A probabilistic molecular fingerprint for big data settings

Daniel Probst\textsuperscript{1,*} and Jean-Louis Reymond\textsuperscript{1,*}

\textsuperscript{1}Department of Chemistry and Biochemistry, National Center for Competence in Research NCCR TransCure, University of Berne, Freiestrasse 3, 3012 Berne, Switzerland

*e-mail: daniel.probst@dcb.unibe.ch, jean-louis.reymond@dcb.unibe.ch

Abstract

Background: Among the various molecular fingerprints available to describe small organic molecules, ECFP4 (extended connectivity fingerprint, up to four bonds) performs best in benchmarking drug analog recovery studies as it encodes substructures with a high level of detail. Unfortunately, ECFP4 requires high dimensional representations ($\geq1,024$D) to perform well, resulting in ECFP4 nearest neighbor searches in very large databases such as GDB, PubChem or ZINC to perform very slowly due to the curse of dimensionality.

Results: Herein we report a new fingerprint, called MHFP6 (MinHash fingerprint, up to six bonds), which encodes detailed substructures using the extended connectivity principle of ECFP in a fundamentally different manner, increasing the performance of exact nearest neighbor searches in benchmarking studies and enabling the application of locality sensitive hashing (LSH) approximate nearest neighbor search algorithms. To describe a molecule, MHFP6 extracts the SMILES of all circular substructures around each atom up to a diameter of six bonds and applies the MinHash method to the resulting set. MHFP6 outperforms ECFP4 in benchmarking analog recovery studies. Furthermore, MHFP6 outperforms ECFP4 in approximate nearest neighbor searches by two orders of magnitude in terms of speed, while decreasing the error rate.

Conclusion: MHFP6 is a new molecular fingerprint, encoding circular substructures, which outperforms ECFP4 for analog searches while allowing the direct application of locality sensitive hashing algorithms. It should be well suited for the analysis of large databases. The source code for MHFP6 is available on GitHub (https://github.com/reymond-group/mhfp).
**Keywords:** Virtual Screening, Similarity Search, Fingerprints, Locality Sensitive Hashing, Approximate k-Nearest Neighbor Search

**Introduction**

Many, if not all uses of cheminformatics require the quantification of the similarity between molecules. As the underlying data structure used to represent molecules is a graph, this problem is equivalent to a subgraph isomerism problem, which is at least NP-complete [1]. Molecular fingerprints reduce this problem to the comparison of vectors, enabling further application of approximation methods and heuristics, thus speeding up the computation [2–5].

Among the assortment of fingerprints for the comparison of molecules in use today, ECFP (extended connectivity fingerprint) is the most prominent due to its outstanding performance in molecular structure comparisons requiring the identification of compounds with similar bioactivity, as assessed in benchmarking studies [6, 7]. However, the performance of ECFP results from a precise encoding of molecular structure, which is achieved by using high-dimensional vectors, typically \( d \geq 1024 \), with the consequence that similarity searches become prohibitively slow when applied to very large databases such as GDB, PubChem or ZINC [8–13]. This problem occurs even when using optimized search algorithms such as k-d or ball trees because their performance degrades to linear time due to the curse of dimensionality [7, 14, 15]. In addition, given the often binary, sparse, and high dimensional nature of ECFP, \( L^p \) metrics generally perform badly, further limiting the number of available optimization techniques. In our effort to facilitate the exploration of very large databases such as GDB, we previously used lower dimensionality fingerprints such as MQN (Molecular Quantum Number, 42D) or SMIfp (SMILES fingerprint, 34D) for similarity searches, however, such fingerprints only encode molecular composition and do not allow precise structural similarity calculation [16–18].
Herein we report a new family of fingerprints termed MHFP (MinHash fingerprint) which combine the circular nature of ECFP with w-shingling and MinHash, which are encoding and comparison methods used in natural language processing and text mining [19–21] (Figure 1). These methods are commonly used in applications such as discarding already indexed web pages during web-crawling, signal processing or plagiarism detection [22–24]. We obtain our MHFP by first writing out circular substructures around each atom as SMILES, a process which we call molecular shingling in analogy to the w-shingling scheme used for the above-mentioned text mining applications. We then apply the MinHash hashing scheme to assign these SMILES to bit values in our MHFP.

MinHash is a locality sensitive hashing (LSH) scheme which applies a family of hashing functions to the substrings in a molecular shingling and stores the minimum hash generated from each hashing function in a set. These sets, containing the minimum hash values, have the interesting property that they can be indexed by an LSH algorithm for approximate nearest neighbor search (ANN), removing the curse of dimensionality [25]. Note that ECFP similarity cannot be computed using LSH due to the nature of the primary hashing scheme used to assign circular substructures to bit values. Furthermore, ECFP encode circular substructures by iteratively hashing atomic invariants, thereby making assumptions regarding the importance of atomic features such as acidity or charge, introducing a potential bias which is entirely avoided in MHFP [6, 26, 27].

To assess the performance of MHFP we compare it to variants of ECFP as well as to a hybrid fingerprint MHECFP which applies MinHash to unfolded ECFP hashes. We find that the performance of MHFP surpasses that of ECFP and MHECFP in a ligand-based virtual screening benchmark [7]. Furthermore, MHFP similarity searches by ANN using the LSH Forest algorithm are 100-fold faster yet have a better accuracy than ECFP nearest neighbor search based on a k-d tree [14, 28]. MHFP6, encoding substructures up to a diameter of 6 bonds, performs best and should be considered as replacement for ECFP4 to speed up
searches in very large databases. The source code for MHFP is available on GitHub (https://github.com/reymond-group/mhfp).

Figure 1 MHFP, ECFP workflow comparison. Comparison of hashing and approximate nearest neighbor search indexing of ECFP with a k-d tree (gray) and MHFP via molecular shingling and MinHash with LSH Forest (orange). In addition, MinHash is applied to unfolded ECFP hashes and indexed using LSH Forest as well (green), resulting in the hybrid fingerprint MHECFP. The latter was used as a control to separate the influences of molecular shingling and applying MinHash on the measured performance.

