Notes about the three research papers

Bilal Akliai

May 2024

Contents

Fingerprints are a way to represent molecules as a sequence of bits, with 1s if a molecule meets a certain property and 0s otherwise.

The Tanimoto coefficient (ranging from 0 to 1) is then used to evaluate the similarity between two molecules (the more similar they are, the closer this coefficient is to 1).

1 Research Paper No. 1 : Why is the Tanimoto index an appropriate choice for fingerprint-based similarity calculations?

The idea of this research paper is to compare 8 similarity metrics. To do this, we will primarily use two statistical methods: SRD and ANOVA.

1.1 SRD

For each metric, we will rank the different compounds (the higher the ranking, the greater the similarity with the reference compound). Once all the rankings are done, we will average these rankings, giving us a composite ranking representing a consensus of these different rankings. Then, we will return to the level of the metrics, and for each of them, we will calculate the difference between the composite ranking and the ranking from that metric: this difference is called SRD (Sum of Ranking Differences) (it is, for a given metric, the sum over all our compounds of the absolute values of the ranking differences between the composite ranking and the one from the considered metric). The metric finally chosen is the one with the lowest SRD.

Illustrative example : Suppose we have three compounds (A, B, C) and three metrics (M1, M2, M3). The rankings of the compounds by each metric are as follows:

- M1: A $>$ B $>$ C (Rankings: A=1, B=2, C=3)
- M2: B $>$ A $>$ C (Rankings: A=2, B=1, C=3)
- M3: A $>$ C $>$ B (Rankings: A=1, B=3, C=2)

The average composite ranking would be:

$$\bar{X}_A = \frac{1+2+1}{3} = 1.33$$
$$\bar{X}_B = \frac{2+1+3}{3} = 2.00$$
$$\bar{X}_C = \frac{3+3+2}{3} = 2.67$$

The ranking differences for each metric compared to the composite:

$$\begin{aligned}
\text{M1} &: |1 - 1.33| + |2 - 2.00| + |3 - 2.67| = 0.33 + 0.00 + 0.33 = 0.66 \\
\text{M2} &: |2 - 1.33| + |1 - 2.00| + |3 - 2.67| = 0.67 + 1.00 + 0.33 = 2.00 \\
\text{M3} &: |1 - 1.33| + |3 - 2.00| + |2 - 2.67| = 0.33 + 1.00 + 0.67 = 2.00
\end{aligned}$$

In this example, M1 has the lowest SRD and is therefore the closest to the composite ranking.

1.2 ANOVA

The idea of this method is to determine if the means of several groups are statistically different from each other. Suppose we initially have similarity scores (previously calculated with a metric) for different groups corresponding to each method. We will first calculate the mean for each group, as well as an overall mean. Then, we will calculate the SSB (variance between groups) and the SSW (variance within groups). Finally, we calculate an F statistic that provides information on the significant impact or not of the studied factor (the higher the F, the more significant the impact of the studied factor and therefore the more it discriminates/differentiates the molecules).

Illustrative example :

Suppose we have two methods of data preprocessing and we want to compare their effects on the similarity scores for several compounds. We have the following data:

- **Method 1**: Similarity scores [10, 12, 15, 14, 13]
- **Method 2**: Similarity scores [22, 20, 19, 21, 23]

1.2.1 Calculating Means

Mean of each group:

$$\begin{aligned}
\text{Mean Method 1}(\bar{X}_1) &= \frac{10 + 12 + 15 + 14 + 13}{5} = \frac{64}{5} = 12.8 \\
\text{Mean Method 2}(\bar{X}_2) &= \frac{22 + 20 + 19 + 21 + 23}{5} = \frac{105}{5} = 21
\end{aligned}$$

Overall mean:

$$\text{Overall mean}(\bar{X}) = \frac{10 + 12 + 15 + 14 + 13 + 22 + 20 + 19 + 21 + 23}{10} = \frac{169}{10} = 16.9$$

1.2.2 SSB (Sum of Squares Between)

SSB represents the variance due to differences between groups (here, the two preprocessing methods). It measures how much each group mean deviates from the overall mean.

$$\text{SSB} = \sum_{i=1}^{k} n_i(\bar{X}_i - \bar{X})^2$$

where k is the number of groups, n_i is the number of observations in group i, $\bar{X}_i$ is the mean of group i, and $\bar{X}$ is the overall mean.

