenging than MMS predictions.

MS-DeepAS 2.0 was trained using 10 863 AS with one to five substitution sites from 854 activity classes, yielding a general model, and tested on corresponding 2716 AS. Given the large number of label tokens, the model's ability to predict final R-group combinations within the top-5 and top-10 ranked tokens was determined. As reported in Table 5, 62.2% and 75.6% of predicted final R-group combinations of training AS were found in the top-5 and top-10 ranked tokens, respectively, and 41.7% and 53.6% of test instances were within the top-5 and top-10 tokens, respectively. In light of the inherent complexity of these predictions, these findings were encouraging (and not necessarily anticipated). Fig. 5 shows exemplary predictions for multi-site AS.

We further analyzed model performance for AS subsets with one to five substitution sites. As reported in Table 6, for AS with increasing number of substitution sites, available training (and test) data decreased, thus further increasing the relative difficulty of predictions as more substitution sites became available. Accordingly, the proportion of training and test AS for which the terminal R-group combination was predicted among the top-5 or top-10 ranked label tokens generally decreased for AS with one to five substitution sites, as expected. For test AS, the proportion of terminal R-group combinations predicted among the top-5 and top-10 tokens declined from 64.7% to 5.4% and 77.5% to 10.4%, respectively, from one to five substitution sites. Global prediction accuracy was dominated by subsets with one to three substitution sites.

Activity class-specific extension of analogue series with multiple substitution sites

Five activity classes with largest numbers of AS having multiple substitution sites we identified (Table 2) were excluded from training and used for class-specific predictions with fine-tuning. These classes with largest numbers of multi-site AS provided as much data for class-specific learning as possible. As reported in Table 7, predictions for these five classes of GPCR ligands produced comparable results, with 20.1% to 28.4% and 26.3% to 39.3% of terminal R-group combinations predicted among the top-5 and top-10 ranked tokens, respectively.

Table 5 Predictions for analogue series with multiple substitution sites

Ranking	Training set	Test set
Top-5	62.2%	41.7%
Top-10	75.6%	53.6%

Reported is the percentage of training and test AS with one to five substitution sites for which MS-DeepAS 2.0 correctly predicted the terminal R-group combination within the top-5 or top-10 ranked R-group combination tokens.

However, the MS-DeepAS 2.0 model only failed to predict the final R-group combination within the top-5 ranked tokens for test AS with four or five substitution sites from two activity classes (but did not fail in these cases when the top-10 ranked tokens were considered). As a control, when the model was fine-tuned only on AS subsets with one to three substitution sites and used to predict test AS with four or five sites, the predictions consistently failed to identify final R-group combinations among the top-10 ranked tokens. Thus, fine-tuning on these activity classes had a detectable effect on the predictions, consistent with the observed class sensitivity of general DeepAS 2.0. However, for multi-site AS predictions, data sparseness limited the magnitude of the effect, especially for AS subsets with four or five substitution sites because in these cases, only ~10–15 training AS were available for activity classes not encountered before.

Conclusions

In this work, we have investigated the extension of AS with potent compounds using CLMs. Inspired by the ability of the original DeepAS RNN model to predict R-groups of potent analogues for series with single substitution sites, providing proof-of-concept for the approach, a BERT-based transformer variant was developed and found to further increase the performance of DeepAS in systematic R-group predictions for MMS. Exploitation of SAR transfer information provides a scientific foundation for the predictive ability of the general DeepAS and DeepAS 2.0 models and additional fine-tuning confirmed compound class sensitivity of DeepAS 2.0, providing opportunities for further developments.

Exploring a methodological framework for covering AS with multiple substitution sites was central to our current study. Therefore, a new AS encoding scheme and R-group data structure were devised and implemented in DeepAS 2.0. Despite the inherent challenges of predicting R-group combinations of potent compounds, the resulting MS-DeepAS 2.0 version accurately prioritized R-group combinations of potent analogues for many multi-site AS. Training data sparseness for AS with more than three substitution sites (which are underrepresented in compound optimization) naturally limited the predictive ability of the model in these cases. Nonetheless, proof-of-concept was also established for the extension of AS with combinations of four or five R-groups. Regardless, for practical applications in compound optimization, predicting combinations of four or five R-groups would typically be an exception. However, given the promising performance of the general MS-DeepAS 2.0 model, meaningful prioritization of varying R-group combinations should often be feasible. In light of the findings reported herein, we intend to further develop the approach by exploring model modifications targeting selected SAR transfer events and considering multi-site AS relationships.

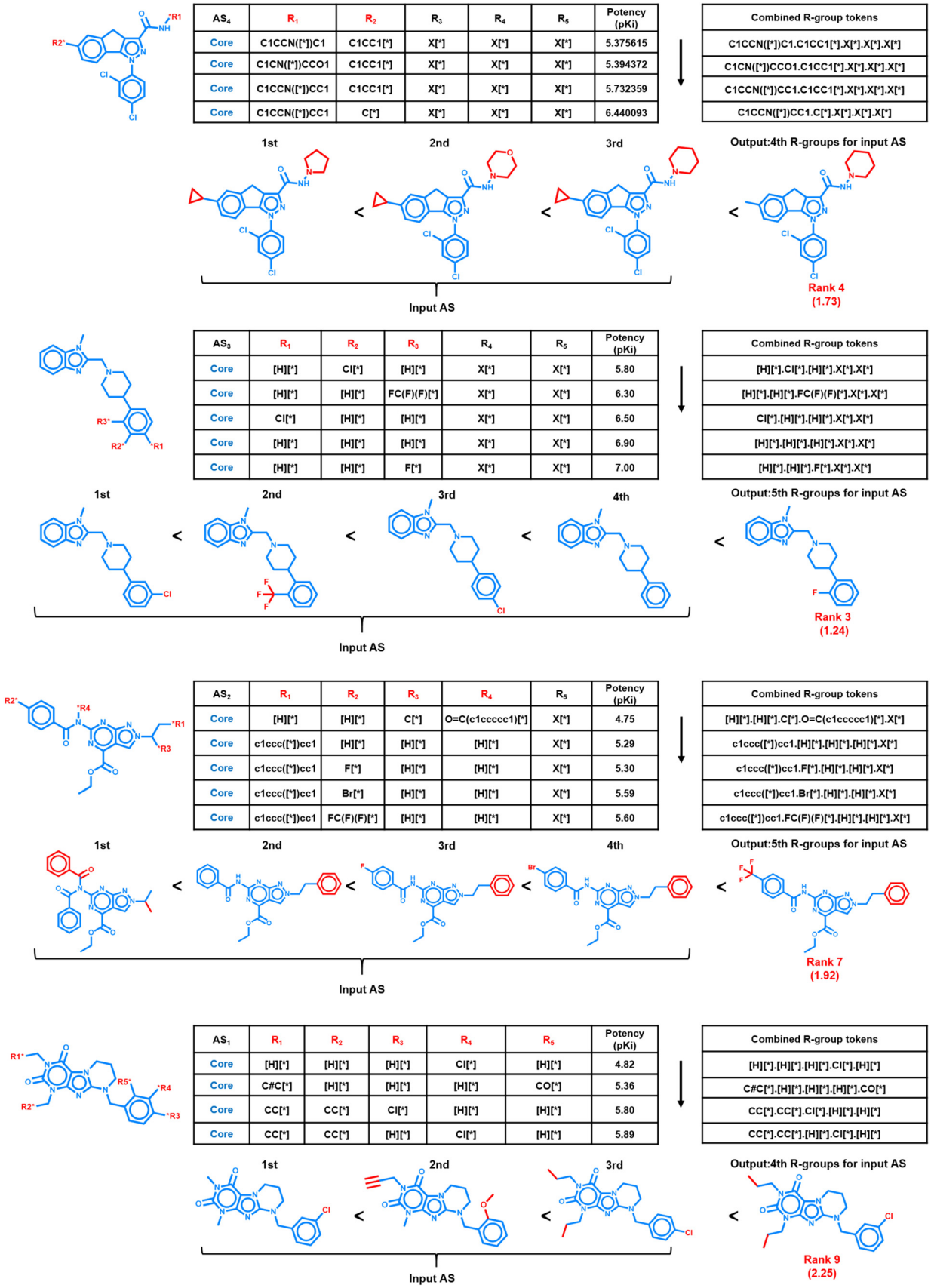


Fig. 5 Extension of exemplary analogue series with multiple substitution sites. From the top to the bottom, exemplary AS with two, three, four, and five substitution sites are shown for which MS-DeepAS 2.0 correctly predicted the R-group combination of the final analogue at rank 4, 3, 7, and 9, respectively (log-likelihood scores are reported in parentheses). Substitution sites and non-hydrogen R-groups are highlighted in red.

Table 6 Predictions for subsets of analogue series with increasing substitution sites

Substitution sites	Training set			Test set		
	AS	Top-5	Top-10	AS	Top-5	Top-10
#1	4900	78.8%	87.0%	1202	64.7%	77.5%
#2	2606	59.9%	79.8%	695	35.1%	49.6%
#3	1504	48.5%	68.3%	384	20.1%	30.7%
#4	964	34.0%	51.1%	214	9.8%	17.3%
#5	889	31.5%	39.5%	221	5.4%	10.4%

Reported is the percentage of training and test AS subsets with one to five substitution sites (#1 to #5) for which MS-DeepAS 2.0 correctly predicted the terminal R-group combination within the top-5 or top-10 ranked R-group combination tokens.

Table 7 Activity class-specific predictions for multi-site analogue series

ChEMBL ID	Top-5	Top-10
226	26.2%	39.3%
234	21.4%	29.1%
251	28.4%	38.0%
256	24.1%	33.3%
264	20.1%	26.3%

Reported is the percentage of test AS for which MS-DeepAS 2.0 correctly predicted the terminal R-group combination within the top-5 or top-10 ranked R-group combination tokens after class-specific fine-tuning.

Data availability

All compound data used in the study are publicly available.

Author contributions

HC: conceptualization, data curation, methodology, formal analysis, writing – original draft, writing – reviewing & editing; AY: methodology, writing – reviewing & editing; JB: conceptualization, methodology, supervision, writing – original draft, writing – reviewing & editing.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

H. C. is supported by the China Scholarship Council (CSC).

References

1 R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams and A. Aspuru-Guzik, *ACS Cent. Sci.*, 2018, **4**, 268–276.
2 W. P. Walters and R. Barzilay, *Acc. Chem. Res.*, 2020, **54**, 263–270.
3 H. Öztürk, A. Özgür, P. Schwaller, T. Laino and E. Ozkirimli, *Drug Discovery Today*, 2020, **25**, 689–705.
4 Z. Liu, R. A. Roberts, M. Lal-Nag, X. Chen, R. Huang and W. Tong, *Drug Discovery Today*, 2021, **26**, 2593–2607.
5 M. H. S. Segler, T. Kogej, C. Tyrchan and M. P. Waller, *ACS Cent. Sci.*, 2018, **4**, 120–131.
6 A. Yoshimori and J. Bajorath, *Bioorg. Med. Chem.*, 2022, **66**, 116808.
7 M. A. Skinnider, R. G. Stacey, D. S. Wishart and L. J. Foster, *Nat. Mach. Intell.*, 2021, **3**, 759–770.
8 F. Grisoni, *Curr. Opin. Struct. Biol.*, 2023, **79**, 102527.
9 J. Bajorath, *Mol. Inf.*, 2024, **43**, e202300288.
10 V. Bagal, R. Aggarwal, P. K. Vinod and U. D. Priyakumar, *J. Chem. Inf. Model.*, 2022, **62**, 2064–2076.
11 E. Mazuz, G. Shtar, B. Shapira and L. Rokach, *Sci. Rep.*, 2023, **13**, 8799.
12 Y. Wang, H. Zhao, S. Sciabola and W. Wang, *Molecules*, 2023, **28**, 4430.
13 L. Chen, Z. Fan, J. Chang, R. Yang, H. Hou, H. Guo, Y. Zhang, T. Yang, C. Zhou, Q. Sui, Z. Chen, C. Zheng, X. Hao, K. Zhang, R. Cui, Z. Zhang, H. Ma, Y. Ding, N. Zhang, X. Lu, X. Luo, H. Jiang, S. Zhang and M. Zheng, *Nat. Commun.*, 2023, **14**, 4217.
14 K. M. Jablonka, P. Schwaller, A. Ortega-Guerrero and B. Smit, *Nat. Mach. Intell.*, 2024, **6**, 161–169.
15 Y. Yu, J. Huang, H. He, J. Han, G. Ye, T. Xu, X. Sun, X. Chen, X. Ren, C. Li, H. Li, W. Huang, Y. Liu, X. Wang, Y. Gao, N. Cheng, N. Guo, X. Chen, J. Feng, Y. Hua, C. Liu, G. Zhu, Z. Xie, L. Yao, W. Zhong, X. Chen, W. Liu and H. Li, *ACS Med. Chem. Lett.*, 2023, **14**, 297–304.
16 L. Huang, T. Xu, Y. Yu, P. Zhao, X. Chen, J. Han, Z. Xie, H. Li, W. Zhong, K. C. Wong and H. Zhang, *Nat. Commun.*, 2024, **15**, 2657.
17 A. M. Wassermann and J. Bajorath, *J. Chem. Inf. Model.*, 2011, **51**, 1857–1866.
18 B. Zhang, Y. Hu and J. Bajorath, *J. Chem. Inf. Model.*, 2013, **53**, 1589–1594.
19 K. Umedera, A. Yoshimori, J. Bajorath and H. Nakamura, *Sci. Rep.*, 2022, **12**, 20915.
20 M. Wawer and J. Bajorath, *J. Med. Chem.*, 2011, **54**, 2944–2951.
21 A. de la Vega de León, Y. Hu and J. Bajorath, *Mol. Inf.*, 2014, **33**, 257–263.
22 A. P. Bento, A. Gaulton, A. Hersey, L. J. Bellis, J. Chambers, M. Davies, F. A. Krüger, Y. Light, L. Mak, S. McGlinchey, M. Nowotka, G. Papadatos, R. Santos and J. P. Overington, *Nucleic Acids Res.*, 2014, **42**, D1083–D1090.
23 J. J. Naveja, M. Vogt, D. Stumpfe, J. L. Medina-Franco and J. Bajorath, *ACS Omega*, 2019, **4**, 1027–1032.

24 D. Stumpfe, D. Dimova and J. Bajorath, *J. Med. Chem.*, 2016, **59**, 7667–7676.

25 X. Q. Lewell, D. B. Judd, S. P. Watson and M. M. Hann, *J. Chem. Inf. Comput. Sci.*, 1998, **38**, 511–522.

26 J. Devlin, M. W. Chang, K. Lee and K. Toutanova, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.

27 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, *Adv. Neural Inf. Process. Syst.*, 2017, **30**, 6000–6010.

28 C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li and P. J. Liu, *J. Mach. Learn. Res.*, 2020, **21**, 1–67.

29 G. Yenduri, M. Ramalingam, G. C. Selvi, Y. Supriya, G. Srivastava, P. K. Maddikunta, C. D. Raj, R. H. Jhaveri, B. Prabadevi, W. Wang, A. V. Vasilakos and T. R. Gadekallu, *IEEE Access*, 2024, **12**, 54608–54649.

30 A. S. Alammary, *Appl. Sci.*, 2022, **12**, 5720.

31 Y. Sun, Y. Zheng, C. Hao and H. Qiu, *arXiv*, 2021, preprint, arXiv:2109.03564, DOI: **10.48550/arXiv.2109.03564**.

32 B. Li, M. Lin, T. Chen and L. Wang, *Briefings Bioinf.*, 2023, **24**, bbad398.

33 R. Irwin, S. Dimitriadis, J. He and E. J. Bjerrum, *Mach. Learn.: Sci. Technol.*, 2022, **3**, 015022.

34 N. Wen, G. Liu, J. Zhang, R. Zhang, Y. Fu and X. Han, *J. Cheminform.*, 2022, **14**, 71.

35 J. Vig, A. Madani, L. R. Varshney, C. Xiong, R. Socher and N. F. Rajani, *arXiv*, 2020, preprint, arXiv:2006.15222, DOI: **10.48550/arXiv.2006.15222**.

36 M. Mao, S. Peng, Y. Yang and D. Park, *J. Inf. Process. Syst.*, 2022, **18**, 549–561.

37 A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, F. Fang, J. Bai and S. Chintala, *Adv. Neural Inf. Process. Syst.*, 2019, **32**, 8026–8037.

