

## Aim of this Talk:

Clinical investigations and research in the biomedical field showed that to treat more complex diseases the administration of just one drug is often not enough. Diseases like HIV, cancer, or kidney failure, to name a few, often require a combination of drugs to achieve satisfactory results or improvements in the patients' health. However, the simultaneous use of multiple drugs often leads to the occurrence of drug-drug interactions (DDI).

While DDI can be taken advantage of to increase a therapeutic effect, they are also a mayor cause for unwanted or unexpected adverse side effects in patients. Hence, extensive knowledge about potential drug-drug interactions is crucial when developing adequate treatment plans. The problem is that traditionally determining potential DDI is a time-consuming and costly process which often only happens in the later stages of drug discovery through extensive biological experiments.

To help reduce said costs and to determine DDI as early as possible, computer-based methods started to be used more and more in drug discovery. Amongst them machine learning methods, like Support Vector Machines (SVM), are highly regarded. Those methods commonly make use of the idea that drugs which are similar behave similar.

Thus, in this talk, we will take a closer look at how we can use a SVM to predict DDI for new drug combinations by comparing them to drug combinations with known DDI with the help of similarity matrices.

## Contents in *Theory*:

- Determining 'similarity' between different drugs.
- Explanation of DDI, their different types, and their effects on medical treatment plans.
- Introduction to SVMs for linear and non-linear classification problems and a more detailed explanation of the kernel trick.
- Detailed workflow on how to use a combined similarity matrix as a pairwise kernel function to determine drug combinations that show DDI based on a set of chosen features with the help of an SVM.
- Overview on the DrugBank database.

## Contents in *Practical*:

- Retrieving necessary data from DrugBank.
- Creating a drug-drug 2D molecular structure similarity database using OpenBabel.
- Creating a drug-drug 3D pharmacophoric similarity database using the Schrödinger package.
- Creating a drug-drug interaction profile database.
- Construction of one pairwise similarity matrix for each feature and combining them into one final similarity matrix with labels which will be used as the kernel for the SVM.
- Modelling and evaluating the SVM.

## References:

- **TODO**

## Theory

### Determining 'similarity' between different drugs.

A base assumption in drug discovery is that similar drugs express similar properties and thus behave similar when introduced to a biological system. Computer-based methods use that assumption to cluster drugs or find molecules that have the potential to have an increased therapeutic effect to already known drugs. As such, it is important to clearly define how 'similarity' is to be judged computationally.

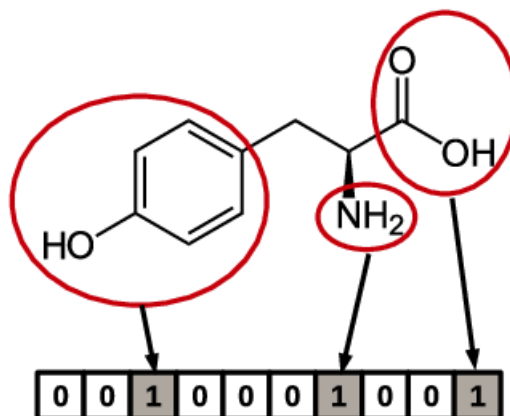
#### Defining 'drug similarity' to a computer

When it comes to comparing different drugs with each other, there are multiple different categories that can be used to define similarity. From comparing their 2D or 3D structures over which pathways are affected to which molecules are targeted by the different drugs, there is a wide range of properties to choose as a means to calculate similarity. Oftentimes, algorithms even use a combination of different properties to improve their results and make better predictions. But how does a computer compare two drugs?

The most common way to do it is by using so-called fingerprints which can then be compared to each other. The Tanimoto coefficient, or Jaccard index, is a widely used method to calculate a similarity score and will also be what we use later during the practical part to build the 2D molecular structure similarity database and the interaction profile database.

#### What is a fingerprint?

A fingerprint is a binary vector whose size equals the number of features you wish to look at for a single property. Each position is then filled with a 1 if the feature related to said position is present in the drug and 0 otherwise. Figure 1 shows an example of a fingerprint for the 2D structure property of Tyrosine.