For our example:

$$\begin{aligned} n_1 &= 5 \\ n_2 &= 5 \end{aligned}$$

Calculating $(\bar{X}_1 - \bar{X})^2$ and $(\bar{X}_2 - \bar{X})^2$:

$$\begin{aligned} (\bar{X}_1 - \bar{X})^2 &= (12.8 - 16.9)^2 = (-4.1)^2 = 16.81 \\ (\bar{X}_2 - \bar{X})^2 &= (21 - 16.9)^2 = (4.1)^2 = 16.81 \end{aligned}$$

Now, calculating SSB:

$$\text{SSB} = n_1 \cdot 16.81 + n_2 \cdot 16.81 = 5 \cdot 16.81 + 5 \cdot 16.81 = 84.05 + 84.05 = 168.1$$

1.2.3 SSW (Sum of Squares Within)

SSW represents the variance due to differences within groups. It measures how much each observation deviates from the mean of its respective group.

$$\text{SSW} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^2$$

where X_{ij} is the observation j in group i.

For Method 1:

$$\begin{aligned} (10 - 12.8)^2 + (12 - 12.8)^2 &+ (15 - 12.8)^2 \\ &+ (14 - 12.8)^2 + (13 - 12.8)^2 \\ &= (-2.8)^2 + (-0.8)^2 + (2.2)^2 \\ &+ (1.2)^2 + (0.2)^2 \\ &= 7.84 + 0.64 + 4.84 \\ &+ 1.44 + 0.04 \\ &= 14.8 \end{aligned}$$

For Method 2:

$$\begin{aligned}(22-21)^2 + (20-21)^2 + (19-21)^2 \\ &+ (21-21)^2 + (23-21)^2 \\ &= (1)^2 + (-1)^2 + (-2)^2 \\ &+ (0)^2 + (2)^2 \\ &= 1+1+4 \\ &+ 0+4 \\ &= 10\end{aligned}$$

Now, calculating SSW:

$$\text{SSW} = 14.8 + 10 = 24.8$$

1.2.4 Calculating the F Statistic

The F statistic is calculated as follows:

$$F = \frac{MSB}{MSW} = \frac{\text{SSB}/\text{df}_{\text{between}}}{\text{SSW}/\text{df}_{\text{within}}}$$

where:

- MSB is the mean square between groups.
- MSW is the mean square within groups.
- $\text{df}_{\text{between}} = k - 1$ (k is the number of groups).
- $\text{df}_{\text{within}} = N - k$ (N is the total number of observations).

For our example:

$$\begin{aligned}\text{df}_{\text{between}} &= 2 - 1 = 1 \\ \text{df}_{\text{within}} &= 10 - 2 = 8\end{aligned}$$

$$\begin{aligned}MSB &= \frac{\text{SSB}}{\text{df}_{\text{between}}} = \frac{168.1}{1} = 168.1 \\ MSW &= \frac{\text{SSW}}{\text{df}_{\text{within}}} = \frac{24.8}{8} = 3.1\end{aligned}$$

Finally, the F statistic:

$$F = \frac{MSB}{MSW} = \frac{168.1}{3.1} \approx 54.23$$

1.2.5 Interpretation

A high F statistic indicates that there is a significant difference between the groups. In our example, the preprocessing method has a significant effect on the similarity scores, meaning one preprocessing method is more effective than the other.

1.3 What we've learned

The Tanimoto index is quite effective for comparing fingerprints of two molecules (to determine their structural similarity).

1.4 Summary

There are two issues: representing a molecule (fingerprints, for example) and measuring the distance/similarity between two molecules (Tanimoto index). There is still much to discover about the impact that the choice of similarity methods has on the results obtained. We use statistical methods (e.g., SRD or ANOVA) to try to choose the best metric.

Combining the results of several fingerprints to calculate a single score only slightly improves the accuracy of the results. However, using different fingerprints separately allows maximizing the diversity of identified molecules and discovering active molecules that other fingerprints would not have detected.

We have very few differences between different fingerprints (compared to the differences between molecules for the same fingerprint). Circular fingerprints are generally the most effective.

