FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Machine-learning prediction of terpene biosynthesis
Student:	Roman Bushuiev
Supervisor:	Tomáš Pluskal, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2021/2022

Instructions

Plant specialized metabolites are an essential source of chemical scaffolds for the development of new medicines. With tens of thousands of unique structures discovered to date, the largest and the most diverse class of plant specialized metabolites are terpenoids. Terpenoids are produced by terpene synthases, which cyclize isoprenoid diphosphate substrates into specific scaffolds. However, only a tiny fraction of known terpene synthase enzymes have been characterized in detail. The objective of the thesis is to develop machine learning models that can predict the chemical structure of a terpene scaffold produced by a terpene synthase enzyme from the amino acid sequence of the enzyme. These models will utilize previously characterized terpene synthase reactions for training.

Electronically approved by Ing. Karel Klouda, Ph.D. on 13 October 2020 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Bachelor's thesis

Machine-learning prediction of terpene biosynthesis

Roman Bushuiev

Department of Applied Mathematics

Supervisor: Mgr. Tomáš Pluskal, Ph.D.

May 12, 2021

Acknowledgements

I would like to express my deepest gratitude to my supervisor Dr. Tomáš Pluskal for providing an opportunity to work on such an exciting project. I cannot imagine better supervision and I am happy to be growing under his guidance. I also want to heartily thank Dr. Josef Šivic for the invaluable pieces of advice and feedback regarding my thesis. I would like to express my gratitude to Raman Samusevich for the regular, unforgettable conversations about machine learning and the continuous feedback regarding my project. Lastly, I want to thank Adéla Tajovská for the creation of the terpene synthase database, which is fundamental for my thesis, and Joshua Smith for the great help with English and writing.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on May 12, 2021 . . . . . .

Czech Technical University in Prague

Faculty of Information Technology

Citation of this thesis

Bushuiev, Roman. *Machine-learning prediction of terpene biosynthesis.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2021.

Abstrakt

Biosyntéza v živých organismech se skládá z komplexních transformací molekul katalyzovaných enzymy. Ačkoli porozumění těmto biochemickým reakcím je zásadní pro moderní medicínu a strojové učení již prokázalo svou účinnost pro rozluštění velmi složitých problémů, predikce biosyntéz dosud nebyla studována. Dokonce i pro dobře definované reakce, jako je biosyntéza terpenů, velmi malé množství dosud charakterizovaných reakcí a komplikovanost jejich složek dělají problém zdánlivě neřešitelným. V této práci se zaměřuji na predikci biosyntézy seskviterpenů a navrhuji řešení nejprve snížením složitosti pomoci modelů strojového učení předtrénovaných na rozsáhlých databázích, a následovně využitím naučených vlastností na řešení primárního úkolu. Výsledky ukazují, že tento přístup umožňuje poměrně dobrou predikci reakcí biosyntézy seskviterpenů s použitím jen 315 trénovacích vzorků, a představuje tedy slibný směr pro další výzkum.

Klíčová slova biochemie, terpen, biosyntéza, strojové učení, Transformer, Variational Autoencoder

Abstract

Biosynthesis in living organisms consists of complex molecular transformations catalyzed by enzymes. Even though deep understanding of such biochemical reactions is essential for modern medicine and machine learning has already proven its efficiency in unraveling complex tasks, the prediction of biosynthesis has not been studied yet. Even for highly conserved reactions, such as terpene biosynthesis, the relatively small amount of reactions characterized to date and the complexity of their components make the problem seem infeasible. In the present work, I focus on the prediction of sesquiterpene biosynthesis and propose a solution by first reducing the problem complexity with machine learning models pre-trained on large databases and then transferring the learned features to the primary task. Results show that the introduced approach allows for reasonable prediction of sesquiterpene biosynthetic reactions using only 315 training samples, which makes it remarkably interesting for further study.

Keywords biochemistry, terpene, biosynthesis, machine learning, Transformer, Variational Autoencoder

Contents

List of Figures

List of Tables

Introduction

Plant specialized metabolites are molecules, which are produced to increase their survivability and fecundity. These compounds are an essential source of chemical scaffolds for the development of new medicines. About 25% of currently produced drugs are directly derived from plants [1]. In the near future, the most sustainable way to produce such molecules will be through biosynthesis in engineered microorganisms. However, this approach demands a comprehensive understanding of the biosynthetic process of the target molecules, which is a challenging task. Terpenes are the largest class of specialized metabolites [2], therefore the ability to predict terpene biosynthesis would be a step towards a new era of drug design. During terpene biosynthesis, terpene synthase enzymes transform substrate molecules into more complex ones – terpenes. The aim of my thesis is to predict a terpene molecule knowing only the substrate and the enzyme. Terpene biosynthesis reactions are relatively simple and conserved compared to other known biosynthetic pathways, which allows studying them with computer science tools. However, the biological complexity and diversity of their components with the low amount of up-to-date characterized reactions make solving the problem extremely hard. Intricacy of enzymes and terpenes is overwhelming even for the traditional computer science approaches.

Machine learning has been rapidly developing for the last decades and has proven its efficiency in various challenging tasks. Nevertheless, even for the machine-learning models, the problems may seem intractable as they require large amounts of training data. Unfortunately, the number of characterized terpene syntheses is less than one thousand leading to an extremely limited set of training data. In this thesis I focus on the biosynthesis of sesquiterpenes (terpenes containing 15 carbons), which constitute the majority of characterized reactions, and keep in mind further generalization to the whole class of terpenes. To overcome the extreme lack of training data I simultaneously uti-

lize two different unsupervised pre-trained machine learning models: one for enzymes and one for sesquiterpenes. These models learn patterns and general structures of complex objects by solving synthetic tasks such as guessing masked parts of the input or learning optimal compression and decompression functions utilizing millions of data samples during the training. It allows breaking down the overwhelming complexity and to operate on learned continuous vectors instead of intricate objects. In the present work, I show that this approach leads to promising results and despite the low amount of training data it is able to capture biosynthesis reactions, which makes it very interesting for further study.

Thesis structure

The thesis starts with Chapter 1 introducing some basic biochemistry terms and selected machine-learning models that are essential for my thesis. In Chapter 2, I outline the current state of the intersection between biochemistry and machine learning and briefly describe several state-of-the-art approaches. Next, I define the problem of terpene biosynthesis prediction by discussing terpene biosynthesis, available data, and the objective of the work (Chapter 3). In the following, Chapter 4, I present the methods I use for solving the problem and in the subsequent Chapter 5 I report the obtained results. Finally, in Chapter 6, I discuss plans for future work.

Mathematical notation

For convenience, I treat bit vectors in Section 4.2 as sets, where each element of the set represents bit index and it's value. For example, the vector 101 can be represented as the set of ordered pairs $\{(0,1),(1,0),(2,1)\}$. Then for such set A, A_+ is a subset containing only positive bits ($\{(0,1),(1,0),(2,1)\}_+ = \{(0,1),(2,1)\}$). I additionally define $|A|_+ := |A_+|$ and $|A|_{/+} := \frac{|A|_+}{|A|}$ for the simplicity. Also, I use operator $\triangle$ for the symmetric difference of two sets satisfying $A \triangle B = (A \setminus B) \cup (B \setminus A)$ and the operator $\times$ for the Cartesian product, defined as $A \times B = \{(a,b) \mid a \in A \ \wedge \ b \in B\}$.

Workflow

I have chosen Python 3 as a programming language, as it has a rich ecosystem for the machine-learning and data processing. The main libraries I use are: pandas [3] (data processing), NumPy (arrays numerical computations), scikit-learn [4] (data analysis tools and machine-learning models), Matplotlib [5] (visualizations), Keras [6] (neural networks), RDKit [7] (chemoinformatics), Beautiful Soup [8] (web-scraping).

Chapter 1

Background

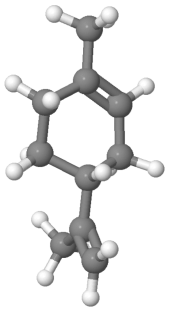
1.1 Biochemistry essentials

In this section, I would like to introduce some key biochemistry terms that I will refer to in the following chapters, which are essential for the discussed problematics. Biochemistry is a complicated science, so I will not be explaining concepts in extraneous detail, but rather cover their understanding from the perspective of the computer science domain.

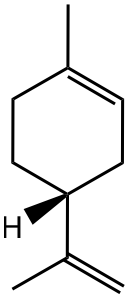
1.1.1 Small molecules

A **molecule** is a compound made up of two or more atoms that are chemically bonded together, while an **atom** is the smallest particle forming a chemical element. All chemical elements have different properties that are often summarized in the periodic table. The structure of a particular molecule can be characterized by its atoms and bonds between them. It leads to the conclusion that any molecule can be represented as a labeled graph, where atoms and bonds are represented as nodes and edges, respectively, and additional labeling mappings are defined for nodes and vertices to take atoms and bonds nature into consideration. Such a representation is usually called **structural formula** (Figure 1.1a). The figure shows a spatial graph representing *(4R)-limonene* - a molecule belonging to the class of terpenes (Section 3.1), it commonly occurs in lemons and has a myriad of pharmacological activities, including apoptosis of breast cancer cells [9]. Grey and white vertices correspond to carbon and hydrogen atoms, and double edges correspond to double bonds between atoms.

Molecules containing carbon-hydrogen bonds are named **organic compounds**. Since they constitute the majority of known chemicals, it is convenient to use a **skeletal formula** – a simplified planar graph representation (Figure 1.1b), where nodes implicitly constitute carbon atoms, but hydrogen



(a) Ball-and-stick representation



(b) Skeletal formula

```
[H][C@]1(CCC(C)=CC1)C(C)=C
```

(c) SMILES string representation

Figure 1.1: Different representations of the same molecule *(4R)-limonene*.

atoms are omitted. This adaptation does not affect the representation's expressivity because hydrogens can be unambiguously filled based on common chemical rules. A spatial arrangement of atoms, **stereochemistry**, is encoded in special types of bonds visualized as either dashed or solid triangles, as is shown in Figure 1.1b. These two types of bonds express opposite spatial directions. In general, different types of bonds are of great importance regarding the properties of a molecule.

A **chemical formula** of *(4R)-limonene* is $C_{10}H_{16}$, which means that it consists of 10 atoms of carbon and 16 atoms of hydrogen. This description of a molecule is compact but far from complete, because plenty of molecules have the same chemical formulas but different structures. Such molecules are referred to as **isomers**. Moreover, isomers do not necessarily share similar chemical or physical properties. Otherwise, this representation is useful for providing the elemental composition of the molecule in a compact way.

Another widely used way to represent a molecule is a **SMILES** (Simplified molecular-input line-entry system) string Figure 1.1c. In terms of a formal language theory, SMILES strings form a context-free language with a generating grammar designed to uniquely encode molecules' structures. It has an easy syntax for both human and machine and is compact, which makes this system suitable for storing large amounts of molecules and using them in various applications.

1.1.2 Proteins

Proteins are large size molecules (macromolecules), formed of repeating structural units called **amino acids** Figure 1.2. There are 20 different amino acids appearing in proteins, arranged in long sequences typically hundreds of units long. These sequences fold into distinct 3D forms, determining their activity. Proteins are present in all living organisms and are required for the structure, function, and regulation of the cells, tissues, and organs, for which they are usually called the building blocks of life.

For the majority of computer science applications, proteins are represented as strings of characters, where each character corresponds to a particular amino acid (for example, S – *Serine*, H – *Histidine*, M – *Methionine*, and so on; Figure 1.3d). Usually, proteins are visualized as 3D constructions according to the geometry of their substructures as shown in Figure 1.3a.

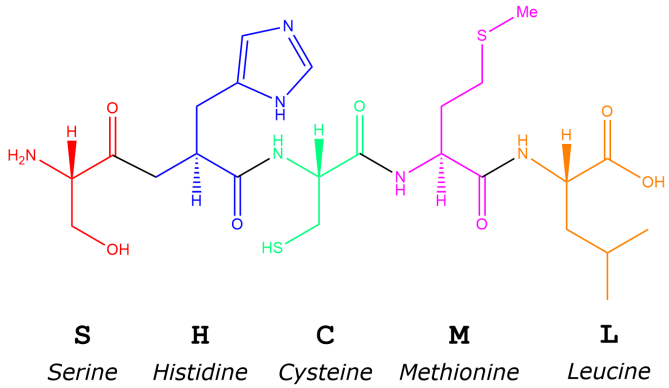


Figure 1.2: Amino acid sequence (single amino acids are highlighted in different colors).

1.1.3 Enzymes

Enzymes are complex proteins catalyzing chemical reactions, during which molecules (**substrates**) are being transformed to different molecules (**products**). Such reactions are referred to as **biosyntheses**. It is important to mention that some enzymes could act on different substrates or mediate in the creation of different products.

As already mentioned, the topology of a protein determines its function. In the case of enzymes, particular folding of an amino acid sequence enables the enzyme to dock a substrate. Thus, it determines which substrates can be accepted by an enzyme and how a catalyzing reaction will occur. Figure 1.4 shows the formed connection between the particular amino acids of the enzyme (*Glucosidase*) and the substrate (sugar maltose) at the so-called active (amino acid) residues. During the reaction catalyzed by *Glucosidase* enzyme,

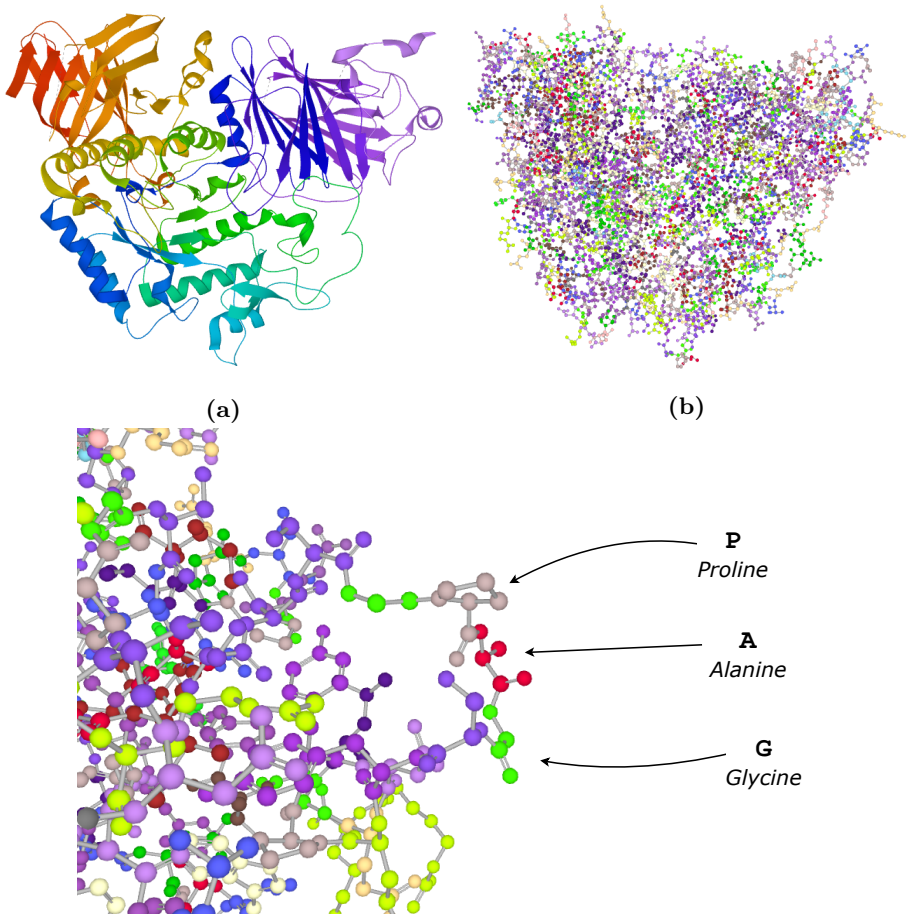


Figure 1.3: Different representations of the same protein *Lysosomal alpha-glucosidase*. **(a)** Visualization based on the geometry of local substructures (secondary structures), color fades along with the sequence of amino acids; **(b)** ball-and-stick visualization showing single atoms, colors display different amino acids; **(c)** zoomed visualization (b) revealing three particular amino acids; **(d)** amino acid sequence string representation, where each character corresponds to a particular amino acid. Source - UniProt [10]: `https://www.uniprot.org/uniprot/P10253`.