[FIGURE 1] An example of a 2D structure fingerprint for Tyrosine.

#### How to calculate the Tanimoto coefficient

To calculate the Tanimoto coefficient (Jaccard index) we need two fingerprints of the same property for two different drugs to compare to each other. It is very important that the vectors are the same size and that each cell in those vectors corresponds to the same feature at the same position, because otherwise a proper similarity score cannot be calculated.

Once we have the two vectors, the Tanimoto coefficient is calculated using the following formula:

$$TC(A, B) = |A \cap B| / |A \cup B|$$

As shown the formula divides the intersection of fingerprints A and B with the number of features present in the union of both fingerprints and calculates the 'similarity' as a float value between 0 and 1 with 0 representing no similarity at all and 1 indicating the two drugs are 'identical' in the given property. Figure 2 shows an example for the comparison between [...] and [...] and the resulting Tanimoto coefficient.

[FIGURE 2]

### **Alternative ways to calculate drug similarity**

Naturally there are other ways to determine the similarity of two drugs for a given property, but those are out of the scope of this talktorial. As you will see in the practical part the above-mentioned method will not be used to calculate the 3D structure similarity database. For this property the Schrödinger 2011 package (available at <http://www.schrodinger.com>) will be used. How this package calculates 3D similarity can be read up in the package's documentation and will not be discussed here.

## **Explanation of DDI, their different types, and their effects on medical treatment plans.**

### **What are drug-drug interactions?**

Drug-drug interactions (DDI) occur when two or more drugs are introduced to the same biological system at the same time or during an overlapping period of their effects. They are caused by one drug interfering with another one in various stages and through that influence the effectiveness of said drug. This means they either cause a medical effect that was unexpected or create a measurable difference of the two drugs in the patient's bloodstream to what would be expected if they had been administered one at a time.

Notably, it does not matter if said influence is beneficial or harmful to the biological system to be classified as a DDI.

### **Which types exist?**

Drug-drug interactions are commonly classified by the cause of their occurrence, meaning they are either caused by pharmacokinetic (PK) or pharmacodynamic (PD) interactions.

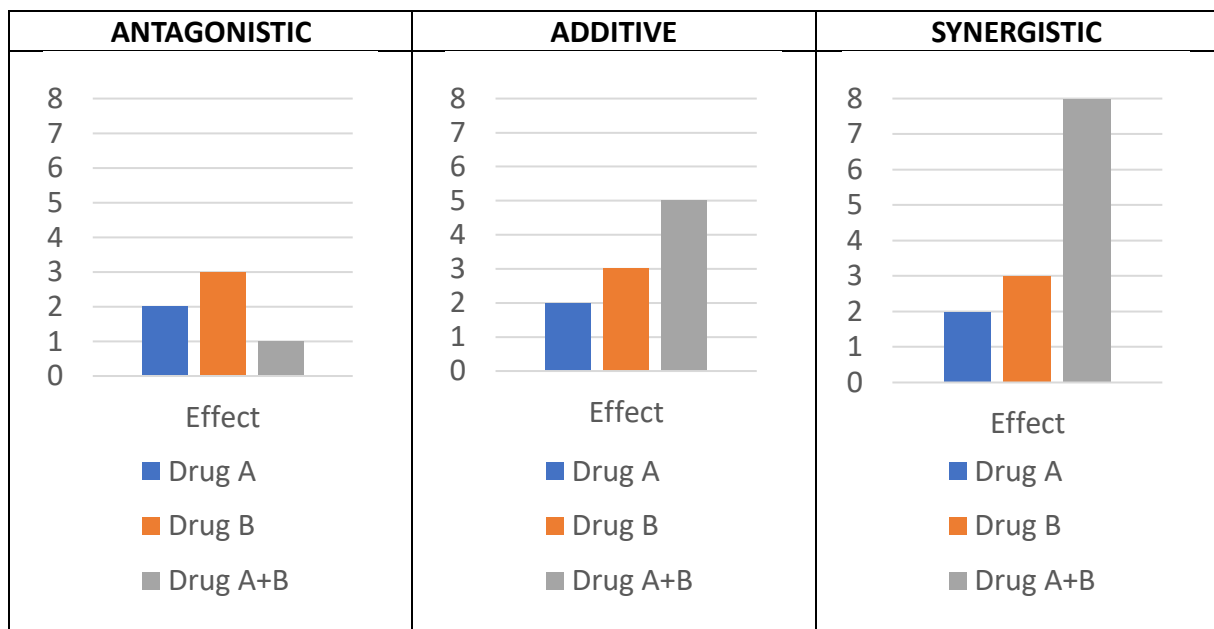
Pharmacokinetic DDI occur when drug A influences drug B's concentration in the blood stream. It does not matter if it's the active component of drug A or one of the additives, added to assist in the delivery of the drug, that is the cause of this interaction.