Similarity metrics are used in many fields outside of cheminformatics. For example, they are used for textured image retrieval (where similar images are sought), web page clustering (to organize websites based on their content), and more, which is why optimizing performance is important.

Several studies have been conducted to try to determine which is the most effective metric. Several similarity metrics stand out, including one based on Tanimoto coefficients. In a 2006 review, Willett maintains, among other conclusions, that "the well-established Tanimoto is the coefficient of choice for computing molecular similarities unless there is specific information about the sizes of the molecules".

Willett's research group has studied data fusion techniques to improve similarity-based virtual screening. They demonstrated that data fusion can improve the performance of similarity-based virtual screening using two distinct approaches: similarity fusion (where multiple similarity measures are used with a single reference structure) and group fusion (where a single similarity measure is used with multiple reference structures). They concluded that group fusion is generally far superior to similarity fusion. Furthermore, in previous work, they identified the Tanimoto coefficient as the best similarity metric for group fusion.

However, the Tanimoto index also has weaknesses: it tends to select small compounds during dissimilarity selection and can produce similarity values around 1/3 even for structurally distant molecules.

This article compares, in a fairly general way (without assumptions about our molecules, without focusing on a particular molecule), 8 similarity measures (Tanimoto, Dice, Cosine, Substructure and Superstructure similarities, and similarity definitions derived from Manhattan, Euclidean, and Soergel distances). About 5 million compounds were analyzed, sorted into 3 categories: Fragments (small molecules or fragments of molecules), Leadlike (molecules with promising characteristics to become "leads," i.e., starting compounds for drug development), and Druglike (molecules with characteristics similar to approved drugs).

Many methods exist to represent a molecule as a sequence of bits (e.g., ECFP4). One characteristic of the latter is that very few bits are set to 1 (about 5 to 10% only). Thus, it results in obtaining the same similarity values, even for different molecules.

Therefore, another method of molecular representation had to be chosen. Our next choice was the Chemaxon chemical fingerprint, a hashed fingerprint whose advantage over ECFPs is that it is "darker" (i.e., there are more bits activated on average) and this "darkness" can even be adjusted by modifying some parameters.

In the context of this study, a "target" is defined as a randomly chosen reference compound for each iteration. 1000 experiments were conducted to minimize bias.

SRD is approximately limited to a hundred molecules (in a reasonable time), so we decided to create 1000 groups of 100 molecules (one reference molecule and 99 to compare with it) and calculate for each of the 8 measures and each of the 1000 groups the SRD. Then, we calculate, for each similarity measure, the average of the SRDs obtained for each molecule, which reduces bias.

We have 3 data preprocessing methods:

- **Interval Scaling**:
  - **Description**: This method transforms values by normalizing them between a minimum and a maximum. It is described by the formula:

$$x_{i,j}(\text{interval scaled}) = \frac{x_{i,j} - \min(x_{i,j})}{\max(x_{i,j}) - \min(x_{i,j})}$$

  - **Usefulness**: This method is used to make values comparable by bringing them to a common scale.

- **Standardization**:
  - **Description**: Standardization transforms values in terms of standard deviations from the mean. It is described by the formula:

$$x_{i,j}(\text{standardized}) = \frac{x_{i,j} - \text{average}(x_i)}{\text{standard deviation}(x_i)}$$

  - **Usefulness**: This method normalizes data by eliminating the effects of different scales and dispersions.

- **Rank Transformation**:

  - **Description**: Rank transformation ranks values in each column, assigning ranks from 1 to 99, where the smallest value takes rank 1 and the largest takes rank 99.
  - **Usefulness**: This method is useful for comparing data based on their relative order rather than their absolute values.

We use the principle of cross-validation to determine if the observed differences are statistically significant or simply due to random variations in the data. This helps to establish the reliability and relevance of the tested similarity metrics. The principle is as follows: we divide our data into 7 sets, and we use 6 of the 7 sets for SRD calculation and one of the 7 for validation (if the gap between the average of the 6 SRDs and the value left for validation is large, then it is considered unreliable). We then plot the SRDs (between 0 and 100) on a graph, along with a Gaussian curve, and we consider that if an SRD overlaps our curve, then the ranking obtained using the similarity measure in question is no better than a random ranking.