38 D. P. Kingma and J. Ba, *arXiv*, 2014, preprint, arXiv:1412.6980, DOI: **10.48550/arXiv.1412.6980**.

Appendix E

Combining a Chemical Language Model and the Structure−Activity Relationship Matrix Formalism for Generative Design of Potent Compounds with Core Structure and Substituent Modifications

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

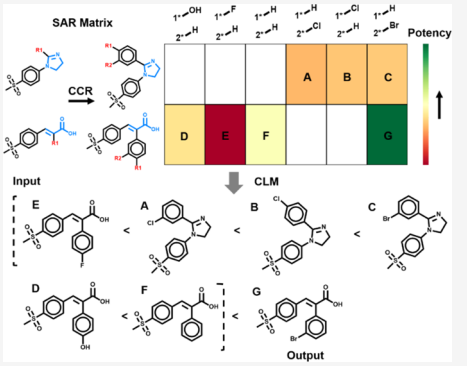
Combining a Chemical Language Model and the Structure−Activity Relationship Matrix Formalism for Generative Design of Potent Compounds with Core Structure and Substituent Modifications

Hengwei Chen and Jürgen Bajorath*

Cite This: J. Chem. Inf. Model. 2024, 64, 8784−8795 Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: In medicinal chemistry, compound optimization relies on the generation of analogue series (AS) for exploring structure−activity relationships (SARs). Potency progression is a critical criterion for advancing AS. During optimization, a key question is which analogues to synthesize next. We introduce a new computational methodology for the extension of AS with potent compounds containing both core structure and substituent modifications at multiple sites, which has been reported for the first time. The approach combines a transformer chemical language model (CLM) with a SAR matrix (SARM) methodology that identifies and organizes structurally related AS. Therefore, the SARM approach was expanded to cover multisite AS. Consensus series extracted from SARMs representing a potency gradient served as input for CLM training to extend test AS with potent analogues. Different model variants were derived and investigated. Both general and fine-tuned models correctly predicted known potent analogues at high positions in probability-based compound rankings and chemically diversified AS through core structure modifications of the generated candidate compounds and substituent replacements at multiple sites.



INTRODUCTION

In medicinal chemistry, active compounds are optimized by generating a series of structural analogues to uncover and exploit structure−activity relationships (SARs). At any stage in this process, decisions must be made as to which analogue(s) to synthesize and evaluate next. Decision support is often provided by quantitative SAR (QSAR) modeling.[1,2] Different analogue series (AS) are typically explored in the course of lead optimization. If such AS exhibit comparable potency progression, one series can often be replaced by another if these compounds have more favorable optimization-relevant characteristics, which is referred to as SAR transfer.[3] AS with SAR transfer potential can be systematically identified by computational analysis of corresponding analogues and their potency progression.[3,4] Although SARs are typically optimized for a given target, SAR transfer can also involve different targets,[5] which is a consequence of generally applied strategies to optimize compound−target interactions.[6] SAR transfer series for different targets are frequently detected.[5]

The advent of deep learning approaches in drug discovery has provided new opportunities for molecular design such as generative modeling.[7−9] Deep neural network architectures originating from natural language processing such as recurrent neural networks (RNNs)[8] and increasingly popular transformer networks[10] learn sequence-to-sequence mappings and can be adapted for new molecular design tasks that have been difficult or impossible to address using conventional machine learning methods. A variety of such models have been derived based on textual representations of molecular structure and properties that are often referred to as chemical language models (CLMs).[11−15]

As a methodological alternative to QSAR modeling, we previously introduced a CLM (termed DeepAS) for the extension of evolving AS that was conceptually based on the notion of SAR transfer across different targets.[16] The DeepAS RNN model was designed to predict the next substituent for AS in which the analogues were ordered according to their potency, thus forming an ascending potency gradient. The approach successfully reproduced AS with activity against a variety of targets from which the terminal (most potent) analogue was removed, thus establishing proof of concept.[16] However, predictions using the original DeepAS model were restricted to AS with single substitution sites that were assembled based on matched molecular pair (MMP) fragmentation of active compounds.[17] Therefore, a second-

Received: September 28, 2024
Revised: October 31, 2024
Accepted: November 11, 2024
Published: November 15, 2024

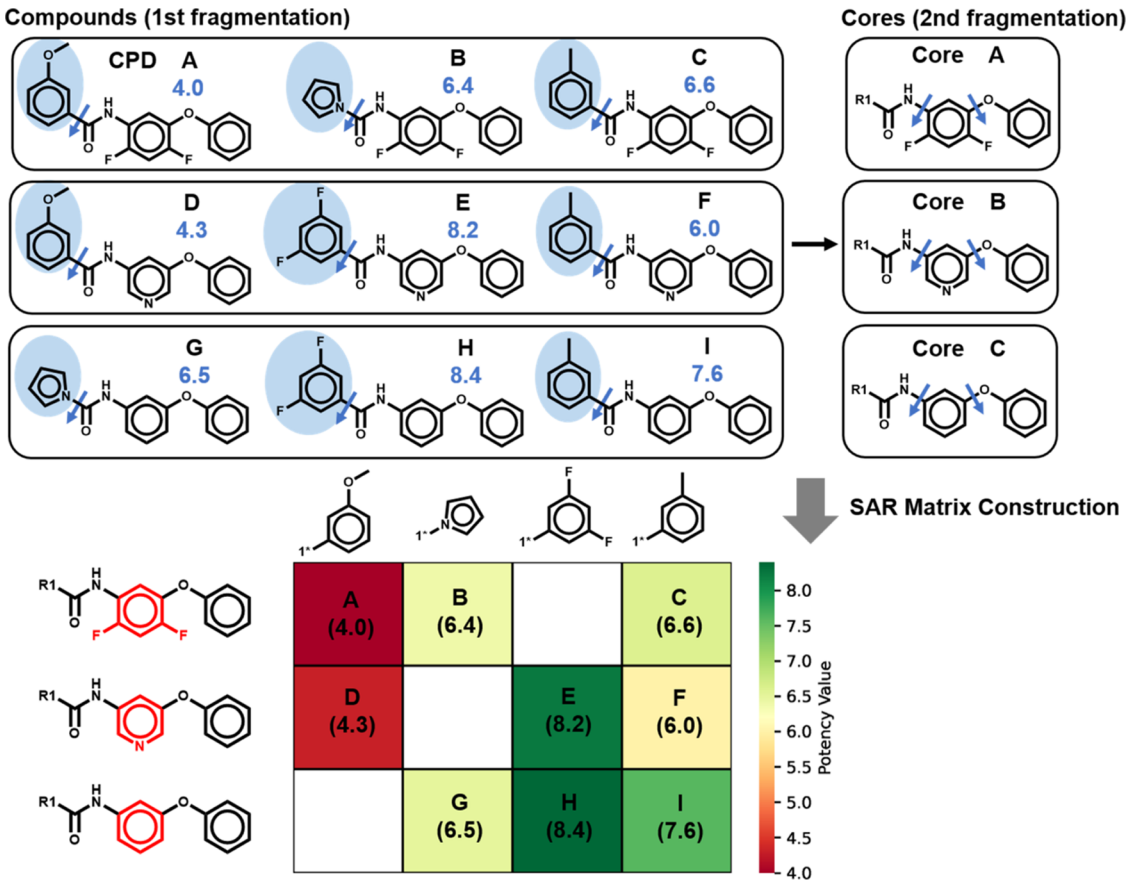


Figure 1. SAR Matrix construction. SARM construction is illustrated using a model data set comprising nine compounds (CPD A–I; pIC_{50} values are reported in blue). Substituents distinguishing analogues are shown on a light blue background. SARM generation is based on a dual-step fragmentation scheme that identifies AS with structurally related cores. Substructures distinguishing cores are shown in red. Each SARM cell represents a unique existing compound (A–I), and empty cells represent a virtual analogue, that is, a currently unexplored combination of a core (key) and substituent (value). Accordingly, virtual analogues are candidate compounds for synthesis that can be prioritized based on their SAR environments in SARMs.

generation model (termed DeepAS 2.0) was designed for the extension of AS with multiple substitution sites in which a transformer replaced the RNN architecture. Proceeding from the extension of AS with single substitution sites to AS with multiple substitution sites substantially increased the complexity of the prediction task, requiring the development of a new encoding scheme to enable the prediction of substituent combinations in terminal analogues. The DeepAS 2.0 transformer model further increased the predictive performance of the original RNN model for AS with single substitution sites and facilitated the prediction of substituent combinations for the extension of multisite AS.[18]

Building upon the successful extension of single- and multisite AS with potent compounds using CLMs, we have aimed to extend the underlying methodology for combined modification of core structures and substituents, hence departing from AS with invariant cores (scaffolds). Modification of core structures in compound series is of high relevance for compound optimization, for instance, to introduce new substitution sites and/or heteroatoms at specific positions, but difficult to model computationally (where core structure modifications are typically attempted through scaffold hopping exercises). To these ends, we have integrated the CLM approach for AS extension with the SAR matrix (SARM) formalism and data structure that was originally introduced for the systematic extraction of AS with single substitution sites from compound collections and the organization of SARMs with structurally related cores in a matrix format.[19] Accordingly, from each SARM capturing a subset of AS with structurally related cores, a potency-ordered compound series with core structure and substituent modifications can be extracted, providing a basis for CLM derivation. For the extension of multisite AS with core structure modifications, the SARM data structure was expanded to organize AS with multiple substitution sites, and a new transformer encoding scheme was devised to represent core structure-substituent combinations. Herein, we report the development of DeepAS 3.0 for structural diversification of AS and prediction of core structure modifications and substitution patterns, yielding potent analogues.

METHODOLOGY

SAR Matrix Concept. The SARM[19] methodology and data structure were originally designed to systematically extract AS with single substitution sites from compound data sets. It identifies AS with structurally analogous cores and organizes them into a matrix format similar to that of R-group tables (also referred to as SARM). Each matrix contains a set of AS with structurally analogous cores. SARMs systematically extract structural relationships from compound data sets. Depending on available structural relationships and the resulting AS, multiple SARMs are typically obtained for a data set.

https://doi.org/10.1021/acs.jcim.4c01781
J. Chem. Inf. Model. 2024, 64, 8784–8795

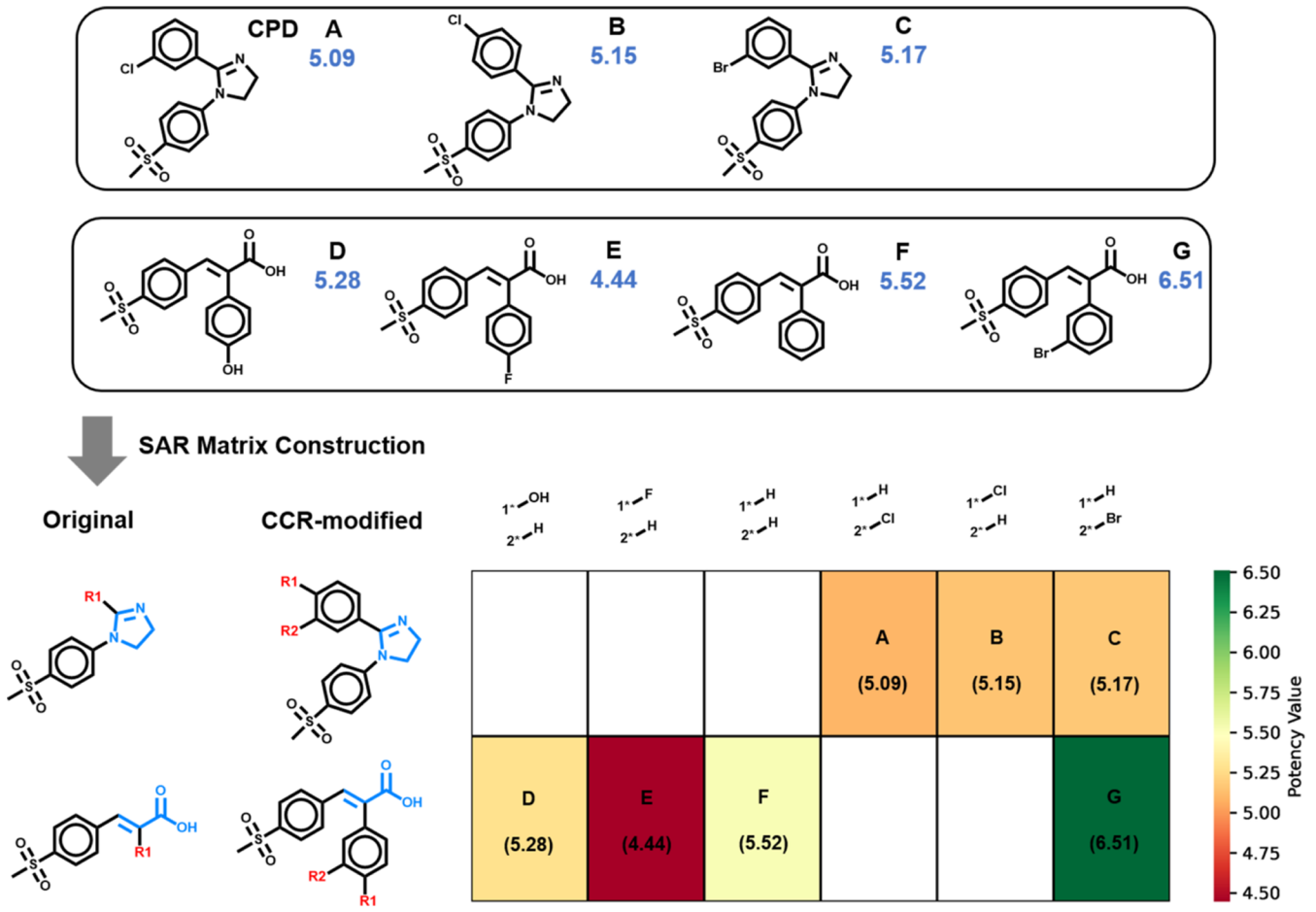


Figure 2. SARMs covering multiple substitution sites. The SARM-CCR approach is illustrated using a model data set comprising seven compounds (CPD A−G; pIC_{50} values are reported in blue) forming two AS (CPD A−C and D−G, respectively). The originally extracted structurally related cores defined the SARM that is displayed. The corresponding AS were then subjected to CCR analysis, producing modified cores with multiple substitution sites representing the SARM. Substructures from original SARM cores that are modified in CCR cores are highlighted in blue and substitution sites in red.

SARM generation follows a dual-step compound fragmentation process adapted from MMP analysis.[17] MMPs are defined as pairs of compounds that are distinguished only by a chemical modification at a single site.[17] In the first step, compounds are fragmented at exocyclic single bonds, producing "keys" (core structures) and "values" (substituents), which are stored in an index table (Figure 1). The cores are then subjected to the same fragmentation protocol again to identify subsets of cores differentiated only by a single chemical change, resulting in a second index table (Figure 1). Each subset of analogous cores and the compounds containing each core form an individual SARM. In this data structure, each row contains an AS, in which all compounds share the same core, while each column contains compounds from different ASs sharing the same substituent (Figure 1). Accordingly, SARMs consist of cells that represent every possible combination of cores and substituents within the subset of related AS. Each cell corresponds to a specific compound (key-value pair), which can be either an existing compound or a virtual analogue, that is, a currently unexplored core-substituent combination.

Thus, the SARM methodology facilitates the systematic detection of structural relationships in large compound collections, extraction of related ASs, and organization of AS with closely related cores in different matrices. Figure 1 illustrates the generation of SARMs and their information content. Cells representing existing compounds can be color-coded by potency values (or other molecular properties), enabling matrix annotations. If potency coloring is used, SARMs effectively visualize SARs contained in compound data sets.

Structurally Related Cores with Multiple Substitution Sites. Given that SARMs originated from a dual MMP fragmentation scheme, AS forming SARMs exclusively contained single substitution sites. However, AS comprising SARMs can be redefined as AS with single or multiple substitution sites and structurally related cores by integrating the SARM methodology with the compound-core relationship (CCR) algorithm,[20] which systematically identifies core structures with variable numbers of substitution sites. Therefore, CCR applies a further extended MMP-based fragmentation approach[21] using retrosynthetic rules,[22] substitution site masking, and indexing to identify multisite AS in compound data sets.[20] The SARM-CCR approach begins with originally defined SARMs and applies the CCR algorithm to the AS forming each SARM to extract redefined multisite cores (with one to three possible substitution sites), as illustrated in Figure 2. Based on these alternative cores, SARMs cover AS with multiple substitution sites having structurally related cores. To our knowledge, this combined SARM-CCR structural decom-

https://doi.org/10.1021/acs.jcim.4c01781
J. Chem. Inf. Model. 2024, 64, 8784−8795

Table 1. Activity Classes Were Selected for Fine-Tuning[a]

			SARM				
UniProt ID	ChEMBL ID	target name	#1	#2	#3	#mix	total
P00533	203	epidermal growth factor receptor erbB1	42	105	67	52	266
P22303	220	acetylcholinesterase	71	98	23	32	224
P35354	230	cyclooxygenase-2	60	116	20	58	254
Q16539	260	MAP kinase p38 α	31	68	43	27	169
P35968	279	vascular endothelial growth factor receptor 2	101	131	48	47	327
Q13547	325	histone deacetylase 1	125	107	25	50	307
P08253	333	matrix metalloproteinase-2	35	69	28	30	162
P06276	1914	butyrylcholinesterase	67	63	16	34	180
P27338	2039	monoamine oxidase B	54	270	20	32	376
P08581	3717	hepatocyte growth factor receptor	45	93	22	27	187

[a]For each activity class, the UniProt ID, ChEMBL target ID, target name, number of SARMs with one, two, or three and varying numbers ("mixed") of substitution sites (#1, #2, #3, and #mix), and the total number of SARMs are reported.

