University of Antwerp

Chemo-informatics and computational drug design

Prof. Dr. Hans De Winter

University of Antwerp

Campus Drie Eiken, Building A

Universiteitsplein 1, 2610 Wilrijk, Belgium

Gefinancierd door de Europese Unie

NextGenerationEU

Chapter 4. Fingerprints, molecular similarity and clustering

1. Molecular fingerprints

In order for computers to handle, search and compare molecules, it is necessary that these molecules are represented in a computer-readable form. Molecular fingerprints are a way of encoding the structure of a molecule, and which involve turning the molecule into a sequence of bits that can then be easily compared between molecules. There are several types of molecular fingerprints depending on the method by which the molecular representation is transformed into a bit string. Most methods use only the 2D molecular graphs (the topology) and are thus called 2D fingerprints. The best known of these 2D fingerprints are the Daylight, Morgan and MACCS fingerprints.

1.1. Linear path-based: Daylight fingerprints

The Daylight fingerprints were originally developed by Daylight Chemical Information Systems, Inc., a US-based company founded in 1987 by the Weininger brothers (http://www.daylight.com/about/index.html). Their fingerprint technology is categorised as topological or path-based fingerprints, and work by analysing all the fragments of the molecule following a linear path up to a certain number of bonds, and then hashing every one of these paths to create the fingerprint (Figure 17). The Daylight fingerprints consist of up to 2,048 bits and encode all possible connectivity pathways through a molecule up to a given length (mostly 7 atoms). Most software packages implement these fingerprints or fingerprints based on them.

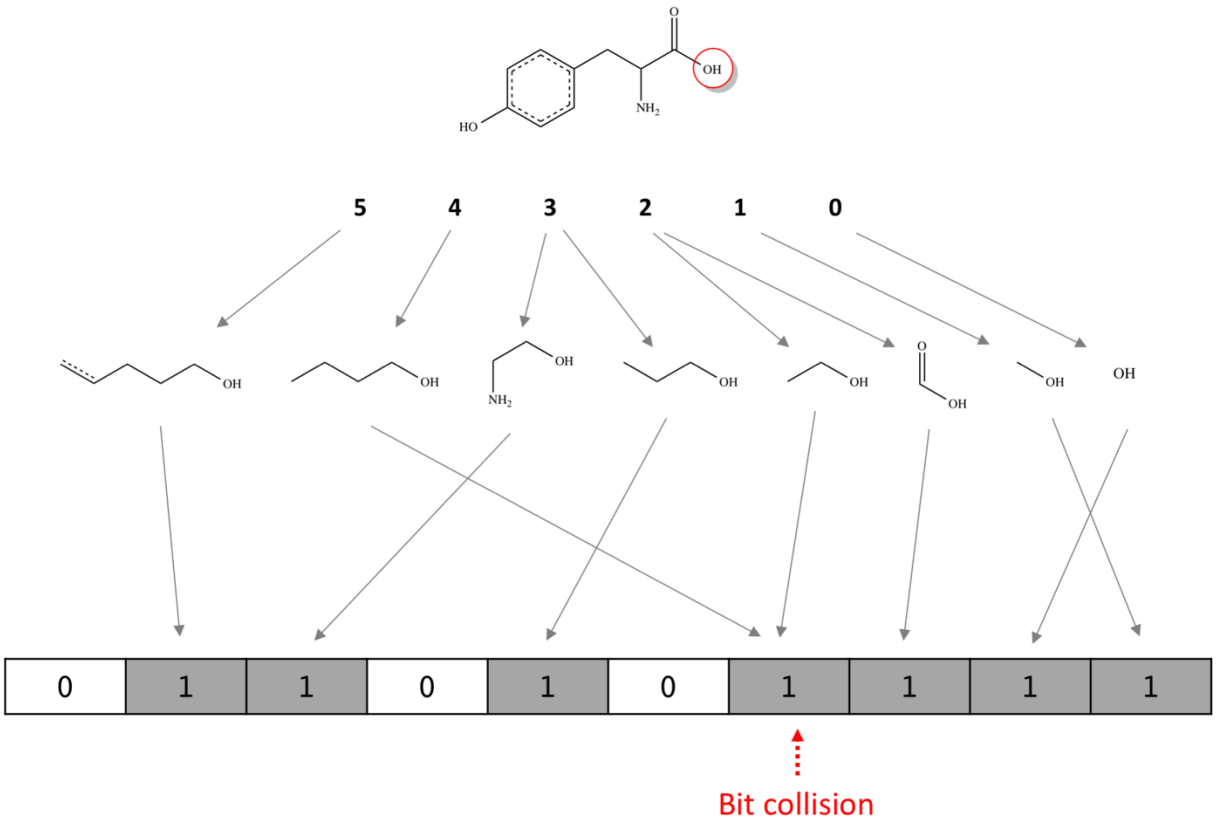

Figure 17. A representation of a hypothetical 10-bit topological fingerprint (a real Daylight fingerprint consists of 2,048 bits), in this case a linear path-based fingerprint with fragments up to a length of 5 (in the Daylight implementation this is normally a length of 7 atoms). All fragments found from the starting atom (circled) are shown, and the fragment length and corresponding bit in the fingerprint are indicated. There is one-bit collision, which are bits that are set by more than one fragment; these are likely in fingerprints with a reduced number of bits. Only fragments and bits for a single starting atom are shown; for the full fingerprint, this process would be carried out for every atom in the molecule. Adapted from reference 4.

1.2. Circular path-based: Morgan fingerprints

Circular fingerprints are also hashed topological fingerprints, but they are different in that instead of looking for paths in the molecule, the environment of each atom up to a determined radius is recorded. They are widely used for full structure similarity searching. The Morgan fingerprints are also called ECFP fingerprints (Extended-Connectivity Fingerprint). They represent circular atom neighbourhoods and produce fingerprints of variable length. They are most commonly used with a diameter of 4 and referred to as ECFP4. A diameter of 6 (ECFP6) is also commonly used.