Methods

Jaccard similarity: The Jaccard similarity is also referred to as Jaccard index, Jaccard similarity coefficient or Tanimoto index. Given two sets $A$ and $B$, the Jaccard similarity coefficient of the molecules is calculated as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$  \hspace{1cm} (1)

The Jaccard distance is a metric defined as $1 - J(A, B)$ [29]. Both, the Jaccard similarity coefficient and distance have been shown to be appropriate for fingerprint-based similarity calculations [30].
**MinHash:** MinHash is used to estimate the Jaccard similarity between two sets [19]. Given sets of integers, such as hash values, MinHash is applied as follows:

Let \( a \) and \( b \) be \( k \)-dimensional vectors with elements set to unique randomly generated integers such that \( a_i, b_i \in \{0, ..., 2^{32} - 1\} \) and let \( H \) be the set of all hash values \( \{0, ..., 2^{32} - 1\} \). Given a family of sets \( F = \{S_1, ..., S_n\} \) over \( H \) where each set represents a molecule, the MinHash function \( h_{\text{min}}(S_i, a, b) \) is applied to each set \( S_i \) in \( F \). Let \( s \) be the vector form of a set \( S \) from \( F \) and \( p \) be the Mersenne prime \( 2^{61} - 1 \). The MinHash of a molecular graph is then calculated as:

\[
h_{\text{min}}(s_i, a, b) = \min \left( \left( (a \cdot s_i^T + b) \mod p \right) \mod (2^{32} - 1) \right)
\]

The set form \( S_{\text{min}} \) of \( s_{\text{min}} \) can then be used to estimate the Jaccard similarity coefficient of two sets \( S_i, S_j \) using Equation 1 [31].

The expected error of estimating the Jaccard similarity coefficient between two sets using MinHash is \( O \left( \frac{1}{\log(n)} \right) \), where \( n \) is the number of hash functions used [32].

**LSH Forest:** The local sensitivity hashing (LSH) forest algorithm is an extension to LSH similarity indexing [28, 33]. Introducing self-tuning indices, the algorithm renders data-dependent manual parameter tuning superfluous by storing the hashes in multiple prefix-trees that make up the LSH Forest.

**Estimate number of hash collisions:** As hash functions for strings are non-injective, so-called hash collisions occur when two or more non-identical strings are being hashed to an identical integer. The number of hash collisions can be estimated through a generalization of the birthday problem [34]:

\[
c(k, N) = k - N \left( 1 - \left( \frac{N - 1}{N} \right)^k \right)
\]

where \( k \) is the number of hashed values and \( N \) is the maximum hash value.
**Statistical methods:** The confidence level $\alpha$ is 0.05 for both the independent (unpaired) $t$-tests and the pairwise post-hoc Friedman tests. The independent (unpaired) $t$-tests are computed using SciPy (1.1.0), the pairwise post-hoc Friedman tests are part of the open-source platform to benchmark fingerprints for ligand-based virtual screening [7].

**Python implementation:** The methods for generating molecular shinglings and computing the MinHash values described above were implemented in a Python (3.6.3) script that takes a SMILES string as an input and returns a NumPy (1.15.1) array of hashes, describing the molecule [35]. The cheminformatics library RDKit (2017_09_1) was used to parse the SMILES and extract substructures form the molecular graph [26]. In order to evaluate the performance of MHFP in combination with LSH-based methods, a Python script implementing the locality sensitive hashing (LSH) forest algorithm for $k$-nearest neighbor searching according to the datasketch Python library was written [28, 36, 37]. The LSH Forest script returns the approximate $k$-nearest neighbors of a query compound encoded as an MHFP fingerprint. In order to compensate for approximation errors, $k_c \cdot k$ neighbors are searched for internally and their actual distance to the query molecule is computed using linear scan. $k_c$ is supplied as an optional parameter that defaults to $k_c = 10$. After this intermediate step, the top $k$ hits are then returned as the result of the LSH Forest query. Both scripts are available on GitHub (https://github.com/reymond-group/mhfp).
Results and Discussion

Fingerprint design

The MinHash fingerprint (MHFP) described herein combines the concept of extended connectivity used for ECFP with MinHash as a hashing scheme to later enable LSH-based ANN searches. As a first step, we enumerate all circular substructures around each atom in a molecule and write these out as SMILES [6]. This operation yields $O(n(r+1))$ SMILES strings for a molecule with a heavy atom count (HAC) of $n$ and a maximum radius $r$. As for either small radii $r$ or macrocycles the ring information of a molecule is lost, we also extract the SMILES string for each ring of the smallest set of smallest rings in the molecule. We then filter the SMILES strings for duplicates and combine them to a set $S(A)$ representing the molecular shingling of the molecule $A$.

We denote the process described above as “shingling of a molecule” and the resulting set $S(A)$ as “molecular shingling”. A molecular shingling differs from the w-shingling of a document, where a w-shingling consists of n-grams with $n = w$, in that it includes SMILES strings of different lengths, with the maximum length depending on the maximum radius $r$ and the size of the rings in the molecule. The number of unique SMILES-encoded molecular subgraphs with radius $r$ grows logarithmically and at a slower rate than ECFP hashes with radius $r$ when processing 1.7 million compounds from ChEMBL24, probably because SMILES do not encode detailed atomic properties as is done in ECFP encoding (Figure S1) [38]. Given the molecular shinglings $S(M_a)$ and $S(M_b)$ of two molecules $M_a$ and $M_b$, the Jaccard similarity coefficient of the molecules is calculated according to Equation 1 (see Methods).