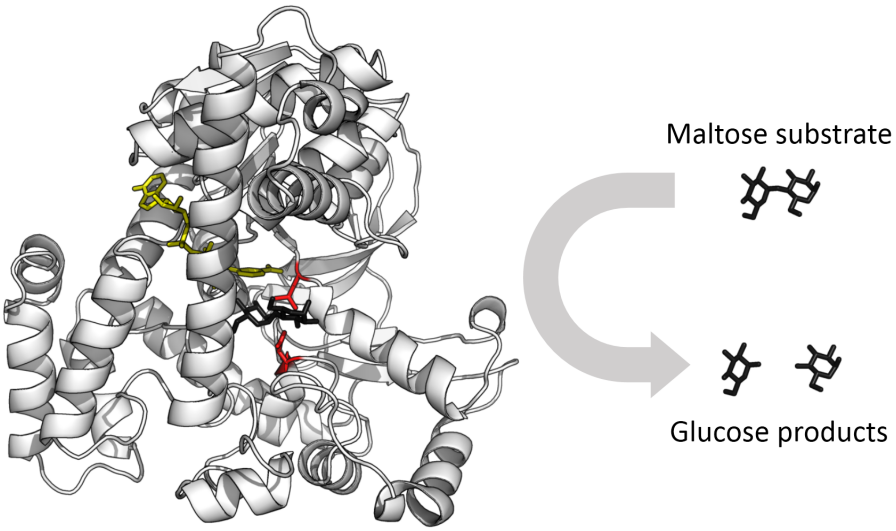


Figure 1.4: The enzyme *Glucosidase* converts the sugar maltose into two glucose sugars. Active site residues are in red, maltose substrate is in black, and *NAD* cofactor is in yellow. Reproduced from Wikipedia.

the substrate is divided into two discrete products. As seen in Figure 1.4 sometimes, enzymes additionally require helping molecules to be able to catalyze reactions. Such molecules are termed **cofactors**. **Intermediate products** are molecules produced during the conversion of substrates to products.

1.2 Selected machine learning models

As described in the previous section, biosyntheses are complex transformations driven by small intricate molecules and complex enzymes. Traditional biochemistry computational approaches usually include large database screening or brute-force alike searches in molecular spaces, which are often time-consuming and extremely limited. At its core, machine learning models typically extract significant features from objects having a large amount of training data, which reduces their complexity and focuses only on desired properties. I assume that the reader is familiar with basic concepts of machine learning. In this section, I directly introduce two types of models, which are essential for my work and allow me to significantly reduce the complexity of terpene synthases and terpenes.

1.2.1 Transformer

Real world data is full of sequences. For example natural language can be represented in a form of word sequences, in addition to, all the terms figuring

[0] `https://en.wikipedia.org/wiki/Enzyme`

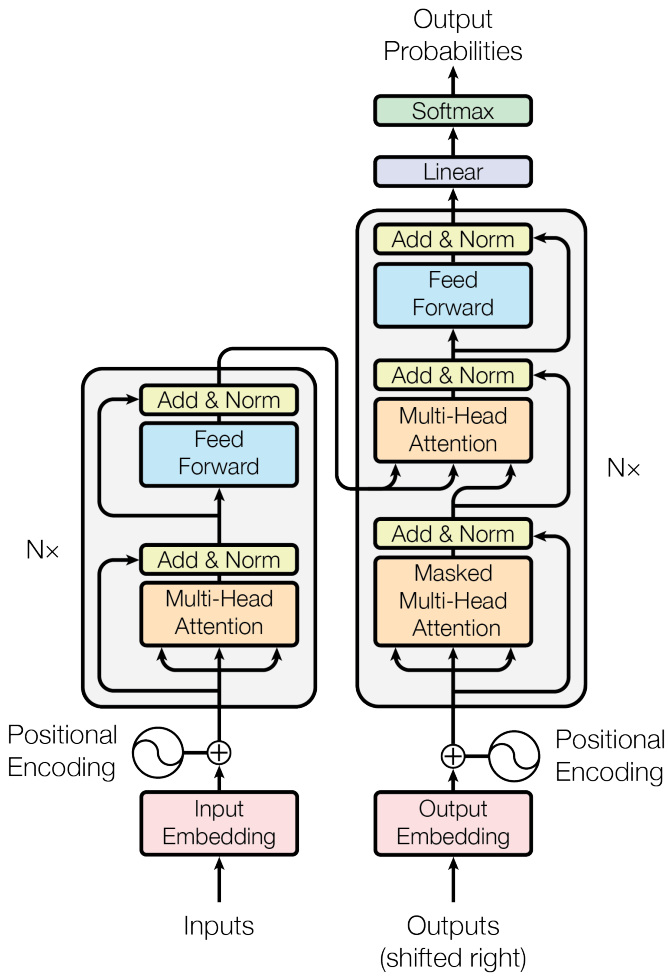


Figure 1.5: The Transformer – model architecture. Reproduced from [11].

in the central dogma of molecular biology[1] (DNA, RNA and protein) are also sequences. Natural language processing is one of the foremost fields of machine learning, because such tasks as translation or text summarization occur every day. Rapid development of this field has led to the emergence of Transformers [11], models currently achieving state-of-the-art performance in various applications on sequences.

Similar to the best earlier sequence to sequence models, the Transformer consists of two main parts: encoder and decoder. However, Transformer does not contain any recurrent layers, which permit recurrent neural networks to process sequences item by item. Instead, they are replaced with attention layers, which among other benefits, allow parallelizing of the model in a much

[1] "DNA makes RNA, and RNA makes protein" (`https://en.wikipedia.org/wiki/Central_dogma_of_molecular_biology`)

faster training process. Figure 1.5 shows the architecture of the Transformer.

At first, a sequence item passes through embedding layers (*Input Embedding*), where it is being converted to a vector of real numbers, for example, with the use of word2vec [12]. The model does not have recurrent layers, but there is a need to provide information about input sequence order. Therefore, the vector is summed up with another vector of the same length, thus encoding the relative position of the item within the whole sequence (*Positional Encoding*). There are different possible choices of this vector, and it can be either fixed or learned, but in the original paper [11], the authors propose to construct a fixed vector by utilizing trigonometric functions. There are sinusoids with different frequencies assigned for all the vector dimensions, and the positional encoding vector is obtained by passing the item's position as an argument to each sinusoid. This approach allows expressing the relative position independent of the vector length, differentiating relative positions by comparing sinusoids' values at certain positions. It was shown that this encoding is easy for the model to learn. Processed by these two operations, the vector is ready to proceed to the encoder. The encoder consists of N (6 in the original paper) consequently connected identical blocks, each containing a **multi-head attention mechanism** (*Multi-Head Attention*) and a regular feed-forward neural network (*Feed Forward*). Attention is a key feature of the Transformers. It allows considering the relation between words within sentences or contacts between amino acids within proteins. All the vectors corresponding to the items of the input sequence can be packed together as rows of some matrix X. Then the scaled dot-product attention can be formally defined in terms of three matrices $Q = XW^Q$, $K = XW^K$ and $V = XW^V$, where W^Q, W^K and W^V are being learned during the training:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V. \tag{1.1}$$

Rows of the Q, K, and V are referred to as query, key, and value vectors, and d_k is the length of key vectors. Matrix QK^T contains dot products of every query vector with all the key vectors in its rows, which can also be interpreted as attention measures between all pairs of items. Notice that, despite the dot product being symmetric, the attention measure is not symmetric regarding the original embedding vectors due to the distinct W^Q and W^K matrices. Additionally, all the values are divided by $\sqrt{d_k}$ to avoid extremely low gradients after applying softmax function[2] on every row, which normalizes the values, making them positive and sum up to 1. Finally, V is multiplied by the softmax($\frac{QK^T}{\sqrt{d_k}}$), which means that the output matrix of the attention function contains sums of value vectors weighted by the obtained attention scores

[2]The softmax function $\sigma : \mathbb{R}^n \to [0,1]^n$ is defined by the formula $\sigma(\mathbf{z})_i := \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$ for $i = 1, \ldots, n$ and $\mathbf{z} = (z_1, \ldots, z_n) \in \mathbb{R}^n$

in its rows. It is important to mention that this mechanism allows for capturing long-range interactions between sequence items as easy as short-range ones. This is a problem for the classical recurrent neural networks because long-range information simply vanishes during the backpropagation pass.

Multi-head attention is a combination of several scaled dot-product attentions, where each one is computed in parallel. All the output matrices are concatenated together and multiplied by another learned matrix:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{1.2}$$

$$\textbf{where } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V). \tag{1.3}$$

h is a number of attention heads, Concat is a matrix concatenation, and W^O is learned during the training. As is seen in Figure 1.5, both multi-head attention and feed-forward stacks are followed by the residual connection [13] and normalization [14] (*Add & Norm*) and are provided to simplify the optimization problem and to decrease the training time. The decoder part of the transformer is very similar to the encoder but has some significant changes. The decoder subsequently constructs the output sequence by having so-far decoded sequence on the input and utilizing the encoder output to create query and key vectors in the additional multi-head attention block. Also, the first attention layer masks undecoded positions of the sequence and considers only the decoded part. In each step, after the decoded vector is calculated, there is a need to get the desired output from the obtained vectors (for example, a word or an amino acid). Thus, for each item, the linear layer (*Linear*) predicts a vector of all possible outputs' scores, and the following softmax layer (*Softmax*) converts them to probabilities. Then the item with the highest probability is chosen as the final output of the decoder.

There are many different ways to train the Transformer end-to-end, but I would like to emphasize **self-supervision**, where some random items of the input sequence are masked, and the model learns to predict these items. This process allows producing a great deal of training data without having any labels. During the training process, the model extracts patterns and features of the sequences so that such a pre-trained model can be further used in various applications. Notice, two facts that encourage this type of training: (*i*) the Transformer's capacity is extremely high as it is a deep and complex model, which means it is designed to be trained on vast datasets, (*ii*) the Transformer is optimized to perform relatively fast training.

1.2.2 Variational autoencoder

Autoencoder is a fully-connected neural network of a symmetric bottleneck diagram that has a dimensionality of a middle hidden layer lower than the

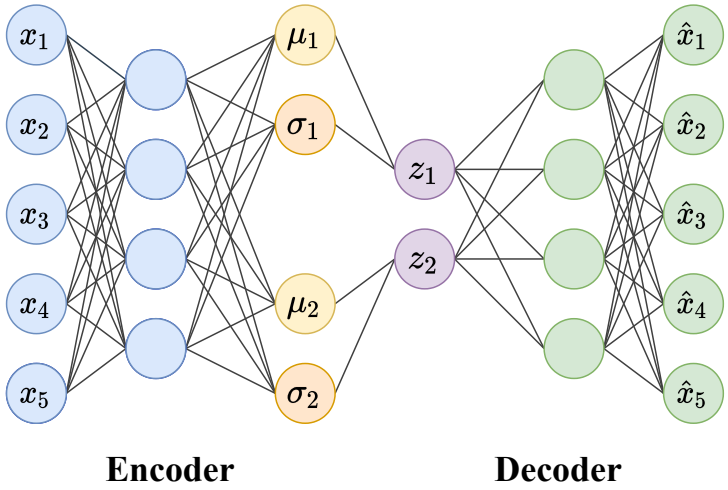


Figure 1.6: Variational autoencoder – example of the architecture with five-dimensional input and two-dimensional code.

input dimension[3]. This type of models is trained on the data x in an unsupervised manner by encoding the data to the lower dimensionality $z = f(x)$ and subsequently decoding it back to the original one $\hat{x} = g(z)$, where f and g are encoder and decoder respectively. The loss function, $\mathcal{L}(x, \hat{x}) = \mathcal{L}(x, g(f(x)))$, is simply a measure of the difference between the original data samples and their decoded versions. In the typical application, one is not interested in the restored input $\hat{x}$, but in the learned low-dimensional representations z (**latent vectors** or code), as to compress and decompress the data, autoencoder should learn significant data features stored in the code. It makes the models efficient for such tasks as dimensionality reduction or feature extraction. However, a space of all latent vectors (**latent space**) is usually formed from separated sparse clusters containing similar input objects, which is natural because such distribution facilitates minimizing the loss function. But suppose that the learned space is represented as a single pre-defined manifold, such that each vector from this manifold can be decoded back to the meaningful object of the original dimensionality. For example, a manifold of small molecules would allow discovering novel compounds by performing simple operations on its vectors. **Variational autoencoders** [16] (VAE) perform this by learning a latent space in a probabilistic manner and thus are known as efficient generative models.

Instead of directly learning latent vectors, VAEs learn parameters of the normal distribution $\mathcal{N}(\mu, \sigma^2)$ corresponding to each element, and elements of

[3]Such autoencoders are referred to as Undercomplete autoencoders. Sometimes autoencoders' code has higher or equal length than the input, but such variants are not important regarding this section. For more details about autoencoders, see for example [15, Chapter 14]

the latent vectors are being sampled from these distributions (Figure 1.6). It forces the learned space to be a continuous manifold, as even the same data sample can be encoded to different vectors, which are bounded by σ distances from the μ values vector. However, this improvement itself is not enough to obtain a manifold with good properties, because the model can learn different mean values and close-to-zero variances, which would make it similar to the classic autoencoder. To avoid this problem, one can force the model to learn, for example, standard normal distributions $\mathcal{N}(0,1)$ which can be formally expressed in terms of the Kullback–Leibler divergence (KL divergence) regularization added to the loss function

$$\mathcal{L}(x,\hat{x}) + \sum_{i\in\{1,\dots,n\}} D_{\mathrm{KL}}(p_i(z|x) \parallel \mathcal{N}(0,1)), \tag{1.4}$$

where n is a length of the latent vector, $p_i(z|x)$ is a probability density of the learned distribution corresponding to the ith element of the latent vector and KL divergence for two continuous probabilities P and Q is defined as $D_{\mathrm{KL}}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$. Such restriction also forces the training latent vectors to have low norms, so they all are localized near the center of the space. Lastly, sampling from the learned distributions makes the backward pass during the training impossible. Thus, a so-called reparameterization trick is employed, which still consists of learning μ_i and σ_i parameters, but obtaining z_i as $z_i = \mu_i + \sigma_i\epsilon_i$, where ϵ_i is sampled from the $\mathcal{N}(0,1)$ and $i \in \{1,\dots,n\}$.

Chapter 2

Related work

Despite the fact that biochemistry arose in the 19th century [17] and machine learning is being actively developed for the last decades, the application of machine learning in biochemistry is still immature. Present work is centered around the machine-learning prediction of biosynthesis, which has not been studied yet. However, in this section, I will discuss works that employ similar methods or data that I use to describe the current state of the intersection between two disciplines.

On the 30th November 2020, DeepMind posted a blog[4] called "AlphaFold: a solution to a 50-year-old grand challenge in biology", which states that the machine learning model learned to predict 3D structures of proteins from the amino acid sequence with a high level of accuracy (Figure 2.1). Biologists can employ the model as a core tool in their scientific research. This breakthrough proves that the employment of machine learning for biochemistry tasks is timely and worth studying. The described work is highly relevant for my thesis, as the enzyme 3D structure determines its activity and strongly affects the product of a biosynthesis reaction. Although the AlphaFold is not published yet, there are other works studying protein properties from the amino acid sequences. They usually employ machine-learning on models self-supervised pre-trainied from vast databases (millions of sequences) and transferring them to a downstream task. For example, study of Strodthoff et al. [18] shows high performance level of the pre-trained Long short-term memory-based [19] model on the three prototypical classification tasks: enzyme class prediction[5], remote homology[6] detection and gene ontology prediction [20]. Another study by Rivese et al. [21] shows the ability of the pre-trained Trans-

[4] https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology

[5] https://science.umd.edu/classroom/bsci424/BSCI223WebSiteFiles/ClassesofEnzymes.htm

[6] Shared ancestry in the evolutionary history of life

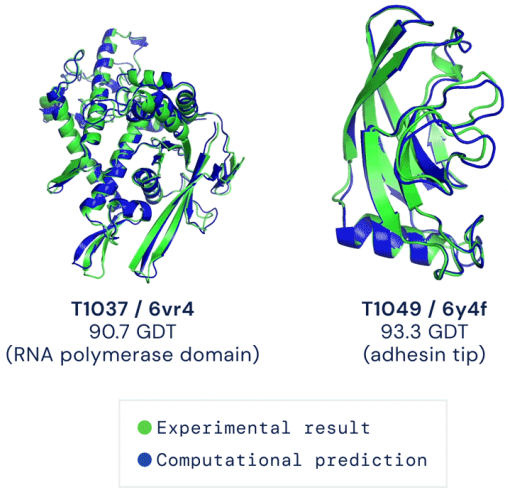


Figure 2.1: AlphaFold predicts highly accurate structures measured against experimental result. Reproduced from the DeepMind blog.

former [11] to learn the information about organizing principles and intrinsic biological properties of the proteins. Machine learning also shows promising results in protein redesign, which involves the development of new proteins with improved properties. Such mutations result in many changes in the protein amino acid sequence that are sometimes difficult to characterize. The number of possible changes is astronomical, so it is challenging to find the functionally interesting mutations. Study of Xu et al. [22] proposes a convolutional neural network that can assist in this task by searching for proteins with specific properties.