ECFPs have two typical representations (Figure 18):

- List of integer identifiers. The natural and accurate representation of ECFPs is by means of varying-length lists of integer identifiers. Each identifier represents a particular substructure, more precisely, a circular atom neighbourhood, which is present in the molecule. The list of integer identifiers is sorted in ascending order. These identifiers can also be interpreted as indexes of bits in a huge virtual bit string. Each position in this bit string accounts for the presence or absence of a specific substructure feature. Since this virtual bit string is extremely large and sparse, it is not stored explicitly, but the indexes of the 1 bits are recorded in a varying-length list. In spite of this interpretation, the feature identifiers are stored as signed values due to technical reasons, that is, they can be either positive or negative. By default, this integer list representation contains only one instance of each identifier. However, in particular applications, it could be beneficial to consider the frequency count of the ECFP features, that is, to record each identifier as many times as the represented feature occurs in the molecule. This variation of ECFP is often denoted as ECFC.
- Fixed-length bit string. Traditional representation of binary molecular fingerprints is by means of fixed-length bit strings. This representation can also be applied to ECFPs by "folding" the underlying virtual bit string into a much shorter bit string of specified length (for example 1,024). Compared to the identifier lists, this representation simplifies the comparison and similarity calculation of ECFPs and it could reduce the required storage space, especially for large molecules. On the other hand, the applied folding operation increases the likelihood of collision, that is, two (or more) different substructure features could be represented by the same bit position. As a result, a certain amount of information is usually lost, which worsens both the quality and interpretability of this representation.

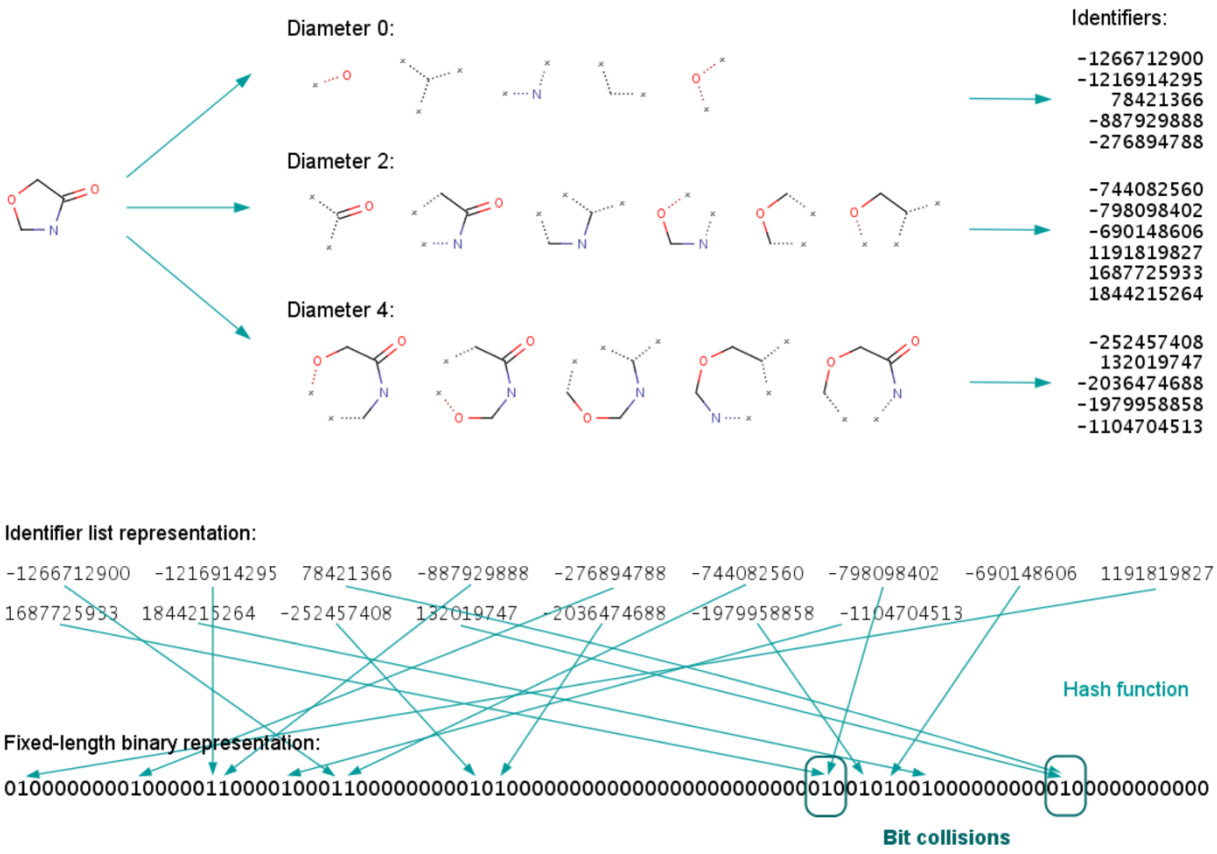

Figure 18. Upper part: ECFP generation process. Lower part: generation of the fixed-length bit string by 'folding'.

1.3. Substructure-based: MACCS keys

Substructure keys-based fingerprints set the bits of the bit string depending on the presence in the compound of certain substructures or features from a given list of structural keys. This usually means that these fingerprints are most useful when used with molecules that are likely to be mostly covered by the given structural keys, but not so much when the molecules are unlikely to contain the structural keys, as their features would not be represented. Their number of bits is determined by the number of structural keys, and each bit relates to presence or absence of a single given feature in the molecule (Figure 19), which is not the case with path-based fingerprints.

The MACCS-key comes in two variants, one with 960 and the other with 166 structural keys based on substructure patterns. The shorter one is the most commonly used, as it is relatively small in length but covers most of the interesting chemical features for drug discovery and virtual screening. The larger variant contains proprietary substructures and cannot be processed by all software packages.