As the MinHash scheme cannot be applied directly to strings, the SMILES in a molecular shingling are first hashed to a 32-bit unsigned integer using a function $f: \Omega \rightarrow$
There is a trade-off when choosing this relatively small 32-bit hash, as the number of collisions (two or more different strings being hashed to the same integer value) during hashing is inversely proportional to the length of the hash. To estimate the number of collisions, molecular shingles with \( r = 2 \) were extracted from 1.7 million ChEMBL24 compounds, yielding a total number of 197,604 unique SMILES. Applying Equation 3 (see Methods), the number of expected collisions yields \( c(k = 197,604, N = 2^{32} - 1) = 4.546 \). Increasing the maximum radius to \( r = 3 \) results in an increase to 2,022,448 unique SMILES and 476.098 expected collisions. The measured numbers of collisions when hashing molecular shinglings from ChEMBL24 were 3 and 481 for \( r = 2 \) and \( r = 3 \), respectively, proving Equation 3 to be a good estimator for SMILES hashing collisions. Substituting the 32-bit (SHA-1) hash with a 64-bit (SHA-1) hash would lower the number of estimated collisions to 0. However, a 64-bit hash would lead to numbers larger than a 64-bit integer during MinHash computation. Thus, SMILES contained within molecular shinglings are hashed to a 32-bit (SHA-1) hash.

To transform the hashed molecular shingling into our final fingerprints, we finally apply MinHash according to Equation 2 (see Methods). In the present study we calculated MinHash fingerprints for hashed molecular shinglings with \( r \in \{2, 3, 4\} \) and \( k \in \{128, 1024, 2048, 4096\} \). We considered radii \( r = 2 \) (MHFP4), \( r = 3 \) (MHFP6), and \( r = 4 \) (MHFP8), resulting in 12 fingerprints with different level of structural encoding and compression (according to common notation, the numbers in the fingerprint names represent the maximum diameter rather than the maximum radius).

**Benchmarking study**

To validate the SMILES-strings based approach as well as the chosen hash function, we used a platform to benchmark fingerprints for ligand-based virtual screening with Jaccard similarity as a metric [7]. The benchmark performs statistically valid comparisons of
fingerprints using structural and activity data drawn from DUD [39], MUV [40] and ChEMBL [41]. The benchmark evaluates 7 metrics: The area under the receiver operating characteristic (ROC) curve (AUC), the enrichment factor (EF) for $\chi = 0.01$ and $\chi = 0.05$, the Boltzmann-enhanced discrimination of ROC (BEDROC) for $\alpha = 20$ and $\alpha = 100$, and the robust initial enhancement (RIE) for $\alpha = 20$ and $\alpha = 100$.

First, we compared the hashed molecular shinglings to ECFP hashes before folding, as well as to ECFP*, a variant of ECFP considering only atomic numbers as invariants, all with $r = 2$ and $r = 3$. This comparison showed that the hashed molecular shingling method with a radius of $r = 3$ is superior to ECFP hashing, as it beats unfolded ECFP (with either radius $r = 2$ or $r = 3$) significantly in 2 out of 7 values (AUC, EF 5%) and with a p-value above 0.05 in 5 out of 7 (EF 1%, BEDROC20, BEDROC100, RIE20 and RIE100) metrics (Figure 2, Figure S5). ECFP* performed significantly worse with both $r = 2$ and $r = 3$ in all metrics compared to molecular shingling with $r = 3$.

![Figure 2 Results of benchmarking hashing methods across 88 benchmark targets.](image)

**Figure 2 Results of benchmarking hashing methods across 88 benchmark targets.** Hashed molecular shingling with $r = 2$ (orange, solid) and $r = 3$ (orange, dashed) are both ranked better than ECFP4/6 (green) and ECFP4/6* (purple) in AUC. However, only hashed molecular shingling with $r = 3$ was ranked better than all other fingerprints in every metric (AUC, EF1, EF5, BEDROC20, BEDROC100, RIE20, and RIE100). The control, a variant of ECFP, ECFP* (purple), considering only atomic numbers as invariants, performed significantly worse than both hashed molecular shingling and ECFP. Pairwise post-hoc Friedman tests of the average rank were performed as part of the benchmark, resulting p-Values shown in Figure S5.
To establish whether results based on evaluating hashed molecular shinglings carry over to minhashed molecular shinglings, we then compared our 12 different MHFPs variants with each other. Comparing these different fingerprints in the benchmark confirmed that MHFP6 (MinHash applied to hashed molecular shinglings with \( r = 3 \)) performed better than both MHFP4 (\( r = 2 \)) and MHFP8 (\( r = 4 \)) for medium (1,024-D, 2,048-D) to high dimensional (4,096-D) variants (Figure 3). The data further suggested that low dimensional variants such as 128-D perform better with \( r = 2 \). As MHFP8 failed to perform better than MHFP6, it was discounted from further experiments. MHFP4, while also performing worse than MHFP6, was kept for further experiments as a comparison to ECFP variants with \( r = 2 \).

![Figure 3 Average ranks of MHFP variants across 88 benchmark targets](image)

**Figure 3 Average ranks of MHFP variants across 88 benchmark targets.** Performance comparison of MHFP variants MHFP4/6/8 across dimensionalities 128-D, 1,024-D, 2,048-D, and 4,096-D. While performance increases with an increase of the radius from \( r = 2 \) to \( r = 3 \), a further increase of the radius to \( r = 4 \) does not translate to further performance gains but a decrease, especially in BEDROC20, BEDROC100, RIE20 and RIE100 rankings. The benchmark used was a platform to benchmark fingerprints for ligand-based virtual screening with Jaccard similarity as a metric [7].

Given the results of benchmarking unfolded ECFP hashes and hashed molecular shinglings (Figure 2), as well as the results of benchmarking different MHFP radii (Figure 3), we finally selected the following fingerprints for a detailed comparison aimed at identifying the best fingerprint: (1) Folded ECFP4 and ECFP6; (2) MinHash molecular shinglings with radii 2
and 3, henceforth denoted MHFP4 and MHFP6; (3) MinHash ECFP4 and ECFP6, henceforth denoted MHECFP4 and MHECFP6, respectively, used here to control for the performance of encoding SMILES (MHFP) as opposed to hashes of invariants (ECFP) by applying the minhashing scheme to unfolded ECFP values (Figure 1). For each fingerprint four different dimensionalities were evaluated.