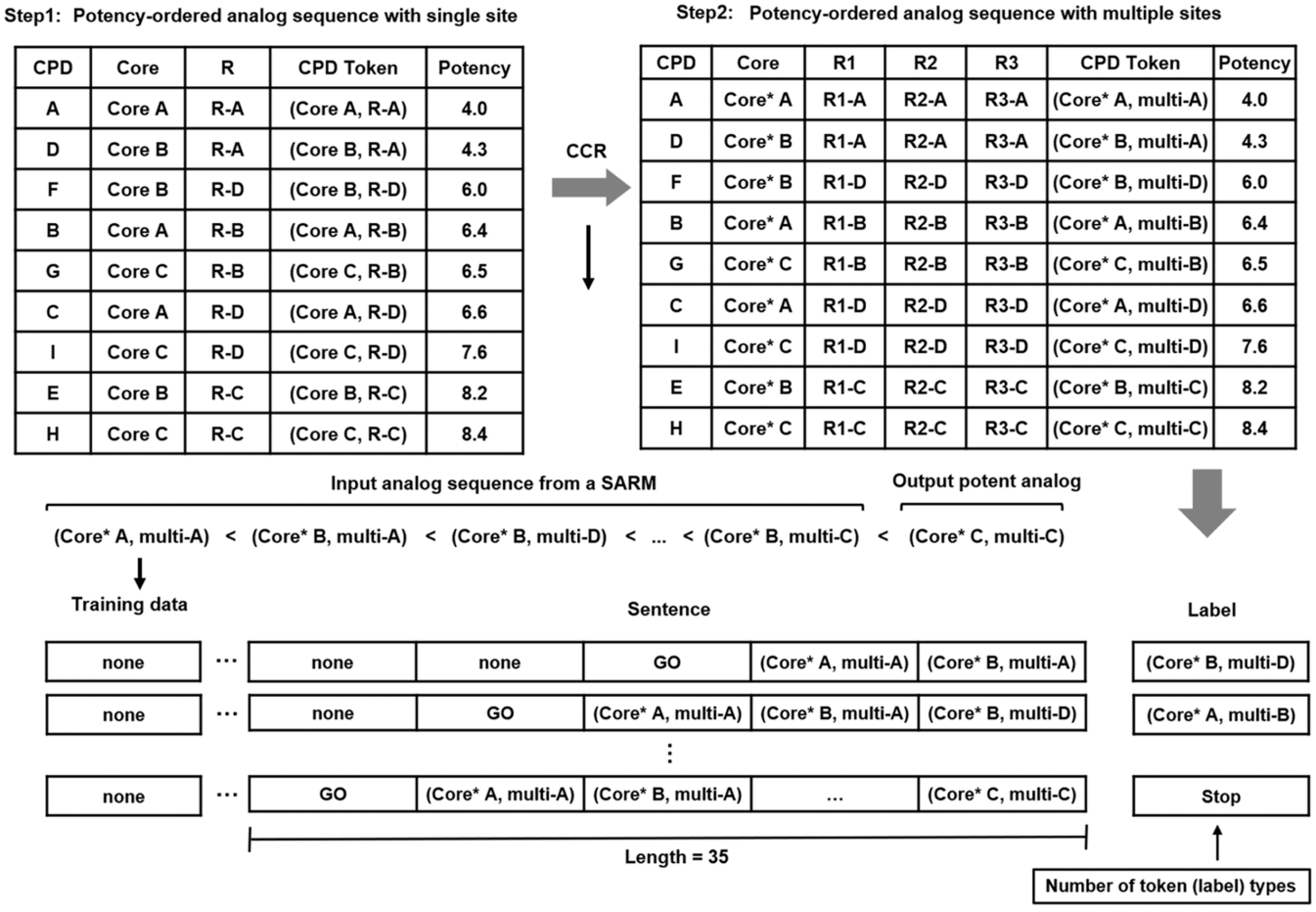


Figure 3. SARM sequence representation. The conversion of a SARM into a tabular format and new textual representation for CLM derivation is illustrated.

position represents a novel concept for systematically identifying multisite AS with structurally related cores and organizing them according to their relationships for SAR exploration and analogue design. As further discussed below, for a given SARM, these AS can be combined and potency-ordered, yielding a "consensus" series of compounds with structurally related cores and varying substitution patterns. We note that a consensus series with core structure modifications does not represent an AS with an invariant core but is still categorized as an AS, given the presence of close (SARM-based) core structure relationships.

Systematic Identification of Analogue Series for Model Derivation and Evaluation. From ChEMBL[23] (release 34), target-based compound activity classes were preselected, applying several criteria to ensure the availability of high-confidence activity data. All active compounds were required to have a molecular mass of less than 1000 Da. Targets regarded as undesirable, such as drug-metabolizing cytochrome P450 isoforms, hERG, or serum albumin, were not selected. Additionally, compounds flagged as "not active," "inactive," "inconclusive," or with potential errors (e.g., author or transcription errors) were disregarded. Only compounds

https://doi.org/10.1021/acs.jcim.4c01781
J. Chem. Inf. Model. 2024, 64, 8784−8795

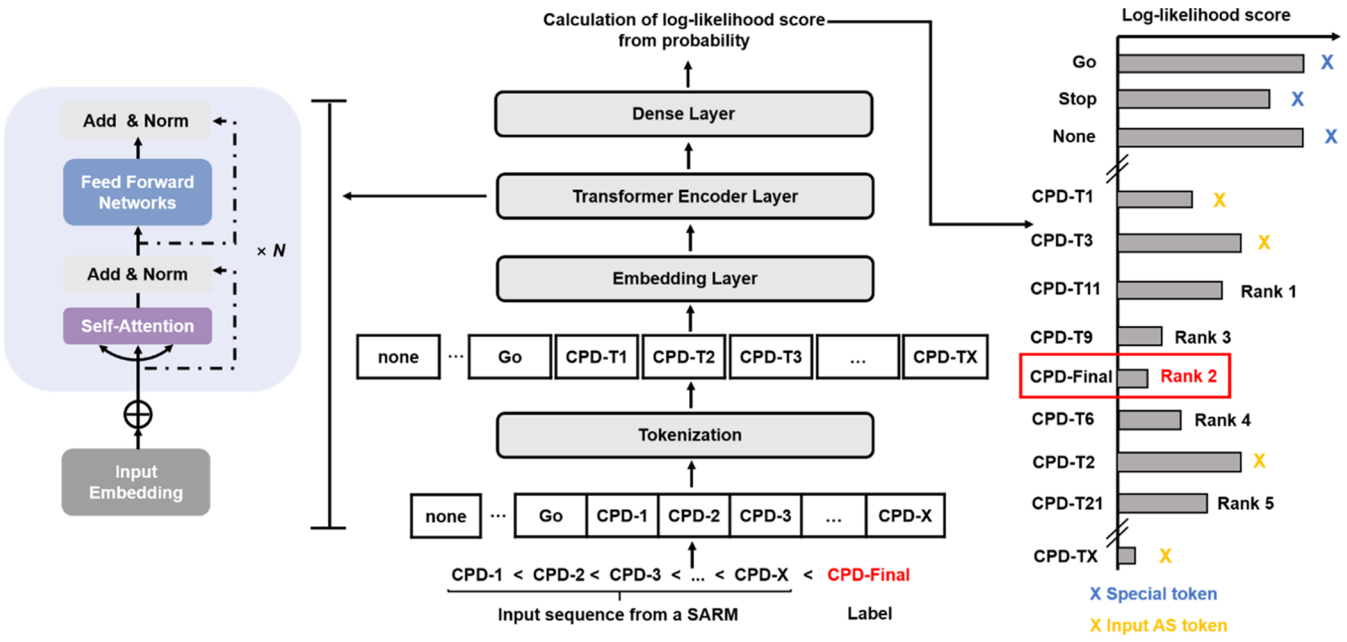


Figure 4. Transformer variant architecture. The BERT model architecture is summarized, and analogue ranking is illustrated (excluding special tokens and analogue tokens from the input sequence).

with direct target interactions (target relationship type: "D") tested in single-protein assays with the highest ChEMBL assay confidence score of 9 were retained. Moreover, availability of a numerically specified IC_{50} potency measurement with standard relation ("=") was required, recorded as a logarithmic pIC_{50} value. For compounds with multiple measurements, the geometric mean was calculated for the final potency annotation, provided all pIC_{50} values fell within the same order of magnitude; otherwise, such compounds were excluded. Finally, activity classes were examined for compounds that might cause assay artifacts using the Lilly medicinal chemistry rules,[24] filters for pan-assay interference compounds,[25] and filters for aggregators.[26]

From the resulting 2895 target-based activity classes, a total of 19,556 SARMs were generated via the SARM-CCR approach, each organizing a subset of multisite AS with related cores. From this large pool, 10 activity classes, each yielding at least 160 SARMs, were randomly selected for CLM model fine-tuning and testing (Table 1). These classes encompassed a range of receptor ligands and enzyme inhibitors. The remaining activity classes were used to generate and evaluate the transformer model for extending multisite AS with scaffold modifications and new R-group combinations.

Representing SARM as Potency-Ordered Analogue Sequences. The analogue design approach focuses on probability-based prediction of core structures and substituent modifications using a predefined pool of chemical vocabulary encompassing all potential core-substituent combinations derived from qualifying SARMs and does not generate SMILES sequences de novo. Consequently, we opted for substructure-based (rather than atom-level) tokenization. For each SARM, multisite AS with structurally related cores were combined, and the analogues were arranged in the order of increasing potency, yielding a consensus series for a SARM, as discussed above. The consensus series represents a potency-ordered analogue sequence as the model input. Accordingly, each analogue was encoded as a compound token consisting of its core and substituent combination. Following our previous approach,[18] R-groups at different substitution sites were represented as an individual (unique) R-group combination, hence ensuring consistent application to AS with varying numbers of substitution sites. For CLM derivation, each SARM-based consensus series was encoded as a sentence in which each analogue was represented as a core-substituent combination by an individual compound token (Figure 3).

Four additional special tokens were introduced, including "Go" and "Stop" to mark the beginning and end of a sentence, respectively, "none" to denote an empty token, and "X" to represent the absence of a substitution site. A sentence was required to contain at least two compound tokens. Each encoded consensus series consisted of a token sequence (sentence) and a terminal label, which indicated the next analogue compound to be added (that is, the prediction for a given input sentence). The sentence length was standardized to 35 tokens, and the total number of label tokens amounted to 95,910, covering all possible analogues extracted from qualifying SARMs and the four special tokens.

Data augmentation involved transforming each analogue sequence into multiple sentences by incrementally adding compound tokens. This expanded each training instance into sentences capturing an increasing number of compound tokens (two, three, ... all compound tokens). The final label represented the next analogue to be predicted, with the "Stop" token indicating the completion of the sequence following analogue predictions.

Model Architecture and Implementation. A transformer[10] variant based upon the Bidirectional Encoder Representations from Transformers (BERT) architecture[27] was adopted from our previous study,[18] as illustrated in Figure 4. BERT consists of three main components, including an embedding layer, multiple transformer encoder layers, and a task-specific output layer.[27] In the embedding layer, each input token is transformed into a continuous vector space using a matrix, and a predefined positional encoding vector is added to each token's embedding to capture sequential information. The transformer encoder layer utilizes the self-attention

https://doi.org/10.1021/acs.jcim.4c01781
J. Chem. Inf. Model. 2024, 64, 8784−8795

mechanism,[11] allowing each token to exchange information with all other tokens in the sequence, thereby enhancing contextual learning. The final layer is a fully connected dense layer that further processes the output from the encoder and focuses on prediction-specific tasks, such as next-sentence prediction, which is conceptually related to AS extension. BERT's bidirectional nature enables the analysis of a sentence in both forward and backward directions, rendering this architecture a preferred solution for predicting new tokens based on the context of an entire sentence.

In our current model, termed DeepAS 3.0, new sentences representing analogue sequences of SARMs were tokenized using the bidirectional maximum matching (BMM) algorithm.[28] Consistent with bidirectional data processing using BERT, the BMM algorithm (originally introduced for a special application in natural language processing[28]) conducts bidirectional token matching, thereby identifying the longest matches while comparing results from both directions to resolve potential ambiguities and derive the optimal token sequence. Therefore, by integrating a predefined vocabulary with BMM, we anticipated achieving high tokenization accuracy for domain-specific chemical language terms. The SARM sentences were transformed into analogue embedding vectors that were concatenated with segmentation embedding vectors and position embedding vectors to form the input sequence. These combined vectors were then submitted to the transformer encoder, where the self-attention mechanism learned the potency-ordered analogue sequences (Figure 4). The self-attention mechanism assigns weights to each analogue in the input sequence, creating a global representation that captures the overall context. Once the input SARM sequence is processed, DeepAS 3.0 predicts the next analogue based on probabilities generated by the softmax function in the dense layer. These model-based probabilities were then converted to log-likelihood scores by applying a negative logarithm such that smaller scores corresponded to higher probabilities. For extending an input SARM, all potential analogue compounds were ranked based on their log-likelihood scores (Figure 4).

DeepAS 3.0 was implemented using PyTorch[29] and the Adam optimizer,[30] with a learning rate of 0.0001 and a batch size of 128. As stated above, softmax served as the activation function in the dense layer. Training was performed on a NVIDIA Tesla A40 (48G) GPU. The model was trained for a minimum of 200 epochs, with a checkpoint saved at the end of each epoch. The final model was selected based on the minimum cross-entropy loss between the ground truth and the predicted output sequence.

Model Derivation and Evaluation. DeepAS 3.0 was initially trained using a global data set of 17,140 SARMs, consisting of 132,310 analogue sequences from 2885 activity classes. For model evaluation, the final analogue token was removed from each potency-ordered test sequence (not encountered during training) and predicted based on the model's conditional probabilities and corresponding log-likelihood scores. The primary validation criterion for the model was its ability to correctly predict the final analogue within the top-ranked tokens. Therefore, model performance was assessed based on the ability to correctly predict the final analogue of the test series within the top-ranked label tokens from the ranking of all possible tokens. As a control, 20 label tokens were randomly selected for each test series and examined for the presence of the correct final analogue. Performance analysis was carried out for the four subsets with varying numbers of substitution sites. Furthermore, individual models were also developed for each subset to explore their relative predictive ability.

Fine-tuning of DeepAS 3.0 was subsequently carried out using analogue sequences from the activity classes in Table 1 that were excluded from pre-training. For each activity class, SARMs from each subset were evenly divided into two nonoverlapping sets of equal size, one for fine-tuning and the other for testing. The corresponding analogue sequences were also divided into two equally sized nonoverlapping fine-tuning and test sets. The individual models fine-tuned on subsets were evaluated in activity-class-specific predictions using 3-fold cross-validation. As a control, the pre-trained global model was also fine-tuned and tested using these subsets. Therefore, for each activity class, the four subsets of SARMs with varying substitution sites were combined into a single general fine-tuning test set. This setup ensured that the same test data were used for evaluating the global and individual models, thereby providing a direct comparison of their performance after fine-tuning.

RESULTS AND DISCUSSION

Generative Analogue Design Concept. The analogue design strategy from our previous study[18] involved probability-based prioritization of substituent combinations to produce potent compounds for extending evolving AS with multiple substitution sites. In this method, relative potency was implicitly captured by ordering the AS according to the increasing compound potency. Accordingly, each AS and its corresponding sequence of substituent combinations followed an ascending potency gradient. Given that CLM-based probabilities of substituent combinations depended on the preceding sequence of combinations, the design of a new substituent combination was order-dependent, following the potency gradient. This approach aligned the design process with the goal of finding increasingly potent analogues, thus bridging between generative modeling and property-based optimization. Leveraging the input data format (sentences), a generative model was supposed to learn conditional probabilities for substituent combinations based on the potency-ordered sequences in which they occurred. In our current study, this approach was further extended to generate core structure modifications within evolving AS, in addition to substituent combinations. These core structure modifications introduce unprecedented chemical diversity in computer-aided lead optimization, for which the assembly of a SARM-based consensus series provided the basis. No current QSAR-based compound design method can introduce core structure modifications. Chemical diversity was further increased by facilitating replacements of the R-group combinations. This extension aimed to further advance generative analogue design through combined chemical modification of related core structures and substitution patterns.

Global Models. We first assessed the performance of the global general DeepAS 3.0 model for extending multisite consensus AS. From potency-ordered test AS, the terminal (last) analogue token (core-substituent combination) was removed and predicted by ranking all 95,910 compound (label) tokens based on log-likelihood scores. The model was trained on consensus series from 13,683 SARMs consisting of 105,857 analogue sequences with one to three substitution sites from 2885 activity classes, yielding a global general model. The test set for the global model comprised 3421 SARMs with

https://doi.org/10.1021/acs.jcim.4c01781
J. Chem. Inf. Model. 2024, 64, 8784−8795

Table 2. Subset-Based Predictions of the Global Model and Individual Models[a]

	training sequences (%)			test sequences (%)		
subsets	top-5	top-10	top-20	top-5	top-10	top-20
single-site	30.0 (39.4)	47.4 (58.9)	59.6 (69.6)	15.6 (21.5)	33.2 (39.3)	39.8 (51.5)
dual-site	36.1 (43.8)	53.7 (60.3)	67.7 (73.2)	23.6 (29.3)	39.9 (42.5)	48.6 (55.1)
triple-site	27.9 (30.8)	40.9 (46.2)	53.6 (59.2)	13.2 (15.9)	28.3 (34.7)	33.7 (39.6)
mixed	32.1 (40.0)	50.6 (56.7)	62.6 (65.7)	16.7 (23.0)	33.4 (36.4)	38.9 (47.1)

[a]For SARM subsets with consensus AS having different number of substitution sites, the percentage of training and test sequences for which the terminal analogue was present among top-5, -10, or -20 ranked tokens is reported for the global general model and individual general models derived for each subset (in parentheses).