Since the SRD method puts all influencing factors on the same scale, a factorial ANOVA was applied to distinguish the effects of the different factors. The effects of the following factors were analyzed:

- **Molecule size classes**: fragment, leadlike, druglike, all
- **Molecule selection method**: random and diverse
- **Scaling options (data preprocessing methods)**: interval scaling, standardization, rank transformation
- **Similarity indices**: Manhattan, Euclidean, Cosine, Dice, Tanimoto, Soergel, Substructure, Superstructure

In the case of interval scaling, the first two factors have a significant influence. The same applies to the other two preprocessing methods.

A three-factor factorial ANOVA was also performed (the first three factors). Two interactions were not significant: the combination of the selection method and the data scaling method, and the combination of all three factors. This last case means that the factor of different data preprocessing methods is not significant in combination with the other two factors. But it should be noted that the factor of different data preprocessing methods is significant on its own. Tanimoto, Soergel, Manhattan, and Euclidean measures seem to be quite close.

1.5 Conclusion

Cosine, Dice, Tanimoto, and Soergel were identified as the best (equally) similarity metrics, while similarity measures derived from Euclidean and Manhattan distances are far from optimal. However, this deviation from other metrics makes them good candidates for data fusion. It is important to note that in this context, "best" means the metric that, on its own, produces the most similar rankings to those produced by the average of the eight studied metrics. In other words, the informative content retrieved by considering all eight metrics is the same as by considering one of the four.

The two-factor ANOVA showed us that the factor of molecule size and the factor of selection method are significant separately and together in each case. This means that the results of the SRD analysis can be influenced by these two factors. Thus, the result depends on the size of the molecules and the selection method. In particular, rankings of Euclidean, Manhattan, Substructure, and Superstructure similarities showed significant dependencies on molecule size. The difference between data preprocessing methods is barely observable.

Possible improvement: extend the comparison to similarity metrics applied to non-dichotomous data and/or using SRD calculations in the case of repeated elements (degeneracies). Another possible extension of this study would involve examining lesser-known similarity metrics.

2 Research Paper No. 2 : Ligand-Based Virtual Screening Using Graph Edit Distance as Molecular Similarity Measure

Screening = the process of selection aimed at identifying elements with specific characteristics.

ErG = Extended Reduced Graph. GED = Graph Edit Distance. SED = String Edit Distance

Traditional similarity measures convert these graphs into 2D vectors (such as fingerprints) before making chemical comparisons. This study examines the effectiveness of a molecular comparison based solely on graphs, without conversion into 2D vectors. It uses extended reduced graphs and graph edit distance methods to calculate molecular similarity. In most cases, the method outperformed other molecular similarity methods that use table representations.

Two main categories:

- Structure-Based Virtual Screening (SBVS): Uses information about the structure of biological targets to search for chemical compounds.
- Ligand-Based Virtual Screening (LBVS): Uses information about the known activity of certain molecules to predict the unknown activity of new molecules.

Extended Reduced Graphs (ErGs) : Useful tool for virtual screening. In this new model, graph edit distance (GED) is used as a similarity measure between molecular structures based on ErGs. GED calculates a cost distance between two graphs, i.e., the minimum number of changes needed to transform one graph into another. Each modification can be one of six operations: insertion, deletion, and substitution for nodes and edges of the graph.

The datasets have been normalized in a ready-to-use format for the LBVS benchmarking platform developed by Skoda and Hoksza. The formatted datasets for this platform include several selections of active and inactive molecules grouped according to different targets.

Partitions are created with the previous selections (for example, if we have 100 active molecules and 100 inactive ones, we will create a partition with 80 active molecules and 80 inactive molecules for training and 20 active and 20 inactive molecules for testing. Then, we can create a second partition with 80 other active molecules and 80 inactive molecules for training (molecules used for the previous test can, for example, now be used for training) and 20 active and 20 inactive molecules for testing. The negative effects of randomness are mitigated by using multiple partitions for each selection.

Selections are also cataloged according to their level of difficulty, estimated by analyzing the performance trends of several commonly used LBVS methods. This performance is measured by the value of the area under the ROC curve (Receiver Operating Characteristic), an indicator of a method's ability to correctly distinguish between active and inactive molecules. A high AUC indicates good performance, while a low AUC suggests the task is more difficult.