PK DDI are separated into four categories depending on which stage of a drug's lifetime circle is affected: absorption, distribution, metabolism, or excretion (ADME).

Pharmacodynamic DDI, on the other hand, occur when the pharmacological effect of drug A influences the pharmacological effect of drug B. This happens when A and B target similar or related biological pathways or targets. However, a PD DDI can also occur when the drugs affected pathways seem to be completely unrelated, but their pharmacological effects still cause an unexpected medical observation.

PD DDI get classified into three groups:

- Antagonistic:  
The combined effect caused by a drug combination is **smaller than the sum** of the pharmacological effects seen when each drug is given alone.
- Additive:  
The combined effect caused by a drug combination is **the sum** of the pharmacological effects seen when each drug is given alone.
- Synergistic:  
The combined effect caused by a drug combination is **greater than the sum** of the pharmacological effects seen when each drug is given alone.



[FIGURE 3]: Graphical representation of the three types of PD DDI.

It is important to note that a DDI can be of both types, signifying that this way of classification is to be seen more as a widely used guideline rather than hard-split categories. Similarly, the words antagonistic, additive, synergistic, and their synonyms are sometimes used to also categorize PK DDI, especially in the medical field where the distinction between PK and PD may not be of importance.

IMPORTANT: For the remainder of this talktorial, unless specified otherwise, DDI will not be differentiated into PK or PD. Likewise, the terms antagonistic, additive, and synergistic – if applied – will be used to describe all DDI as smaller, equal, and greater than the sum of their therapeutic effect respectively as to not distract from the actual topic of this talktorial.

### How do they affect medical treatment plans?

From ensuring that the patient doesn't take drugs which are known to negatively affect each other (e.g., causing adverse reactions to the drugs, making one drug unusable for the body, potentiating side effects to a severely harmful degree), over ensuring that existing medical conditions do not worsen, up to taking advantage of synergistic DDI to allow for the administration of smaller drug dosages or give better treatment. There are many ways in which knowledge about DDI can and should be applied by doctors when creating personalized medical treatment plans. As such analysing new drugs or drug combinations for potential DDI or finding synergistic DDI amongst existing drugs is an important aspect in the advancement of personalized medicine.

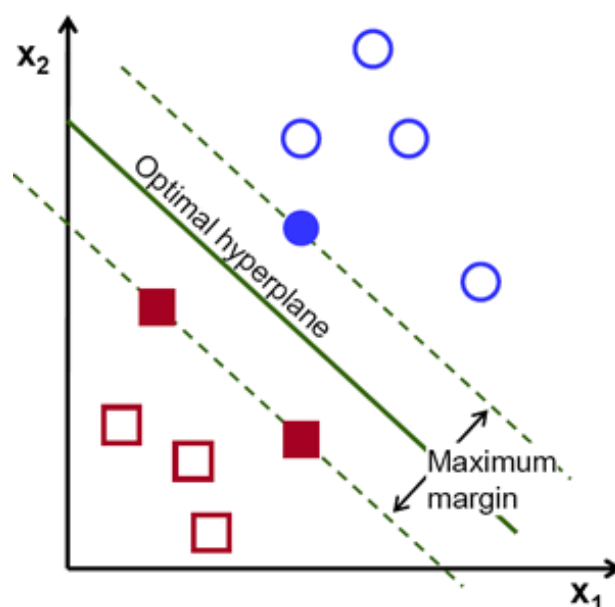
Since finding and verifying DDI is a costly process due to the amount of in vitro and in vivo experiments, computational means started to develop to filter the large number of potential drug combinations for those who are more likely to show advantageous DDI with others or to remove those combinations which have a high risk of harming a potential patient. Amongst these computational methods, many of them employing machine-learning techniques, we will take a closer look at Support Vector Machines in the following chapters.