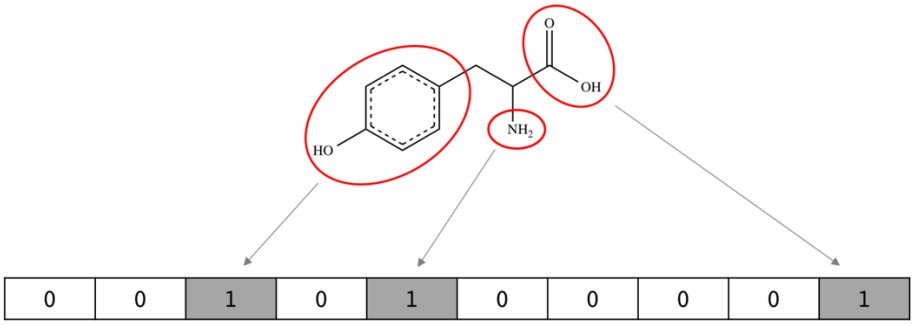

Figure 19. A representation of a hypothetical 10-bit substructure fingerprint, with three bits set because the substructures they represent are present in the molecule (circled). Adapted from reference 4.

2. Similarity metrics

Quantifying the similarity of two molecules is a key concept and a routine task in cheminformatics. Its applications encompass a number of fields, mostly medicinal chemistry-related, such as virtual screening. A virtually infinite number of methods have been described to calculate similarity and dissimilarity, but the Tanimoto index and Euclidean distance metrics being the most widely used.

2.1. Tanimoto

The Tanimoto similarity metric is commonly used in conjunction with bitwise fingerprints, such as the linear hash-based Daylight and the fixed-length Morgan fingerprints. Represented as a mathematical equation:

$$T(a,b) = \frac{N_c}{N_a + N_b - N_c}$$

with N_a and N_b representing the number of bits set (state 1) in fingerprints ***a*** and ***b***, respectively, and N_c the number of common bits set. The Tanimoto distance runs from 0 to 1.

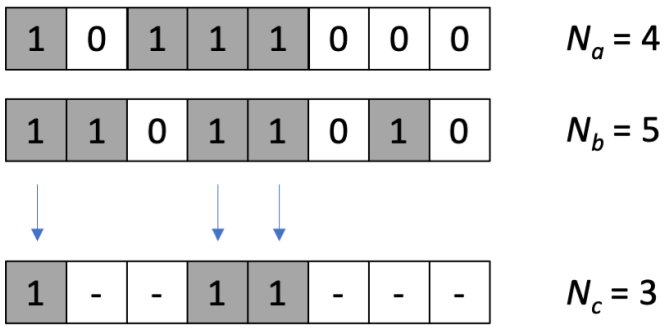

Figure 20. Illustration of the Tanimoto index on a hypothetical example of two bit strings of 8 bits each. T(a,b) = 3 / (4 + 5 – 3) = 0.5. Perfect similarity results in an index of 1, while perfect dissimilarity results in an index of 0.

2.2. Tversky

The Tversky similarity measure is an asymmetrical metric, meaning that the resulting value depends on the order of the two fingerprints:

$$T(a,b) = \frac{N_c}{\alpha O_a + \beta O_b + N_c}$$

with O_a and O_b being the number of bits that are only set (state 1) in the fingerprints of ***a*** and ***b***, respectively. N_c is defined as above. The factors α and β are weighing factors that can be used to put more weight on the fingerprints ***a*** and ***b***, respectively. Setting the parameters $\alpha = \beta = 1.0$ is identical to using the Tanimoto measure. The Tversky metric runs from 0 to 1, with 0 indicating total dissimilarity and 1 indicating equality.

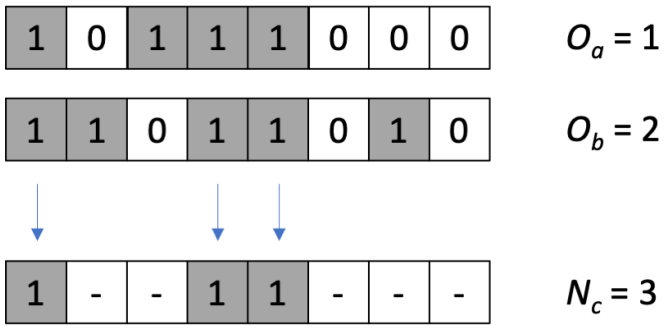

*Figure 21. Illustration of the Tversky index on a hypothetical example of two bit strings of 8 bits each. With $\alpha = 0.9$ and $\beta = 0.1$, T(a,b) = 3 / (0.9 * 1 + 0.1 * 2 + 3) = 0.73. However, with $\alpha = 0.1$ and $\beta = 0.9$, T(a,b) = 3 / (0.1 * 1 + 0.9 * 2 + 3) = 0.61. Perfect similarity results in an index of 1, while perfect dissimilarity results in an index of 0.*

The actual choice of the α and β values depends on the research question one wants to answer. If fingerprint $\boldsymbol{a}$ is the fingerprint of a query molecule, and $\boldsymbol{b}$ is the fingerprint of a molecule in a database in which one wants to look for molecules that are similar to $\boldsymbol{a}$, then setting α to a large value (e.g. 0.9) will give large values of *T(a,b)* for database molecules that are superstructures of the query $\boldsymbol{a}$, while setting α to a small value (e.g. 0.1) will give large values of *T(a,b)* for database molecules that are rather substructures of $\boldsymbol{a}$.

2.3. Application of similarity metrics

Similarity or distance measures have been used widely to calculate the similarity or dissimilarity between two samples of dataset. Cheminformatics is known as the domain that dealing with chemical information and both similarity and distance coefficient have been an important role for matching, searching and classification of chemical information. Similarity metrics are applied in a number of tasks:

- Selection of a set of diverse compounds for *in vitro* screening. In case one wants to purchase a small subset of chemical compounds from a large database of potential structure, it is often advisable to select a small but diverse subset of these structures in order to increase the likelihood of identifying a potential hit compound that is active against the *in vitro* screen one is investigating. Diversity-based selection is often performed by means of compound clustering, an approach that is further described in section 3.
- Selection of a set of similar compounds for hit conformation and the development of quantitative structure-activity relationships (QSAR). The chemical similarity principle, which states that compounds with similar structure will probably have similar bioactivities, is an underlying assumption of similarity-based virtual screening.
- Development of machine learning models in QSAR. Machine learning techniques can be broadly classified as supervised or unsupervised learning.