Reduced graphs condense essential information from chemical graphs into characteristic nodes. For virtual screening, they locate features likely to interact with receptors while preserving their spatial and topological distribution. The method used, described by Stiefl et al., is called ErG and combines reduced graphs and binding property pairs, allowing representation of both the spatial side of the molecule and its function (its effects...). Nodes can represent one or more of the following features: hydrogen bond donor, hydrogen bond acceptor, positive charge, negative charge, hydrophobic group, and aromatic ring system. Some nodes without specific features serve as links to these relevant features and can be carbon or non-carbon binding nodes.

There are 3 different methods for measuring similarity between two ErGs (two already existing, and one new described in the paper):

- Fingerprint-Based: the molecule is represented as a bit array, called a fingerprint. Each bit represents the presence or absence of a certain substructure. The similarity between two molecules is then compared using the Tanimoto similarity index. Fingerprints can be described in two ways: either as atometypei-(distance)-atometypej where the distance is the minimum number of bonds between atoms of type i and type j. Another description is: as propertypoint1-(distance)-propertypoint2. Each time we encounter a property-property-distance triplet, we increase a number associated with it by one.

- SED-Based: the distance between two molecules A and B is defined as the maximum of the minimum costs after comparing all pairs of terminal nodes (degree one node) in A and B. For each molecule and for each pair of nodes, we calculate the minimum distance between these two nodes. Finally, for each pair of nodes, we take the minimum distance (between that in A and that in B). And the distance between molecules A and B will be defined as the maximum of these minimums.

- GED-Based: the distance is defined as the minimum number of modifications to transform one molecule into the other. This is weighted with costs to disfavor certain transformations, and the total cost (sum of the costs of each transformation) is divided by the average total number of nodes in our graphs (to avoid favoring small graphs).

The screening phase involves using all the active and inactive compounds in the test set to compare them with the active compounds in the training set.

For performance analysis, it is recommended to provide performance values for the area under the ROC curve (AUC) and one of the "early recognition" methods (enrichment factor (EF) or Boltzmann-enhanced discrimination of ROC (BEDROC), chosen here).

To evaluate the effectiveness of a virtual screening method, several performance measures are used.

AUC (Area Under the ROC Curve)

- ROC (Receiver Operating Characteristic): This is a curve that represents the performance of a classification model by plotting the true positive rate (sensitivity) against the false positive rate (1 - specificity) at different decision thresholds.
- AUC (Area Under the Curve): The area under the ROC curve is an overall measure of model performance. It ranges from 0 to 1:
  - An AUC of 1 indicates perfect classification.
  - An AUC of 0.5 indicates random classification.
  - An AUC of less than 0.5 means the model performs worse than random classification.
- Early Recognition Methods : These methods evaluate the ability of a model to quickly identify active molecules among a large number of candidates. This is crucial in virtual screening where the goal is to find promising compounds as early as possible.
- Enrichment Factor (EF) : The enrichment factor measures how many times better a model is than random at identifying active compounds among the top-ranked results. For example, if the model identifies 30% of active compounds in the top 10% of results, and random chance expects 10%, then the enrichment factor would be 3 (30% / 10%).

Boltzmann-Enhanced Discrimination of ROC (BEDROC)

- BEDROC is a modified version of ROC analysis that gives more weight to early-ranked results. This is particularly useful in virtual screening scenarios where it is important to find active molecules as early as possible.
- Parameter α: In BEDROC, the parameter α controls the importance given to early results. A higher α gives more weight to the early positions.

We compare our 3 methods (SED, FP, and GED) using AUC and BEDROC results. To analyze this statistically, we perform a Friedman test and a Wilcoxon signed-rank test. In the latter test, the smaller the value, the closer the two molecules in question are. For these tests, we compare, for each group of molecules, the ranking differences between two methods (for example, between SED and FP, then between SED and GED...).

Finally, GED seems to be the more efficient.

3 Research Paper No. 3 : Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs

Approximate Nearest Neighbor Search (K-NN): This is a method used to find the closest data points (neighbors) to a given point in a dataset. Instead of finding the exact neighbors, this method finds approximate neighbors, which is faster and often sufficient.