Terpenes are small molecules and are not represented as sequences of repeating fragments like proteins. Although the prediction of their biosyntheses has not been studied yet, there are other researches involving applications of machine learning on small molecules. For example, the work of Stokes et al. [23] shows a novel machine learning method for the discovery of new antibiotics. They utilized a recently developed Graph convolution neural network (GCN) and trained it to predict whether a given molecule inhibits the growth of *E.coli* bacteria. Subsequently, databases of molecules were screened, and the best candidate was tested *in vivo* on mice. Experiments showed that the molecule is effective not only against *E.coli* but can be considered as a broad-spectrum antibiotic (it was named *halicin*). The work of Gomez-Bombarelli et al. [24] describes the Variational Autoencoder (VAE) that learned a continuous space containing vector representations of molecules by encoding and decoding millions of synthetic compounds from the large ZINC database [25] (Figure 2.2). It allowed to automatically generate novel chemical structures by performing simple operations in the learned space, such as decoding ran-

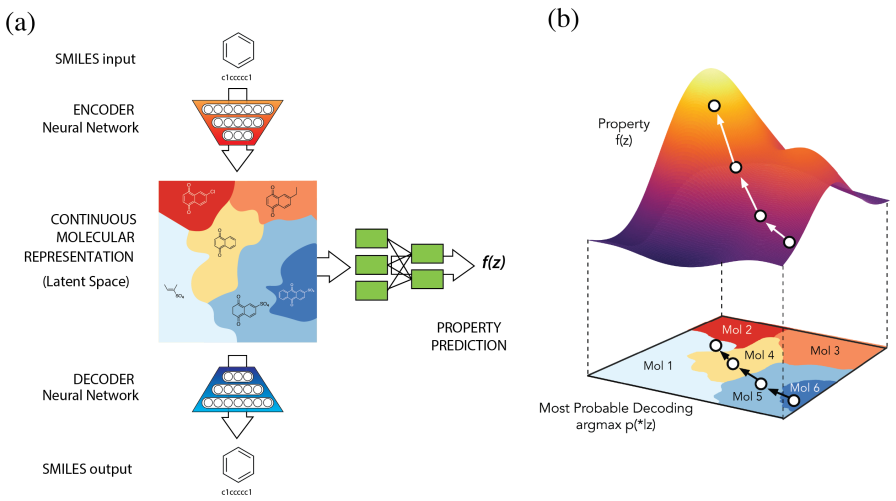


Figure 2.2: **(a)** A diagram of the Variational autoencoder for molecular design, including the joint property prediction model. Starting from a discrete molecular representation, such as a SMILES string, the encoder network converts each molecule into a vector in the latent space, which is effectively a continuous molecular representation. Given a point in the latent space, the decoder network produces a corresponding SMILES string. Another network estimates the value of target properties associated with each molecule. **(b)** Gradient-based optimization in continuous latent space. After training a surrogate model $f(z)$ to predict the properties of molecules based on their latent representation z, we can optimize $f(z)$ with respect to z to find new latent representations expected to have high values of desired properties. These new latent representations can then be decoded into SMILES strings, at which point their properties can be tested empirically. Reproduced from [24].

dom vectors, perturbing known chemical structures, or interpolating between molecules. The learned space of VAE can also be used to measure a molecular similarity, which is a core problem of chemoinformatics [26]. Finally, I want to emphasize the work of Eguchi et al. [27] as it utilizes an approximately equal number of training data as I have used in my thesis. Furthermore, they utilized alkaloids, which are similar to terpenes. Authors trained a GCN to classify alkaloids by the starting substances of their biosynthetic pathways and the model achieved an accuracy of 97.5%. The classification task is much easier than generation of molecules, but still, the work has many similar aspects to the present one.

Chapter 3

Problem definition and dataset

In this chapter, I define the problem studied in this thesis. First, I discuss terpenes and their biosynthesis. Second, I analyze a dataset of characterized terpene biosyntheses, which forms a base for me to expand on. Finally, I formulate an objective of the present work.

3.1 Terpene biosynthesis

Terpenes are a diverse and significant class of organic compounds from plants. They are volatile and produce odors that discourage herbivores or insects from attacking the plant and in general, are unappealing to their predators. The essential oils of many plants, flowers, and trees, are made up of terpenes and their related compounds [28].

Terpenes are simple molecules made up only of carbon and hydrogen atoms and derived from isoprene units forming chains, branches, or cycles. Depending on the number of isoprene units, terpenes are classified into distinct categories. Most common are **monoterpenes** (2 isoprene units), **sesquiterpenes** (3 isoprene units), **diterpenes** (4 isoprene units) and **triterpenes** (6 isoprene units). Terpenes satisfy the chemical formula $(C_5H_8)_n$, where C_5H_8 is a chemical formula of an isoprene unit and n corresponds to the number of units. Figure 3.1 shows structural formulas of isoprene and some particular terpenes belonging to different classes. When terpenes are modified, such as by attachment of an oxygen or rearrangement of the carbon atoms, the resulting compounds are generally referred to as **terpenoids**. Despite the chemical difference between compounds, these terms are sometimes used interchangeably.

Terpenes have various medical uses. Table 3.1 shows different types of terpenes along with some examples of their usage in medicine. Antimicrobial properties or the ability to kill or stop the growth of a microorganism has

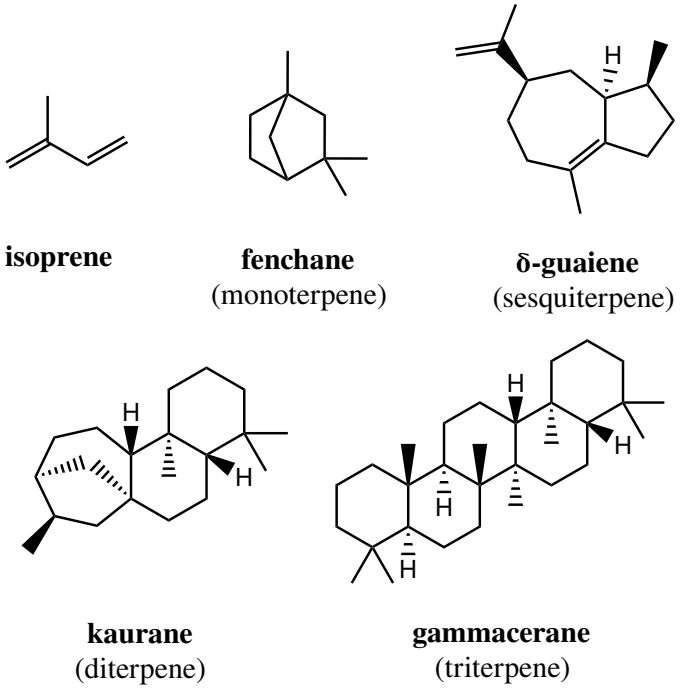


Figure 3.1: Skeletal formulas of isoprene (the bulding block of terpenes) and selected terpenes belonging to different classes.

Classification	Carbon atoms	Species produced from	Medicinal uses	References
Monoterpenes	C_{10}	*Quercus ilex*	Fragrances, repellent	Loreto et al. [29]
Sesquiterpenes	C_{15}	*Helianthus annuus*	Treat malaria, treat bacterial infections, and migraines	Chadwick et al. [30]
Diterpenes	C_{20}	*Euphorbia, salvia miltiorrhiza*	Anti-inflammatory, cardiovascular diseases	Vasas and Hohmann [31], Zhang et al. [32]
Triterpenes	C_{30}	*Centella asiatica*	Wound healing, increases circulation	James and Dubery [33]

Table 3.1: Different types of terpenes and their pharmaceutical properties. Reproduced from [34].

been seen in these compounds. Also, they are commonly used in traditional and modern medicine [35]. Terpenes are widely acclaimed for their anticancer activity. An early 1997 study concluded that a combination of monoterpenes, diterpenes, and sesquiterpenes can effectively be used to treat cancers that occur in the colon, brain, prostate gland, and bones [34]. Researches also

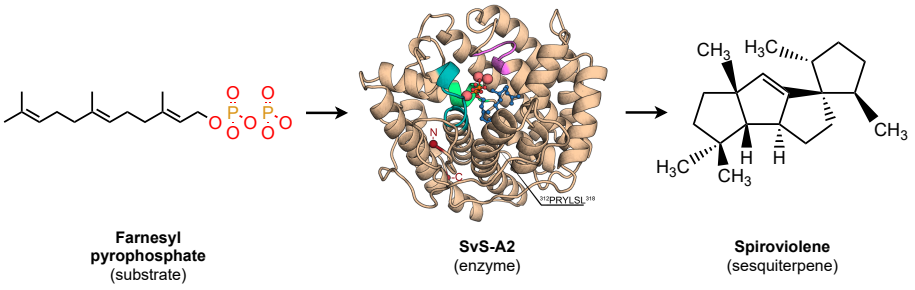


Figure 3.2: Biosynthesis of the sesquiterpene *spiroviolene*. Enzyme structure is reproduced from [38].

show that some terpenes possess antiviral activities [36, 37].

Terpene biosynthesis is the key subject of this work. Compared to other enzymatic synthesis, these reactions are relatively strict and defined, which motivates studying them with machine learning methods. In general, they are derived from simple substrates, formed as plain carbon sequences with a *pyrophosphate* at the end (for example, *geranyl pyrophosphate* or *farnesyl pyrophosphate*[7]). During the synthesis, the *pyrophosphate* is being detached and the rest of the substrate is being cycled to a more complex terpene molecule (Figure 3.2). Typically, these reactions are not straightforward and clear, because terpene biosynthesis usually consists of multiple steps and may have several exceptions. The majority of terpenes share the same substrates among their types. For example, almost all the sesquiterpenes are derived from the *farnesyl diphosphate*, but some enzymes can produce sesquiterpenes from other substrates. Although the terpene chemical formula is strictly defined, some enzymes require additional H_2O cofactors in order to catalyze a reaction, resulting in the additional oxygen atoms in the product[8]. Further, some terpene synthases can produce multiple of different compounds, but others – only a single one. Such compounds usually differ in the types of bonds or the number of bonds and atoms present, the number depending on the compared reaction.

Regarding my task, I focus on three main components of a terpene biosynthesis: substrate, enzyme, and product (terpene). While substrates can be distinguished as just categories – because they strongly correlate with product terpene types and their exact molecular structures are relatively simple – terpene synthases and product terpenes are extremely complicated objects. Terpene synthases are sequences typically consisting of 400 to 800 amino acids and the whole sequences do not possess any simply interpretable semantics, which means that their complexity cannot be easily reduced. Terpenes are

[7] See, for example Figure 2.
[8] See, for example, `https://www.uniprot.org/uniprot/A0A1D6EFT8`.

represented as intricate skeletal structure graphs on tens of vertices or complex SMILES strings encoding their structural features in tens of characters. Biosynthesis prediction implies a generation of such graphs or SMILES strings from a substrate and an amino acid sequence on the input. The number of possible amino acid sequences of length 600 is equal to 20^{600} and according to the MOLGEN [39] software estimation only the number of sesquiterpenes is larger than 61,101,340 (number of possible $C_{15}H_{24}$ isomers). This means that the core problem is to find a mapping between two immense spaces.

3.2 Terpene synthases database

The basis of the work is a comprehensive database of exhaustively manually collected characterized terpene syntheses referred to as the **TPS database**. All the belonging terpene synthases were manually collected from the UniProt [10] database and the products were referenced in ChEBI [40] and PubChem [41] databases. Training data is a cornerstone of machine learning; however, the database contains only 750 entries. Therefore, I would like to describe it more in detail to form a better understanding of the problem. Each entry of the dataset corresponds to a particular biosynthesis reaction and has 13 string features that are explained in the following list with examples in brackets:

Uniprot ID — Terpene synthase Uniprot database ID (*B5HDJ6*)

Name — Terpene synthase name (*Selina-4(15),7(11)-diene synthase*)

Amino acid sequence — Terpene synthase amino acid sequence (*MEPELTVP PLFSPIRQAIHP...*)

Species — Latin name of a species, in which biosynthesis passes (*Streptomyces pristinaespiralis*)

Kingdom — Kingdom which a species belongs to (Bacteria)

Type — Product terpene type (mono-, di-, sesquiterpene...) (*sesq*)

Substrate — Biosynthesis substrate shortened IUPAC name (*(2E,6E)-FPP*)

Cofactors — Cofactors required for the enzyme's activity (*H2O*)

Name of intermediate — Name of intermediate product of a synthesis (*(+)-copalyl diphosphate*)

SMILES of intermediate	SMILES string of intermediate product (*[H][C@@]1(CCC(C)=CC1)C(=C)CCC=C(C)C*)
Name of product	Names of product terpenes separated with semicolon and sorted by the natural abundance rate (*7-epi-ent-eudesmane-5,11-diol*)
Chemical formula of product	Chemical formula of product terpene (*C15H24*)
SMILES of product	SMILES strings of product terpenes separated with semicolon and sorted by the natural abundance rate (*[C@@]12([C@@](CCC[C@@H]1C)(CC[C@H](C2)C(O)(C)C)C)O*)

Such features as *Name*, *Species*, *Name of intermediate* and *Name of product* are valuable for a human but do not provide useful information for a machine learning algorithm. Thus, I do not use them in my work. Feature *Kingdom* can be helpful for some applications, but it is not relevant for my task. There are three features that reflect the specificity of a reaction and are present in only a portion of the entries: *Cofactors*, *Name of intermediate* and *SMILES of intermediate*. Since they represent some special cases and they are not of much importance for the present problem, I ignore them. As well as the *Chemical formula of product*, because it can be programmatically derived from the *SMILES of product*. *Uniprot ID* is a handy feature for my experiments, as long as it can provide an easy identification of a synthase. Although the dataset consists of 13 features, only 3 of them are significant one that I will use: *Substrate*, *Amino acid sequence* and *SMILES of product*.