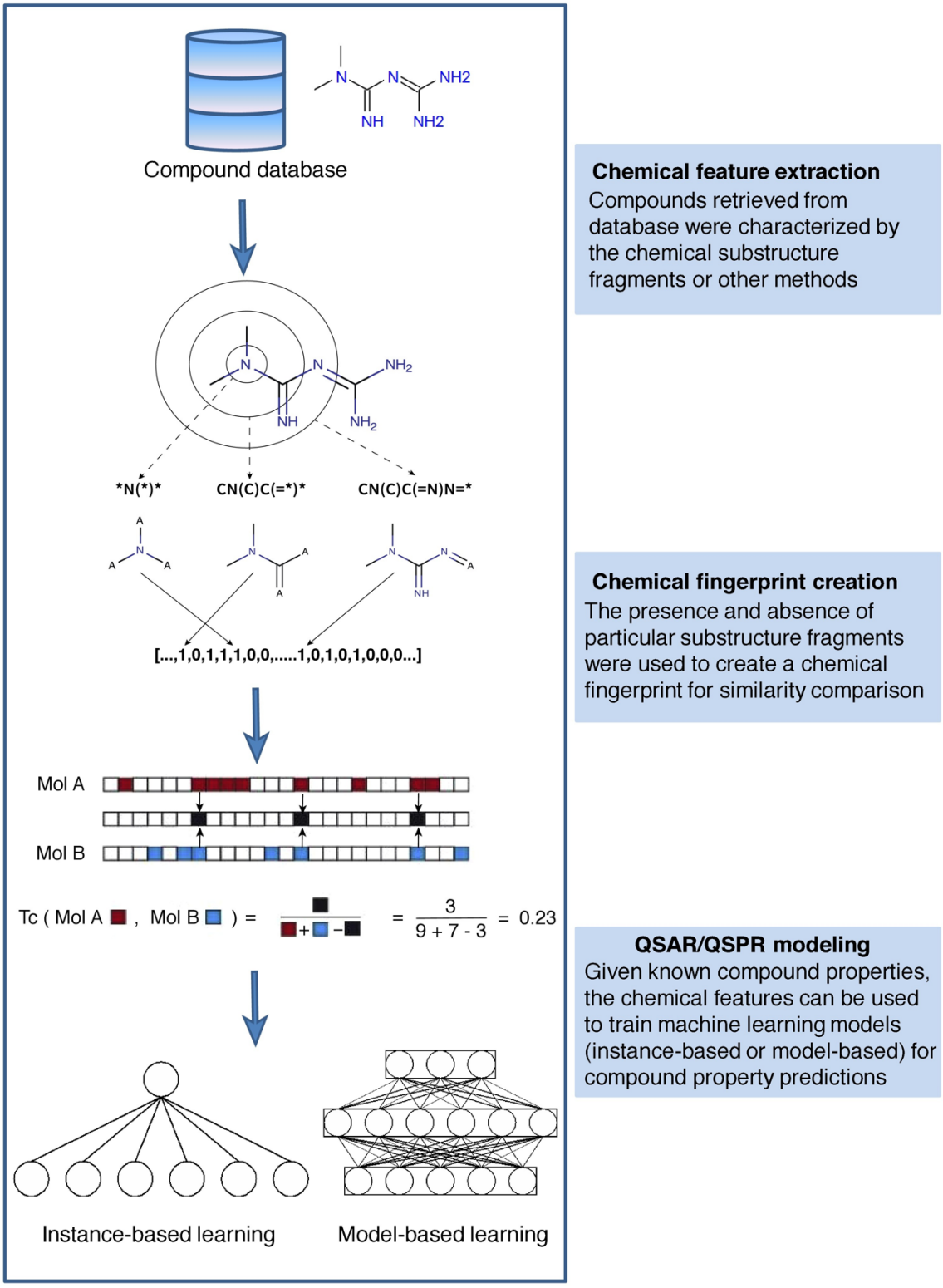

Figure 22. Overview of the process that is commonly used in pharmaceutical drug discovery. Compound databases are converted in chemical fingerprints, which are in turn used to search for similar compounds or for QSAR model building.

3. Maximum common substructure

The maximum common substructure (MCSS) is the largest substructure (graph) which can be identified between two or more molecules.

To illustrate the MCSS concept, consider the example in Figure 23. In this example, the MCSS between morphine, codeine and heroine is shown in red.

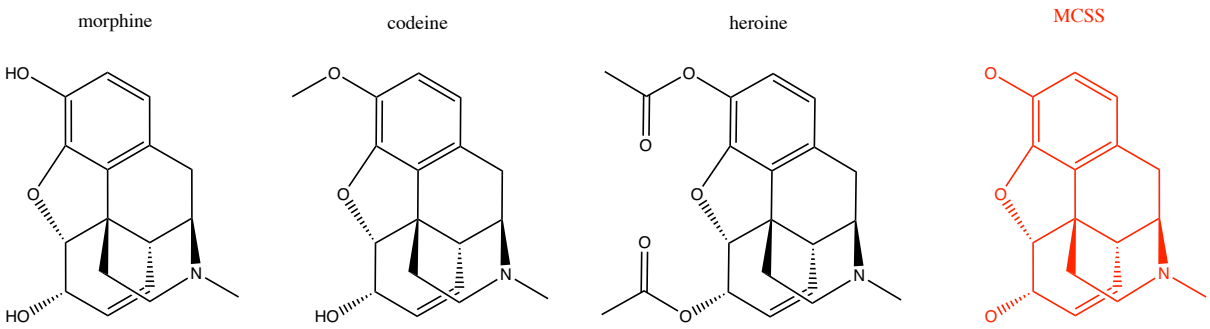

Figure 23. The maximum common substructure (red) between morphine, codeine and heroine. RDKit was used to calculate this MCSS.