The HNSW method progressively builds a multi-layer structure. Each layer is a proximity graph representing subsets of stored elements. The highest level in which an element is present is selected randomly with an exponentially decreasing probability distribution. This means that elements are distributed in different layers based on their probability, creating graphs similar to previously studied NSW structures while separating links by their characteristic distance scales. This allows an efficient search : the search starts from the upper layer and uses scale separation to improve performance compared to NSW, allowing logarithmic search complexity. Using a heuristic to select neighbors in the proximity graph significantly increases performance, especially when aiming for high recall and in the case of heavily clustered data.

When adding an element, it is more likely to be added to the lowest layer. However, during the search, we start from the highest layer.

The goal is the search for the K nearest neighbors (K-Nearest Neighbor Search (K-NNS)). The naive way to proceed is too costly but provides an exact solution. Instead, we calculate an approximation of this solution (of Approximate Nearest Neighbors Search (K-ANNS)). To measure the efficiency of our approximation, we calculate the number of correct neighbors found divided by K. In this article, we propose the Hierarchical Navigable Small World (Hierarchical NSW, HNSW), a new incremental structure for approximate nearest neighbor search (K-ANNS).

The greedy algorithm involves choosing a random vertex, analyzing its neighbors, then choosing the closest neighbor. We then take this neighbor and search among its neighbors for the closest to our initial query, and so on... We stop with a stopping condition (e.g., the number of distance calculations performed). This allows constructing a k-NN graph (k-nearest neighbors), where each node (or point) is connected to the k closest neighbors based on the stored points' distances.

A Delaunay graph is a type of graph that ensures for a set of points, if a greedy search algorithm is used to find the nearest neighbor, the actual nearest neighbor will always be found. However, constructing a Delaunay graph efficiently requires prior information about the structure of the space in which the points are located, which can be complicated and costly in terms of computation. Therefore, we can instead construct a k-NN graph, providing a good approximation of the Delaunay graph (if we have the k closest neighbors approximately, we can roughly determine which is the nearest neighbor of a vertex, or at least one that is likely to be, quite efficiently).

Navigable Small World (NSW): Navigable graphs are graphs where the number of hops needed during greedy traversal increases logarithmically or polylogarithmically with the network size. This means that even if the graph becomes very large, the number of hops needed to find a close neighbor remains relatively low. The NSW graph is constructed by sequentially inserting elements in a random order. Each new element is bidirectionally connected to the M closest neighbors among the already inserted elements. To find the M closest neighbors, the algorithm uses a search procedure in the graph structure, which is a variant of the greedy search starting from several random entry nodes. This construction creates bridges between network hubs : the links to the closest neighbors of the elements inserted at the beginning of the construction later become bridges between the network hubs. These bridges maintain the overall connectivity of the graph and allow the logarithmic scaling of hops during greedy routing. Several parts of the construction can be executed simultaneously without requiring global synchronization, allowing time savings. This remains as accurate and is a very good option for distributed search systems (multiple processors, for example).

Kleinberg Spatial Models: Regular grid graph in a d-dimensional vector space with long-range links following a specific distribution $r^{-\alpha}$. For $\alpha = d$, the number of hops to reach the target by greedy routing scales polylogarithmically. The limitations are that it is necessary to know the data distribution in advance. Moreover, greedy routing in Kleinberg graphs suffers from polylogarithmic complexity. Power-law search complexity is even worse and requires global knowledge of the data distribution. In comparison, the NSW algorithm uses a simpler model of navigable networks, allowing decentralized graph construction and adaptation to data in arbitrary spaces. It has been suggested that the NSW network formation mechanism could be responsible for the navigability of large-scale biological neural networks. However, the NSW model also suffers from polylogarithmic search complexity.

There are two phases in finding a minimum:

- Zoom-out phase: starting from a low-degree node and moving by gradually increasing the degree of the visited nodes until the characteristic length scale of the node links reaches the scale of the distance to the query.

- Zoom-in phase: focusing and refining the search until the exact target is found.

- Problem: getting stuck in a local minimum. It is useful to have long-range links. How to solve this problem? Start with a high-degree node, thus directly beginning the zoom-in phase. However, evolution will still be of polylogarithmic complexity at best for an individual greedy search and offers lower performance on high-dimensional data compared to the Hierarchical NSW algorithm.