Introduction to SVMs for classification problems and a more detailed explanation of the kernel trick.

### Basic Principle of a Support Vector Machine

Support Vector Machines (SVM) are supervised learning models, used to analyse data for classification and regression purposes. Initially developed by Vladimir Vapnik and colleagues during the 1990s, SVMs became more and more popular as one of the most robust prediction methods in machine learning.

Using a set of training examples and a binary label system, the SVM training algorithm maps each labelled training example to a point in space and then marks the middle of the gap between the two groups as the optimal hyperplane. The points closest to the hyperplane are called support vectors and an optimal hyperplane maximizes the distance it has from the support vectors on both sides. That distance is called margin. As such SVMs use a maximal margin classifier to determine the allegiance of new data points.



[FIGURE 5] A schematic example of an optimal separating hyperplane.

Considering that the optimal separating hyperplane is chosen to maximize the margin between the two groups of the training data, the placement of the hyperplane gets influenced by the presence of outliers. This is especially true for when the SVM uses a Hard Margin Classifier where no labelled data points are allowed to lay either in the area of the margin or on the wrong side of the hyperplane. Hence, to ensure that the positioning of the optimal hyperplane won't be as sensitive to outliers, SVM can

allow misclassifications by employing a Soft Margin Classifier which allows both within the constraints defined by the algorithm. Said constraints have to be chosen carefully to prevent either overfitting or a lack of specificity and is part of handling the Bias/Variance Trade-off typical for machine learning models.

[FIGURE 6] A comparison between a Hard Margin Classifier and a Soft Margin Classifier.

Notably the figures above all show examples of linear classification problems, meaning the two classes can be easily separated from each other with a linear hyperplane. However, this does not work when faced with a classification problem as depicted in the figure below.

[FIGURE 7] An example of a non-linear classification problem

For problems like this Support Vector Machines have to make use of the kernel trick to change a non-linear classification problem into a linear one before finding the optimal separating hyperplane.

### **The Kernel Trick**

To turn a non-linear classification problem into a linear classification problem, SVMs take the data points and artificially move them into a higher dimension with the help of a kernel function which plots the data points in a manner to make them linearly separable. In this higher dimension, the SVM will then find an optimal separating hyperplane as described in the previous chapter. In the last step, said hyperplane will get transformed back into the original dimension where it may no longer be linear.

A simple example of how the kernel trick works is detailed in Figure 8, where we artificially plot the datapoints from a one-dimensional space into a two-dimensional space with a polynomial kernel function, determine the hyperplane, and then transform the data back into the original space.

[FIGURE 8] <https://www.andreaperlato.com/theorypost/introduction-to-support-vector-machine/>

Since the kernel function does not actually transform the data into a higher dimension and instead only calculates the relationships between every pair of points as if they were in the higher dimension, the whole method is called the Kernel Trick.

### **The Kernel Function**

There are many different ways of creating a kernel function. The example above was a polynomial kernel function where the degree of the polynomial was set to  $d=2$  for easier understanding. Usually, the best value for  $d$  is found via cross validation during the training process of the model.

In this talktorial, however, we focus on a pairwise kernel method which uses a similarity matrix of drug pairs to transform our data points into a higher dimension to find the optimal separating hyperplane. How exactly we calculate the pairwise kernel will be discussed in the corresponding chapter in Practical.