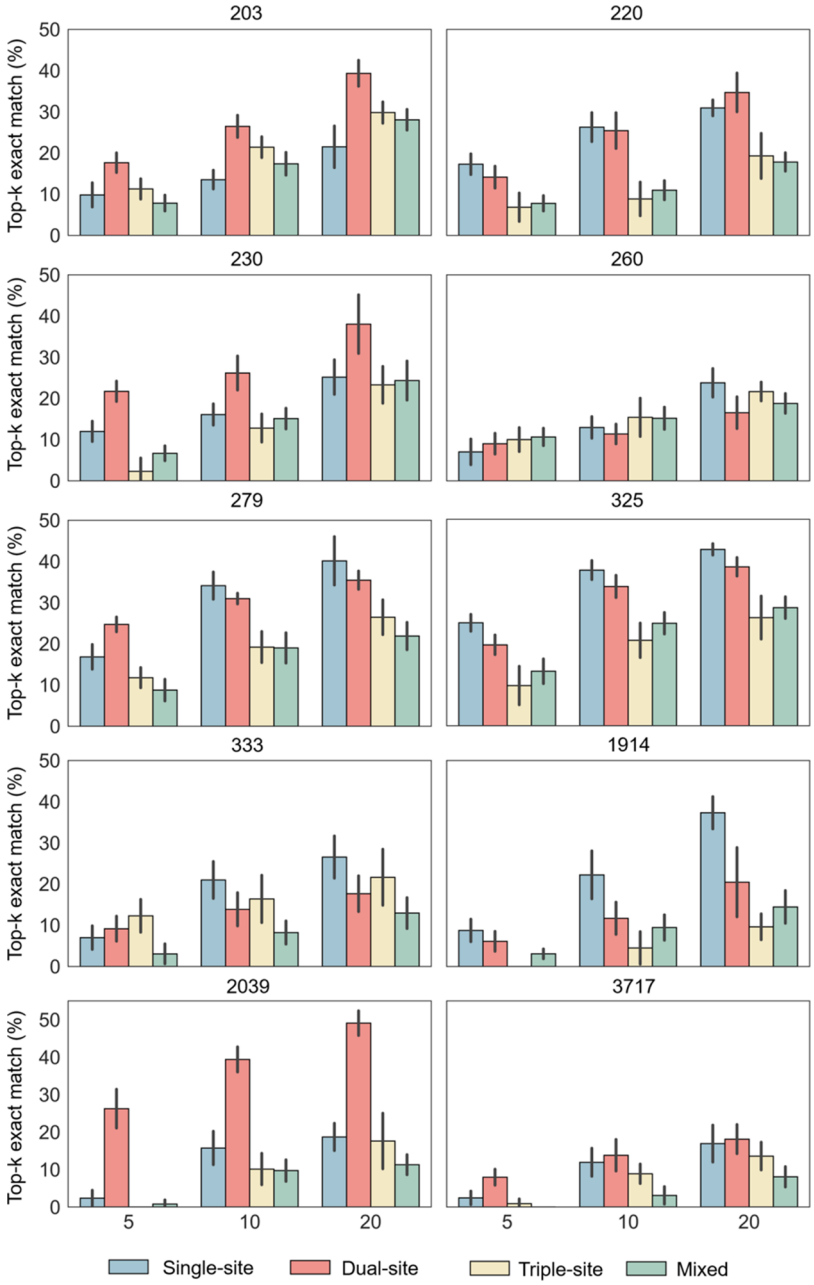


Figure 5. Performance of fine-tuned individual models. For each activity class, the performance of the fine-tuned subset-based models is summarized. On the *x*-axis, top-5, -10, and -20 token rankings are separately shown for all four models, reporting the percentage of correct terminal analogues of test sequences present in each ranking (*y*-axis). The mean values and standard deviations (error bars) are provided following cross-validation. Activity classes are identified according to Table 1.

26,453 analogue sequences. Given the very large set of label tokens, the model's ability to predict correct terminal core-substituent combinations within the top-5, top-10, and top-20 ranked tokens was determined, respectively. For 32.3, 50.3, and

https://doi.org/10.1021/acs.jcim.4c01781
J. Chem. Inf. Model. 2024, 64, 8784−8795

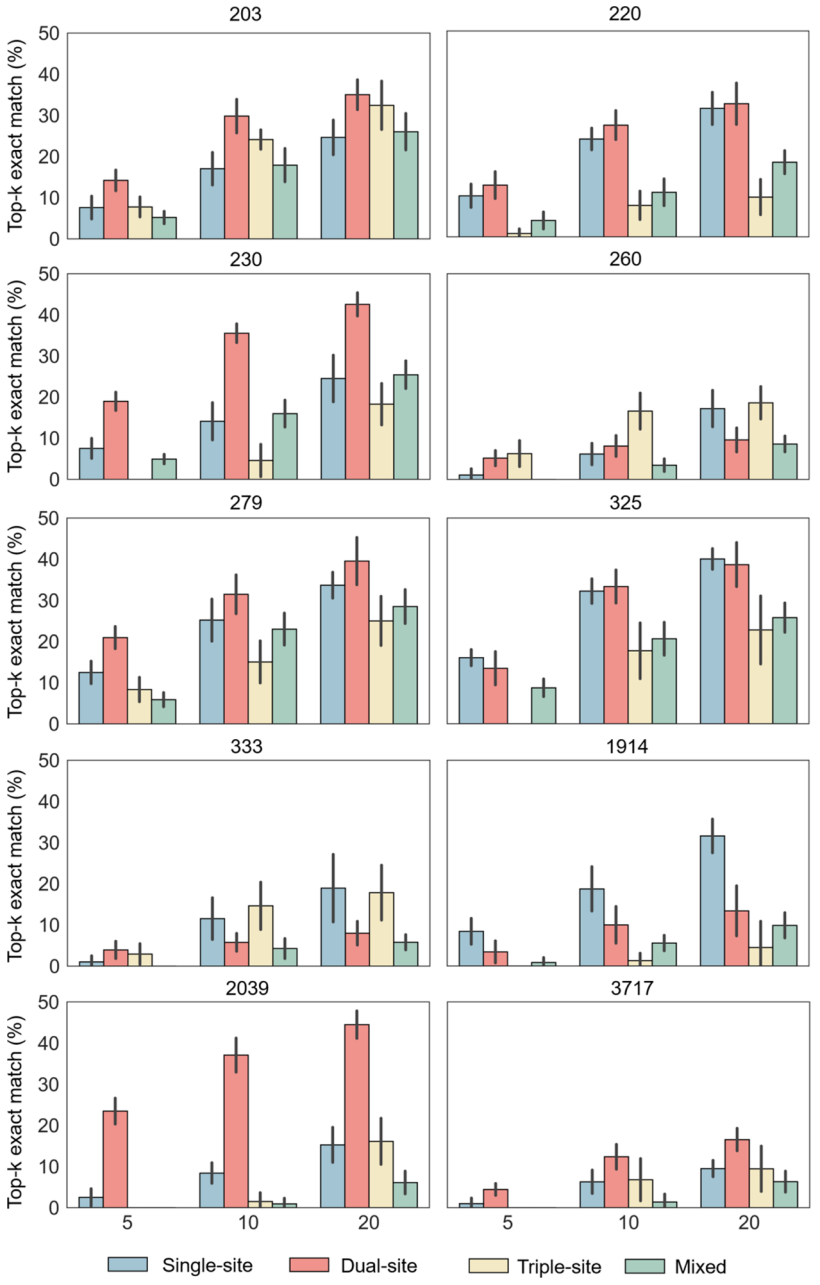


Figure 6. Performance of the fine-tuned global model. For each activity class, the performance of the fine-tuned global model monitored across different subsets is summarized. The presentation is the same as in Figure 5. The mean values and standard deviations (error bars) are provided following cross-validation. Activity classes are identified according to Table 1.

63.2% of the training analogue sequences, the correct terminal analogues were contained in the top-5, top-10, and top-20 ranked tokens, respectively. Furthermore, for 19.7, 33.1, and 39.8% of the test analogue sequences, the correct terminal analogues were found within the top-5, top-10, and top-20 tokens, respectively. As a control, a random selection of 20 label tokens yielded the correct terminal analogues for only very small numbers of four, 10, and 20 training sequences and two, five, and seven test sequences within the top-5, top-10, and top-20 tokens, respectively. Considering the inherent complexity of core-substituent pattern predictions, these results were considered promising. Correctly predicted training sequences within the top-5, top-10, and top-20 ranked tokens covered 29.7, 47.6, and 58.3% of all SARM-based series, respectively. For correctly predicted test sequences, the corresponding numbers were 18.5, 31.7, and 35.0% of all series, indicating broad coverage of different SAR environments captured by SARMs.

Training and test sets of the global model were then organized into four subsets according to the number of substitution sites in SARM-based AS, including single-site, dual-site, triple-site, and "mixed" series (that is, combining AS with different numbers of substitution sites). Dual-site SARMs formed the largest subset, with a total of 8250 SARMs with 49,469 AS, followed by 3941 single-site SARMs (31,007 AS), 2585 mixed (34,616 AS), and 2328 triple-site SARMs (17,218 AS).

The performance of the global model was analyzed for each subset, as reported in Table 2. For consensus AS from dual-site SARMs, correctly predicted terminal analogues were contained

https://doi.org/10.1021/acs.jcim.4c01781
J. Chem. Inf. Model. 2024, 64, 8784−8795

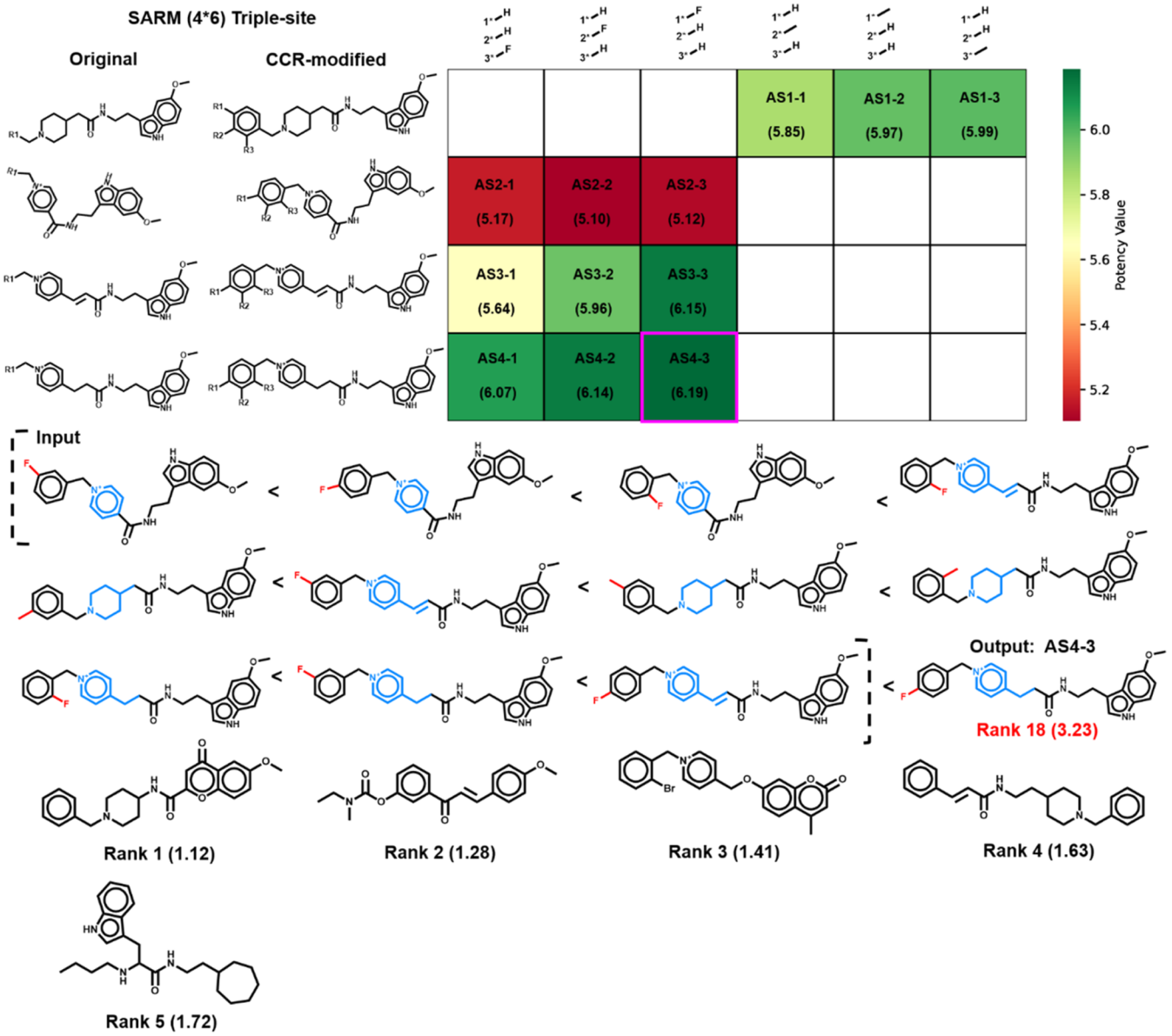


Figure 7. Extension of the consensus series from a triple-site SAR matrix. Shown are a CCR-modified triple-site SARM of the target butyrylcholinesterase and the consensus series obtained by combining and potency ordering the SARM AS. Substructures from original SARM cores that were extended in multisite cores and varying substituents are colored blue and red, respectively. The terminal analogue was predicted by the fine-tuned subset-based model at rank 18 (log-likelihood score in parentheses). At the bottom, the top-5 ranked candidate compounds generated by the model are shown.

in the top-5, top-10, and top-20 tokens for 36.1, 53.7, and 67.7% of all training sequences and 23.6, 39.9, and 48.6% of all test sequences, respectively. For consensus AS from triple-site SARMs, the corresponding percentages were 27.9, 40.9, and 53.6% for all training and 13.2, 28.3, and 33.7% for all test sequences. The reduction observed for sequences from triple-site SARMs was expected because less training data were available than for dual-site SARMs. The prediction accuracy for consensus AS from single-site SARMs fell between those from dual- and triple-site SARMs. For sequences with varying numbers of substitution sites (from mixed SARMs), the global model predicted terminal analogues of 16.7, 33.4, and 38.9% of all test instances within the top-5, top-10, and top-20 ranked tokens, despite the increasing complexity of the predictions.

For comparison, we then built individual general models for each subset based on the respective SARMs and corresponding consensus series reported above. Sequences from single-, dual-, triple-site, and mixed SARMs contained a total of 23,291, 36,560, 14,727, and 26,794 compound tokens, respectively. Compared to the global model, there was a consistent increase in the percentage of sequences with highly ranked correct terminal analogues for the subset-based models of ∼3 to ∼12% (with less than 10% in most cases), as also reported in Table 2. While less training data were available for the individual models, their prediction tasks were simpler than for the global model, also taking into consideration that much lower numbers of label tokens were available for the subset-based models. However, since the consistent improvements were only moderate, there was no substantial advantage of the subset-based general models compared to the global general model. The proportion of SARMs covered by the predictions was also

https://doi.org/10.1021/acs.jcim.4c01781
J. Chem. Inf. Model. 2024, 64, 8784−8795

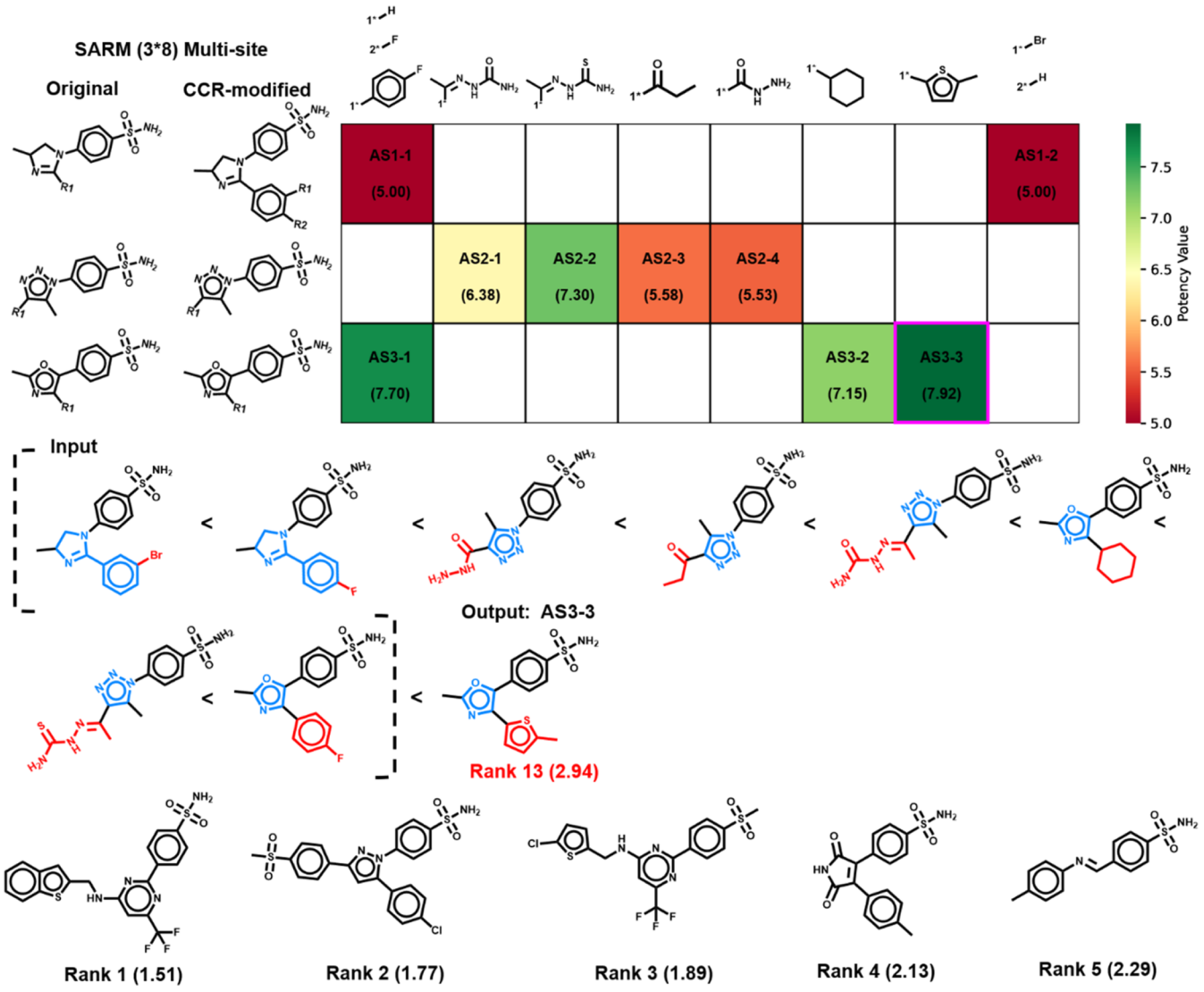


Figure 8. Extension of a consensus series from a multisite SAR matrix. Shown is a CCR-modified multisite SARM for the target cyclooxygenase-2 and the consensus series obtained by combining and potency ordering the SARM AS. Substructures from original SARM cores that were extened in multisite cores and varying substituents are colored blue and red, respectively. The terminal analogue was predicted by the fine-tuned subset-based model at rank 13 (log-likelihood score in parentheses). At the bottom, the top-5 ranked candidate compounds generated by the model are shown.

similar for the individual models and the global model (as reported above).