The RDKit code to calculate the MCSS is rather straightforward but may take some time for complex molecule sets (normally in the order of microseconds, but it may go to minutes in some rare cases):

```
from rdkit.Chem import rdFMCS

morphine = Chem.MolFromSmiles("CN1CC[C@]23C4=C5C=CC(O)=C4O[C@H]2[C@H](C=C[C@H]3[C@H]1C5)O")
codeine = Chem.MolFromSmiles("CN1CC[C@]23[C@@H]4[C@H]1CC5=C2C(O[C@H]3[C@@H](O)C=C4)=C(OC)C=C5")
heroine =
Chem.MolFromSmiles("CN([C@H](CC(C=C1)=C23)[C@@H]4C=C[C@@H]5OC(C)=O)CC[C@]43[C@H]5OC2=C1OC(C)=O")

mols = [morphine, codeine, heroine]
mcss = rdFMCS.FindMCS(mols)
```

There exist many variations in the algorithms to calculate the MCSS of molecules, but the majority is based on some kind of backtracking approach in which an exhaustive search on all combinations is performed. Each molecule is converted into a graph representation (a graph is a set of edges [bonds] and nodes [atoms]) that is traversed iteratively to identify common edges and nodes.

4. Clustering

With the increase in the amount of chemical information available, methods that can organize chemical structures and their associated data are essential. Cluster analysis refers to a group of statistical methods that are used for identifying groups ('clusters') of similar items in multidimensional space. They require a measure of similarity between items, hence the Tanimoto or Euclidean distance measures are wide used for this. In cheminformatics, clustering methods are used for three main purposes:

- Grouping compounds into chemical series (or something approximating to this), as a way of organizing large datasets. For example, it is easier for a chemist to browse through 500 clusters (where the molecules in a cluster are similar) than 50,000 arbitrarily ordered compounds
- Identifying new bioactive molecules: if a compound with unknown activity is in a cluster that is biased towards compounds with known activity, we can make a prediction of the probability of activity of the unknown compound (for example, if 75% of the compounds in the cluster are active, we might say the probability of activity is 75%).
- Picking representative subsets: if we cluster a set of compounds, we can then take one compound from each cluster as a 'representative' of this cluster, and the total set of representative compounds as a representative subset of the whole dataset. This is sometimes more useful than random selection.

4.1. Hierarchical clustering

Clustering methods can be either **hierarchical** or **non-hierarchical**. Hierarchical clustering creates a tree of clusters, with, at the bottom level, every item in its own cluster, and at the top level all items in one cluster. Algorithmically,

this can be done either by starting at the bottom and progressively merging clusters (agglomerative) or starting at the top and breaking up clusters (divisive).

Mostly, the hierarchical agglomerative methods work in the same algorithmic fashion, but differ in the way that they decide which clusters to merge at each level.

1) The **Ward's method** has been used widely in cheminformatics, and is distinguished by merging clusters which, when merged, have the smallest increase in variance from the mean (i.e. create the 'tightest' cluster when merged).

2) Other methods include **single linkage** (the clusters are merged with the minimum distance between the nearest two points in each cluster); complete linkage (the clusters are merged with the minimum distance between the farthest points in each cluster); and group average (the minimum value of the mean distance between all pairs in the two clusters). Due to their computational complexity and memory requirements, hierarchical methods do not scale well to very large datasets, and thus they are giving way to faster, non-hierarchical methods. In order to create a partitioned grouping of a dataset (i.e. where every item is in one and only one cluster, or is a singleton), one must select a horizontal slice from this tree (Figure 24).

In order to extract a partition from the hierarchy (i.e. a grouping of compounds where every compound is in one and only one cluster), we need to select a 'level' from this hierarchy. This can be done intuitively or by using some algorithms.