An average rank comparison according to the benchmark is shown in Figure 4. Comparing the average ranking of the fingerprints as a function of the chosen radius, both ECFP4 and MHECFP4 perform marginally better than their respective counterparts, ECFP6 and MHECFP6, in the vast majority of cases. In contrast, MHFP6 generally performs better than MHFP4. This result confirms the observations from Figure 2 where hashed molecular shinglings performed better with $r = 3$ than with $r = 2$, while the ECFP4 hashes outperformed ECFP6 hashes. With the exception of the 128-D variant, MHFP4/6 exhibit strictly better performance in AUC compared to both MHECFP4/6 and ECFP4/6, while both MHFP4/6 and MHECFP4/6 perform better than ECFP4/6 in early recognition metrics EF1 and EF5, suggesting that the AUC performance gains are a result of the molecular shingling approach, while the gains in early recognition can be attributed to minhashing. Note that MHFP6 (both 2,048-D and 4,096-D) did not perform significantly worse than path-based methods (TT and AP, ref. [7]) in AUC, while performing generally significantly better in other metrics, which is in contrast to ECFP fingerprints, which perform worse in AUC benchmarks than path based fingerprints (Figure S4).

The above comparisons established that MHFP6 provided the best overall performance across all fingerprints considered, with the 2,048-D offering a good compromise between performance and size. In detail, 2,048-D MHFP6 significantly outperformed 2,048-D ECFP4 in AUC, EF1 and EF5, while performing non-significantly better in BEDROC20, BEDROC100, RIE20 and RIE100. In fact, 2,048-D MHFP6 was comparable to 16,384-D ECFP4, although it still performed better in terms of BEDROC20 and RIE20. 2,048-D
MHFP6 also performed significantly better in AUC than 2,048-D MHECFP4 while non-significantly better in EF1, EF5, BEDROC100 and RIE100 and worse in BEDROC20 and BEDROC100. While 2,048-D MHFP6 ranked significantly worse than 4,096-D MHECFP4 in AUC, 4,096-D MHFP6 significantly outranked 4,096-D MHECFP4 in AUC (Figures S6). Further analysis of the data suggested that gains by MHFP6 over ECFP4 was largely due to better performance on benchmark targets selected from ChEMBL24, while performing approximately equal on DUD and MUV data (Figure 5, see full target-level performance comparisons between 2,048-D MHFP6 and 2,048-D ECFP4 and MHECFP4 in Figures S2 and S3, respectively).

![Figure 4](https://example.com/figure4.png)

**Figure 4** Average ranks of ECFP4/6, MHECFP4/6 and MHFP4/6 across 88 benchmark targets (REF 7 here). The benchmark was run for a total of 24 fingerprint variants. MHFP6 generally outperforms MHFP4, while ECFP4 and MHECFP4 are always ranked equal or better than ECFP6 and MHECFP6, respectively. MHFP6 matches or outperforms ECFP4/6 and MHECFP4/6 in virtually all metrics across benchmarked dimensionalities (pairwise post-hoc Friedman tests of the average rank were performed as part of the benchmark, resulting p-Values in Figure S6). (*) The 4,096-D variants of MHECFP4/6 and MHFP4/6 were compared to the 16,384-D variant of ECFP4/6 as this is the highest reported dimensionality applied with ECFP.
Figure 5 Performance comparison between MHFP6 2,048-D and ECFP4 2,048-D. Colors highlighting the difference in the AUC, EF1 and BEDROC20 values for 88 targets between MHFP6 2,048-D (orange) and ECFP4 2,048-D (green). MHFP6 significantly outperforms MHECFP4 in the AUC and EF1 metrics (see pairwise post-hoc Friedman tests of the average rank results in Figure S6a). Comparisons to EF5, BEDROC100, RIE20 and RIE100 can be found in Figure S2, comparisons of all metrics between MHFP6 and MHECFP4 (both 2,048-D) in Figure S3.

Approximate k-nearest neighbor (ANN) searches

In the context of big data, the key advantage of our MHFP over ECFP consists in the implementation of MinHash, which enables the use of the LSH Forest algorithm to perform ANN, while ECFP can only be searched using spatial indexing methods, such as k-d tree. To quantify the advantage gained by MinHash, we compared the performance of ANN with 2,048-D MHFP6 and MHECFP4 by LSH Forest to that of 2,048-D ECFP4 using a k-d tree [14, 28]. A benchmark was set up, consisting of 20 randomly selected target sets from DUD-E [42]. From each active set, 20 actives were randomly selected, resulting in a total of 400 query compounds. Next, the active and decoy sets for each target were merged. For each of the 400 query compounds, the Jaccard indices to all compounds from their respective combined sets were calculated using brute-force linear scan, resulting in 400 sorted lists. These steps were
performed for MHFP6, MHECFP4, and ECFP4 based Jaccard indices. Finally, the recovery rates of \(k\)-nearest neighbors for \(k \in \{5, 10, 50, 100\}\) of approximate \(k\)-nearest neighbor algorithms (LSH Forest for MHFP6 and MHECFP4, and k-d tree for ECFP4) were calculated. The parameter \(k_c\) for the LSH Forest \(k\)-nearest neighbor searches was set to 10.

The MHFP6 and MHECFP4-based LSH Forest algorithms were benchmarked with 8, 16, 32, 64, 128, 256, and 512 prefix trees. As expected, a greater number of prefix trees led to a higher recovery rate, a shorter search time, and longer indexing / build time (Figure 6a, c, d). While the number of prefix trees \(l\) had a relatively small influence on search time (Figure 5d), it mainly impacted the recovery rate (Figure 6a), overtaking k-d tree with \(l = 16\) for MHFP6 and \(l = 32\) for MHECFP4. In terms of search time, LSH Forest outperformed k-d tree by two orders of magnitude for all \(l\) (Figure 6d). Interestingly, k-d tree recovery rates decreased with increasing \(k\), while the combined LSH Forest recovery rates remained stable (Figure 6b). Comparing the performance of the LSH Forest algorithms based on MHFP6 and MHECFP4, MHECFP4 expressed similar degrading recovery rates with \(k = 50\) and \(k = 100\) (Figure 6b). In terms of recovery rate as a function of the number of prefix trees \(l\), MHFP6 based LSH Forest generally performed better than the MHECFP4-based variant (Figure 6a).