Fine-Tuned Models. The individual models were then used for fine-tuning on the activity classes in Table 1 that were excluded from pre-training. The number of SARMs (corresponding AS) used for fine-tuning and testing varied from 8 to 135 (45 to 625), depending on the activity class and subset. For each activity class, the four subsets of consensus series originating from SARMs with different numbers of substitution sites were combined and then partitioned into single fine-tuning and test sets for fine-tuning and evaluation of the global general model. Figure 5 summarizes the results for fine-tuned subset-based models. For seven of 10 activity classes, terminal core-substituent combinations were found within the top-5 ranked label tokens across all subsets. For eight activity classes, model performance was higher for the single- and dual-site subsets than for the others. Specifically, for the single-site subsets, an average of 2.3−24.9% of the terminal analogues were found within the top-5 ranked tokens, 11.9−37.7% within the top-10, and 16.9−42.7% within the top-20. For the dual-site subsets, an average of 6.1−26.2% of the terminal analogues were present in the top-5 ranked tokens, 11.4−39.4% in the top-10, and 16.5−49.1% in the top-20. For the triple-site and mixed subsets, the predictive performance was, on average, overall lower by ∼10%. The triple-site subset model failed to predict correct terminal analogues within the top-5 ranked tokens for test sequences from two activity classes, and the mixed subset model failed for test sequences from one class. For mixed subsets, prediction accuracy was inherently limited by the increasing complexity of label token space compared to individual subsets and, in addition, by the sparseness of training data for AS with three substitution sites. Nonetheless, both the triple-site and mixed subset models also consistently predicted terminal analogues within the top-10 ranked tokens for all activity classes.

As a control, the global general model was also fine-tuned using a single fine-tuning set for each activity class and evaluated using the corresponding test set. For each activity class, the predictive performance was separately monitored for the four different subsets, as shown in Figure 6. Here, correct

https://doi.org/10.1021/acs.jcim.4c01781
J. Chem. Inf. Model. 2024, 64, 8784−8795

terminal analogues within the top-5 ranked label tokens were only detected across all subsets for three of 10 activity classes, and the predictive performance was overall lower than observed for the fine-tuned individual models, albeit by only small margins in several cases. Given the smaller number of label tokens available for subset models compared to the global model, as discussed above, the performance of the fine-tuned global model was similar to that of subset models. However, for fine-tuning on individual activity classes, we assign preference to subset models in light of their more consistent predictions of terminal analogues among the top-5 ranked label tokens.

Extension of Consensus Series with Substituent and Core Structure Modifications. A hallmark of the DeepAS 3.0 design approach is the ability to generate compounds with multiple substituents as well as core structure modification, setting it apart from its precursor, which extended AS with invariant cores and multiple substitution sites. The additional ability to generate core structure modifications represents a key feature of DeepAS 3.0 methodology. This was facilitated through the (i) CCR-based modification of core structures from SARMs to obtain individual AS with multiple substitution sites and (ii) generation of consensus series by combining all multisite AS from a given SARM and potency ordering of their analogues. Accordingly, consensus series serving as input for generative design consisted of compounds with structurally related yet distinct cores and varying substitution sites. In our calculations, DeepAS 3.0 was found to consistently generate candidate compounds with chemical modification (diversification) of cores and substituent patterns, consistent with the underlying design ideas. Figure 7 shows an exemplary CCR-modified triple-site SARM and the corresponding consensus series for which the terminal analogue was predicted at rank 18. In addition, the top-5 ranked candidate compounds are shown, illustrating the chemical diversification potential. Furthermore, Figure 8 shows an exemplary multisite SARM and its consensus series, for which the terminal analogue was predicted at rank 13. The top-5 candidate compounds also displayed desired chemical modifications of cores and substituents, thus reinforcing the DeepAS 3.0 approach.

CONCLUSIONS

In this work, we have introduced a methodology for the extension of compound series with potent analogues containing core structure and substituent modifications at multiple sites. The prediction of compounds with core structure and substituent modifications based on template series representing potency gradients was a challenging task. By adding the new SARM-CCR approach and data structure as a front end to a transformer CLM, structurally related AS organized in SARMs were combined into consensus series as input for CLM derivation. For this purpose, the SARM formalism was further advanced by devising a new compound decomposition protocol to cover AS with multiple substitution sites. Furthermore, a new CLM coding and tokenization scheme was designed to represent core structure-substituent combinations. We derived a global general CLM and four other models for subsets of AS with different numbers of substitution sites, thus simplifying the prediction task. The global and subset-based models predicted terminal analogues of test series at high ranks based on log-likelihood scores from model-internal probability distributions. Overall, the predictive performance of the global general model and the general subset-based models was similar, demonstrating the ability of the bidirectional transformer to learn the chemical space of compound series with extensive structural variations. Fine-tuning of models on AS from activity classes excluded from pre-training also yielded promising predictions, with a confined but consistent performance increase for the subset-based models over the global fine-tuned model. Test calculations using the general and fine-tuned models yielded a wealth of candidate compounds and confirmed the ability of these models to introduce a variety of core structure and substituent modifications and further chemically diversify input series. In light of our findings, the DeepAS 3.0 approach further advances the extension of AS with invariant cores by introducing core modifications in AS with multiple substitution sites and should have considerable potential for practical applications in medicinal chemistry and drug design.

ASSOCIATED CONTENT

Data Availability Statement

All data and code used for our analysis are freely available via the following link: https://uni-bonn.sciebo.de/s/xOXKsBgUisQCD9j.

AUTHOR INFORMATION

Corresponding Author

Jürgen Bajorath − *Department of Life Science Informatics and Data Science, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, University of Bonn, D-53115 Bonn, Germany; Lamarr Institute for Machine Learning and Artificial Intelligence, University of Bonn, D-53115 Bonn, Germany;* orcid.org/0000-0002-0557-5714; Phone: 49-228-7369-100; Email: bajorath@bit.uni-bonn.de

Author

Hengwei Chen − *Department of Life Science Informatics and Data Science, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, University of Bonn, D-53115 Bonn, Germany; Lamarr Institute for Machine Learning and Artificial Intelligence, University of Bonn, D-53115 Bonn, Germany;* orcid.org/0009-0008-5678-0874

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.4c01781

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

H.C. was supported by the China Scholarship Council.

REFERENCES

(1) Lewis, R. A. A General Method for Exploiting QSAR Models in Lead Optimization. *J. Med. Chem.* **2005**, *48*, 1638−1648.

(2) Cumming, J. G.; Davis, A. M.; Muresan, S.; Haeberlein, M.; Chen, H. Chemical Predictive Modelling to Improve Compound Quality. *Nat. Rev. Drug Discovery* **2013**, *12*, 948−962.

(3) Wassermann, A. M.; Bajorath, J. A Data Mining Method to Facilitate SAR Transfer. *J. Chem. Inf. Model.* **2011**, *51*, 1857−1866.

(4) Zhang, B.; Wassermann, A. M.; Vogt, M.; Bajorath, J. Systematic Assessment of Compound Series with SAR Transfer Potential. *J. Chem. Inf. Model.* **2012**, *52*, 3138−3143.

(5) Yoshimori, A.; Bajorath, J. Computational Method for the Systematic Alignment of Analogue Series with Structure-Activity Relationship Transfer Potential Across Different Targets. *Eur. J. Med. Chem.* **2022**, *239*, No. 114558.

(6) Topliss, J. G. A Manual Method for Applying the Hansch Approach to Drug Design. *J. Med. Chem.* **1977**, *20*, 463−469.
(7) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **2018**, *4*, 268−276.
(8) Segler, M. H. S.; Kogej, T.; Tyrchan, C.; Waller, M. P. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Cent. Sci.* **2018**, *4*, 120−131.
(9) Walters, W. P.; Barzilay, R. Applications of Deep Learning in Molecule Generation and Molecular Property Prediction. *Acc. Chem. Res.* **2021**, *54*, 263−270.
(10) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000−6010.
(11) Skinnider, M. A.; Stacey, R. G.; Wishart, D. S.; Foster, L. J. Chemical Language Models Enable Navigation in Sparsely Populated Chemical Space. *Nat. Mach. Intell.* **2021**, *3*, 759−770.
(12) White, A. D. The Future of Chemistry is Language. *Nat. Rev. Chem.* **2023**, *7*, 457−458.
(13) Grisoni, F. Chemical Language Models for de Novo Drug Design: Challenges and Opportunities. *Curr. Opin. Struct. Biol.* **2023**, *79*, No. 102527.
(14) Bajorath, J. Chemical Language Models for Molecular Design. *Mol. Inf.* **2024**, *43*, No. e202300288.
(15) Jablonka, K. M.; Schwaller, P.; Ortega-Guerrero, A.; Smit, B. Leveraging Large Language Models for Predictive Chemistry. *Nat. Mach. Intell.* **2024**, *6*, 161−169.
(16) Yoshimori, A.; Bajorath, J. DeepAS − Chemical Language Model for the Extension of Active Analogue Series. *Bioorg. Med. Chem.* **2022**, *66*, No. 116808.
(17) Hussain, J.; Rea, C. Computationally Efficient Algorithm to Identify Matched Molecular Pairs (MMPs) in Large Data Sets. *J. Chem. Inf. Model.* **2010**, *50*, 339−348.
(18) Chen, H.; Yoshimori, A.; Bajorath, J. Extension of Multi-site Analogue Series with Potent Compounds using a Bidirectional Transformer-based Chemical Language Model. *RSC Med. Chem.* **2024**, *15*, 2527−2537.
(19) Wassermann, A. M.; Haebel, P.; Weskamp, N.; Bajorath, J. SAR Matrices: Automated Extraction of Information-Rich SAR Tables from Large Compound Data Sets. *J. Chem. Inf. Model.* **2012**, *52*, 1769−1776.
(20) Naveja, J. J.; Vogt, M.; Stumpfe, D.; Medina-Franco, J. L.; Bajorath, J. Systematic Extraction of Analogue Series from Large Compound Collections Using a New Computational Compound−Core Relationship Method. *ACS Omega* **2019**, *4*, 1027−1032.
(21) Stumpfe, D.; Dimova, D.; Bajorath, J. Computational Method for the Systematic Identification of Analog Series and Key Compounds Representing Series and Their Biological Activity Profiles. *J. Med. Chem.* **2016**, *59*, 7667−7676.
(22) Lewell, X. Q.; Judd, D. B.; Watson, S. P.; Hann, M. M. RECAP - Retrosynthetic Combinatorial Analysis Procedure: A Powerful New Technique for Identifying Privileged Molecular Fragments with Useful Applications in Combinatorial Chemistry. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 511−522.
(23) Bento, A. P.; Gaulton, A.; Hersey, A.; Bellis, L. J.; Chambers, J.; Davies, M.; Krüger, F. A.; Light, Y.; Mak, L.; McGlinchey, S.; Nowotka, M.; Papadatos, G.; Santos, R.; Overington, J. P. The ChEMBL Bioactivity Database: An Update. *Nucleic Acids Res.* **2014**, *42*, D1083−D1090.
(24) Bruns, R. F.; Watson, I. A. Rules for Identifying Potentially Reactive or Promiscuous Compounds. *J. Med. Chem.* **2012**, *55*, 9763−9772.
(25) Baell, J. B.; Holloway, G. A. New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays. *J. Med. Chem.* **2010**, *53*, 2719−2740.
(26) Irwin, J. J.; Duan, D.; Torosyan, H.; Doak, A. K.; Ziebart, K. T.; Sterling, T.; Tumanian, G.; Shoichet, B. K. An Aggregation Advisor for Ligand Discovery. *J. Med. Chem.* **2015**, *58*, 7076−7087.
(27) Devlin, J.; Chang, M. W.; Lee, K.; Toutanova, K.In *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019; pp 4171−4186.
(28) Mao, M.; Peng, S.; Yang, Y.; Park, D. Bi-directional Maximal Matching Algorithm to Segment Khmer Words in Sentence. *J. Inf. Process. Syst.* **2022**, *18*, 549−561.
(29) Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, F.; Bai, J.; Chintala, S. PyTorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026−8037.
(30) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization, 2014. arXiv:1412.6980. https://arxiv.org/abs/1412.6980v9.

https://doi.org/10.1021/acs.jcim.4c01781
J. Chem. Inf. Model. 2024, 64, 8784−8795

Appendix F

Generative Design of Compounds with Desired Potency from Target Protein Sequences using a Multimodal Biochemical Language Model

Chen and Bajorath
Journal of Cheminformatics (2024) 16:55
https://doi.org/10.1186/s13321-024-00852-x

Journal of Cheminformatics

RESEARCH

Open Access

Generative design of compounds with desired potency from target protein sequences using a multimodal biochemical language model

Hengwei Chen[1] and Jürgen Bajorath[1*]

Abstract

Deep learning models adapted from natural language processing offer new opportunities for the prediction of active compounds via machine translation of sequential molecular data representations. For example, chemical language models are often derived for compound string transformation. Moreover, given the principal versatility of language models for translating different types of textual representations, off-the-beaten-path design tasks might be explored. In this work, we have investigated generative design of active compounds with desired potency from target sequence embeddings, representing a rather provoking prediction task. Therefore, a dual-component conditional language model was designed for learning from multimodal data. It comprised a protein language model component for generating target sequence embeddings and a conditional transformer for predicting new active compounds with desired potency. To this end, the designated "biochemical" language model was trained to learn mappings of combined protein sequence and compound potency value embeddings to corresponding compounds, fine-tuned on individual activity classes not encountered during model derivation, and evaluated on compound test sets that were structurally distinct from training sets. The biochemical language model correctly reproduced known compounds with different potency for all activity classes, providing proof-of-concept for the approach. Furthermore, the conditional model consistently reproduced larger numbers of known compounds as well as more potent compounds than an unconditional model, revealing a substantial effect of potency conditioning. The biochemical language model also generated structurally diverse candidate compounds departing from both fine-tuning and test compounds. Overall, generative compound design based on potency value-conditioned target sequence embeddings yielded promising results, rendering the approach attractive for further exploration and practical applications.

Scientific contribution

The approach introduced herein combines protein language model and chemical language model components, representing an advanced architecture, and is the first methodology for predicting compounds with desired potency from conditioned protein sequence data.

Keywords Deep learning, Molecular design, Protein language model, Conditional transformer, Active compounds

*Correspondence:
Jürgen Bajorath
bajorath@bit.uni-bonn.de
Full list of author information is available at the end of the article

Introduction

In drug discovery, compound optimization requires the comprehensive evaluation of multiple physicochemical and in vivo properties such as affinity, hydrophobicity, solubility, toxicity, pharmacogenetics, and pharmacodynamics [1]. Experimental efforts to assess and optimize these molecular properties are supported by computational approaches [2], with quantitative structure–activity relationship (QSAR) analysis being a classical methodology for compound affinity prediction [3, 4], mostly focusing on congeneric compounds and progression of hit-to-lead or lead series.