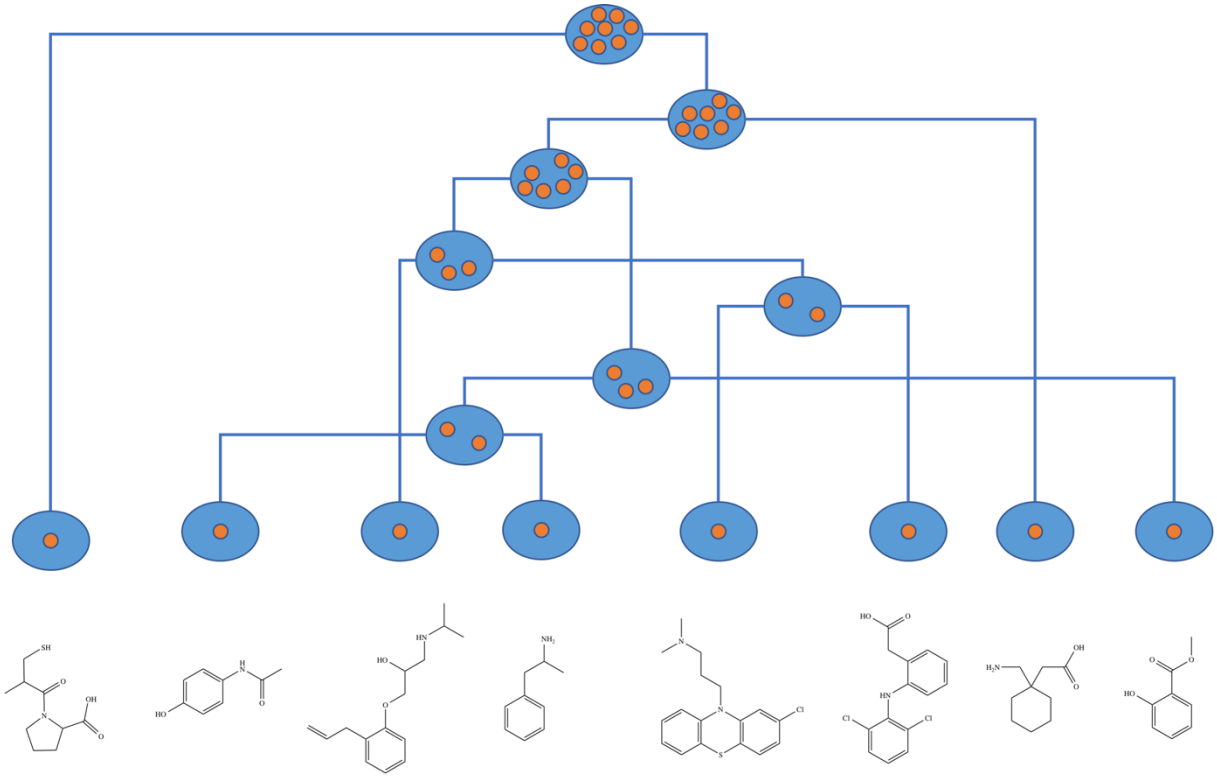

Figure 24. Hierarchical clustering work by initially putting every item in a cluster by itself, at the bottom level (with n clusters, where n is the number of points). It then identifies the two clusters to merge (depending on the methods as described above) and merges them to form a new cluster at the next level. The next level up will therefore consist of one cluster with two points, and all the rest of the points in clusters by themselves (i.e. there will be n-1 clusters). The process repeats until there is just one cluster at the top containing all the points, resulting in a cluster hierarchy.

4.2. Non-hierarchical clustering

Non-hierarchical methods can use a variety of algorithms, but they generally all produce a single partitioning of the dataset into clusters (versus a tree which can result in many partitions). Some of the more common non-hierarchical methods are Jarvis-Patrick, *K*-means and *K*-medoids.

1) **Jarvis-Patrick (JP)** is a non-hierarchical method where, for each compound in a set, the *j* nearest neighbours (that is the *j* other compounds in the dataset that are the most similar) are identified. Two compounds cluster

together if they 1) are in each other's list of *j* nearest neighbours, and 2) have k_{min} of their *j* nearest neighbours in common (Figure 25). This method doesn't require level selection, but does require *j* and k_{min} to be predefined. For example, one might choose a nearest neighbour list of size 20 (*j*) and put molecules in the same cluster if they share more than 10 nearest neighbours (k_{min}). Depending on the types of fingerprints used, Tanimoto is usually used as the measure of similarity. JP is fast, but has had mixed results in cheminformatics in terms of quality.

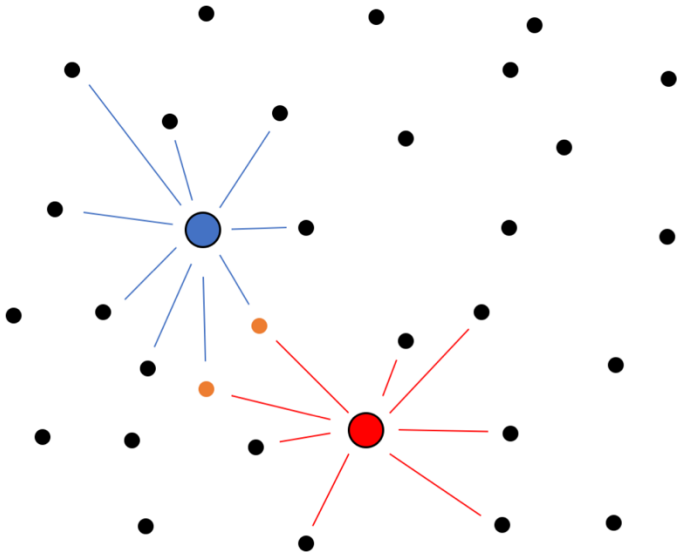

Figure 25. Illustration of the Jarvis-Patrick algorithm. Each dot represents a molecule and the distance between the dots corresponds to the Tanimoto similarity between the compounds (larger distance, lower similarity). For each of the blue and red compounds, the j=8 nearest neighbors are shown using lines. The two compounds that are shared by both nearest neighbor lists are shown in orange. The blue and red compound would cluster together if k_{min} would be set to 1 or 2, but not if $k_{min} > 2$.

2) ***K*-means** clustering is more widely used than JP. It requires that the number of desired clusters *k* be known in advance. The algorithm proceeds by alternating between these steps:

- Assignment step: assign each observation to the cluster whose mean is nearest;
- Update step: calculate the new cluster means to be the centroids of all molecules in the cluster.

The algorithm has converged when the assignments no longer change, however there is no guarantee that the optimum is found using this algorithm and the result may depend on the initial clusters. Generally, only a few (<100, often <10) iterations are required to settle (Figure 26).

The assignment of the initial *k* cluster centroids is normally done in a *random fashion*. However, alternative methods exist, such as the *Random Partition method* in which each molecule is initially assigned random to one of the *k* clusters, and then calculating the mean from each of the cluster's randomly assigned points to get the initial cluster centres. A consequence of the Random Partition method is that the initial cluster centres are all close to the centre of the dataset (Figure 27).

3) ***K*-medoids** is a derivative of K-means, but differs from *K*-means in that it uses real compounds, or 'medoids', to represent cluster centres, rather than centroids.