Table 3.2 includes the following quantitative characteristics of each feature: count of present values (*Count*), number of unique values (*Unique*),

Feature name	Count	Unique	Top value	Top frequency
Uniprot ID	750	714	H8ZM70	3
Name	750	414	Terpene synthase	26
Amino acid sequence	750	704	MATLRISSALIYQNTLTHHFR...	4
Species	750	281	Arabidopsis thaliana	36
Kingdom	750	7	Plants	594
Type	748	8	sesq	411
Substrate	750	30	(2E,6E)-FPP	403
Cofactors	11	3	H2O	9
Name of intermediate	40	8	ent-copalyl diphosphate	17
SMILES of intermediate	40	9	[C@@H]1(CC/C(/C)=C/CO...	17
Name of product	750	336	ent-kaurene	23
Chemical formula of pr...	750	31	C15H24	344
SMILES of product	750	334	[H][C@]12CC[C@@]34C[C...	24

Table 3.2: Quantitative characteristics of the TPS database features

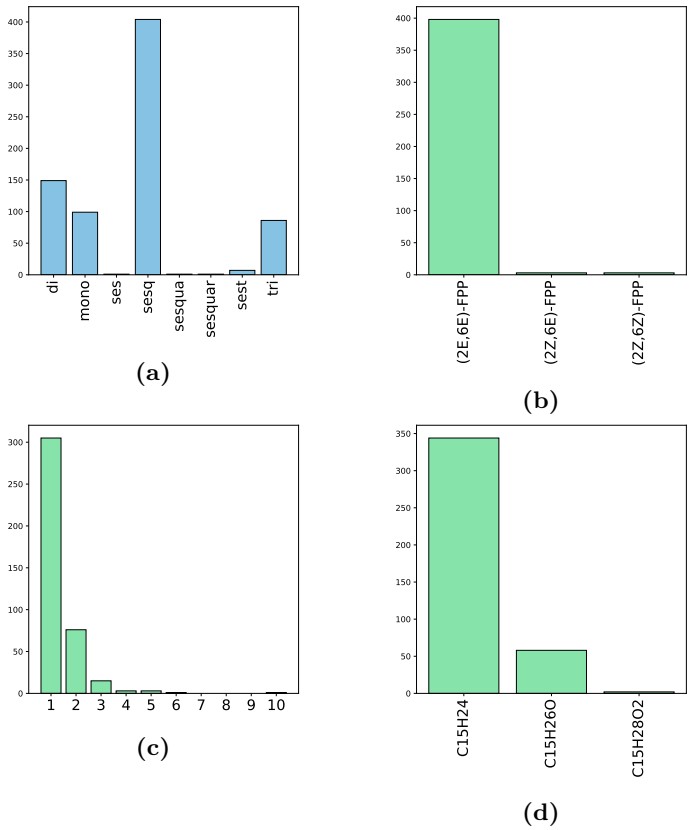


Figure 3.3: TPS database histograms, blue – whole database, green – sesquiterpenes only. **(a)** Terpene types; **(b)** substrates; **(c)** number of possible products; **(d)** chemical formulas of products.

mostly occurring value (*Top value*), and the frequency of its occurrences (*Top frequency*). It can be seen that all the features except for enzyme-related ones (*Uniprot ID* and *Amino acid sequence*) repeat through the dataset, and only a small fraction of synthases require cofactors or produce intermediate products. The most valuable observation from the table is that although there are eight different terpene types present in the TPS dataset, sesquiterpene biosynthesis form the largest part of the reactions (Figure 3.3a depicts a histogram corresponding to all the eight types). Figure 3.3b, Figure 3.3c and Figure 3.3d additionally show distributions of sesquiterpene biosynthesis substrates, numbers of possible products and chemical formulas respectively. This implies that the **synthases producing single possible sesquiterpene from the *(2E,6E)-FPP* substrate constitute the majority of terpene synthases characterized to date**.

3.3 Objective of the thesis

The thesis aims to study the capability of machine learning regarding the prediction of products related to terpene biosynthesis. My objective can be formulated in terms of a question to the machine learning model – **providing a substrate and enzyme as the input, what are the products of the output?** Considering the facts that the type of biosynthesis can be determined from the synthase sequence (Section 4.3.1) and that sesquiterpene syntheses constitute the majority of characterized reactions, I focus on the prediction of sesquiterpene biosynthesis as a proof of concept with aims to illustrate that any terpene biosynthesis can be predicted. An accurate prediction of sesquiterpenes would indicate that the prediction of terpenes, in general, can be achieved by applying analogous approaches for all types independently. Notice, that the problem is not of a classification but of a generative character. This means that the part of the objective is to develop a method allowing to precisely compare specific molecules in order to assess the performance of machine learning models.

Chapter 4

Methods and experimental setup

In this chapter I describe the methods I employ and the experiments I conduct for the machine-learning prediction of sesquiterpene biosynthesis. I start with a general motivation and intuition behind the proposed solution and then subsequently describe all the components.

Considering only sesquiterpenes, the problem reduces to the prediction of a biosynthesis product based on a synthase, as almost all sesquiterpenes share the same substrate. The input is an enzyme represented as a long sequence of characters over the dictionary of 20 amino acids and the output is a sesquiterpene represented as a SMILES string. Machine learning models often operate only on continuous vectors; however, my 318 training samples from the TPS database are evidently not enough to obtain meaningful numerical reproductions (embeddings) from the before-mentioned sophisticated representations. I propose to solve this problem by employing pre-trained machine learning models on vast databases for both input enzyme sequences and output sesquiterpenes' SMILES strings. Pre-training usually implies extracting general features from the large amounts of similar objects in an unsupervised or self-supervised manner. There are millions of publically available proteins and small molecules, which can be used for such pre-training. It means that sesquiterpene synthases and sesquiterpenes can be converted into vectors encoding their significant properties. Therefore, I can solve the problem of sesquiterpene prediction in a transfer-learning fashion by fine-tuning the models for the sesquiterpene prediction task and operating between the corresponding vector spaces. The full proposed solution is depicted in Figure 4.1 and explained in the next paragraph, while all three components are described more in detail in the Section 4.3.

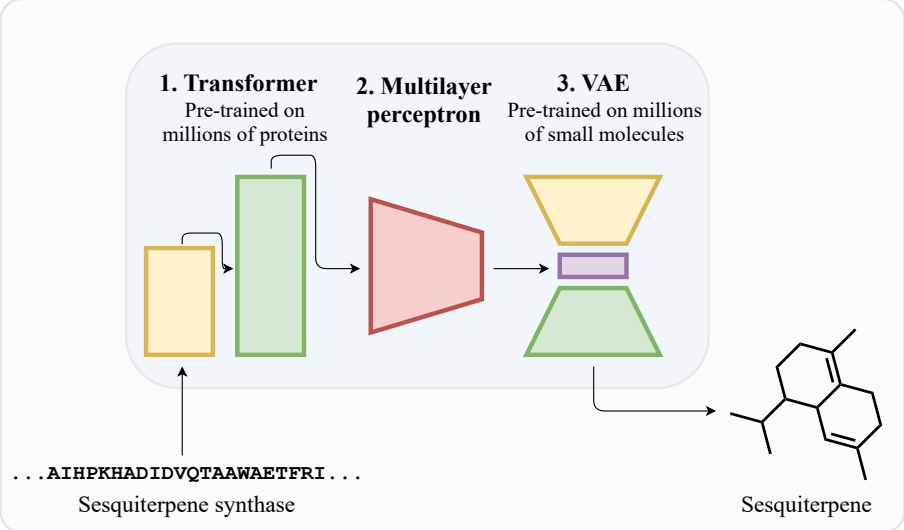


Figure 4.1: Pipeline of the proposed machine learning solution for the sesquiterpene biosynthesis prediction. First, the Transformer (**1**) is pre-trained on millions of proteins, and the Variational Autoencoder (VAE) (**3**) is pre-trained on millions of small molecules. In order to predict the sesquiterpene, Multilayer perceptron (**2**) is trained on the TPS database to produce a latent vector of the VAE having synthase embedding on the input. Finally, the predicted molecule is obtained by decoding the latent vector of VAE.

Despite the protein complexity, obtaining enzymes' embeddings is relatively straightforward, as it requires only meaningful encoding, which is a common practice in modern applications of machine learning. Transformers have already proven their efficiency in such tasks and are being actively studied for applications on protein sequences. Thus, I can apply a pre-trained Transformer to encode synthases (**1**). However, it is challenging to create a vector space of small molecules, which would allow me to predict vectors corresponding to sesquiterpenes and decode them back to an interpretable form. A model not only needs to associate meaningful continuous representations with the molecules, but also return them back to the original representations. Variational Autoencoders (VAEs) are powerful models for such a purpose and showed a high level of performance in numerous applications including molecule generation. It means that I can use a latent space produced by pre-trained VAE to operate on small molecules (**3**). Having sesquiterpene synthases' embeddings and learned latent space of VAE, the task is to find a mapping between two vector spaces, which is a typical task for the classic Multilayer perceptron[9](**2**).

[9]From here on out I use the *Multilayer perceptron* as a synonym for the *Artificial feed-forward neural network*.

4.1 Data preparation

The TPS database is the foundation of this work. In order to employ it for the training of machine learning models and to perform their assessment, I needed to rationally split a total of 745 valid samples into folds. First of all, 86 of them were manually selected as a *hidden test fold*, which is interesting from the biology perspective and reasonable for the final testing reactions. This fold will be used to test the model efficiency only after achieving high prediction accuracy on the rest of the dataset. The remaining 659 samples were divided into 10 folds of approximately equal size (9 folds of 66 samples and 1 fold of 65 samples). One of them constitutes the *test fold*, and the rest are employed for the *k-Fold Cross-Validation*, where $k = 9$. This method is a conventional approach for the model training and validation on a small dataset, as each sample affects the validation accuracy exactly once. Additionally, all the sesquiterpenes were evenly distributed across the folds (34 to 37 molecules in each fold), which allows me to utilize them in sesquiterpene-specific tasks.

The TPS dataset contains only terpenes for which the biosyntheses are characterized, however there are many more terpenes and similar compounds in public databases. For example, ChEBI database provides ChEBI Ontology, which is a structured classification of the molecules[10] contained within the whole database. It can be perceived as a directed tree graph, where each node corresponds to a particular molecule and edges directed from leaves to root express the *is* relation. For example, *limonene* (CHEBI:15384) *is* a *monoterpene* (CHEBI:35187) and *monoterpene* (CHEBI:35187) *is* a *terpene* (CHEBI:35186). I web-scraped all the molecules recursively starting from the *terpene* node, which resulted in a dataset of 3637 molecules. In other words, all

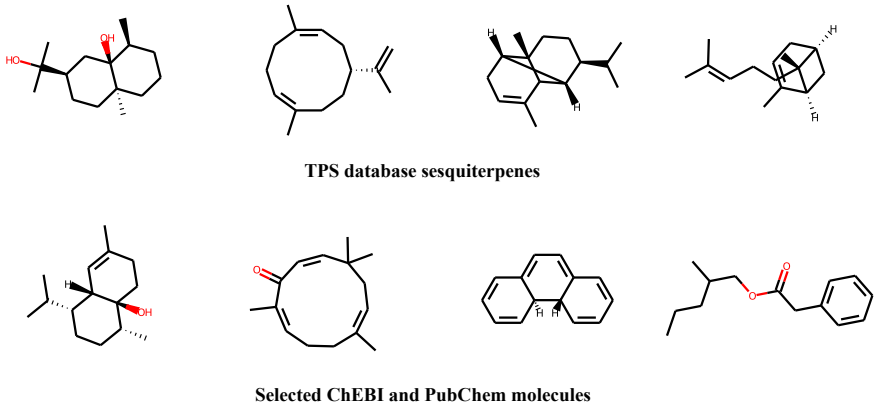


Figure 4.2: Examples of the molecules belonging to the SesqSim dataset.

[10]Here and further by the molecule I mean a corresponding SMILES string.

the collected molecules *are* transitively *terpenes*. Additionally, I downloaded 400,000 PubChem molecules containing exactly 13, 14, 15 and 16 carbon atoms (100,000 each) to have more compounds similar to sesquiterpenes.

Based on the sesquiterpenes of the TPS database and the downloaded ChEBI terpenes, I created a dataset containing compounds the most similar to sesquiterpenes. I filtered these datasets by putting the following constraints on their entries: (*i*) molecule contains only carbons, hydrogens, and oxygens; (*ii*) number of carbons is greater than 13 and less than 17; (*ii*) number of oxygens is less than 3; (*iii*) molecule represents a connected graph (SMILES string does not contain . symbol). Since the obtained dataset contained only 725 molecules, I additionally added PubChem molecules with 14, 15 and 16 carbons while also satisfying the other previously mentioned constraints. It resulted in a dataset of 16,311 molecules, which I will refer to as a **SesqSim** dataset. Figure 4.2 provides examples of the molecules belonging to the SesqSim dataset. Furthermore, I will refer to **SmallSesqSim** as a subset of SesqSim with a reduced number of molecules containing 14 and 16 carbons to have a smaller dataset with the most similar compounds. Additionally, I will refer to **SesqSim13** as a SesqSim dataset without molecules containing 14 and 16 carbons, but comprising additional compounds with 13 carbons. This dataset generally has smaller, similar to sesquiterpenes, molecules than SesqSim and SmallSesqSim.

4.2 Evaluation metric selection

Since this work is about the prediction of particular molecules, I need to be able to determine the similarity between predicted and actual compounds. In other words, I need to define an evaluation metric for machine learning models. Measuring similarity between molecules is an essential problem of biochemistry. It commonly occurs in such applications as predicting the properties of chemical compounds or discovering new antibiotics by screening large databases. Such researches is based on the principle that **structurally similar compounds are more likely to have similar properties** [42]. In general, any similarity score is subjective and depends on a purpose, especially for such abstract objects as graphs. However, graphs of chemical structures are strongly constrained by chemical rules, and despite the molecular diversity, different compounds contain identical substructures.

A **molecular fingerprint** of a molecule is the fixed-length bit vector, where in the simplest case each bit indicates the presence or absence of a particular substructure of the molecule. This approach enables the comparison of compounds by juxtaposing corresponding bit vectors. This way of encoding molecules is a golden standard in various applications, because it is a relatively simple general framework that produces vectors, interpretable by both

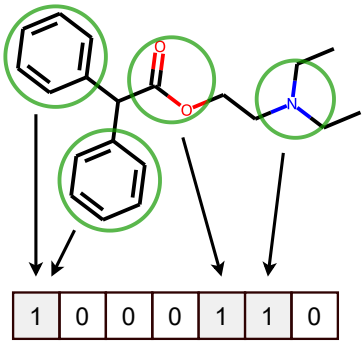


Figure 4.3: Visual explanation of the molecular fingerprint. True bits correspond to the present substructures.

human and machine. However, it has several weaknesses, which are revealed in Figure 4.3. The bit vector does not contain information about the number of present fragments, consider the topology of the molecule, and takes into account only specific substructures.

Molecular fingerprints are widely used, and there are many publications proving their efficiency. There exist different types of fingerprints varying in the determination of molecular substructures. The following list enumerates the three most popular types with a short informal algorithm description in brackets:

- MACCS key (determines the presence or absence of 166 predefined fragments and produces a bit vector with each bit corresponding to a particular fragment)
- Circular fingerprint (determines all subgraphs induced by the neighborhoods of each heavy[11] atom up to the given radius and hashes them into a bit vector of fixed length)
- Path-based fingerprint (similar to the circular fingerprint, but determines all possible paths up to the given length instead of neighborhoods)

Molecular fingerprints is an active area of research and more sophisticated types of algorithms are being actively developed. For example, the study [43] proposes a 71,375 bits-long fingerprint formed as a concatenation of 24 different fingerprints, including ones enumerated above. It is designed specifically for deep learning models to provide them as much information about molecules as possible and to let them extract task-specific features in deep stacks of layers. Another example is described in the work by Seo et al. [44], where authors

[11] Any atom that is not hydrogen

designed a fingerprint that is more appropriate for natural compounds then the regular ones. It mainly contains information about substructures such as complex fused rings[12] or fragments with a large number of oxygen atoms, which typicaly occure in nature.