In recent years, machine learning (ML) including deep learning (DL) has increasingly been considered for activity and property predictions in drug discovery [5], leading to the application of various neural network (NN) methods such as convolutional NN (CNN) [6], recurrent neural NN (RNN) [7], graph convolutional network (GCN) [8], or message passing NN (MPNN) [9]. DL methods including those employed for property predictions generally benefit from the availability of large data sets for learning the multitude of internal weights they require. However, such data sets are for the most part unavailable in early-phase drug discovery where data sparseness often hinders the use of DL models and limits the accuracy of their predictions [10]. In addition, the assessment of ML methods for quantitative compound potency predictions in typical benchmark settings poses considerable challenges. Notably, benchmark potency predictions by ML/DL models of varying complexity and randomized predictions are often only differentiated by small error margins [11], thus complicating an unambiguous assessment of relative method performance [11]. As a consequence of data sparseness and intrinsic limitations in method evaluation and comparison, there currently are no generally applicable criteria or guidelines available for prioritizing ML approaches for quantitative molecular property predictions in drug discovery.

Property predictions can also be combined with generative modeling of new compounds [12], which provides a conceptual alternative to conventional property prediction strategies. For example, to this end, we have developed specialized transformer models, as further detailed below. In computer science, transformers originated from the field of natural language processing where they were used for the conversion of an input sequence of characters into an output sequence with the aid of self-attention (importance) mechanisms [13]. Transformer architectures are increasingly employed in other fields for various machine translation tasks. A transformer-based compound design concept investigated in our laboratory was semi-quantitative in nature. It aimed at deriving models for predicting potent compounds for targets of interest without specifying numerical potency values across wide ranges, thereby circumventing some of the obstacles associated with benchmark compound potency predictions [11]. Previously, we derived transformer-based chemical language models (CLMs) for molecular string-to-string conversion conditioned on potency differences between pairs of structural analogues [14, 15]. So-called conditional transformer models not only learn conditional probabilities for character sequence translation, but also for other context-dependent rules (such as molecular property constraints). Our rules included potency difference thresholds required for the formation of activity cliffs (i.e., analogue pairs having largest potency differences in compound activity classes) [14] or -in a generalized form- desired potency difference thresholds structural analogues [15]. In the latter case, transformer models were trained based on large numbers of analogue pairs with greatly varying potency differences. In both instances, conditional transformers consistently reproduced highly potent compounds from activity cliffs or other compound pairs for a variety of activity classes, thus providing proof-of-principle, and generated other structurally diverse candidate compounds [14, 15]. On the basis of these findings, we extended this transformer architecture for generative modeling of potent compounds by a meta-learning framework for modeling in low compound data regimes [16].

In addition to learning compound-to-compound mappings for predicting new active or highly potent compounds, various attempts have been made to establish direct links between biological targets and chemical entities with DL models using representations combining protein sequence and compound information [17–22]. These models were often derived to distinguish true target-ligand complexes from false (randomly assembled) complexes. Potential applications of such models include target validation or compound repurposing. Furthermore, in recent studies, transformer-based language models have been employed to learn mappings of protein sequences to compounds [22–25]. In the following, models using protein sequence data as input are termed protein language models (PLMs), regardless of the nature of the output sequences. Sequence-to-compound modeling aimed to revitalize the concept of sequence-based compound design [22] that was investigated during the early days of drug design but was then for long out of fashion in drug discovery settings, for scientific reasons. Notably, only limited numbers of residues in protein sequences are typically implicated in ligand binding and only high global sequence similarity indicates similar ligand binding characteristics of targets. Hence, designing active compounds based on sequence data is challenging and partly controversial, perhaps not even possible without

additional knowledge, and difficult to pursue using standard ML methods. However, the advent of PLMs has made it possible to have a fresh look at this scientifically provoking design task. For example, a transformer was adapted to associate the primary structures of target proteins with known active compounds and predict new ones [23]. Compounds were represented as Simplified Molecular Input Line Entry System (SMILES) strings [26], a mainstay textual representation. In another study, an Lmser network-based transformer variant incorporating multi-head cross attention blocks was developed to map complete protein sequences to active compounds [24]. The encoder processed information from the protein sequence and the resulting latent space was decoded into compound SMILES. In addition, compound generation was combined with Monte Carlo tree search [24]. In both of these studies, conventional protein–ligand docking scores were used to guide compound prioritization. In a different investigation, a transformer was derived to associate extended sequence motifs of ligand binding sites with active compounds [25]. In this case, the ability of the model to exactly reproduce ATP site-directed inhibitors of different kinases not included in model training was used as a proof-of-concept criterion (instead of hypothetical scoring). Notably, the definition of sequence motifs directly implicated in compound binding requires prior (structural) knowledge.

Following principles from natural language processing, PLMs embed long protein sequences as sentences of characters in which one or more residues form words [27, 28]. The resulting sequence embeddings are thought to implicitly capture much information concerning structural and functional characteristics of proteins, rendering these embeddings attractive for a variety of applications [29, 30].

Given our previous studies of chemical language models for predicting potent compounds and the applications of PLMs discussed above, we have been interested in exploring the possibility to combining these approaches and investigating whether compounds with pre-defined potency could also be designed using a conditional transformer architecture and protein sequence data. To this end, we have developed and assessed a new so-called biochemical language model for learning from multimodal data, as presented in the following.

Methods

Targets, compounds, and activity data

Compounds with high-confidence activity data were selected from ChEMBL (release 33) [31]. Only compounds engaged in direct interactions (assay relationship type: "D") with human targets at the highest assay confidence level (assay confidence score 9) were considered. Potency measurements were restricted to numerically specified equilibrium constants (K_i values) and recorded as negative logarithmic pK_i values. In cases where multiple measurements were available for the same compound, the geometric mean was calculated as the final potency annotation, contingent on all values falling within the same order of magnitude; otherwise, the compound was excluded from further consideration. Qualifying compounds were divided into target-based activity classes. Only targets with a maximal (monomer) sequence length of 4000 residues were considered. On the basis of these data curation criteria, 1575 activity classes were obtained, comprising a total of 87,839 unique compounds. For each activity class, the protein sequence of the target was extracted in FASTA format from UniProt [32] using an in-house script. Compounds were represented as canonical SMILES strings generated using RDKit [33]. From the large activity class pool, 10 classes with at least close to 400 compounds were randomly selected as test cases for generative design (Table 1). These activity classes

Table 1 Activity classes for model evaluation

ChEMBL ID	Target name	Compounds
204	Thrombin	454
218	Cannabinoid CB1 receptor	1118
234	Dopamine D3 receptor	1529
244	Coagulation factor X	702
251	Adenosine A2a receptor	1825
1862	Tyrosine-protein kinase ABL	499
4005	PI3-kinase p110-alpha subunit	576
5113	Orexin receptor 1	1086
1,075,104	Leucine-rich repeat serine/threonine-protein kinase 2	397
1,908,389	Mitogen-activated protein kinase kinase kinase 12	404

For each of 10 activity classes, the number of compounds, ChEMBL target ID, and target name are reported

included ligands G protein-coupled receptors and inhibitors of different enzymes.

Model architecture

For our prediction task, we devised a new multimodal conditional compound generator combining two language model components. Its characteristic feature is the design of compounds with desired potency based on protein sequence information conditioned on compound potency values. To our knowledge, this scheme represents a previously unconsidered design concept and, in addition, the first instance of a language model conditioned on molecular context rules from chemistry applied to biological sequences (thus also incorporating multimodality). The model architecture is schematically depicted in Fig. 1. A pre-trained PLM generating protein sequence embeddings (component 1) was combined with a conditional transformer (component 2) challenged to learn mappings of combined protein and potency values embeddings to compounds (SMILES strings) with corresponding activity against a given target. Accordingly, the transformer should predict compounds from target sequence embeddings having a desired potency level. Since the generator bridges between protein sequence information with compound activity constraints and chemical structure, it is termed a "multimodal biochemical language model". In the following, the two model components are described in more detail.

Protein language model for generating embeddings

Sequence embeddings should capture distributions of vast numbers of amino acid sequences of proteins, residue frequencies, and positional dependencies. Hence, they should implicitly encode characteristic features related to biophysical properties, structure, and function. For our study, we adapted as model component 1 the pre-trained ProtT5XLUniref50 PLM from ProtTrans [29] with default dimensionality of 1024. ProtTrans PLMs were originally derived based on ultra-large sequence data sets from UniRef [34] and BFD [35], comprising up to 2122 million proteins and 393 billion amino acids. Each protein sequence was initially tokenized and then subjected to positional encoding. The resulting vector was processed to generate context-aware embeddings for each input token (amino acid). These embeddings, extracted from the last hidden state of a PLM's attention

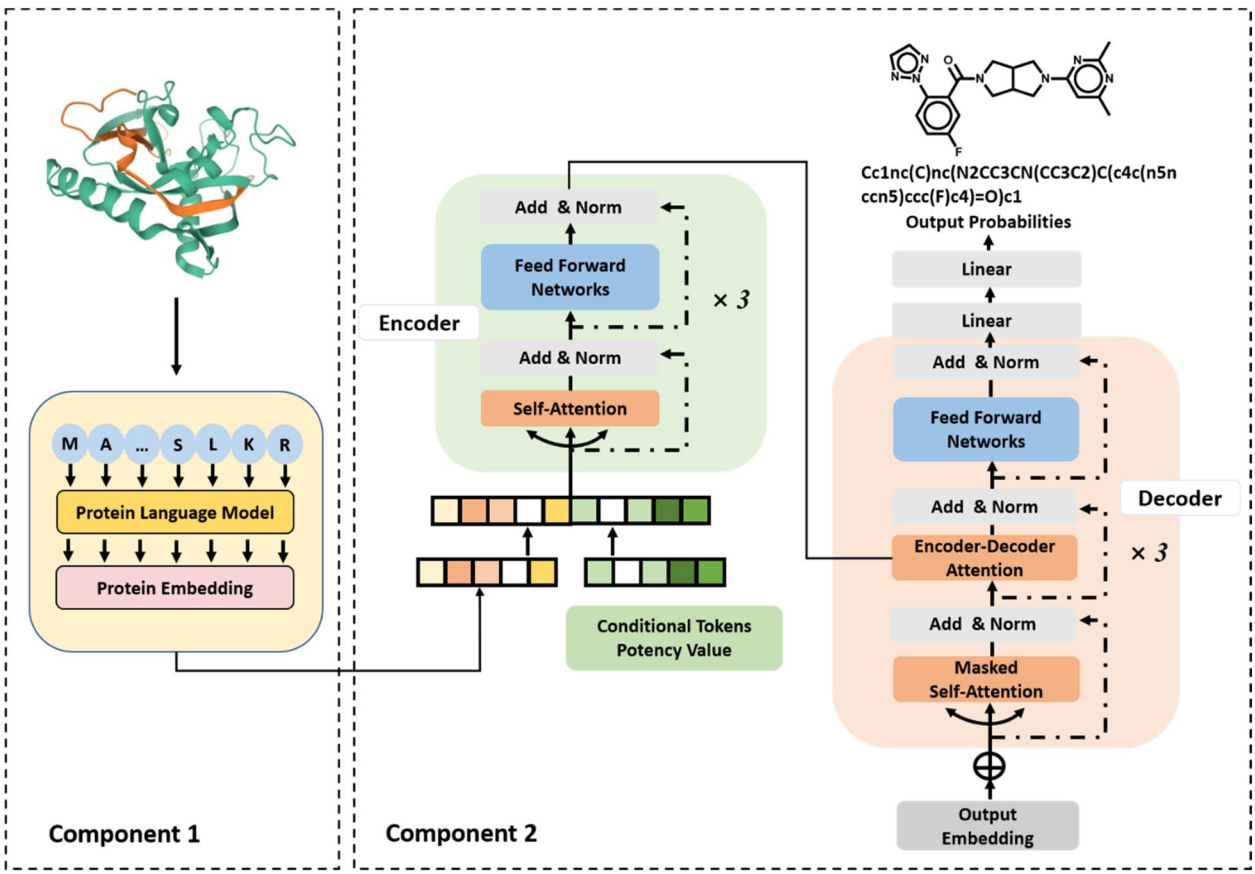


Fig. 1 Architecture of the biochemical language model

stack, were concatenated and pooled along the length dimension. This pooling approach generated a fixed-size embedding, regardless of the input length [29]. ProtTrans embeddings are considered one of the pioneering developments in the field. In our work, ProtT5XLUniref50 protein embeddings of constant dimensionality were generated for each target and concatenated with conditional token embeddings representing compound potency values (see below). The resulting combined embedding vectors provided the input for the encoder of the conditional transformer (model component 2). The ProtTrans PLM was only used for calculating protein sequence embeddings and not involved in model derivation, optimization, or fine-tuning.

Conditional transformer

The architecture of the conditional transformer was adapted from our previous study predicting highly potent compounds from weakly potent templates [15] and modified for generative design of compounds based on sequence data. The transformer was implemented using PyTorch [36]. It consisted of three encoder and three decoder modules with self-attention mechanism. Each encoder module included a multi-head self-attention sub-layer and a fully connected feed-forward neural network sub-layer. The encoder converted the input embedding into a context vector in its final hidden state, serving as input for the decoder. Each decoder contained two multi-head self-attention sub-layers and a feed-forward sub-layer. It transformed the context vector into a sequence of tokens. The masked self-attention sublayer processed the output of the preceding attention sub-layer to prevent translation errors. Compounds were predicted from a given protein sequence embedding conditioned on desired potency via the following triple:

(*Protein sequence embedding, Potency embedding*) → (*Compound*).

For a given protein sequence, representation vectors of the sequence embedding were initially computed using the ProtTrans PLM. Subsequently, the output protein embedding was concatenated with the potency embedding, forming combined representations as input for transformer encoder that were converted into a latent representation. The decoder then iteratively generated an output SMILES sequence until the stop token was obtained. Multinomial sampling was employed to increase output diversity during decoding (hence, in this case, the chemical diversity of candidate compounds). Conditional probabilities for SMILES tokens were derived by the Softmax function of the decoder.

The conditional transformer component was trained on a large number of target-compound triples (see below). The model was then applied to sample candidate (output) compounds for (*Protein sequence embedding, Potency embedding*) input instances.

Tokenization

For model training, protein sequences, compounds, and potency values must be tokenized. Specifically, protein sequences were represented as standard uppercase residue symbols and tokenized using a single space. The vocabulary consisted of 21 tokens including the 20 natural amino acids plus "X" for rare amino acids. Compounds were encoded as canonical SMILES strings. Atoms were represented as single-character tokens (e.g., "C" or "N"), two-character tokens (e.g., "Cl" or "Br"), or tokens enclosed in brackets (e.g., "[nH]" or "[O-]"). Potency values were tokenized based on potency range binning [15, 16, 37]. Therefore, the globally observed potency range of [4.00, 12.52] pK_i units was divided into 852 bins with a constant width of 0.01. This granularity (resolution) captures the limits of experimental potency annotations. Each bin was encoded as a single token, and each potency value was assigned to the corresponding token. Additionally, two special tokens, i.e., "start" and "end," were defined to mark the beginning and end point of a sequence, respectively. This tokenization scheme was introduced previously for the successful generation of potent compounds [15].

Model derivation and evaluation

The conditional transformer variant was trained using the Adam optimizer with a learning rate of 1e-5 and 1024 dimensions for the hidden states, thus precisely matching the settings of the ProtTrans PLM to prevent information loss through the connection. A batch size of 1 was chosen to place the longest protein sequence into GPU memory, and a gradient accumulation scheme was employed to achieve an effective batch size of 64. Training was carried out on a single NVIDIA Tesla A40 (48G) GPU. Throughout the training process, the cross-entropy loss between the ground truth and the output sequence was minimized. The model was trained for at least 50 epochs and at the end of each epoch, a checkpoint was saved. The final model was selected based on minimal cross-entropy loss. The training procedure included pre-training and fine-tuning.

The data set for model pre-training consisted of 212,004 target-compound pairs from 1565 activity classes. For each target-compound pair, triples were generated, as described above:

(*Protein sequence embedding, Potency embedding*) → (*Compound*).

For each pre-training and fine-tuning compound, its experimental potency value was embedded.

As a control, an unconditional transformer with the same architecture but without potency information was also derived from all compounds-target pairs:

(*Protein sequence embedding*) → (*Compound*).

For model fine-tuning and evaluation, each of the 10 activity classes in Table 1 was separately used. Importantly, model fine-tuning and testing were carried out on structurally distinct activity class subsets. Therefore, for each class, a systematic search for analogue series (AS) was conducted using the compound-core relationship (CCR) algorithm [38]. This method employs an extended modified matched molecular pair (MMP) fragmentation procedure [39] based on retrosynthetic rules [40] to systematically identify AS with single or multiple (up to five) substitution sites. The core structure of an AS was required to contain at least twice the number of non-hydrogen atoms of the combined substituents [38]. AS obtained for each activity class were then randomly divided into 50% fine-tuning and 50% test instances, ensuring no overlap in core structures between these sets. Consequently, the fine-tuning and test sets were structurally distinct. Figure 2 shows two exemplary AS.

For each test compound, a (*Protein sequence embedding, Potency embedding*) input instance was generated using its experimental potency value. Then, maximally 100 valid compounds (valid SMILES) were sampled, and these candidates were compared to all test compounds. The model's capacity to exactly reproduce known compounds was determined as the most stringent criterion for model validation. Additionally, for each activity class, 1-nearest neighbor (1-NN) similarity was calculated to compare the generated candidate compound structures with known test compounds. 1-NN similarity was quantified using the Tanimoto coefficient (Tc) [41], calculated based on 2048-bit Morgan fingerprints [42] with a bond radius of 3.