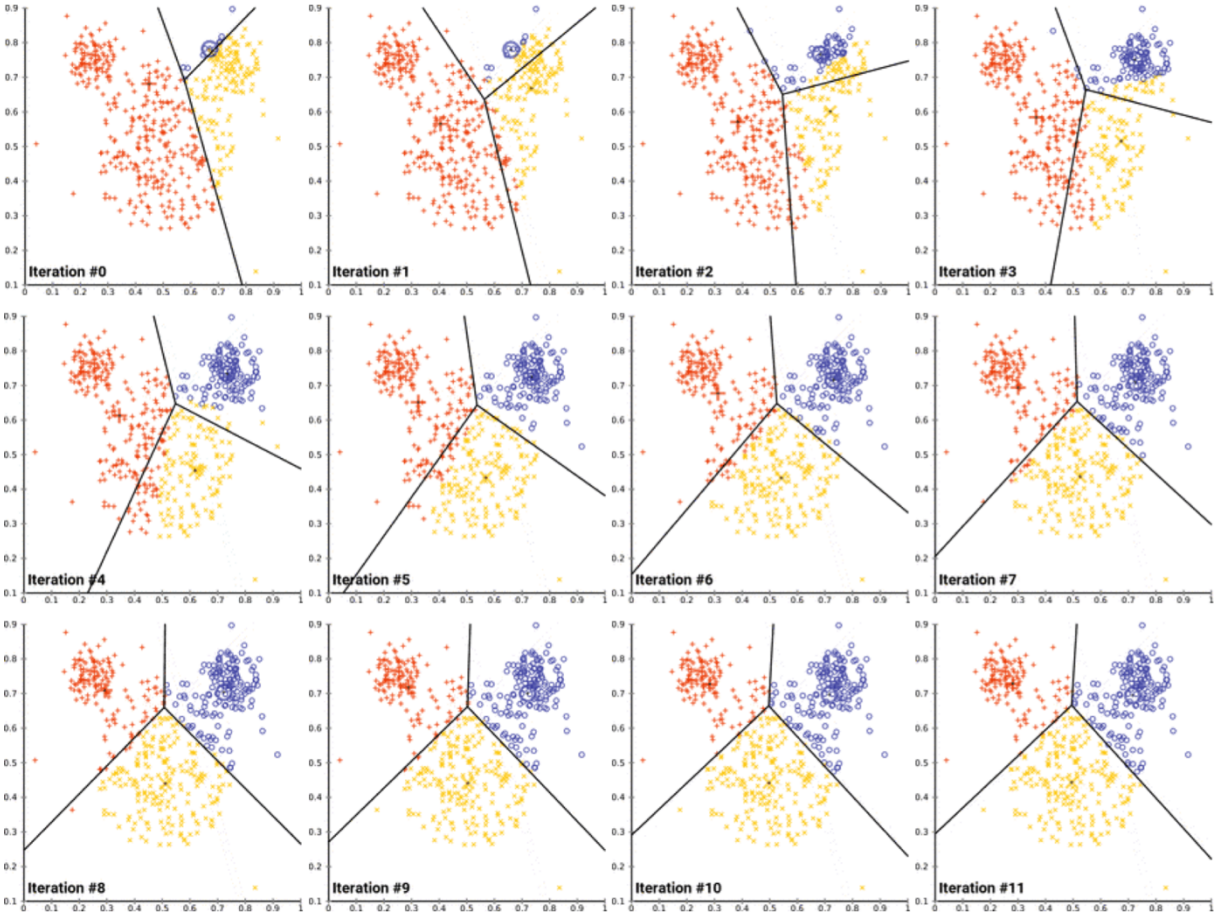

Figure 26. Illustrating the iterative process of the k-means clustering algorithm (in this case, k=3). The centroids of each cluster are shown as black dots.

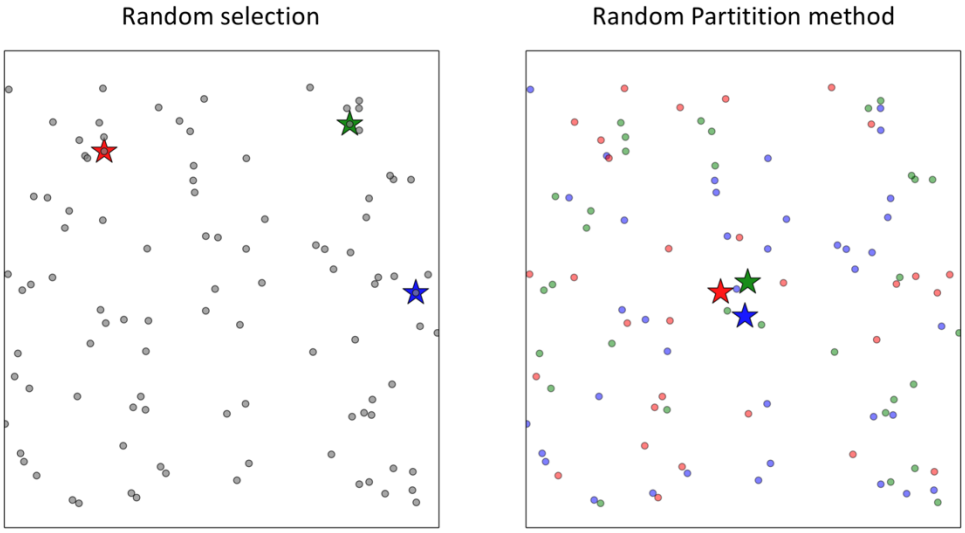

Figure 27. Initialization step of the k-means clustering method. The random selection method is nothing more than a random selection of k cluster centers from the dataset, while the 'Random Partition' method assigns each molecule initially random to one of the k *clusters, and then calculating the mean from each of the cluster's randomly assigned points to get the initial cluster centres.*

4.3. Self-organising maps (SOM) or Kohonen networks

SOM's are a type of artificial neural networks that are trained using unsupervised learning to produce a two-dimensional and discretised representation of the input molecules. The artificial neural network introduced by the Finnish professor Kohonen in the 1980s and is therefore sometimes called a Kohonen map or network. Kohonen

maps can be used in conjunction with spectrophore fingerprints, which makes it an attractive method to cluster compounds based on their spectrophore similarities (Figure 28).