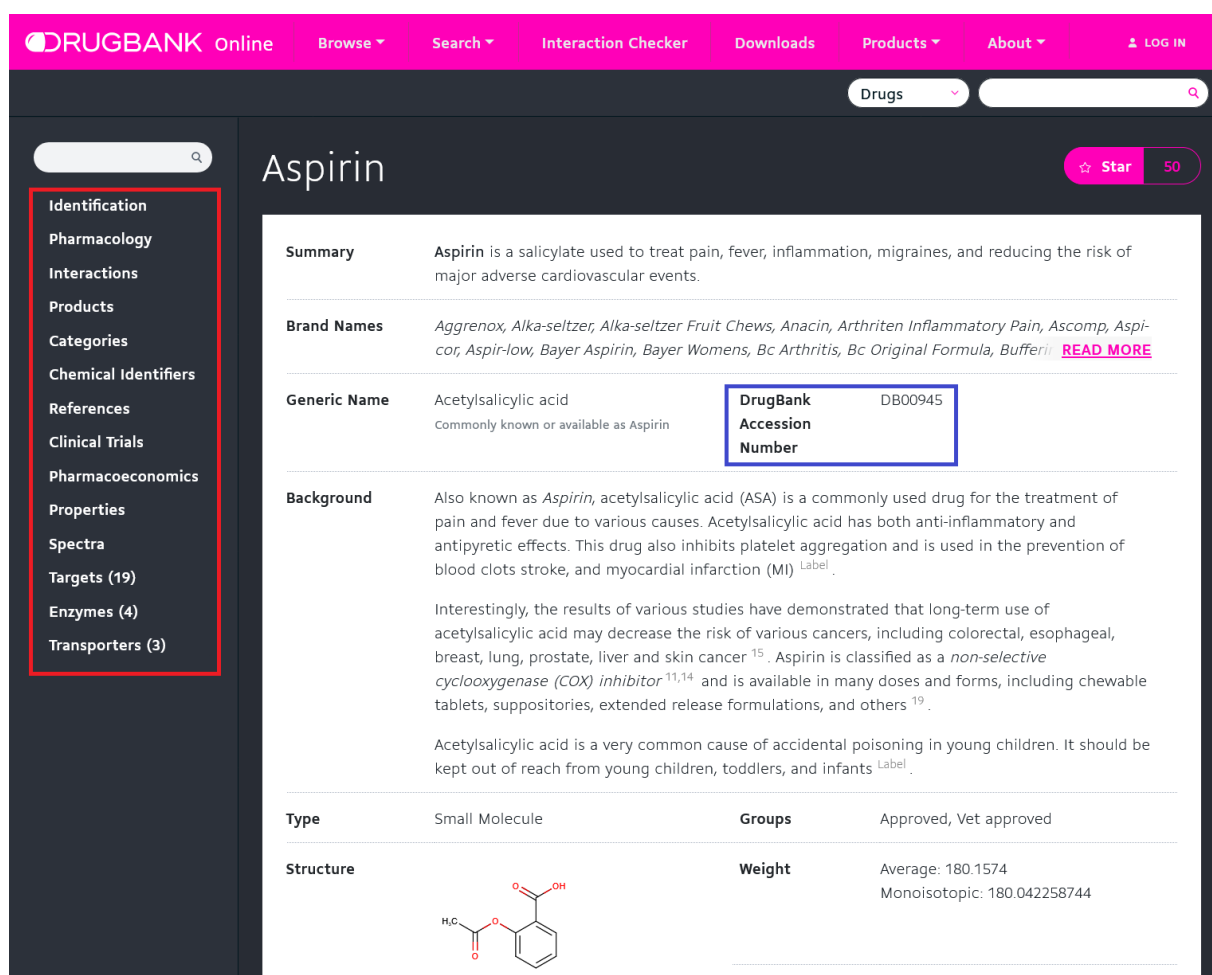
Detailed workflow on how to use a combined similarity matrix as a pairwise kernel function to determine drug combinations that show DDI based on a set of chosen features with the help of an SVM.

### **TODO**

Overview on the DrugBank database.