Results and discussion

Study concept

Our study had four primary objectives. (1) Conceptualize target-based compound generation as a machine translation task from a "protein language" to a "chemical language". Therefore, protein representation learning was employed through the incorporation of a PLM. (2) Investigate if compound design across different activity classes could be facilitated on the basis of sequence-based protein representations (embeddings), without reliance on prior knowledge of ligand binding sites (for example, by defining characteristics sequence motifs of binding regions). (3) Evaluate the effects of potency value conditioning on generative compound design. (4) Assess model performance in a most rigorous manner. To address the first two objectives, which were central to our study, we designed a new dual-component conditional biochemical language model to process data of different modality. The model was challenged to learn mappings of protein embeddings conditioned on molecular potency values to active compounds. To address the third objective, we repeated the calculations using a corresponding unconditional model without context-dependent potency conditioning. To address the fourth objective, exact reproduction of known active compounds not encountered during training was set as the most stringent proof-of-concept criterion for the ability of the biochemical language model to correctly predict compounds with desired potency from protein sequence data. To this end, we ensured that fine-tuning and test sets for activity classes were structurally distinct by systematically identifying AS and partitioning them into non-overlapping subsets for fine-tuning and testing, respectively. There also was no compound overlap between activity classes.

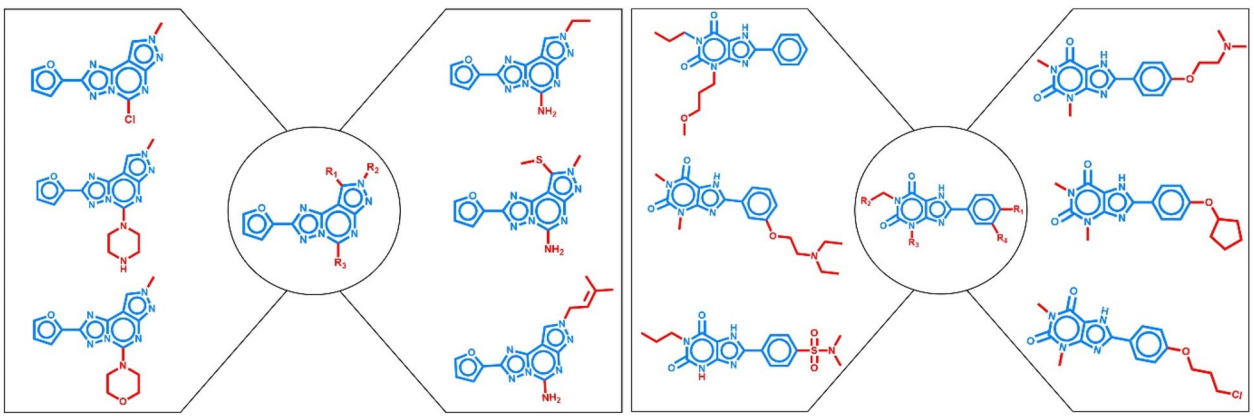


Fig. 2 Exemplary analogue series. On the left and right, two distinct AS are shown consisting of six compounds each. In the center, the common core structure is displayed and all substitution sites are indicated. In the analogues, distinguishing substituents are colored red

Reproducibility of known compounds

The results of the systematic search for AS across 10 activity classes are presented in Table 2. The number of AS per activity class varied from 64 to 312 ("singleton" compounds not participating in any AS were disregarded). AS-based partitioning resulted in 74 to 619 compounds for fine-tuning and 318 to 1206 compounds for model evaluation, depending on the activity classes. In each case, AS were evenly divided (50/50%) and the subset with the smaller and larger total number of compounds was used for fine-tuning and testing, respectively. For each test instance, maximally 100 candidate compounds were sampled, canonicalized, and compared to compounds in the test set to identify exactly reproduced compounds. As reported in Table 2, both the conditional model and the unconditional model produced a substantial number of candidate compounds on the basis of target sequence embeddings. Specifically, depending on the activity class, the conditional model and unconditional model produced from 1789 to 7880 and from 769 to 4206 candidate compounds, respectively. As also reported in Table 2 (last two columns on the right), both the conditional and the unconditional model correctly reproduced multiple test compounds for each activity class; an encouraging finding. For the conditional model, the number of reproduced known compounds ranged from 10 to 115, with on average 43 per class, while the unconditional model generated between 3 and 57 known compounds, with on average 16 per class. Thus, the conditional model consistently reproduced ~ 2- to ~ 4-times more compounds per class than the unconditional model. By design, exact reproduction of test compounds ensured that these compounds had the desired potency value. Hence, these findings revealed a clear effect of compound potency conditioning on multimodal learning. Figure 3 shows exemplary predictions.

In Table 2, for each of 10 activity classes (ChEMBL target ID according to Table 1), the number of AS, number of compounds from AS for fine-tuning and testing, number of compounds produced by the conditional and unconditional model, and number of known test compounds exactly reproduced by the conditional and unconditional model are reported.

As a control, we also used the conditional model without fine-tuning to predict the test sets of three exemplary activity classes (204, 218, and 234). In these cases, the model sampled a total of 3082, 4328, and 8932 valid candidate compounds, respectively. However, no test compounds were reproduced in these calculations, as anticipated, thus confirming an essential role of class-specific fine-tuning.

Potency value conditioning

In Fig. 3, exemplary pairs of reproduced compounds and their most similar fine-tuning compounds are shown for each activity class. In each pair, the reproduced compound is displayed on the right side of the arrow, and its most similar fine-tuning compound is on the left side. In addition, for each pair, the 1-NN similarity is reported, ranging from 0.52 to 0.76 depending on the activity classes. These examples illustrate the recurrent successful reproduction of test compounds from combined target sequence and compound potency embeddings. Moreover, the comparison of most similar fine-tuning and test compounds also indicated that test compounds correctly reproduced by the model had at least comparable, but often higher potency than the corresponding fine-tuning compounds. Notably, higher potency of predicted compared to fine-tuning compounds was not encoded as a conditional constraint. In Fig. 4, boxplots compare the potency value distributions of fine-tuning and test compounds from all activity classes with the potency value

Table 2 Composition of fine-tuning and test sets and reproducibility of known active compounds

ChEMBL ID	Number of AS	Fine-tuning compounds	Test compounds	Sampled compounds		Reproduced compounds	
				Conditional	Unconditional	Conditional	Unconditional
204	130	134	320	2531	1181	16	4
218	250	285	833	2905	1730	75	29
234	213	499	1030	7880	4021	91	21
244	92	188	514	5163	1990	34	11
251	312	619	1206	7077	4206	115	57
1862	64	100	399	1789	894	21	7
4005	125	149	427	3592	2135	30	13
5113	155	288	798	3869	2021	25	10
1,075,104	114	74	323	1940	769	10	3
1,908,389	78	86	318	2324	1092	13	3

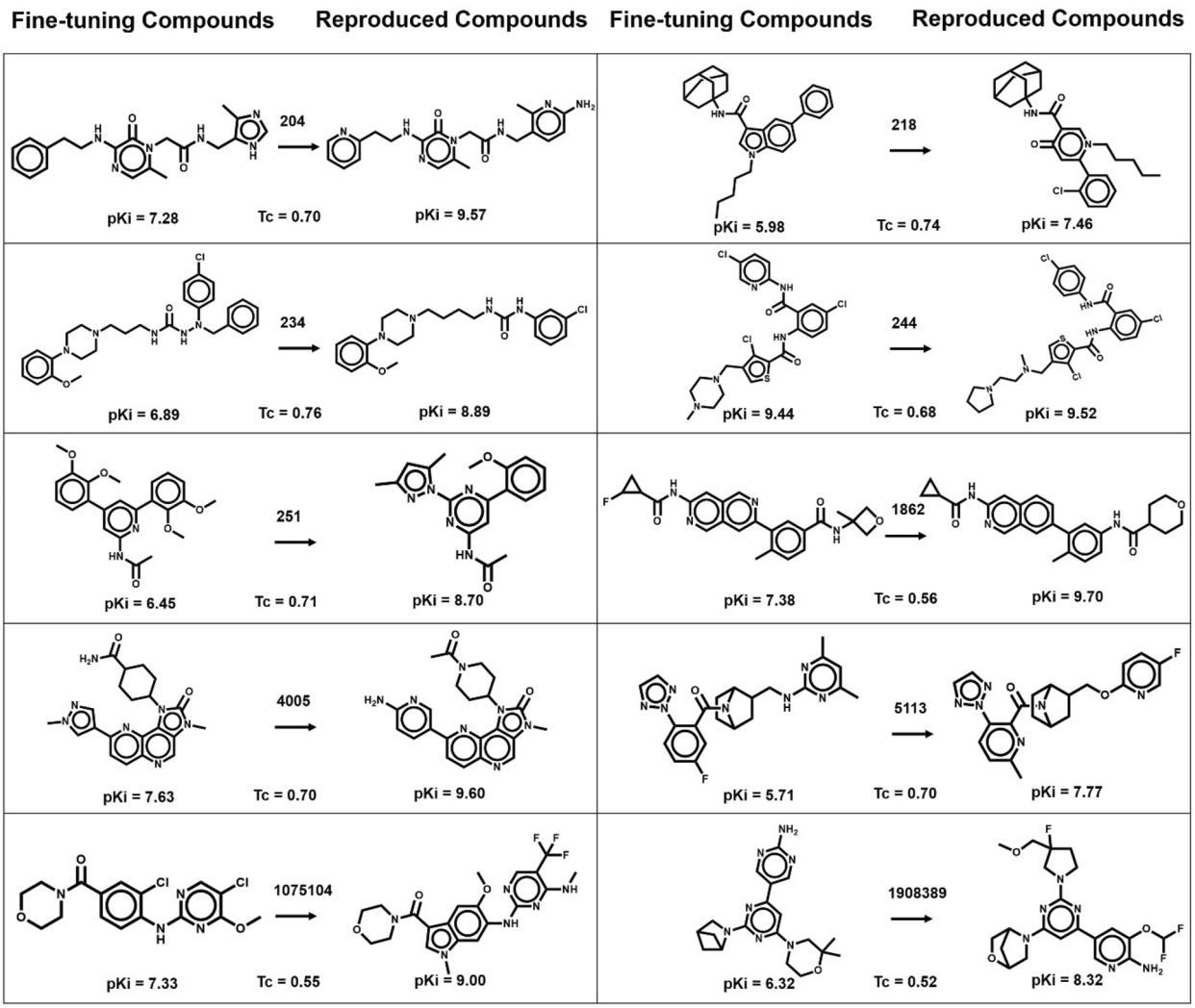


Fig. 3 Exemplary predictions. For each activity class, exemplary test compounds are shown (right of the arrow) that were exactly reproduced using the conditional model together with the most similar fine-tuning compounds (left). For each test/fine-tuning compound pair, the Tanimoto similarity value is reported. ChEMBL IDs on arrows identify activity classes according to Table 1

distributions of test compounds correctly predicted by the conditional transformer and the unconditional model.

The comparison showed that potency value distributions and the resulting median values of fine-tuning and test compounds differed depending on the activity class, as one would expect. In some instances, the median potency of test compounds was higher than of fine-tuning compounds and vice versa. However, for most activity classes, the potency distributions of test compounds correctly predicted by the conditional model closely matched the potency distributions of all test compounds, consistent with the desired effects of potency conditioning. By contrast, the unconditional model mostly reproduced smaller numbers of compounds with lower median potency than those correctly predicted by the conditional model, thus revealing a tendency to under-predict compound potency values in the absence of potency conditioning. Notably, the absence of statistical significance of potency differences between compounds reproduced with the conditional and unconditional model was mostly a consequence of the imbalanced sample sizes, including very small samples for the unconditional model (Table 2).

Similarity analysis

In addition to identifying and characterizing correctly reproduced test compounds, the 1-NN similarity of all sampled candidate compounds to test compounds was determined. Importantly, for rigorously establishing proof-of-concept of the approach, it was essential to

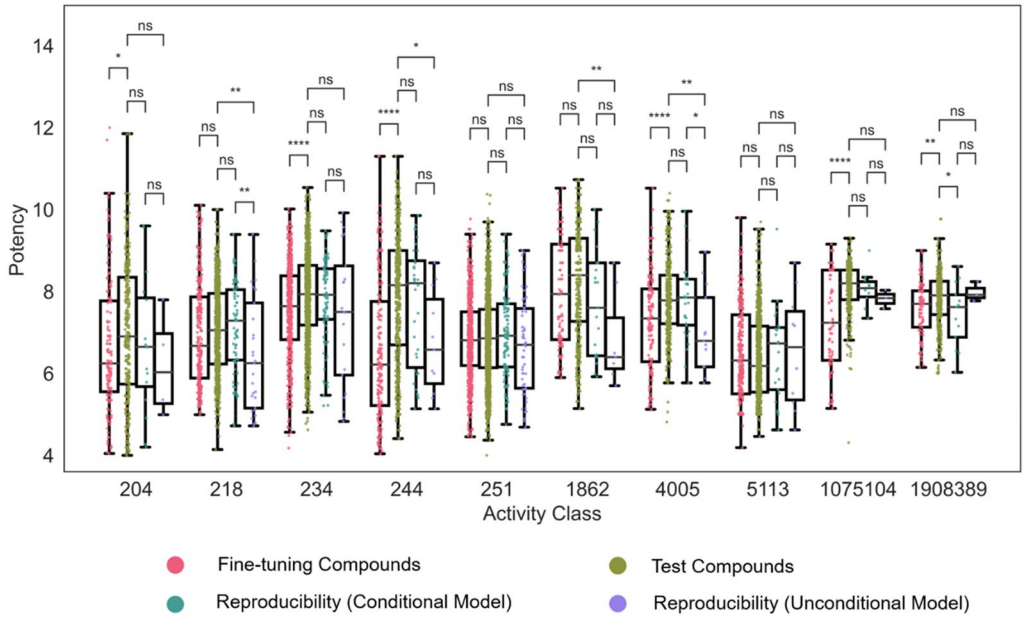


Fig. 4 Potency value distributions of different compound subsets. For each activity class, boxplots compare logarithmic potency value distributions for all fine-tuning and test compounds and for test compounds correctly predicted by the conditional transformer and the unconditional model. To assess the statistical significance of differences between potency value distributions, independent-samples t-tests were conducted: $0.05 < p \leq 1.00$ (ns), $0.01 < p \leq 0.05$ (*), $0.001 < p \leq 0.01$ (**), $0.0001 < p \leq 0.001$ (***), $p \leq 0.0001$ (****). Stars denote increasing levels of statistical significance and "ns" stands for "not significant"

confirm the ability of the biochemical language model to exactly reproduce known active compounds. However, for the practical relevance of the model and its design capacity, generalization potential should also be assessed. Ideally, a model with generalization ability should diversify candidate compounds (i.e., structurally abstract from fine-tuning and test compounds). Hence, the generation of candidate compounds with increasing structural diversity compared to known compounds also represented an important evaluation criterion. Therefore, we first systematically compared newly generated candidate compounds to test compounds. Figure 5 shows the distribution of 1-NN similarities of predicted candidate compounds compared to test compounds across the 10 activity classes. The predicted compounds consistently exhibited a variety of 1-NN similarities to test compounds, ranging from identical (or nearly identical) structures (100% 1-NN similarity) to distinct structures (~10% similarity). The most frequently observed 1-NN similarities ranged from ~30% to ~60%, depending on the activity class. These findings underscored the capability of the biochemical language model to not only reproduce known compounds but also generate structurally diverse candidate compounds.

Secondly, we also examined the distribution of 1-NN similarities for reproduced test compounds compared to fine-tuning compounds across the 10 activity classes. The reproduced compounds also exhibited a wide range of 1-NN similarities compared to fine-tuning compounds, from (~18%, ~56%) to (~40%, ~70%) across all activity classes. Here, the most frequently observed 1-NN similarities varied from ~25% to ~65%, depending on the activity class. Hence, these findings also confirmed the ability of the approach to abstract from fine-tuning compounds.

Synthetic accessibility

While exact reproduction of known test compounds represents the ultimate criterion for establishing proof-of-concept for the design approach, newly generated candidate compounds also provide a resource for synthesis. Therefore, we have compared the synthetic accessibility (SA) of all sampled candidate compounds to the existing fine-tuning compounds using a well-established scoring scheme [43]. The results in Fig. 6 show that the SA score distributions for fine-tuning and candidate compounds sampled with both the conditional and unconditional model were nearly indistinguishable, thus indicating high SA for the newly generated candidate compounds.