Molecular fingerprint selection

Since I will use fingerprints only to compare model performances, I need to choose a relatively simple and well-recognized fingerprint independent of the models used in the primary experiments. The similarity is supposed to be measured on sesquiterpenes and molecules similar to them, therefore a fingerprint will be selected reflecting features specific for this class of compounds. Thus, I conducted a brief survey on the most common types of fingerprints in order to determine which one is the most sensitive to sesquiterpenes. Sensitivity is defined as high values for the three factors calculated on the set containing fingerprints of all the unique sesquiterpenes from the TPS database (410 molecules) and additional 410 hydrocarbons from the SesqSim dataset. The factors being: (i) fraction of different bits within all fingerprints, (ii) mean fraction of different bits between all pairs of sesquiterpenes and all the selected SesqSim molecules, (iii) mean fraction of positive bits within all fingerprints. These factors can be formally defined using a set bit-vectors notation described in the Mathematical notation as following:

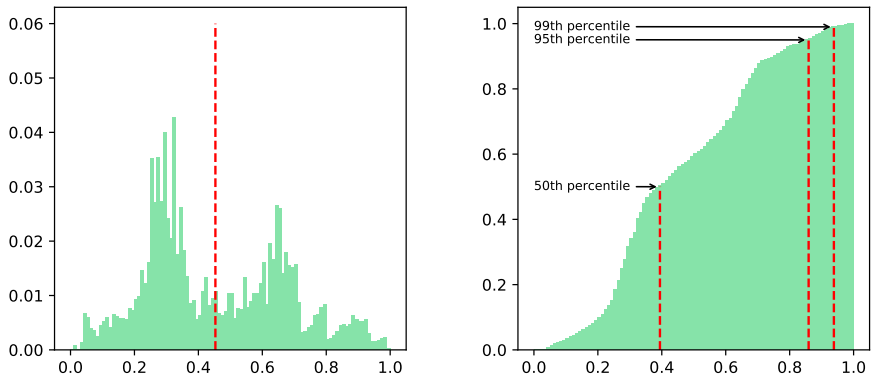
$$g(A) = 1 - \frac{1}{|F_1|}\left|\bigcap_{F \in A} F\right|, \text{ where } F_1 \in A, \tag{4.1}$$

$$l(A,S) = \frac{1}{|A \times S|}\sum_{(F_1,F_2)\in A\times S}\frac{|F_1 \triangle F_2|}{|F_1 \cup F_2|}, \tag{4.2}$$

$$p(A) = \frac{1}{|A|}\sum_{F \in A}|F|_{/+}, \tag{4.3}$$

where A is a set of all fingerprints and $S \subset A$ is a subset of the TPS database sesquiterpenes fingerprints. Notice that $g(A)$ can be perceived as a global measurement of present sesquiterpene information, since the Equation 4.1 shows how many bits differ within all the fingerprints. However, a fingerprint leading to the $g(A)$ equal to 0.5 is not automatically better than the one leading to 0.2, because there is a chance that in the first case the majority of bits corresponding to the 0.5 fraction are equal within almost all the vectors with some rare exceptions. In the second case the bits from the 0.2 fraction could frequently change within the vectors. Hence $l(A,S)$ factor is required,

[12] `https://www.qmul.ac.uk/sbcs/iupac/fusedring/FR1.html`



(a) **(b)**

Figure 4.4: Distribution of the evaluation metric on all pairs of selected sesquiterpenes and similar compounds(330,229 pairs): **(a)** empirical density function, red line depicts mean value; **(b)** empirical cumulative distribution function, red lines depict percentiles.

which in some sense shows the local difference as it expresses a mean number of different bits within pairs of vectors. I compute $l(A, S)$, but not $l(A, A)$, as one of the arguments of the evaluation metric may always be a TPS database sesquiterpene fingerprint. It is important to mention that the $l(A, S)$ is not enough by itself since, as a worst scenario, all the different bits could share the same positions between pairs. That is why both $l(A, S)$ and $g(A)$ are needed to assess the fingerprint. Additionally, $p(A)$ is considered to avoid bit vectors' sparsity, which is important for further fingerprint comparison as discussed in the next paragraph. To combine three factors into a single real number, I compute their harmonic mean, which defined as $h(x_1, x_2, x_3) = \frac{3x_1x_2x_3}{x_1x_2+x_1x_3+x_2x_3}$ for three numbers x_1, x_2 and x_3, because the ranges of possible values are not independent. Table 4.1 contains calculated values for the tested fingerprints. The 1800 bits long Path-based fingerprint with the maximum path length equal to 18 has the maximum harmonic mean of the three factors, making it the best fingerprint for the evaluation metric on machine learning models.

Fingerprints metric selection

After the type of fingerprint is determined, there is a need to choose a metric on bit vectors to have a real number expressing a measure of similarity between molecules. In the same manner, as I have chosen a simple and widely used fingerprint, I am going to use a Tanimoto similarity coefficient, which is defined as

$$T(A, B) = \frac{|A_+ \cap B_+|}{|A_+ \cup B_+|}, \tag{4.4}$$

Name	Total number of bits	g	l	p	Harmonic mean
MACCSKeys	167	0.473	0.108	0.140	0.162
Path-based 14	1800	1.000	0.435	0.616	0.609
Path-based 18	1500	1.000	0.413	0.664	0.609
Path-based 18	**1800**	**1.000**	**0.433**	**0.626**	**0.611**
Path-based 18	2100	1.000	0.446	0.587	0.607
Path-based 22	1800	1.000	0.433	0.626	0.611
Circular 4	1000	0.978	0.045	0.030	0.053
Circular 4	4000	0.603	0.011	0.008	0.014
Circular 12	1000	1.000	0.075	0.047	0.085
Circular 12	4000	0.947	0.020	0.012	0.022
Circular 16	1000	1.000	0.076	0.048	0.085
Circular 16	4000	0.948	0.020	0.012	0.022

Table 4.1: Sesquiterpenes fingerprints survey results. The number after the fingerprint name represents the maximum length of the path for the Path-based fingerprints and the maximum diameter for the Circular fingerprints. *Path-based 18* fingerprint highlighted in red has the maximum harmonic mean of the g, l and p values.

where A and B are some fingerprints. It is a straightforward way to measure molecular similarity because it indicates the fraction of common fragments concerning all fragments of two fingerprints. In general, Tanimoto similarity is an appropriate choice for fingerprint-based similarity calculations, because it retrieves the maximum information content of the total information carried by eight different tested metrics [45]. The definition implies that $(\forall A, B \neq \emptyset)(T(A, B) \in [0, 1])$, where $T(A, B) = 0$ means that A and B have no fragments in common and $T(A, B) = 1$ means that A and B contain only the same fragments.

Evaluation metric interpretation

While the Tanimoto similarity's theoretical background is transparent, it is not obvious how to interpret empirically obtained value from the $[0, 1]$ interval. Thus I propose to analyze the empirical distribution of the Tanimoto similarities on the chosen fingerprint by comparing the same 330,229 pairs that were selected for the $l(A, S)$ calculation. Figure 4.4 shows the plot of an empirical density function. Although the selected compounds are similar, Tanimoto similarities are distributed across the entire range of possible values. The mean value of the distribution is equal to 0.45, which satisfies an intuitive expectation, that on average, two random molecules should have a similarity score close to 0.5. Figure 4.4b depicts an empirical cumulative distribution function, which helps understand how the values can be interpreted. For example, two compounds having a similarity score of 0.4 are more similar than

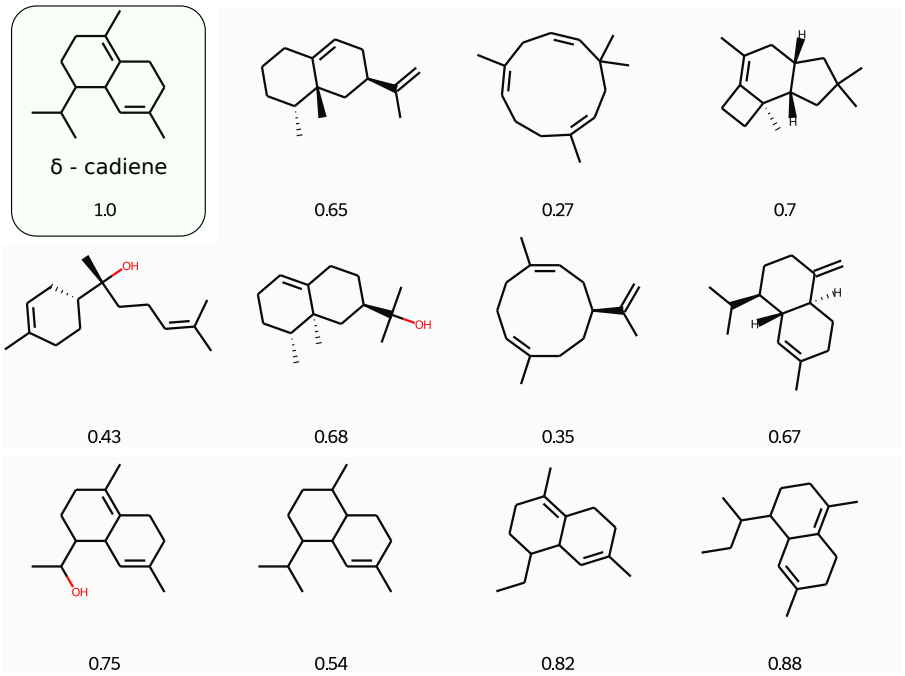


Figure 4.5: Evaluation metric on *δ-cadinene* (green box) with other selected molecules. The blue box contains other sesquiterpenes, and the orange box contains distorted *δ-cadinene* molecules (without consideration of chemical rules): change of a carbon atom to an oxygen, change of a double bond to a single bond, deletion of a bond, addition of a bond (changes are enumerated from left to right).

the 50% of other pairs. In terms of this work's main task, the Tanimoto similarity score higher than 0.86 (95th percentile) between actual and predicted products means that the predicted molecule belongs to the 1% of the best possible predictions among similar compounds. Figure 4.5 shows similarity scores between *δ-cadinene* and other selected molecules. The values confirm that the selected approach to compare molecules indeed indicates the measure of similarity and is sensitive for the sesquiterpenes.

I choose the Tanimoto similarity score on 1800 bits long Path-based fingerprints (maximum path length equals 18) as the evaluation metric for machine learning models, as it is a discriminative metric with respect to sesquiterpenes. Also, I want to emphasize two facts. (*i*) Since the distribution discussed above was obtained empirically, the interpretation of the similarity scores should not be perceived with great precision. The goal was to choose a rational evaluation metric and form an overall understanding of the score. (*ii*) Selected fingerprint and metric have good properties for comparison of sesquiterpenes and compounds similar to them, but it is not necessarily true for arbitrary molecules, which corresponds to the aim of the whole analysis. Additionally, I define the evaluation metric score to equal zero

if any of the argument molecules is not valid. In the further text, I will refer to the *evaluation metric score* as a mean score of all the validation samples across the folds discussed in the previous section.

4.3 Machine learning models pipeline

The following sections describe each part of the machine learning models pipeline (Figure 4.1) proposed as a solution for the prediction of sesquiterpene biosynthesis. Additionally, I describe important experiments elucidating the choice of particular models.

4.3.1 ESM-1b Transformer

In order to obtain sesquiterpene synthases embeddings, I employ Facebook AI Research ESM-1b Transformer used in the [21]. It has already proven its efficiency in such tasks as predicting protein substructures or amino acid contacts, making the model ideal for terpene synthases based on the structure-function relationship. The ESM-1b model was pre-trained on 250 million Uniparc [10] protein sequences on the masked language modeling task. For each input sequence 15% of amino acids were selected, and from that 15%, 80% were substituted with a special "masking" token, 10% were changed to a randomly chosen alternate amino acid and the rest 10% were left unchanged. Since the task was to predict a probability for amino acids substituting masked tokens, the authors selected an average cross-entropy for each training batch as a loss function. So, for each prediction of amino acid, it can be defined as

$$H(p, q) = -\sum_{x \in \mathcal{X}} p(x) \log q(x), \tag{4.5}$$

where $\mathcal{X}$ is a discrete random variable expressing possible amino acid tokens, p is a true distribution over this variable and q is a predicted one. The model produces 1280-dimensional vectors with close-to-zero values.

TPSs embeddings space visualization

To test whether TPSs embeddings, obtained from the pre-trained ESM-1b Transformer, capture substrate-specific folding, I applied the Uniform Manifold Approximation and Projection (UMAP) algorithm to reduce the dimensionality of all vectors from 1280 to 2 and visualized the reduced space (Figure 4.6). It can be seen that synthases' embeddings form clusters by substrates they take, which means that the vectors at least partially capture enzyme primary function.

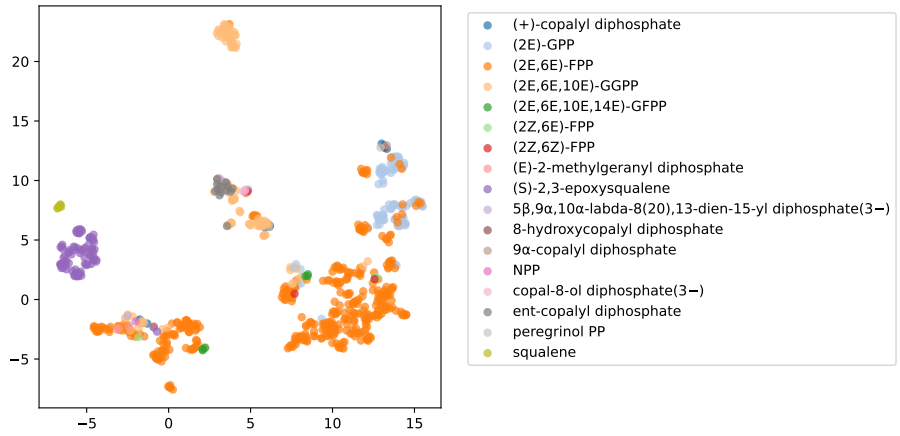


Figure 4.6: Two-dimensional UMAP on ESM-1b Transformer TPSs embeddings. Single colors correspond to different substrates (extremely rare substrates were excluded).

Terpene type classification

Since I focus on the prediction of sesquiterpenes it is important to test whether embeddings can be classified by a product terpene type (monoterpene, diterpene, sesquiterpene, etc.). With an accurate classification by type and successful sesquiterpene synthesis prediction, I could apply the same approach for each terpene type independently. Therefore, I trained a random forest classifier with 30 trees and entropy as a splitting decision function to classify the synthases by the product type. The model achieved a mean accuracy score of 0.95 for both validation and test folds (rare types were excluded, see Figure 3.3a), which means that the **sesquiterpene biosynthesis prediction can be perceived as a proof of concept that any terpene biosynthesis can be predicted**.

Prediction of the evaluation metric fingerprint

Before solving the primary task, I tested whether obtained TPSs embeddings contain information relevant to the primary task. For this purpose, I trained relatively simple machine learning models to directly predict fingerprints selected for the evaluation metric, which is a simplified primary task requiring the same properties of enzymes encoded in the embeddings. More precisely, the task is to predict a 1,800 bits-long bit vector based on a continuous vector of a length 1,280. I examined logistic and linear regressors, trained to predict each bit separately, and Multilayer Perceptron (MLP) with a single hidden layer of 1,500 nodes with ReLU activations. As a loss function for the MLP, I chose a binary cross-entropy (a special case of Equation 4.5 for the binary random variable), as it is a natural way to penalize wrong classification predictions in an exponential manner. Since there are only 321 training samples in

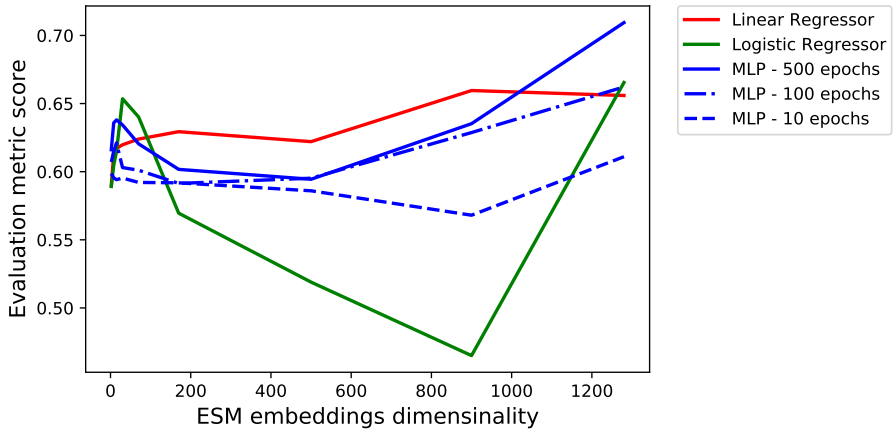


Figure 4.7: Performance of selected models on prediction of evaluation metric fingerprint from the ESM-1b Transformer TPSs embeddings as a function of embeddings dimensionality. A UMAP algorithm was applied to reduce the dimensionalities to 2, 8, 15, 30, 70, 170, 500 and 900; the dimensionality of 1,280 corresponds to the original embeddings' size.

the TPS database, which points to the presence of the *curse of dimensionality*, I additionally applied UMAP to the input vectors and examined the performance of the models having different dimensionalities of UMAP embeddings on the input. Figure 4.7 shows evaluation metric scores of selected models as a function of the embeddings reduced dimensionality. MLP trained on 500 epochs achieved the best score of 0.71, which means that embeddings indeed encode information about terpene synthesis. Logistic regressor reached similar scores on 30-dimensional reduced vectors and the original 1,280-dimensional embeddings, however, it is not true for other models. Consequently, original embeddings are the best fit for the primary task. Remarkably, the linear regressor achieved high scores, although it is not a model specific for the binary classification. Further, MLP performance grew with the number of training epochs. Even though there are only 321 training samples, the model does not tend to overfit. It points to the high information content encoded in the embeddings. Conclusively, results prove that the **TPSs embeddings obtained from the ESM-1b Transformer are appropriate for sesquiterpene prediction**, thus I employ them in further experiments.