Additional calculations were run to confirm the measured results using the ChEMBL24 compound database as a data source. From ChEMBL24, 20 subsets of size 100,000 were randomly exported. From each subset 20 compounds were randomly selected as the query compound. Performance data was gathered analogous to the previously described benchmark. The results confirmed the observations from the DUD-E based benchmark (Figures S8 and S9, see also Table S1 for associated p-Values and \(t\) statistics for all benchmarks).
Figure 6 Performance analysis of MHFP6-based k-nearest neighbor search ($k_c = 10$). Comparison of k-nearest neighbor search performance between MHFP6 with LSH Forest (orange), MHECFP4 with LSH forest (green), and ECFP4 with k-d tree (purple). All fingerprints were calculated with a dimensionality of 2,048. (a) MHFP6 and MHECFP4 equalize and surpass ECFP4 recovery rate with more than 16 and 32 prefix trees, respectively. (b) MHFP6 significantly outperforms both MHECFP4 and ECFP4 with $k = 50$ and $k = 100$. (c) The build / indexing time for MHFP6 and MHECFP4 increases with the number of prefix trees. (d) MHFP6 and MHECFP4 outperform ECFP4 by two orders of magnitude in terms of search time. Search time drops with increasing number of prefix trees. t-test results and p-values can be found in Table S1.

The results gathered in the DUD-E and ChEMBL benchmarks illustrate the impact of the chosen number of prefix trees $l$ as well as the number of approximate nearest neighbors $k_c \cdot k$ to be retrieved from which the $k$ nearest neighbors are then selected using linear scan (Figures 6, S8, and S9). Both search speed and search quality can be tuned by increasing or decreasing these two parameters. In order to analyze the interaction of these parameters and their impact on search speed and recovery rate, the ChEMBL24 benchmark was rerun and the parameter $k_c$ was varied from 10 to 100 by steps of 10. Additionally, in order to control for the influence of data set size on speed and recovery rate, 20 smaller subsets of size 10,000 were created. According to our results (Figure 7 and S10), data set size and composition both influence the speed and recovery rate of the algorithm, as size alone fails to account for changes in performance given the observed data. The suggested effect of data set composition
is especially visible in $k$-nearest neighbor searches for $k = 5$ and $k = 10$ where set size and recovery rate do not seem to correlate for lower values of $k_c$ and $l$ (Figure S10).

However, overall, the size of the data set is poised to be the principal influence on both recovery rate and to a lesser degree search speed, especially in combination with high values for $k$, $k_c$, and $l$. While increasing the number of prefix trees $l$ increases recovery rate as well as search speed, increasing parameter $k_c$ increases recovery rate and decreases search speed marginally (Figures 6, 7, S8, S9 and S10).

Figure 7 Performance as a function of parameter $k_c$. A recovery study was repeated twice for 20 subsets containing 10,000 (a, c) and 100,000 (b, d) compounds, respectively. The performance of our implementation of the LSH Forest algorithm depends on the parameters $k_c$ and $l$. While speed is only marginally affected by changes in both parameters, the recovery rate rises with the increase of both values, approaching a perfect recovery rate. Data set size and composition both influence the speed and recovery rate of the algorithm, as size alone fails to account for changes in performance.
Conclusion

MHFP6 is a new fingerprint based on the circular nature of ECFP combined with methods from natural language processing and data mining. The data presented here and the earlier benchmark study [7] demonstrate that MHFP6 performs better than any currently used fingerprint in a ligand-based virtual screening. Furthermore, MHFP6 has the potential to speed up $k$-nearest neighbor searches by orders of magnitude by enabling the direct application of ANN algorithms such as LSH Forest, thereby successfully removing the curse of dimensionality under which spatial indexing methods, such as k-d trees, suffer. In addition to improving $k$-nearest neighbor search speed by two orders of magnitude, LSH Forest, in combination with MHFP6, also significantly increases search accuracy compared to spatial indexing methods for ANN. The remarkable performance of MHFP6 makes the new fingerprint a highly recommended alternative to ECFP4 for virtual screening experiments as well as for querying and analyzing large chemical databases. The source code for MHFP6 is available on GitHub (https://github.com/reymond-group/mhfp).

Acknowledgments. This work was supported financially by the Swiss National Science Foundation, NCCR TransCure.

Conflict of interest statement. The authors declare no conflict of interest.

Data and material availability. Data and materials are available on GitHub (https://github.com/reymond-group/mhfp)

Author contributions. DP designed and realized the study and wrote the paper. JLR supervised the study and wrote the paper.
| Abbreviation | Description |
|--------------|-------------|
| ANN          | Approximate Nearest Neighbor |
| AP           | Atom Pair fingerprint |
| AUC          | Area Under the Curve |
| BEDROC       | Boltzmann-Enhanced Discrimination of the Receiver Operating Characteristic |
| DUD          | Directory of Useful Decoys |
| DUD-E        | Directory of Useful Decoys Enhanced |
| ECFP         | Extended Connectivity Fingerprint |
| EF           | Enrichment Factor |
| GDB          | Generated DataBase |
| HAC          | Heavy Atom Count |
| LSH          | Locality Sensitive Hashing |
| MHECFP       | MinHash Extended Connectivity Fingerprint |
| MHFP         | MinHash Fingerprint |
| MQN          | Molecular Quantum Numbers |
| MUV          | Maximum Unbiased Validation Data Sets |
| RIE          | Robust Initial Enhancement |
| ROC          | Receiver Operating Characteristic |
| TT           | Topological Torsion fingerprint |
| SMIfp        | SMILES Fingerprint |
| SMILES       | Simplified molecular-input line-entry system |
| ZINC         | ZINC is not commercial |
References

1. Grohe M, Rattan G, Woeginger GJ (2018) Graph Similarity and Approximate Isomorphism. In: Graph Similarity and Approximate Isomorphism. Schloss Dagstuhl--Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp 20:1--20:16