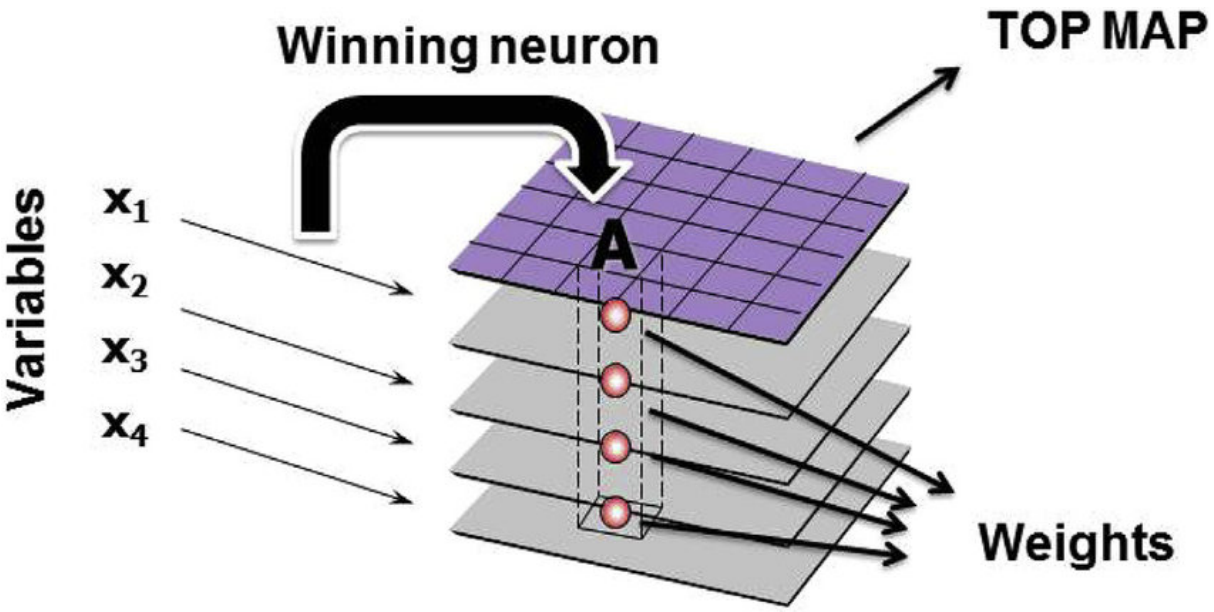

Figure 28. Illustration of a SOM network consisting of 6 x 6 cells, each represented by a weight vector of size 4. Adapted from https://goo.gl/images/15mSAo .

- Initialisation phase. The user needs to define the desired map size. In most cases, a grid of 10 by 10 cells is often sufficient, but other size may be tested as well. Each grid cell is composed of a vector of 48 values (the size of the spectrophore vector), and all values of these 100 vectors are initially assigned with random numbers.
- Step 1. From the database of compounds that need to be clustered, an input molecule (spectrophore) is selected.
- Step 2. The Euclidean distance between the input spectrophore and each of the 100 cells is calculated, after which the cell with the smallest distance is selected.
- Step 3. The 48 values of the selected cell are updated according a user-defined update function. Many flavours exist for this function, but a good start is to use an average function (each of the values in the cell are assigned a new value which is nothing more than the average between the old value and the corresponding value of the input spectrophore).
- Step 4. Stop if the maximum number of iterations has been reached, otherwise continue with step 1.

5. Diversity analysis

Diversity analysis gained popularity in the late 1990's in response to the following needs in the pharmaceutical industry:

- There was much interest as to how well the corporate collections of compounds held by pharmaceutical companies 'covered' possible chemistry/drug space (Figure 29).
- Combinatorial chemistry experiments were producing many new compounds, and people wanted to know if these compounds added anything new (in terms of chemical or biological functionality) to their corporate collections, i.e. if they made the datasets more diverse, or just replicated what was already in there?
- Libraries of thousands of compounds became available for purchase – are they worth the money?

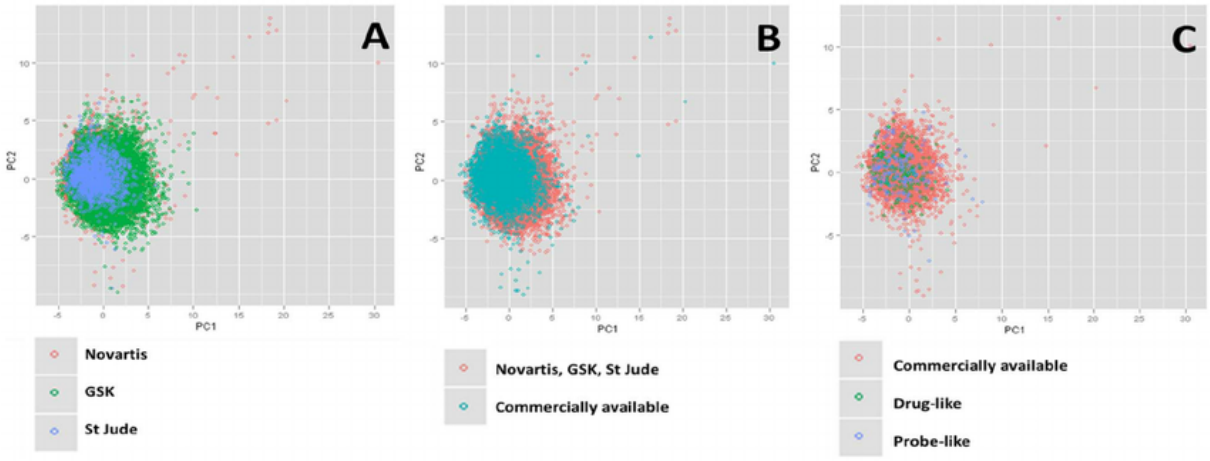

Figure 29. Principal Component Analysis plots. Chemical diversity of the GSK, Novartis and St-Jude libraries displayed (panel A); Overlap in chemical diversity of the combined datasets and the commercially available compounds (panel B); Overlap in chemical diversity of the commercially available compounds where the drug-like and probe-like chemotypes were annotated (panel C). Adapted from reference 5.