4.3.2 Chemical VAE

To be able to encode small molecules to continuous vectors and then decode them back to the original form I employ Chemical VAE used by Gomez-Bombarelli et al. [24] for the automatic chemical design. It was pre-trained on 250,000 synthetic molecules from the ZINC database and showed impressive results in the generation of novel chemical structures. The model was

trained to compress SMILES strings to the 196-dimensional latent space and to decompress the vectors back to SMILES. The 1D convolution layers were chosen for the encoder and GRU [46] layers for the decoder.

Fine-tuning Chemical VAE

The authors of the Chemical VAE estimate that the model trained on 250,000 compounds from ZINC encodes approximately 7.5 million distinct molecules. However, I observe that the learned subspace corresponding to sesquiterpenes is sparse, due to the model not being able to encode and decode many sesquiterpenes from the TPS dataset. Such observation is normal and expected, since the latent space learned during pre-training captures general properties of various molecules, but not specific for some class of compounds. Since the aim is to precisely predict sesquiterpenes I fine-tune the VAE by additionally training it on sesquiterpenes and compounds similar to them. I examine fine-tuning on the full SesqSim, SmallSesqSim and SesqSim13 datasets with the same parameters as during the pre-training, but on a lower number (1 to 12) of epoch and at a 10 times lower learning rate.

To test whether fine-tuned model captures structural properties of the target molecules and that fine-tuning process has not annihilated pre-trained weights, I encode all the SesqSim compounds, reduce their dimensionality from 196 to 2 by the UMAP and visualize the obtained space (Figure 4.9). Plots show that the learned space captures such complex structural features as the size of the smallest set of smallest rings (SSSR) and the fraction of double bonds with respect to all bonds. To additionally ensure that the space is dense enough to capture small changes, I sample random vectors from the neighborhood of the *(+)-(R)-germacrene A* embedding and decode them to molecules. Figure 4.8 satisfies expectations that the sampled molecules do indeed differ in small changes. Also, one molecule contains a nitrogen atom, which is a result of a large quantity of compounds containing nitrogens in ZINC.

Decoding latent vectors

Since SMILES strings are sensitive to small changes and VAEs are probabilistic, there are typically multiple attempts needed to decode a valid molecule. Also, the learned latent space is not perfectly continuous and the final predicted vectors may be imperfect. Therefore, to decode a latent vector I sample 500 vectors from its neighborhood with the Euclidean distances from the original vector given by a normal distribution and decode all of them. Then I select a successfully decoded SMILES string corresponding to the vector with the lowest distance as the final decoding.

Since $C_{15}H_{24}$, $C_{15}H_{26}O$ and $C_{15}H_{28}O_2$ are the only three possible chem-

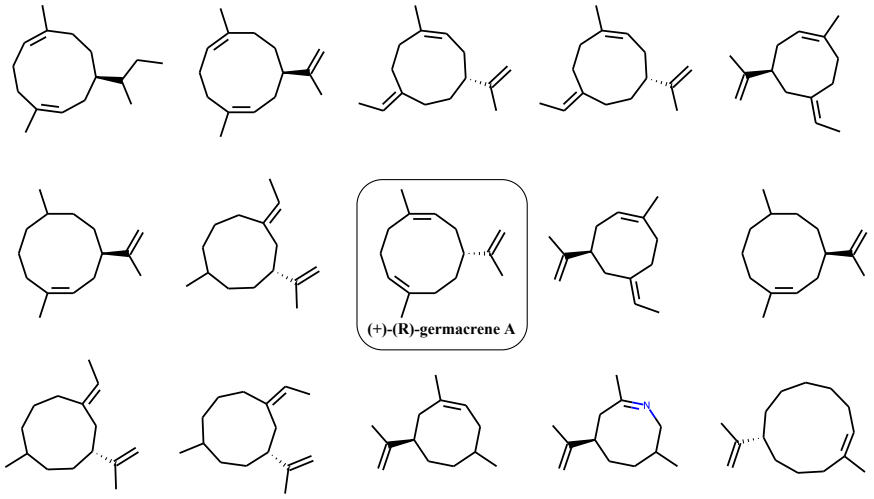
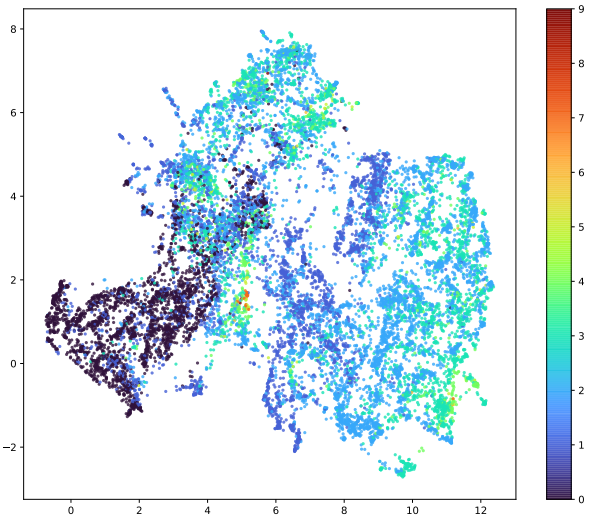


Figure 4.8: Molecules randomly sampled from the neighbourhood of *(+)-(R)-germacrene A* encoding in the latent space of fine-tuned Chemical VAE.

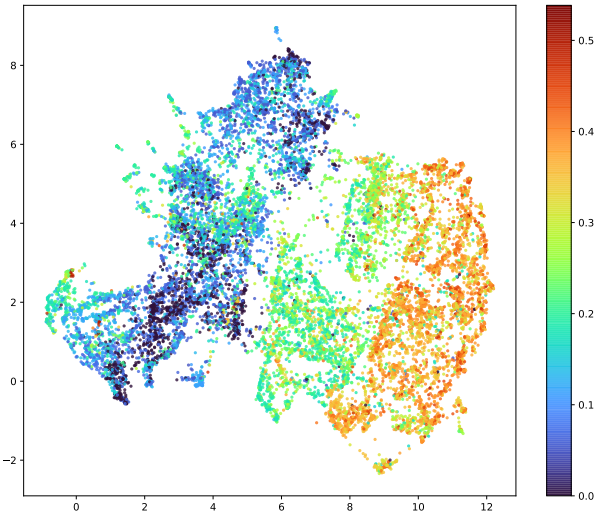
ical formulas of TPS database sesquiterpenes, I apply an additional post-processing step on the SMILES string after decoding. I substitute all heavy atoms to carbons with an exception of a maximum of two oxygens allowed.

4.3.3 Multilayer perceptron

After obtaining the synthases embeddings and a latent space for sesquiterpenes, there is a need to find a mapping between two spaces. For this purpose I employed a Multilayer perceptron, as it is designed for the approximation of continuous functions. I experimented with different hyperparameters and loss functions to find the most appropriate model. I tested MLP with one hidden layer of 725 nodes, two hidden layers of 900 and 500 nodes, three hidden layers of 1009, 738, 467 and, finally, four hidden layers of 1009, 738, 738, 467 with ReLU activation functions. I experimented with training the models on a different number of training epochs from 10 to 3,000. I used Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and the learning rate equal to 0.01 for the training. For the loss functions I used either Mean squared error (MSE) or Mean absolute error (MAE). Additoinally, I examined the linear regressor, as it showed high performance on directly predicting evaluation metric fingerprints. Finally, to have a baseline model I performed a random sampling from the latent space of Chemical VAE, which consists of obtaining random vectors by sampling each element from the $\mathcal{N}(0, 2)$ distribution.



(a)



(b)

Figure 4.9: UMAP of embeddings of all SesqSim molecules obtained from the fine-tuned Chemical VAE. Colors represent selected structural properties: **(a)** size of the smallest set of smallest rings (SSSR), **(b)** fraction of double bonds with respect to all bonds.

Chapter 5

Results

In this chapter, I first summarize quantitative results of the sesquiterpene biosynthesis prediction obtained on validation folds and choose a model leading to the best scores. Second, I analyze the performance of the model on the test fold by interpreting particular predictions.

Quantitative analysis of the results

For the prediction of sesquiterpene biosynthesis, I examined different models mapping ESM-1b synthases' embeddings to the latent space of fine-tuned Chemical VAE and measured their scores on validation folds. For each model, I computed the mean evaluation metric score (*Score*), which is considered as the main value expressing the performance level of the model. Using the same score, while considering only positive values (*Positive score*), can assess the quality of only valid decoded molecules. Additionally, I counted the number of perfect predictions (evaluation metric score equals to 1.0; *#1*) and the number of invalid predictions that cannot be decoded from the latent space of VAE (evaluation metric score equals to 0.0; *#0*).

Table 5.1 summarizes scores of selected tested models. The linear regressor trained only on the ZINC database without further fine-tuning (*b*) achieved approximately the same mean score as the linear regressors additionally trained on two epochs utilizing SesqSim and SmallSesqSim datasets (*c* and *d*). However, the fine-tuning increased *#1* scores, which proves that the latent space of fine-tuned Chemical VAE is more appropriate for the prediction of sesquiterpenes. It can be seen that the fine-tuning on SesqSim13 (*f*) leads to significantly better results comparing to SesqSim and SmallSesqSim (especially for the *Score* and *#0*). In addition, scores of linear regressors slightly improve with the number of fine-tuning epochs (*e* to *i*); however, it is not true for the Multilayer perceptron with a single hidden layer trained

Model	VAE fine-tuning		Score	Positive score	#1	#0
	Dataset	# epochs				
a. Random sampling	–	–	0.08 ± 0.04	0.25 ± 0.08	*0*	*207*
b. Linear regressor	–	–	0.35 ± 0.09	0.50 ± 0.05	3	91
c. Linear regressor	SmallSesqSim	2	0.34 ± 0.10	0.52 ± 0.06	13	105
d. Linear regressor	SesqSim	2	0.37 ± 0.11	0.54 ± 0.06	22	101
e. Linear regressor	SesqSim13	1	0.38 ± 0.09	0.50 ± 0.06	10	74
f. Linear regressor	SesqSim13	2	0.44 ± 0.11	0.55 ± 0.07	29	65
g. Linear regressor	SesqSim13	5	0.44 ± 0.11	0.54 ± 0.08	33	59
h. Linear regressor	SesqSim13	8	0.46 ± 0.12	0.60 ± 0.08	40	70
i. Linear regressor	SesqSim13	12	0.47 ± 0.12	0.59 ± 0.08	43	64
j. MLP 1L 100E	SesqSim13	2	0.45 ± 0.09	0.52 ± 0.07	34	41
k. MLP 1L 2000E	SesqSim	2	0.41 ± 0.09	0.51 ± 0.06	21	65
l. MLP 1L 2000E	SesqSim13	2	0.45 ± 0.09	0.53 ± 0.07	30	46
m. MLP 1L 3000E	SesqSim13	2	0.46 ± 0.10	0.55 ± 0.07	33	53
n. MLP 1L 1000E	SesqSim13	5	0.46 ± 0.10	0.53 ± 0.07	43	41
o. MLP 1L 2000E	SesqSim13	5	0.51 ± 0.09	0.55 ± 0.08	41	21
p. MLP 1L 3000E	SesqSim13	5	0.50 ± 0.11	0.56 ± 0.08	44	36
q. MLP 1L 4000E	SesqSim13	5	0.47 ± 0.11	0.56 ± 0.08	42	52
r. MLP 1L 2000E	SesqSim13	12	0.48 ± 0.10	0.55 ± 0.08	48	39
s. MLP 2L 2000E	SesqSim13	5	0.53 ± 0.10	0.56 ± 0.09	59	17
t. MLP 3L 1500E	SesqSim13	5	0.53 ± 0.10	0.54 ± 0.1	62	11
u. **MLP 3L 2000E**	**SesqSim13**	**5**	$\mathbf{0.56 \pm 0.10}$	$\mathbf{0.57 \pm 0.10}$	**73**	**6**
v. MLP 3L 2500E	SesqSim13	5	0.53 ± 0.11	0.55 ± 0.10	77	8
w. MLP 4L 2000E	SesqSim13	5	0.54 ± 0.10	0.55 ± 0.10	75	6

Table 5.1: Examined models validation scores. *Score* stands for the mean evaluation metric score across validation folds with corresponding variances, and *Positive score* – for mean positive evaluation metric score with variances. *# 1* and *# 0* represent numbers of perfect predictions (evaluation metric score equals to 1) and not decoded predicted latent vectors (evaluation metric score equals to 0) respectively (from the total number of 315 samples). *MLP 3L 2000E* (*u*) achieved the highest *Score* and is selected as a final model (*MLP* stands for the Multilayer perceptron, *3L* – 3 hidden layers and *2000E* – 2000 training epochs). Notice that random baseline scores are higher than expected due to the specificity of the selected evaluation metric.

for 2,000 epochs (*l*, *o*, *r*). Table 5.1 shows that the fine-tuning on five epochs (*o*) leads to a much better relation between *#0* and *#1* scores. Therefore, I trained MLPs with more layers – consuming more time for training – with the SesqSim13 dataset and 5 training epochs as a fine-tuning setup. Conclusively, the Multilayer perceptron with 3 hidden layers (*u*) outperformed other tested models in the *Score* and is remarkably better when comparing all four scores together. Due to this increase in score, I selected it as the final model for the prediction of sesquiterpene biosynthesis. **The chosen MLP achieved a *Score* of 0.55 on the so far hidden test fold and perfectly predicted 11 molecules out of 34 in total**.

Detailed analysis of the predictions

Figure 5.1, Figure 5.2, and Figure 5.3 show skeletal formulas of predicted and real molecules for the whole test fold. It can be seen that the majority of predicted compounds preserve expected structural features. Molecules *1* to *11* have evaluation metric scores equal to 1.0. Despite their chemical equivalence, some of the depictions are not identical, which is a result of possible skeletal formula invariants of the same compound. For example, the predicted molecule *4* and the corresponding real molecule are two opposite projections of the same spatial conformation. Remarkably, the majority of other pairs possess equal structures or substructures and only three predictions were not decoded from the latent space of VAE (*32*, *33* and *34*). For instance, molecules *17* and *25* differ from the actual molecule only by bond types. Further, such predictions as *13*, *18* and *19* are remarkably interesting, since despite their incorrectness they preserve structures and features of real molecules: *13* – two rings with two agreeing stereochemical bonds to two hydrogens, *18* – all fragments branch from two rings with similar stereochemistry, *19* – a large ring with three branching moieties with similar structure. Ultimitally, it means that the employment of Variational Autoencoder as a model producing space for predictions is appropriate and is very interesting for future and deeper investigation.

The most valuable observation is that from the total 34 predicted molecules, 14 satisfy the chemical formula of the sesquiterpene $C_{15}H_{24}$. Of those 14: 10 are perfect predictions (*1* to *8*, *10* and *11*), 2 have identical structures (*17* and *25*) but differ in bond types, and the remaining 2 preserve global features (*16* and *20*). Since the analogous observation is true for the validation fold, it is logical to generalize it into the following statement: **a predicted molecule satisfying the chemical formula $C_{15}H_{24}$ is most likely a perfect prediction**. Therefore, a domain knowledge about the sesquiterpene chemical formula can be used as a prediction confidence score that allows determining perfect predictions without any information about expected ones. Together with the observation that the fraction of perfect predictions on the validation folds constitutes 23% and 32% on the test fold, there can be derived an even stronger statement: **each fifth prediction is most likely perfect and can be determined without the knowledge of a real compound**. It implies that the proposed approach allows prediction of approximately 20% of uncharacterized sesquiterpene syntheses and the majority of the remaining 80% may be relevant as well.