2. Nilakantan R, Bauman N, Dixon JS, Venkataraman R (1987) Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors. J Chem Inf Comput Sci 27:82–85. https://doi.org/10.1021/ci00054a008

3. Morgan HL (1965) The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. Journal of Chemical Documentation 5:107–113. https://doi.org/10.1021/ci60017a018

4. Awale M, Reymond J-L (2014) Atom Pair 2D-Fingerprints Perceive 3D-Molecular Shape and Pharmacophores for Very Fast Virtual Screening of ZINC and GDB-17. J Chem Inf Model 54:1892–1907. https://doi.org/10.1021/ci500232g

5. Kpodjedo S, Galinier P, Antoniol G (2010) On the use of similarity metrics for approximate graph matching. Electronic Notes in Discrete Mathematics 36:687–694. https://doi.org/10.1016/j.endm.2010.05.087

6. Rogers D, Hahn M (2010) Extended-Connectivity Fingerprints. J Chem Inf Model 50:742–754. https://doi.org/10.1021/ci100050t

7. Riniker S, Landrum GA (2013) Open-source platform to benchmark fingerprints for ligand-based virtual screening. Journal of Cheminformatics 5:26. https://doi.org/10.1186/1758-2946-5-26

8. Cereto-Massagué A, Ojeda MJ, Valls C, et al (2015) Molecular fingerprint similarity search in virtual screening. Methods 71:58–63. https://doi.org/10.1016/j.ymeth.2014.08.005

9. Gütlein M, Kramer S (2016) Filtered circular fingerprints improve either prediction or runtime performance while retaining interpretability. Journal of Cheminformatics 8:. https://doi.org/10.1186/s13321-016-0173-z

10. Muegge I, Mukherjee P (2016) An overview of molecular fingerprint similarity search in virtual screening. Expert Opinion on Drug Discovery 11:137–148. https://doi.org/10.1517/17460441.2016.1117070

11. Irwin JJ, Shoichet BK (2005) ZINC – A Free Database of Commercially Available Compounds for Virtual Screening. J Chem Inf Model 45:177–182

12. Ruddigke L, van Deursen R, Blum LC, Reymond J-L (2012) Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. J Chem Inf Model 52:2864–2875. https://doi.org/10.1021/ci300415d

13. Wang Y, Xiao J, Suzek TO, et al (2009) PubChem: a public information system for analyzing bioactivities of small molecules. Nucleic Acids Res 37:W623–W633. https://doi.org/10.1093/nar/gkp456
14. Bentley JL (1975) Multidimensional binary search trees used for associative searching. Communications of the ACM 18:509–517. https://doi.org/10.1145/361002.361007

15. Weber R, Schek H-J, Blott S (1998) A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In: Proceedings of the 24rd International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 194–205

16. Schwartz J, Awale M, Reymond J-L (2013) SMIfp (SMILES fingerprint) Chemical Space for Virtual Screening and Visualization of Large Databases of Organic Molecules. J Chem Inf Model 53:1979–1989. https://doi.org/10.1021/ci400206h

17. Awale M, van Deursen R, Reymond J-L (2013) MQN-Mapplet: Visualization of Chemical Space with Interactive Maps of DrugBank, ChEMBL, PubChem, GDB-11, and GDB-13. J Chem Inf Model 53:509–518. https://doi.org/10.1021/ci300513m

18. Nguyen KT, Blum LC, van Deursen R, Reymond J-L (2009) Classification of Organic Molecules by Molecular Quantum Numbers. ChemMedChem 4:1803–1805. https://doi.org/10.1002/cmdc.200900317

19. Broder AZ (1998) On the resemblance and containment of documents. In: Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171). IEEE Comput. Soc, Salerno, Italy, pp 21–29

20. Manber U (1994) Finding Similar Files in a Large File System. In: Usenix Winter 1994 Technical Conference. pp 1–10

21. Damashek M (1995) Gauging Similarity with n-Grams: Language-Independent Categorization of Text. Science 267:843–848. https://doi.org/10.1126/science.267.5199.843

22. Brin S, Page L (1998) The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems 30:107–117. https://doi.org/10.1016/S0169-7552(98)00110-X

23. Hu S (2005) Efficient video retrieval by locality sensitive hashing. In: Proceedings. (ICASSP ’05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. p ii/449-ii/452 Vol. 2

24. Crussell J, Gibler C, Chen H (2013) AnDarwin: Scalable Detection of Semantically Similar Android Applications. In: Crampton J, Jajodia S, Mayes K (eds) Computer Security – ESORICS 2013. Springer Berlin Heidelberg, pp 182–199

25. Indyk P, Motwani R (1998) Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing. ACM, New York, NY, USA, pp 604–613

26. (2018) RDKit: Open-source cheminformatics. http://www.rdkit.org. Accessed 31 Aug 2018

27. (2018) ChemAxon JChem. http://www.chemaxon.com. Accessed 31 Aug 2018
28. Bawa M, Condie T, Ganesan P (2005) LSH forest: self-tuning indexes for similarity search. In: Proceedings of the 14th international conference on World Wide Web - WWW '05. ACM Press, Chiba, Japan, p 651

29. Kosub S (2016) A note on the triangle inequality for the Jaccard distance. arXiv:161202696 [cs, stat]

30. Bajusz D, Rácz A, Héberger K (2015) Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? Journal of Cheminformatics 7:. https://doi.org/10.1186/s13321-015-0069-3

31. Broder AZ, Charikar M, Frieze AM, Mitzenmacher M (1998) Min-wise Independent Permutations (Extended Abstract). In: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing. ACM, New York, NY, USA, pp 327–336

32. Christiani T, Pagh R (2016) Set Similarity Search Beyond MinHash. arXiv:161207710 [cs]

33. Atkinson MP, International Conference on Very Large Databases (1999) Very large data bases: proceedings of the Twenty-fifth International Conference on Very Large Data Bases, Edinburgh, Scotland, 7 - 10th September 1999. Morgan Kaufmann, Orlando, Fla

34. Mckinney EH (1966) Generalized Birthday Problem. The American Mathematical Monthly 73:385. https://doi.org/10.2307/2315408