Conclusion

In this work, we have explored a new concept for predicting compounds with activity against given targets and desired potency from sequence embeddings with

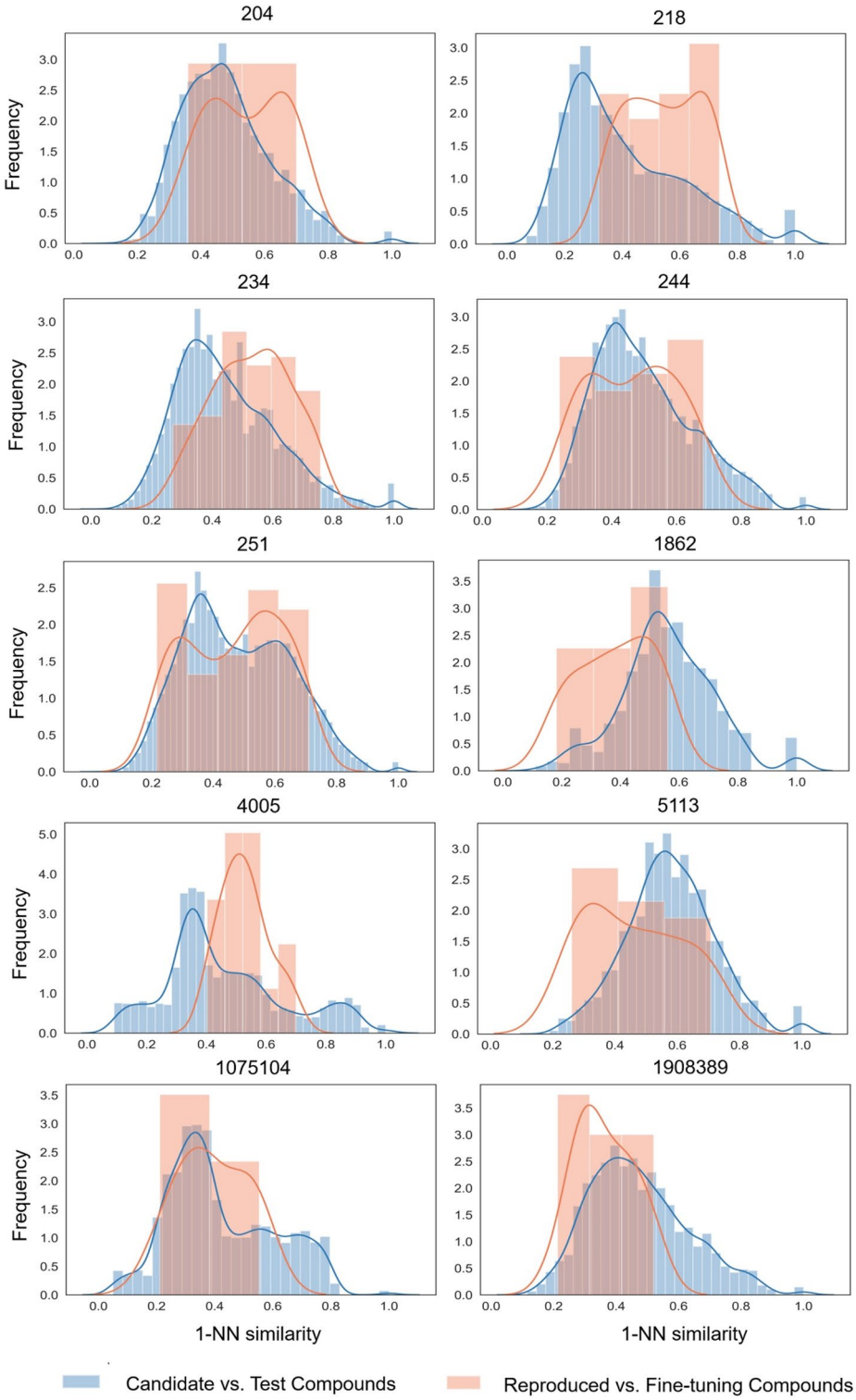


Fig. 5 Distribution of 1-nearest neighbor similarities. For each activity class, blue and orange value distributions show 1-NN similarities of sampled candidate compounds vs. test compounds and correctly reproduced compounds vs. fine-tuning compounds, respectively

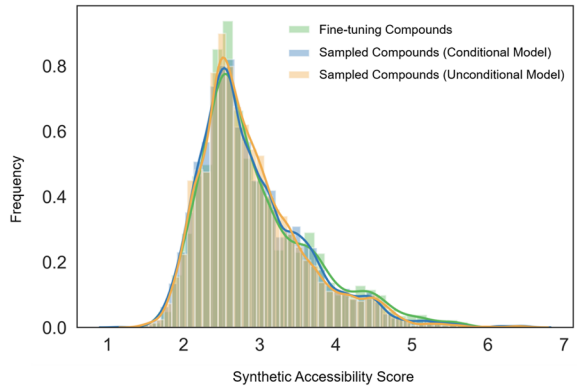


Fig. 6 Synthetic accessibility. Compared are distributions of SA scores calculated for fine-tuning compounds and candidate compounds sampled with the conditional and unconditional model, respectively

potency conditioning. For this purpose, a dual-component biochemical language model was designed for multimodal learning. The model included a pre-trained PLM (component 1) for protein representation learning and a conditional transformer (component 2) operating on the output of the PLM. The transformer was trained to learn mappings of target sequence embeddings conditioned on potency values to active compounds. Accordingly, the model input for generative design was heterogeneous, combining a sequence embedding with a molecular property constraint. The model was individually fine-tuned on 10 different target-based activity classes not included in model derivation. Model fine-tuning and evaluation were carried out on structurally distinct compound subsets generated by comprehensive AS identification and AS-based compound splitting. As the most rigorous proof-of-concept criterion for the approach, the ability of the biochemical language model to exactly reproduce known active compounds not encountered during training was determined. By design, exactly reproduced compounds had desired potency. The biochemical language model consistently reproduced varying numbers of known active compounds for all activity classes; an encouraging finding. Moreover, compared to an unconditional model used as a control, the conditional transformer consistently reproduced larger numbers of known compounds, thus revealing a clear positive effect of potency value conditioning on successful predictions. In addition, for most activity classes, the potency distribution of correctly reproduced compounds closely matched the potency distribution of all test compounds, consistent with reproducing compounds at different potency levels. Subsequent molecular similarity analysis showed that the biochemical language model was also capable of generating structurally diverse candidate compounds departing from both fine-tuning and test compounds; an indicator of model generalization potential.

Generative modeling compounds with desired potency from compound potency-conditioned target sequence embeddings was an unusual design task that might be expected to fail, for the scientific reasons discussed, and that could not possibly be addressed using standard ML approaches. Rather, for this challenging task, a language model was required to learn mappings of conditioned sequence data to active compounds, providing an example for a new potential opportunity provided by language models in compound design. Assessing whether or not such models might be predictive required a well-defined system set-up and rigorous evaluation criteria. The detected ability of the two-component biochemical language model to exactly reproduce compounds with pre-defined potency was not expected initially. Encouragingly, however, exact reproduction of test compounds was consistently observed across different activity classes, establishing proof-of-concept for such predictions.

Taken together, the results of our study suggest that compound design based on conditioned target sequence embeddings using language models merits further consideration. Currently, origins of correct compound reproduction remain model-internal and are non-transparent. Therefore, subsequent studies will be devised to explore the learning characteristics of the biochemical language model, rationalize correct predictions, and identify their input determinants. Furthermore, having established proof-of-principle at the methodological level, the approach will need to be prospectively assessed. For practical applications, it is straightforward, for example, to direct generative design towards highly potent compounds by setting corresponding potency thresholds. Furthermore, other context-dependent rules (such as different molecular property constraints) can be investigated in conjunction with target sequence embeddings. Moreover, the demonstrated ability of the biochemical language model to generate structurally diverse candidate compounds can also be explored in prospective applications by tesing new candidates. Therefore, given that the methodology is made freely available as a part of this study, there are ample opportunities for further research and applications.

Abbreviations

QSAR	Quantitative structure–activity relationship
ML	Machine learning
DL	Deep learning
CNN	Convolutional neural network
RNN	Recurrent neural network
GCN	Graph convolutional network
MPNN	Message passing neural network
SMILES	Simplified molecular input line entry system
PLM	Protein language model
AS	Analogue series

CCR	Compound-core relationship
1-NN	1-Nearest neighbor
MMP	Matched molecular pair
Tc	Tanimoto coefficient

Acknowledgements
The authors thank Martin Vogt for many helpful suggestions.

Author contributions
Both authors designed and conducted the study, analyzed the results, and prepared the manuscript.

Funding
Open Access funding enabled and organized by Projekt DEAL. H.C. is supported by the China Scholarship Council (CSC).

Availability of data and materials
Calculations were carried out using publicly available programs and compound data. Python scripts generated for the study, the models, all pre-training and fine-tuning data, and newly generated compounds are available via the following link: https://uni-bonn.sciebo.de/s/Z9O2ZqKoA2cS7B1.

Declarations

Competing interests
The authors declare no competing financial interest.

Author details
[1]Department of Life Science Informatics and Data Science, B-IT, Lamarr Institute for Machine Learning and Artificial Intelligence, LIMES Program Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Friedrich-Hirzebruch-Allee 5/6, 53115 Bonn, Germany.

Received: 6 December 2023 Accepted: 9 May 2024
Published online: 22 May 2024

References

1. Keserü GM, Makara GM (2009) The influence of lead discovery strategies on the properties of drug candidates. Nat Rev Drug Discov 8:203–212. https://doi.org/10.1038/nrd2796
2. Ferreira LLG, Andricopulo AD (2019) ADMET modeling approaches in Drug Discovery. Drug Discov Today 24:1157–1165. https://doi.org/10.1016/j.drudis.2019.03.015
3. Lewis RA, Wood D (2014) Modern 2D QSAR for drug discovery. WIREs Comput Mol Sci 4:505–522. https://doi.org/10.1002/wcms.1187
4. Muratov EN, Bajorath J, Sheridan RP et al (2020) QSAR without borders. Chem Soc Rev 49:3525–3564. https://doi.org/10.1039/d0cs00098a
5. Vamathevan J, Clark D, Czodrowski P et al (2019) Applications of machine learning in drug discovery and development. Nat Rev Drug Discov 18:463–477. https://doi.org/10.1038/s41573-019-0024-5
6. Huo X, Xu J, Xu M, Chen H (2023) An improved 3D quantitative structure-activity relationships (QSAR) of molecules with CNN-based partial least squares model. Artif Intell Life Sci 3:100065. https://doi.org/10.1016/j.ailsci.2023.100065
7. Li Y, Xu Y, Yu Y (2021) CRNNTL: Convolutional recurrent neural network and transfer learning for QSAR modeling in organic drug and material discovery. Molecules 26:7257. https://doi.org/10.3390/molecules26237257
8. Wang F, Lei X, Liao B, Wu F-X (2022) Predicting drug–drug interactions by graph convolutional network with multi-kernel. Brief Bioinform. https://doi.org/10.1093/bib/bbab511
9. Tang M, Li B, Chen H (2023) Application of message passing neural networks for molecular property prediction. Curr Opin Struct Biol 81:102616. https://doi.org/10.1016/j.sbi.2023.102616
10. Pasupa K, Sunhem W. A comparison between shallow and deep architecture classifiers on small dataset. 8th International Conference on Information Technology and Electrical Engineering, 2016; pp 1–6. https://doi.org/10.1109/iciteed.2016.7863293
11. Janela T, Bajorath J (2022) Simple nearest-neighbour analysis meets the accuracy of compound potency predictions using complex machine learning models. Nat Mach Intell 4:1246–1255. https://doi.org/10.1038/s42256-022-00581-6
12. Walters WP, Barzilay R (2020) Applications of deep learning in molecule generation and molecular property prediction. Acc Chem Res 54:263–270. https://doi.org/10.1021/acs.accounts.0c00699
13. Hirschberg J, Manning CD (2015) Advances in natural language processing. Science 349:261–266. https://doi.org/10.1126/science.aaa8685
14. Chen H, Vogt M, Bajorath J (2022) DeepAC – conditional transformer-based chemical language model for the prediction of activity cliffs formed by bioactive compounds. Digital Discov 1:898–909. https://doi.org/10.1039/d2dd00077f
15. Chen H, Bajorath J (2023) Designing highly potent compounds using a chemical language model. Sci Rep 13:7412. https://doi.org/10.1038/s41598-023-34683-x
16. Chen H, Bajorath J (2023) Meta-learning for transformer-based prediction of potent compounds. Sci Rep 13:16145. https://doi.org/10.1038/s41598-023-43046-5
17. Chen L, Tan X, Wang D et al (2020) TransformerCPI: improving compound–protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments. Bioinformatics 36:4406–4414. https://doi.org/10.1093/bioinformatics/btaa524
18. Nguyen T, Le H, Quinn TP et al (2020) GraphDTA: predicting drug–target binding affinity with graph neural networks. Bioinformatics 37:1140–1147. https://doi.org/10.1093/bioinformatics/btaa921
19. Öztürk H, Özgür A, Ozkirimli E (2018) DeepDTA: deep drug–target binding affinity prediction. Bioinformatics 34:i821–i829. https://doi.org/10.1093/bioinformatics/bty593
20. Karimi M, Wu D, Wang Z, Shen Y (2019) DeepAffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. Bioinformatics 35:3329–3338. https://doi.org/10.1093/bioinformatics/btz111
21. Zhao Q, Zhao H, Zheng K, Wang J (2022) HyperAttentionDTI: improving drug–protein interaction prediction by sequence-based deep learning with attention mechanism. Bioinformatics 38:655–662. https://doi.org/10.1093/bioinformatics/btab715
22. Chen L, Fan Z, Chang J et al (2023) Sequence-based drug design as a concept in computational drug design. Nat Commun 14:4217. https://doi.org/10.1038/s41467-023-39856-w
23. Grechishnikova D (2021) Transformer neural network for protein-specific de novo drug generation as a machine translation problem. Sci Rep 11:321. https://doi.org/10.1038/s41598-020-79682-4
24. Qian H, Lin C, Zhao D et al (2022) AlphaDrug: protein target specific de novo molecular generation. PNAS Nexus. https://doi.org/10.1093/pnasnexus/pgac227
25. Yoshimori A, Bajorath J (2023) Motif2Mol: prediction of new active compounds based on sequence motifs of ligand binding sites in proteins using a biochemical language model. Biomolecules 13:833. https://doi.org/10.3390/biom13050833
26. Weininger D (1988) SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Comput Sci 28:31–36. https://doi.org/10.1021/ci00057a005
27. Rives A, Meier J, Sercu T et al (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proc Natl Acad Sci USA 118:e2016239118. https://doi.org/10.1073/pnas.2016239118
28. Bepler T, Berger B (2021) Learning the protein language: evolution, structure, and function. Cell Syst 12:654-669.e3. https://doi.org/10.1016/j.cels.2021.05.017
29. Elnaggar A, Heinzinger M, Dallago C et al (2022) ProtTrans: toward understanding the language of life through self-supervised learning. IEEE Trans Pattern Anal Mach Intell 44:7112–7127. https://doi.org/10.1109/tpami.2021.3095381
30. Singh R, Sledzieski S, Bryson B et al (2023) Contrastive learning in protein language space predicts interactions between drugs and protein targets. Proc Natl Acad Sci USA 120:e2220778120. https://doi.org/10.1073/pnas.2220778120
31. Bento AP, Gaulton A, Hersey A et al (2014) The CHEMBL bioactivity database: an update. Nucleic Acids Res 42:D1083–D1090. https://doi.org/10.1093/nar/gkt1031

32. UniProt Consortium (2019) UniProt: a worldwide hub of protein knowledge. Nucleic Acids Res 47:D506–D515. https://doi.org/10.1093/nar/gky1049
33. RDKit: cheminformatics and machine learning software. 2021. http://www.rdkit.org/.
34. Suzek BE, Wang Y, Huang H et al (2015) UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. Bioinformatics 31:926–932. https://doi.org/10.1093/bioinformatics/btu739
35. Steinegger M, Söding J (2018) Clustering huge protein sequence sets in linear time. Nat Commun 9:2542. https://doi.org/10.1038/s41467-018-04964-5
36. Paszke A, Gross S, Massa F et al (2019) PyTorch: An imperative style, high-performance deep learning library. Adv Neural Inf Process Syst 32:8026–8037
37. He J, You H, Sandström E et al (2021) Molecular optimization by capturing chemist's intuition using deep neural networks. J Cheminform 13:26. https://doi.org/10.1186/s13321-021-00497-0
38. Naveja JJ, Vogt M, Stumpfe D et al (2019) Systematic extraction of analogue series from large compound collections using a new computational compound–core relationship method. ACS Omega 4:1027–1032. https://doi.org/10.1021/acsomega.8b03390
39. Stumpfe D, Dimova D, Bajorath J (2016) Computational method for the systematic identification of analog series and key compounds representing series and their biological activity profiles. J Med Chem 59:7667–7676. https://doi.org/10.1021/acs.jmedchem.6b00906
40. Lewell XQ, Judd DB, Watson SP, Hann MM (1998) RECAP—retrosynthetic combinatorial analysis procedure: a powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. J Chem Inf Comput Sci 38:511–522. https://doi.org/10.1021/ci970429i
41. Bajusz D, Rácz A, Héberger K (2015) Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? J Cheminform 7:20. https://doi.org/10.1186/s13321-015-0069-3
42. Cereto-Massagué A, Ojeda MJ, Valls C et al (2015) Molecular fingerprint similarity search in virtual screening. Methods 71:58–63. https://doi.org/10.1016/j.ymeth.2014.08.005
43. Ertl P, Schuffenhauer A (2009) Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. J Cheminform 1:8. https://doi.org/10.1186/1758-2946-1-8

Publisher's Note

Additional Publications

Umedera, K.; Yoshimori, A.; Chen, H.; Kouji, H.; Nakamura, H.; Bajorath, J. "DeepCubist: Molecular Generator for Designing Peptidomimetics Based on Complex Three-Dimensional Scaffolds. *J Comput-Aided Mol Des* **2023**, *37*, 107-115".
DOI: 10.1007/s10822-022-00493-y

Yoshimori, A.; Chen, H.; Bajorath, J. "Chemical Language Models for Applications in Medicinal Chemistry. *Future Med Chem* **2023**, *15*, 119-121".
DOI: 10.4155/fmc-2022-0315