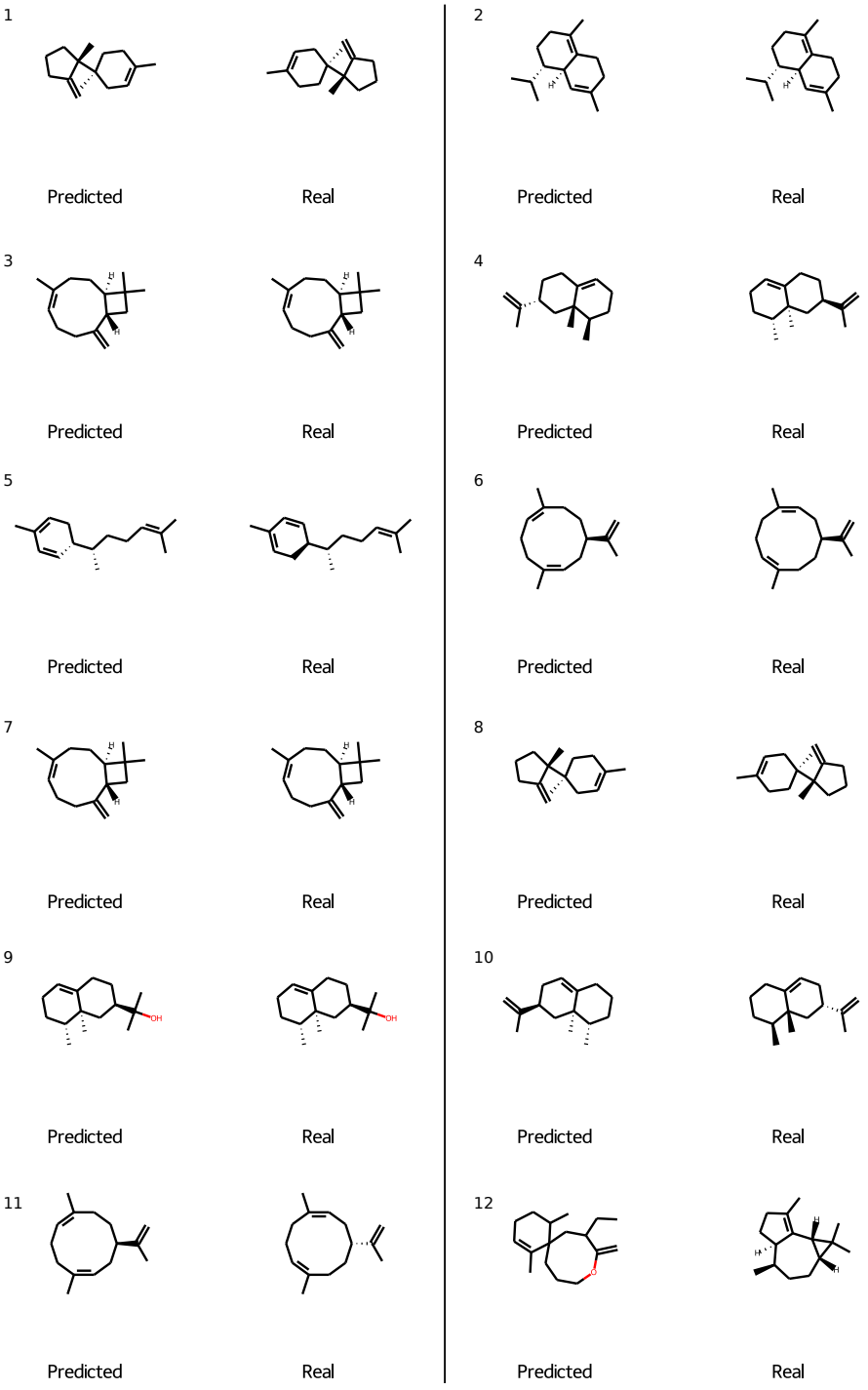


Figure 5.1: Test fold predictions. Part 1/3. Although some real products are the same (for example *3* and *7*), corresponding biosyntheses are catalyzed by different enzymes.

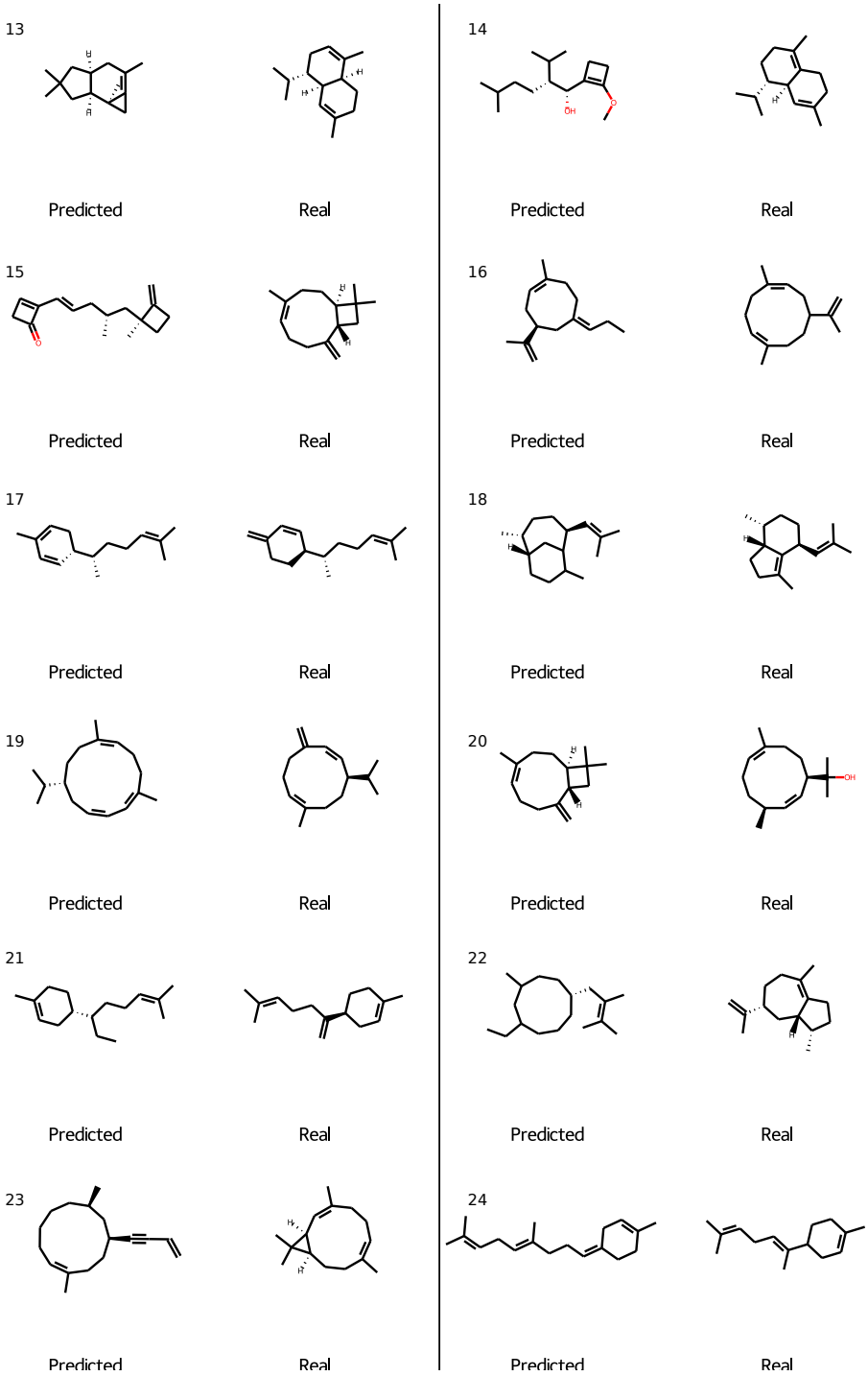


Figure 5.2: Test fold predictions. Part 2/3.

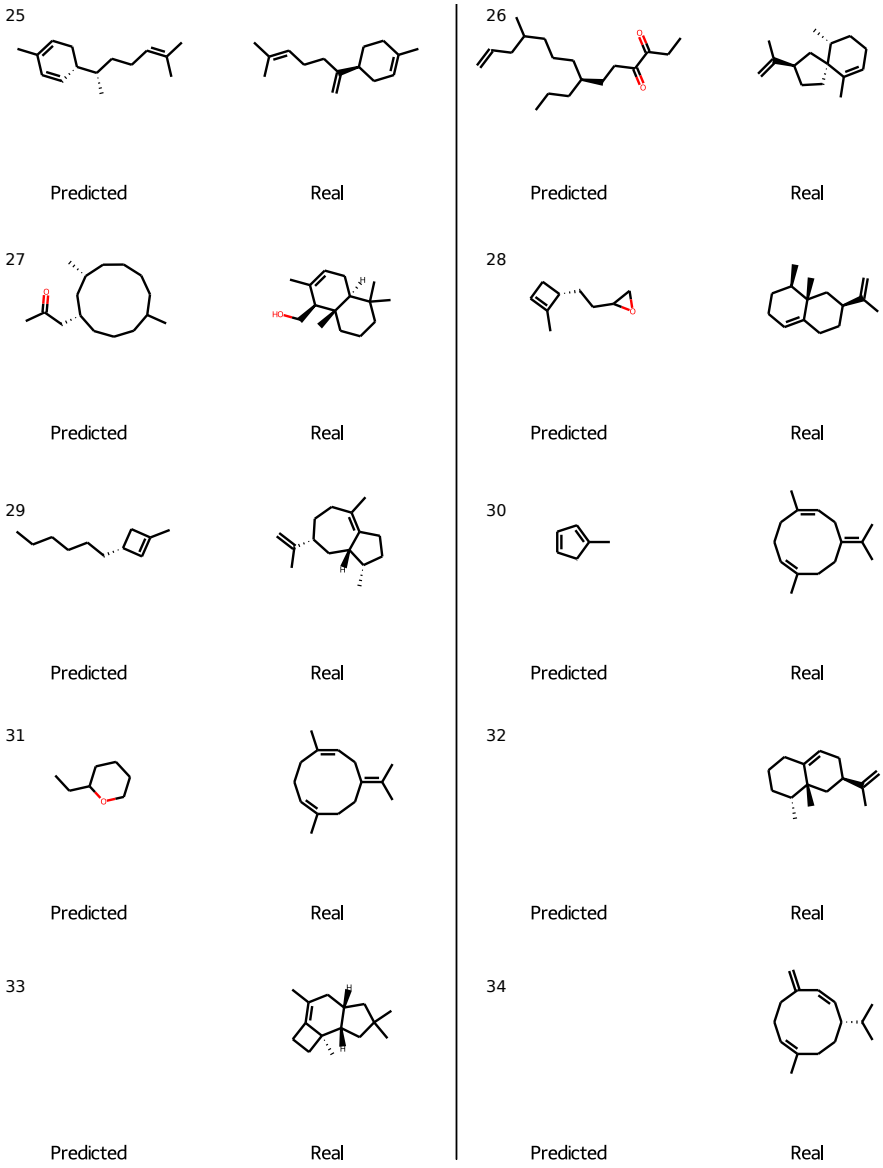


Figure 5.3: Test fold predictions. Part 3/3. Predictions *32*, *33* and *34* are not decodable by VAE.

Chapter 6

Future work

Terpene biosynthesis prediction has not been studied before and the amount of time allocated for the work on bachelor thesis is highly limited. Regardless of time constraint, I have unfolded many improvements of the introduced approach. Thus, in the following sections I will describe potential future work on the problem.

Improved datasets

The number of characterized terpene syntheses grow over time, resulting in an increase of the size of the TPS database. SesqSim dataset can be significantly improved by manually selecting the most similar sesquiterpenes groups of compounds in public databases or leveraging such tools as MOLGEN, which would allow obtaining millions of desired isomers. I can not only collect a dataset for fine-tuning but also build a huge database for the primary training purpose. For example a vast database of hydrocarbons may be much more appropriate than a synthetic ZINC database of arbitrary molecules. Further, I can collect a large database of tens of thouthands terpene synthases and similar enzymes, which would allow me to fine-tune a Transformer for my purpose instead of just using a pre-trained model.

Upgraded machine learning models

The obvious disadvantage of the current implementation is the fact that the VAE operates on SMILES strings. These strings are extremely sensitive to any changes. Typically, substitution or deletion of an arbitrary character leads to an invalid molecule, when combined with the probabilistic essence of VAEs, this becomes a serious complication for this type of model. Generally, representing small molecules as graphs instead of strings encoding their structures is a more natural way and has obvious advantages. Neural networks do not

have to learn the fragile syntax during the training, but can directly manipulate graphs. Graph neural networks (GNN) have been actively studied and developed in recent years. In the work of Gilmer et al. [47] authors propose a Message-passing neural network (MPNN) framework, which was derived as a generalization of efficient neural networks on graphs and is beneficial in the application of molecules. This approach is based on sending messages between atoms within bonds, resulting in the model directly learning the molecular structure. Later, this approach evolved into other novel variants, such as for example Directed MPNN (D-MPNN) [48]. These evolved models are even more convenient for learning structures of molecules. I can therefore build a Variational autoencoder on graphs inspired by the work of Kipf et al. [49], but with a D-MPNN encoder. I believe that this upgrade will significantly improve the results of my work. Also, I can experiment with Generative adversarial networks (GANs) [50] for molecule generation. This was already done by Prykhodko et al. [51], where they used GAN to operate on a latent space of VAE.

Generalization of the proposed approach

Considering the fact that the current result was achieved without leveraging guaranteed improvements, confident prediction of sesquiterpenes is only a matter of time. The next step is the prediction of any terpene biosynthesis. While I pointed to the generalization of other terpene types by creating independent models analogous to the one described in the thesis, a much more interesting approach is one leveraging the power of the proposed pipeline of machine learning models. The essence of the approach is a transformation of enzymes and small molecules to continuous vectors, which allows expressing biochemistry in terms of vector spaces and operations over them. At its core biosynthesis is a transformation of a substrate molecule to a product molecule catalyzed by an enzyme. If we denote x and y as vectors from the same space assigned to a substrate and a product respectively and a – as an embedding of an enzyme, then the prediction of biosynthesis is equivalent to the approximation of y with a neural network $f(x, a)$. So the general form of a loss function can be represented as

$$\mathcal{L}(x, a, y) = \|f(x, a) - y\| \tag{6.1}$$

From this perspective, the prediction of sesquiterpene biosynthesis discussed in the thesis is a special case of a more general approach (Figure 6.1) because in the case of sesquiterpenes the term x is omitted. Term x is omitted since it is constant for the prevailing majority of sesquiterpene biosyntheses and does not affect learning optimization problem. The fact that x, y, and $f(x, a)$ are vectors belonging to the same vector space plays an important role and provides a great advantage for the neural network. For example, vectors assigned

to substrate *geranyl pyrophosphate* and product *limonene* will have partially similar encodings. During the training neural network will capture the fact that the molecules differ in the presence/absence of a *phosphate* (which is typically true for all terpene syntheses) and several other bond types. It implies that the model will learn the transformation of a molecule in a natural way by employing an enzyme embedding as it is designed by nature. I believe that the proposed general solution has far reaching potential, as it is very robust and allows utilizing various biochemistry domain knowledge. For example, enzyme cofactors can be additionally encoded to the before-mentioned vector space or multiple synthesis products can be located in its pre-defined subspace. Moreover, this approach is scalable regarding different biosyntheses and diverse training data might remarkably increase the overall performance due to its general expressivity. It means that **the neural network could learn a function directly displaying enzyme activity**.

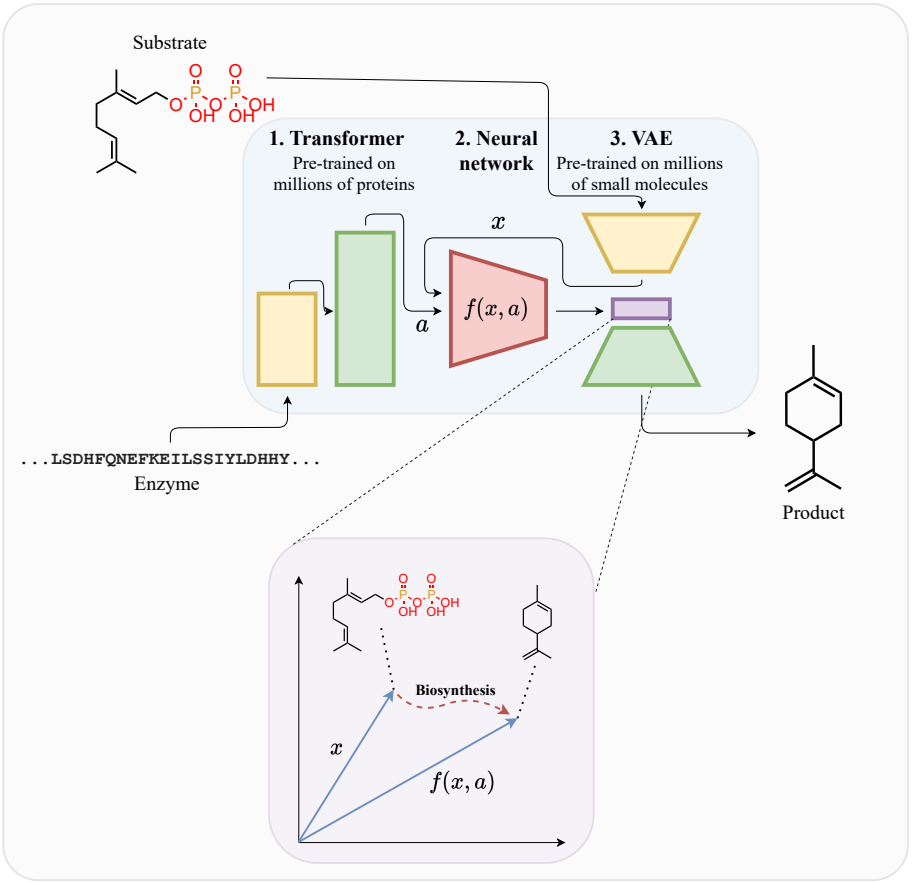


Figure 6.1: General proposed approach for the biosynthesis prediction. First, Transformer (**1**) is pre-trained on millions of proteins, and Variational autoencoder (**2**) is pre-trained on millions of small molecules. In order to predict a biosynthesis, a substrate is encoded by VAE, and an enzyme is encoded by Transformer. Then the Multilayer perceptron (**3**) is trained to predict a product in a latent space of the VAE based on obtained encodings. Finally, a predicted molecule is obtained by decoding the predicted latent vector.