35. Van Der Walt S, Colbert SC, Varoquaux G (2011) The NumPy array: a structure for efficient numerical computation. Computing in Science & Engineering 13:22–30. https://doi.org/10.1109/MCSE.2011.37

36. Zhu E, Markovtsev V (2017) datasketch: Big Data Looks Small

37. Andoni A, Razenshteyn I, Nosatzki NS (2017) LSH Forest: Practical Algorithms Made Theoretical. In: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, pp 67–78

38. Papadatos G, Davies M, Dedman N, et al (2016) SureChEMBL: a large-scale, chemically annotated patent document database. Nucleic Acids Res 44:D1220–D1228. https://doi.org/10.1093/nar/gkv1253

39. Irwin JJ (2008) Community benchmarks for virtual screening. Journal of Computer-Aided Molecular Design 22:193–199. https://doi.org/10.1007/s10822-008-9189-4

40. Rohrer SG, Baumann K (2009) Maximum Unbiased Validation (MUV) Data Sets for Virtual Screening Based on PubChem Bioactivity Data. Journal of Chemical Information and Modeling 49:169–184. https://doi.org/10.1021/ci8002649

41. Gaulton A, Bellis LJ, Bento AP, et al (2012) ChEMBL: a large-scale bioactivity database for drug discovery. Nucleic Acids Research 40:D1100–D1107. https://doi.org/10.1093/nar/gkr777

42. Mysinger MM, Carchia M, Irwin JJ, Shoichet BK (2012) Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. Journal of Medicinal Chemistry 55:6582–6594. https://doi.org/10.1021/jm300687e
Supplementary Information

Figure S1 Number of ECFP hashes and unique SMILES extracted from ChEMBL. For both radii $r = 2$ and $r = 3$, the number of unique SMILES extracted from ChEMBL (pink and orange) is lower than the number of ECFP hashes (purple and green). However, both approaches exhibit a similar pattern as can be seen by their deviations from the fitted curve between fractions 0.2 and 0.7. All four variants shown follow a logistic growth as a function of the fraction of the dataset processed.
Figure S2 Performance comparison between MHFP6 2,048-D and ECFP4 2,048-D. Colors highlighting the difference in the AUC, EF1, EF5, BEDROC20, BEDROC100, RIE20 and RIE100 values for 88 targets between MHFP6 2,048-D (orange) and ECFP4 2,048-D (green). MHFP6 significantly outperforms ECFP4 in the AUC, EF1 and EF5 metrics (see pairwise post-hoc Friedman tests of the average rank results in Figure S6a).
Figure S3 Performance comparison between MHFP6 2,048-D and MHECFP4 2,048-D. Colors highlighting the difference in the AUC, EF1, EF5, BEDROC20, BEDROC100, RIE20 and RIE100 values for 88 targets between MHFP6 2,048-D (orange) and MHECFP4 2,048-D (green). MHFP6 significantly outperforms MHECFP4 in the AUC, EF1 and EF5 metrics (see pairwise post-hoc Friedman tests of the average rank results in Figure S6a).
Figure S4 Average ranks of (L)ECFP4, (L)MHFP6, and path-based methods across 88 benchmark targets. Comparisons to path-based methods AP (Atom Pair fingerprint), TT (Topological Torsion fingerprint) and RDK5 (RDKit implementation of the Daylight fingerprint) show that the AUC performance of (L)MHFP6 is not significantly different from that of path-based methods but significantly better than that of (L)ECFP4 (Figure S7). However, (L)MHFP6 outperforms path-based methods in other metrics. ECFP4 and LECFP4 are 1,024-D and 16,384-D, respectively. MHFP4 and LMHFP4 correspond to 2,048-D and 4,096-D, respectively. MACCS and ECFC0 were used as a baseline.

Figure S5 Results of benchmarking hashing methods across 88 benchmark targets. Results from benchmarking hashing approaches without further dimensionality reduction [7]. (a) Comparison of hashed molecular shinglings with $r = 2$ with ECFP4/6* and ECFP4/6 hashes. (b) Comparison of hashed molecular shinglings with $r = 3$ with ECFP4/6* and ECFP4/6. As a control, variants of ECFP4/6, ECFP4/6*, considering only atomic numbers as invariants was benchmarked. Green and orange colors indicate molecular shingling hashes being ranked lower and higher than ECFP hashes, respectively. Data points below the dashed line are 0.05.
Figure S6 Pairwise post-hoc Friedman tests of the average rank. Statistical tests were run as part of the benchmark and visualized for easier comprehension. (a) Relative ranking and p-values of 2,048-D MHFP6 compared to ECFP4 (2,048-D), LECFP4 (16,384-D), MHECFP4 (2,048-D), and LMHECFP4 (4,096-D). (b) Relative ranking and p-values of 4,096-D MHFP6 compared to ECFP4 (2,048-D), LECFP4 (16,384-D), MHECFP4 (2,048-D), and LMHECFP4 (4,096-D). Orange color corresponds to MHFP6 being ranked higher than the other fingerprint, while green color indicates a lower ranking. P-values below 0.05 (dashed horizontal line) indicate significance.

Figure S7 Pairwise post-hoc Friedman tests of the average rank (path-based methods). Statistical tests were run as part of the benchmark and visualized for easier comprehension. (a) Relative ranking and p-values of 2,048-D MHFP6 compared to AP (Atom Pair fingerprint), ECFP4 (1,024-D), RDK5 (RDKit implementation of the Daylight fingerprint) TT (Topological Torsion fingerprint), and LECFP4 (16,384-D). (b) Relative ranking and p-values of 4,096-D MHFP6 compared to AP (Atom Pair fingerprint), ECFP4 (1,024-D), RDK5 (RDKit implementation of the Daylight fingerprint) TT (Topological Torsion fingerprint), and LECFP4 (16,384-D). Orange color corresponds to MHFP6 being ranked higher than the other fingerprint, while green color indicates a lower ranking. P-values below 0.05 (dashed horizontal line) indicate significance.
Figure S8 Performance analysis of MHFP6-based k-nearest neighbor search (ChEMBL, $k_c = 10$). Running the benchmark on randomized ChEMBL subsets of size $n = 100000$ with $k_c = 10$ shows an overall decrease in recovery rate (a, b) compared to the DUD-E benchmark (Figure 5a, b). However, in terms of speed (c, d) the LSH Forest algorithm performed as expected. $t$-test results and $p$-values can be found in Table S1.