The total number of distance calculations is approximately proportional to the product of the average number of hops in the greedy path and the average degree of the nodes on this path. These two elements of the product increase logarithmically with network size because we tend to go through the hubs, and the number of hubs increases logarithmically.

The idea is to separate the links into different layers according to their length. We place the longest links in the highest layer (which will logically contain fewer links): this top layer corresponds to the zoom-in phase. Once we have found the local minimum of a layer, we move to the layer below.

Formation of a Layered Structure: Links are explicitly defined with different length scales by introducing layers. For each element, an integer level l is selected, defining the highest layer to which the element belongs. For all elements in a layer, a proximity graph is constructed incrementally. This graph contains only "short" links that approximately form a Delaunay graph. By defining an exponentially decreasing probability for l (following a geometric distribution), we obtain a logarithmic scale of the expected number of layers in the structure.

If we merge the connections of all layers in Hierarchical NSW, the resulting structure becomes similar to the standard NSW graph. In this case, the level of an element in Hierarchical NSW would correspond to the degree of nodes in NSW.

In the NSW algorithm, to ensure a good distribution of connections and avoid biased structures, it is necessary to shuffle the elements before inserting them into the graph. For Hierarchical NSW, randomization is introduced through the random selection of levels for each element during their insertion (each element receives a random level determining the highest layer to which it will belong). Thanks to the randomization of levels, Hierarchical NSW allows for truly incremental indexing (elements can be added as they arrive, without needing to reorganize or shuffle existing elements).

The idea of Hierarchical NSW is very similar to the probabilistic skip list structure in 1D (several levels of linked lists), but in Hierarchical NSW, linked lists are replaced by proximity graphs. When inserting an element, connections are created with other nearby elements: a connection is created with an element only if it is closer than the elements already connected to the base element (the inserted element).

Heuristic fot the selection of neighbors: When there are enough candidates (potential nodes), the heuristic used allows creating an exact relative neighborhood graph. A relative neighborhood graph is a minimal subgraph of the Delaunay graph, constructed only from the distances between the nodes. This graph easily maintains a global connected component, even if the data is heavily clustered. For example, if you have multiple clusters of data, this graph helps ensure all clusters remain connected. The heuristic creates additional edges compared to exact relative neighborhood graphs. This allows controlling the number of connections, which is crucial for search performance. For one-dimensional (1D) data, it allows creating exactly a Delaunay subgraph. In the 1D case, the Delaunay subgraph coincides with the relative neighborhood graph. This means we can directly transition from the Hierarchical NSW structure to the probabilistic skip list algorithm in 1D using only the distances between the elements.

The construction parameters of the algorithm m_L (nomalization factor for level generation) and M_{max0} are responsible for maintaining the navigability of the small world in the constructed graphs. Setting m_L to zero (which corresponds to a single layer in the graph) and M_{max0} to M leads to the production of directed k-NN graphs with power-law search complexity (assuming the use of Algorithm 3 for neighbor selection). Setting m_L to zero and M_{max0} to infinity leads to the production of NSW graphs with polylogarithmic complexity. Finally, setting m_L to a non-zero value leads to the emergence of controllable hierarchical graphs that allow logarithmic search complexity through the introduction of layers.

To maximize the hierarchy, we need to minimize the overlap of neighbors between different levels (i.e., the number of neighbors that belong to other levels as well). Therefore, to do this, m_L must be reduced, but this also increases the average number of hops during a greedy search, hence the existence of an optimal value. A simple choice for the optimal is $m_L = 1/\ln(M)$.

A small note: Recall is a measure used in machine learning to evaluate the performance of a classification or search model. It represents the model's ability to retrieve all relevant elements in a dataset. Specifically, recall is the number of relevant elements retrieved among all the actual relevant elements in the dataset.

For complexity, through calculations and the approximation of the Delaunay graph, we find $O(\log(N))$, where N is the number of elements in the graph. However, even if the Delaunay graph is not exact, the algorithm still reliably retrieves relevant elements in the data. It is only for large dimensions that problems arise. For the complexity during element insertion, we find $O(N \log(N))$. For memory complexity, each element costs at most $M_{max0} + m_L M_{max}$ bytes to store.