DrugBank is a comprehensive, free-to-access, online database containing information on drugs and drug targets. First established in 2006 by Dr David Wishart's lab at the University of Alberta as a project to help academic researchers to get easier and detailed, structured information about drugs, DrugBank grew in size and popularity thanks to the backing of various research organisations as well as government funding. Now in its 5<sup>th</sup> version (version 5.1.10 as of 1<sup>st</sup> April 2023), Drug Bank contains over 15,000 drug entries with almost 5,296 non-redundant protein sequences being linked to them. Each entry contains more than 200 data fields and combines detailed drug data (chemical, pharmacological, pharmaceutical, etc.) with comprehensive drug target data like sequence, structure, and pathway, collected from bioinformatics and cheminformatics resources.

To give an easy access to researchers from various fields, DrugBank provides various ways to browse, search and filter for information on its website ([www.drugbank.ca](http://www.drugbank.ca)). All entries are likewise clearly structured, systemically ordered, and quickly accessible over the navigation toolbar at its left side, and possess a unique number through which they can be addressed within the database.



**DRUGBANK** Online Browse Search Interaction Checker Downloads Products About LOG IN

Drugs

Aspirin ☆ Star 50

**Identification**  
Pharmacology  
Interactions  
Products  
Categories  
Chemical Identifiers  
References  
Clinical Trials  
Pharmacoeconomics  
Properties  
Spectra  
Targets (19)  
Enzymes (4)  
Transporters (3)

**Summary** Aspirin is a salicylate used to treat pain, fever, inflammation, migraines, and reducing the risk of major adverse cardiovascular events.

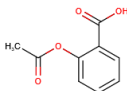
**Brand Names** Aggrenox, Alka-seltzer, Alka-seltzer Fruit Chews, Anacin, Arthriten Inflammatory Pain, Ascomp, Aspicor, Aspir-low, Bayer Aspirin, Bayer Womens, Bc Arthritis, Bc Original Formula, Bufferin [READ MORE](#)

**Generic Name** Acetylsalicylic acid  
commonly known or available as Aspirin

**DrugBank Accession Number** DB00945

**Background** Also known as *Aspirin*, acetylsalicylic acid (ASA) is a commonly used drug for the treatment of pain and fever due to various causes. Acetylsalicylic acid has both anti-inflammatory and antipyretic effects. This drug also inhibits platelet aggregation and is used in the prevention of blood clots stroke, and myocardial infarction (MI) <sup>Label</sup>.  
  
Interestingly, the results of various studies have demonstrated that long-term use of acetylsalicylic acid may decrease the risk of various cancers, including colorectal, esophageal, breast, lung, prostate, liver and skin cancer <sup>15</sup>. Aspirin is classified as a *non-selective cyclooxygenase (COX) inhibitor* <sup>11,14</sup> and is available in many doses and forms, including chewable tablets, suppositories, extended release formulations, and others <sup>19</sup>.  
  
Acetylsalicylic acid is a very common cause of accidental poisoning in young children. It should be kept out of reach from young children, toddlers, and infants <sup>Label</sup>.

**Type** Small Molecule **Groups** Approved, Vet approved

**Structure**  **Weight** Average: 180.1574  
Monoisotopic: 180.042258744

[FIGURE X] Screenshot of DrugBank Entry of Aspirin. The red-framed toolbar provides easy access to the different categories of drug information provided by DrugBank. The blue square marks the unique accession number through which each drug or small molecule can be uniquely addressed within the database.

## Practical

Retrieving necessary data from DrugBank.

**TODO**

Creating a drug-drug 2D molecular structure similarity database using OpenBabel.

**TODO**

Creating a drug-drug 3D pharmacophoric similarity database using the Schrödinger package.

**TODO**

Creating a drug-drug interaction profile database.

**TODO**

Construction of one pairwise similarity matrix for each feature and combining them into one final similarity matrix with labels which will be used as the kernel for the SVM.

**TODO**

Modelling and evaluating the SVM.

**TODO**

## Discussion

**TODO**

## Quiz

- Why is determining drug-drug interactions (DDI) relevant?
- How do you determine similarity between two drugs?
- How do you create a similarity matrix for drug-pairs?
- Can you briefly explain how a Support Vector Machine (SVM) can be used to predict DDI?