Figure S9 Performance analysis of MHFP6-based k-nearest neighbor search (ChEMBL, $k_c = 100$). Running the benchmark on randomized ChEMBL subsets of size $n = 100000$ with $k_c = 100$ shows an overall increase in recovery rate (a, b) compared to LSH Forest with $k_c = 10$ (Figure S8). The performance of the LSH Forest algorithm increases compared to the DUD-E benchmark (Figure 5a, b). While the search time (d) for $k_c = 100$ increases slightly compared to $k_c = 10$, the build / index time (c) remains the same as it is not influenced by changes to $k_c$. $t$-test results and $p$-values can be found in Table S1.
Figure S10 Performance as a function of parameter $k_c$. A recovery study was repeated twice for 20 subsets containing 10,000 (a, c) and 100,000 (b, d) compounds, respectively. The performance of our implementation of the LSH Forest algorithm depends on the parameters $k_c$ and $l$. While speed is only marginally affected by changes in both parameters, the recovery rate rises with the increase of both values, approaching a perfect recovery rate. The number of nearest neighbors $k$ to be searched for also influences recovery rate in a manner not correlating to the size of the data set (a, b).

Table S1 Performance statistics of MHFP6-based k-nearest neighbor search

| MHFP6 compared to | DUD-E | ChEMBL (10) | ChEMBL (100) |
|-------------------|-------|-------------|--------------|
| MHECFP4 (Recovery Rate) |       |             |              |
| # Nearest Neighbors | p-Value | t | p-Value | t | p-Value | t |
| 5                  | 9.498E-01 | 0.063 | 1.348E-04 | 3.82 | 4.150E-07 | 5.068 |
| 10                 | 9.382E-01 | -0.078 | 1.245E-07 | 5.294 | 1.452E-06 | 4.823 |
| 50                 | 2.773E-05 | 4.195 | 1.316E-08 | 5.693 | 3.952E-15 | 7.879 |
| 100                | 2.905E-08 | 5.555 | 3.710E-08 | 5.512 | 1.079E-29 | 11.383 |
| MHECFP4 (Recovery Rate) |       |             |              |
| # Prefix Trees | p-Value | t | p-Value | t | p-Value | t |
| 8                  | 8.760E-06 | 4.453 | 4.805E-06 | 4.581 | 2.301E-31 | 11.775 |
| 16                 | 9.317E-03 | 2.602 | 3.514E-01 | 0.932 | 1.848E-10 | 6.394 |
| 32                 | 4.179E-03 | -2.866 | 5.667E-01 | -0.573 | 7.303E-01 | 0.345 |
| 64                 | 9.221E-02 | -1.684 | 5.697E-01 | -0.569 | 1.860E-02 | 2.355 |
| 128                | 6.943E-01 | 0.393 | 2.492E-02 | 2.244 | 8.795E-06 | 4.452 |
| 256                | 9.609E-02 | 1.665 | 4.885E-06 | 4.578 | 6.384E-10 | 6.2 |
| 512                | 7.953E-18 | 8.651 | 1.892E-57 | 16.301 | 3.308E-32 | 11.945 |
| MHECFP4 (Search Speed) |       |             |              |
| # Prefix Trees | p-Value | t | p-Value | t | p-Value | t |
| 8                  | 1.000E-01 | 1.645 | 4.773E-02 | 1.981 | 1.412E-01 | 1.472 |
| 16                 | 4.311E-01 | 0.787 | 1.663E-01 | 1.384 | 4.352E-02 | 2.019 |
| 32                 | 1.137E-01 | 1.582 | 8.489E-01 | 0.19 | 7.870E-01 | -0.27 |
### ECFP4 (Recovery Rate)

| # Nearest Neighbors | p-Value | t     | p-Value | t     | p-Value | t     |
|---------------------|---------|-------|---------|-------|---------|-------|
| 5                   | 5.688E-02 | -1.905 | 1.418E-03 | -3.194 | 2.158E-29 | 11.37 |
| 10                  | 3.897E-01 | -0.86  | 2.938E-03 | -2.976 | 1.594E-51 | 15.376|
| 50                  | 7.62E-16  | 8.102  | 2.365E-02 | 2.264  | 4.900E-248| 36.846|
| 100                 | 5.29E-50  | 15.13  | 5.418E-13 | 7.244  | 0.000E+00 | 51.446|

### ECFP4 (Search Speed)

| # Prefix Trees    | p-Value | t     | p-Value | t     | p-Value | t     |
|-------------------|---------|-------|---------|-------|---------|-------|
| 8                 | 1.697E-24 | -10.3 | 1.364E-66 | -17.648 | 1.618E-38 | 13.152|
| 16                | 3.82E-02  | -2.073 | 1.949E-27 | -10.954 | 1.233E-113| 23.598|
| 32                | 1.436E-01 | 1.463  | 1.155E-05 | -4.393  | 7.426E-191| 31.589|
| 64                | 1.228E-18 | 8.867  | 6.200E-02 | 1.867   | 7.302E-313| 42.463|
| 128               | 5.05E-47  | 14.639 | 1.334E-22 | 9.857   | 0.000E+00 | 52.195|
| 256               | 9.537E-90 | 20.74  | 7.761E-61 | 16.811  | 0.000E+00 | 57.261|
| 512               | 3.02E-110 | 23.208 | 7.108E-33 | 12.078  | 0.000E+00 | 59.449|

The p-values and t statistics of independent t-tests for the means of two independent samples for 2,048-D MHFP6 versus 2,048-D MHECFP4 and 2,048-D ECFP4. Significant values are shown in bold and color. Values where MHFP6 performed better or worse are colored in blue and red respectively.