Conclusions

In the present work, I studied the capabilities of machine learning regarding the prediction of sesquiterpene biosynthesis. I proposed a pipeline of machine learning models consisting of Transformer, Variational Autoencoder, and Multilayer perceptron. A pre-trained Transformer was used to obtain embeddings of sesquiterpene synthases and a pre-trained VAE to obtain a continuous space encoding structural properties of small molecules. Multilayer perceptron was used to predict a sesquiterpene vector in a latent space of VAE, which could be subsequently decoded into a SMILES string of a molecule. Despite having an extremely low amount of training data, the approach showed its efficiency in decoding biosynthesis reactions encoded in sesquiterpene synthase sequence. The best model achieved a Tanimoto similarity score of 0.55 on sesquiterpene-sensitive fingerprints for the test fold. It managed to perfectly predict 11 sesquiterpenes out of 34, while the majority of other predicted molecules preserve structural features of actual compounds.

I showed that the domain knowledge about the chemical formula of sesquiterpene can be used as a confidence score of the prediction, which implies that the model can be applied to the prediction of unstudied sesquiterpene syntheses. More precisely, on average, the model can perfectly predict one of five uncharacterized sesquiterpene syntheses with a high level of confidence. Although the mean numerical scores are not perfect, the features of similarity between predicted and real compounds are fairly impressive. Typically, the model properly predicts the general structural properties of molecules, which means that it can be employed as an assistant for biochemistry experts in practical applications.

I have studied a novel field of research and I believe that my thesis is a pioneering contribution to further work on the prediction of biosyntheses. The proposed pipeline constitutes a framework that allows operating on vector spaces instead of row enzyme sequences and small molecules, which makes it especially interesting for the generalization to other biosyntheses – which I have pointed to – and future work in general.

Bibliography

[1] S.M.K Rates. Plants as source of drugs. *Toxicon*, 39(5):603–613, 2001. ISSN 0041-0101. doi: https://doi.org/10.1016/S0041-0101(00)00154-9. URL https://www.sciencedirect.com/science/article/pii/S0041010100001549.

[2] Shagufta Perveen. *Introductory Chapter: Terpenes and Terpenoids*. 12 2018. ISBN 978-1-78984-776-5. doi: 10.5772/intechopen.79683.

[3] The pandas development team. pandas-dev/pandas: Pandas, February 2020. URL https://doi.org/10.5281/zenodo.3509134.

[4] 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, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[5] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

[6] François Chollet et al. Keras. https://keras.io, 2015.

[7] Greg Landrum. Rdkit: Open-source cheminformatics software. 2016. URL https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_4.

[8] Leonard Richardson. Beautiful soup documentation. *April*, 2007.

[9] Ethan B Russo. Taming thc: potential cannabis synergy and phytocannabinoid-terpenoid entourage effects. *British Journal of Pharmacology*, 163(7):1344–1364, 2011. doi: https://doi.org/10.1111/j.1476-5381.2011.01238.x. URL https://bpspubs.onlinelibrary.wiley.com/doi/abs/10.1111/j.1476-5381.2011.01238.x.

[10] The UniProt Consortium. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489, 11 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa1100. URL https://doi.org/10.1093/nar/gkaa1100.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[12] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL http://arxiv.org/abs/1301.3781.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385?hl=zh-cn.

[14] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

[15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014. URL https://arxiv.org/abs/1312.6114.

[17] PN Campbell. Biochemistry and molecular biology. *Biochemical Education*, 20(3):158–165, 1992. ISSN 0307-4412. doi: https://doi.org/10.1016/0307-4412(92)90061-P. URL https://www.sciencedirect.com/science/article/pii/030744129290061P.

[18] Nils Strodthoff, Patrick Wagner, Markus Wenzel, and Wojciech Samek. UDSMProt: universal deep sequence models for protein classification. *Bioinformatics*, 36(8):2401–2409, 01 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa003. URL https://doi.org/10.1093/bioinformatics/btaa003.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

[20] The Gene Ontology Consortium. The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Research*, 47(D1):D330–D338, 11 2018. ISSN 0305-1048. doi: 10.1093/nar/gky1055. URL https://doi.org/10.1093/nar/gky1055.

[21] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), 2021. ISSN 0027-8424. doi: 10.1073/pnas.2016239118. URL https://www.pnas.org/content/118/15/e2016239118.

[22] Yuting Xu, Deeptak Verma, Robert P. Sheridan, Andy Liaw, Junshui Ma, Nicholas M. Marshall, John McIntosh, Edward C. Sherer, Vladimir Svetnik, and Jennifer M. Johnston. Deep dive into machine learning models for protein engineering. *Journal of Chemical Information and Modeling*, 60(6):2773–2790, 2020. doi: 10.1021/acs.jcim.0c00073. URL https://doi.org/10.1021/acs.jcim.0c00073. PMID: 32250622.

[23] Jonathan M. Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M. Donghia, Craig R. MacNair, Shawn French, Lindsey A. Carfrae, Zohar Bloom-Ackermann, Victoria M. Tran, Anush Chiappino-Pepe, Ahmed H. Badran, Ian W. Andrews, Emma J. Chory, George M. Church, Eric D. Brown, Tommi S. Jaakkola, Regina Barzilay, and James J. Collins. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702.e13, 2020. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2020.01.021. URL https://www.sciencedirect.com/science/article/pii/S0092867420301021.

[24] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, Jan 2018. ISSN 2374-7951. doi: 10.1021/acscentsci.7b00572. URL http://dx.doi.org/10.1021/acscentsci.7b00572.

[25] Teague Sterling and John J. Irwin. Zinc 15 – ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 55(11):2324–2337, 2015. doi: 10.1021/acs.jcim.5b00559. URL https://doi.org/10.1021/acs.jcim.5b00559. PMID: 26479676.

[26] Soumitra Samanta, Steve O'Hagan, Neil Swainston, Timothy J. Roberts, and Douglas B. Kell. Vae-sim: A novel molecular similarity measure

based on a variational autoencoder. *Molecules*, 25(15), 2020. ISSN 1420-3049. doi: 10.3390/molecules25153446. URL https://www.mdpi.com/1420-3049/25/15/3446.

[27] Ryohei Eguchi, Naoaki Ono, Aki Hirai Morita, Tetsuo Katsuragi, Satoshi Nakamura, Ming Huang, Md. Altaf-Ul-Amin, and Shigehiko Kanaya. Classification of alkaloids according to the starting substances of their biosynthetic pathways using graph convolutional neural networks. *BMC Bioinformatics*, 20(1):380, Jul 2019. ISSN 1471-2105. doi: 10.1186/s12859-019-2963-6. URL https://doi.org/10.1186/s12859-019-2963-6.

[28] E. Michael Thurman. Chapter seven - analysis of terpenes in hemp (cannabis sativa) by gas chromatography/mass spectrometry: Isomer identification analysis. In Imma Ferrer and E. Michael Thurman, editors, *Analysis of Cannabis*, volume 90 of *Comprehensive Analytical Chemistry*, pages 197–233. Elsevier, 2020. doi: https://doi.org/10.1016/bs.coac.2020.04.013. URL https://www.sciencedirect.com/science/article/pii/S0166526X20300374.

[29] F. Loreto, A. Förster, M. Dürr, O. Csiky, and G. Seufert. On the monoterpene emission under heat stress and on the increased thermotolerance of leaves of quercus ilex l. fumigated with selected monoterpenes. *Plant, Cell & Environment*, 21(1):101–107, 2002. doi: https://doi.org/10.1046/j.1365-3040.1998.00268.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1046/j.1365-3040.1998.00268.x.

[30] Martin Chadwick, Harriet Trewin, Frances Gawthrop, and Carol Wagstaff. Sesquiterpenoids lactones: Benefits to plants and people. *International Journal of Molecular Sciences*, 14(6):12780–12805, 2013. ISSN 1422-0067. doi: 10.3390/ijms140612780. URL https://www.mdpi.com/1422-0067/14/6/12780.

[31] A. Vasas and J. Hohmann. Euphorbia diterpenes: isolation, structure, biological activity, and synthesis (2008-2012). *Chem Rev*, 114(17):8579–8612, Sep 2014.

[32] Yong Zhang, Peixin Jiang, Min Ye, Sung-Hoon Kim, Cheng Jiang, and Junxuan Lü. Tanshinones: Sources, pharmacokinetics and anti-cancer activities. *International Journal of Molecular Sciences*, 13(10):13621–13666, 2012. ISSN 1422-0067. doi: 10.3390/ijms131013621. URL https://www.mdpi.com/1422-0067/13/10/13621.

[33] Jacinda T. James and Ian A. Dubery. Pentacyclic triterpenoids from the medicinal herb, centella asiatica (l.) urban. *Molecules*, 14(10):3922–3941, 2009. ISSN 1420-3049. doi: 10.3390/molecules14103922. URL https://www.mdpi.com/1420-3049/14/10/3922.

[34] Destinney Cox-Georgian, Niveditha Ramadoss, Chathu Dona, and Chhandak Basu. Therapeutic and medicinal uses of terpenes. *Medicinal Plants: From Farm to Pharmacy*, pages 333–359, Nov 2019. doi: 10.1007/978-3-030-31269-5_15. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7120914/. PMC7120914[pmcid].

[35] Masaki Himejima, Kenneth Hobson, Toshikr Otsuka, David Wood, and Isao Kubo. Antimicrobial terpenes from oleoresin of ponderosa pine tree pinus ponderosa– a defense mechanism against microbial invasion. *Journal of chemical ecology*, 18:1809–1818, 10 1992. doi: 10.1007/BF02751105.

[36] Adriana Pliego Zamora, Judith H. Edmonds, Maxwell J. Reynolds, Alexander A. Khromykh, and Stephen J. Ralph. The in vitro and in vivo antiviral properties of combined monoterpene alcohols against west nile virus infection. *Virology*, 495:18–32, 2016. ISSN 0042-6822. doi: https://doi.org/10.1016/j.virol.2016.04.021. URL https://www.sciencedirect.com/science/article/pii/S0042682216300794.

[37] Ming Chen, Li-Feng Chen, Man-Mei Li, Ni-Ping Li, Jia-Qing Cao, Ying Wang, Yao-Lan Li, Lei Wang, and Wen-Cai Ye. Myrtucomvalones a–c, three unusual triketone–sesquiterpene adducts from the leaves of myrtus communis 'variegata'. *RSC Adv.*, 7:22735–22740, 2017. doi: 10.1039/C7RA02260C. URL http://dx.doi.org/10.1039/C7RA02260C.

[38] K. Schriever, P. Saenz-Mendez, R. S. Rudraraju, N. M. Hendrikse, E. P. Hudson, A. Biundo, R. Schnell, and P. O. Syrén. Engineering of Ancestors as a Tool to Elucidate Structure, Mechanism, and Specificity of Extant Terpene Cyclase. *J Am Chem Soc*, 143(10):3794–3807, Mar 2021.

[39] Ralf Gugisch, Adalbert Kerber, Axel Kohnert, Reinhard Laue, Markus Meringer, Christoph Rücker, and Alfred Wassermann. Chapter 6 - molgen 5.0, a molecular structure generator. In Subhash C. Basak, Guillermo Restrepo, and José L. Villaveces, editors, *Advances in Mathematical Chemistry and Applications*, pages 113–138. Bentham Science Publishers, 2015. ISBN 978-1-68108-198-4. doi: https://doi.org/10.1016/B978-1-68108-198-4.50006-0. URL https://www.sciencedirect.com/science/article/pii/B9781681081984500060.

[40] Janna Hastings, Gareth Owen, Adriano Dekker, Marcus Ennis, Namrata Kale, Venkatesh Muthukrishnan, Steve Turner, Neil Swainston, Pedro Mendes, and Christoph Steinbeck. Chebi in 2016: Improved services and an expanding collection of metabolites. *Nucleic acids research*, 44(D1): D1214—9, January 2016. ISSN 0305-1048. doi: 10.1093/nar/gkv1031. URL https://europepmc.org/articles/PMC4702775.

[41] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen,

Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E Bolton. PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Research*, 49(D1):D1388–D1395, 11 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa971. URL https://doi.org/10.1093/nar/gkaa971.

[42] Gilles Klopmand. Concepts and applications of molecular similarity, by mark a. johnson and gerald m. maggiora, eds., john wiley & sons, new york, 1990, 393 pp. price: $65.00. *Journal of Computational Chemistry*, 13(4):539–540, 1992. doi: https://doi.org/10.1002/jcc.540130415. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.540130415.

[43] Lagnajit Pattanaik and Connor W. Coley. Molecular representation: Going long on fingerprints. *Chem*, 6(6):1204–1207, 2020. ISSN 2451-9294. doi: https://doi.org/10.1016/j.chempr.2020.05.002. URL https://www.sciencedirect.com/science/article/pii/S2451929420301984.

[44] Myungwon Seo, Hyun Kil Shin, Yoochan Myung, Sungbo Hwang, and Kyoung Tai No. Development of natural compound molecular fingerprint (nc-mfp) with the dictionary of natural products (dnp) for natural product-based drug development. *Journal of Cheminformatics*, 12(1): 6, Jan 2020. ISSN 1758-2946. doi: 10.1186/s13321-020-0410-3. URL https://doi.org/10.1186/s13321-020-0410-3.

[45] Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7(1):20, May 2015. ISSN 1758-2946. doi: 10.1186/s13321-015-0069-3. URL https://doi.org/10.1186/s13321-015-0069-3.

[46] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014. URL https://nyuscholars.nyu.edu/en/publications/empirical-evaluation-of-gated-recurrent-neural-networks-on-sequen.

[47] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1263–1272. JMLR.org, 2017.

[48] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019. doi: 10.1021/acs.jcim.9b00237. URL https://doi.org/10.1021/acs.jcim.9b00237. PMID: 31361484.

[49] Thomas N. Kipf and Max Welling. Variational graph auto-encoders, 2016.

[50] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[51] Oleksii Prykhodko, Simon Viet Johansson, Panagiotis-Christos Kotsias, Josep Arús-Pous, Esben Jannik Bjerrum, Ola Engkvist, and Hongming Chen. A de novo molecular generation method using latent vector based generative adversarial network. *Journal of Cheminformatics*, 11(1):74, Dec 2019. ISSN 1758-2946. doi: 10.1186/s13321-019-0397-9. URL `https://doi.org/10.1186/s13321-019-0397-9`.

Acronyms

MAE Mean absolute error

MLP Multilayer perceptron

MSE Mean squared error

TPS Terpene synthase

UMAP Uniform Manifold Approximation and Projection

VAE Variational Autoencoder

Contents of enclosed CD

`thesis` ..... the directory with contents of enclosed CD
- `experiments` ..... the directory with a source code of experiments
- `data` ..... the directory with datasets and data-related source code
  - `datasets` ..... the directory with datasets
- `tex` ..... the directory of LaTeX source codes of the thesis
- `thesis.pdf` ..... the thesis text in PDF format