Hashing Algorithms and Data Structures for Rapid Searches of Fingerprint Vectors

Ramzi Nasr, Daniel S. Hirschberg, and Pierre Baldi*

School of Information and Computer Sciences, Institute for Genomics and Bioinformatics, University of California, Irvine, Irvine, California 92697-3435

Received April 6, 2010

In many large chemoinformatics database systems, molecules are represented by long binary fingerprint vectors whose components record the presence or absence of particular functional groups or combinatorial features. To speed up database searches, we propose to add to each fingerprint a short signature integer vector of length M. For a given fingerprint, the i component of the signature vector counts the number of 1-bits in the fingerprint that fall on components congruent to i modulo M. Given two signatures, we show how one can rapidly compute a bound on the Jaccard−Tanimoto similarity measure of the two corresponding fingerprints, using the intersection bound. Thus, these signatures allow one to significantly prune the search space by discarding molecules associated with unfavorable bounds. Analytical methods are developed to predict the resulting amount of pruning as a function of M. Data structures combining different values of M are also developed together with methods for predicting the optimal values of M for a given implementation. Simulations using a particular implementation show that the proposed approach leads to a 1 order of magnitude speedup over a linear search and a 3-fold speedup over a previous implementation. All theoretical results and predictions are corroborated by large-scale simulations using molecules from the ChemDB. Several possible algorithmic extensions are discussed.

INTRODUCTION

We consider the problem of efficiently searching large databases of vectors, which occurs in various areas of information retrieval. In chemoinformatics, we assume that we have a large database of small molecules.[1−5] It is common practice to represent these molecules using long fingerprint vectors, where the components of a vector correspond to binary or integer variables, associated with the presence or absence or the number of occurrences of a particular feature in a molecule.[6−12] For simplicity, in the rest of the paper, we consider the most frequently used binary fingerprints, but most of the ideas can be extended to integer valued fingerprints. Typical features used in the literature and existing chemoinformatics systems correspond, for instance, to all possible labeled paths or trees up to a certain depth.[13−16] The exact nature of the fingerprints is not important for what follows, and the derivations can be applied to any kind of fingerprint vectors. Thus, for any molecule $\mathcal{A}$, we assume that we have a corresponding binary fingerprint vector $\vec{A} = (A_i)$ of length N with $A = \sum_{i=1}^{N} A_i$. We call A the size or weight of the corresponding fingerprint. Assuming that $\mathcal{A}$ is the query molecule, we are interested in rapidly retrieving all the molecules $\mathcal{B}$ in the database that are similar to $\mathcal{A}$. The approach we propose is described using the most widely used Jaccard−Tanimoto similarity measure given by

$$S(\mathcal{A}, \mathcal{B}) = T(\vec{A}, \vec{B}) = \frac{A \cap B}{A \cup B} \tag{1}$$

where $A \cap B$ and $A \cup B$ denote the size of the intersection and union of $\vec{A}$ and $\vec{B}$. In the case of integer- or real-valued vectors, this can be generalized by

$$S(\mathcal{A}, \mathcal{B}) = T(\vec{A}, \vec{B}) = \frac{\sum_{i=1}^{N} \min(A_i, B_i)}{\sum_{i=1}^{N} \max(A_i, B_i)} \tag{2}$$

But the same ideas can be adapted to other similarity measures.[17] We will assume that one is interested in retrieving all the molecules $\mathcal{B}$ with similarity to $\mathcal{A}$ above a certain Jaccard−Tanimoto threshold $0 \leq t \leq 1$. In large chemoinformatics databases, the overwhelming majority of the molecules will *not* be similar to the query. Thus, the basic idea pursued here is the idea of database pruning, which seeks to develop fingerprint representations and algorithms to prune the search space and avoid having to search the space linearly and compute the Jaccard−Tanimoto similarity for all the molecules in the database. More precisely, the idea is to derive short signatures from the long fingerprints and use the shorter signatures to both derive efficient bounds on the similarity measures and organize the set of fingerprints to facilitate rapid searches. Molecules for which the bound is unfavorable can be discarded from the search. In other words, for a search based on a similarity threshold t, we are interested in rapidly identifying a collection of molecules that are guaranteed to have a similarity to $\mathcal{A}$ lower than t and discard them from the search.

One general approach of deriving fingerprint signatures is to partition the components of the fingerprints into M sets ($1 \leq M \leq N$) and, for each molecule, use a simple function

* To whom correspondence should be addressed. E-mail: pfbaldi@ics.uci.edu.

10.1021/ci100132g
Published on Web 08/03/2010

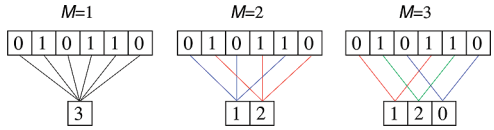

Figure 1. An illustration of modulo hashing at $M = 1, 2,$ and 3. For demonstration purposes, a small 6-bit fingerprint $\vec{A} = (0, 1, 0, 1, 1, 0)$ is used.

f to summarize the complement of 1-bits that are observed in each set. While uneven partitions are possible and briefly discussed at the end of this article, here we consider even partitions where all the sets have the same size (N/M). Since random permutations are often applied to fingerprint vectors, we can assume without any loss of generality that the partition i corresponds to taking components that are congruent to i modulo M. Several methods proposed in the literature can be viewed as special cases of this framework:

(1) When $M = 1$ and f is the sum, the signature corresponds to the total number of bits set to 1 in the fingerprint, and one then can use the bounds derived in ref 18 to prune the database.

(2) When f is the logical *OR* operator capturing whether there is at least one 1-bit in a given partition (with, for instance, $M = 1024$), one obtains the lossy OR-compressed fingerprint used in the Daylight system,[8] and the Jaccard−Tanimoto similarity computed on these compressed fingerprints can be used to search the database directly, although even better results can be obtained with a systematic correction.[19]

(3) When f is the logical *XOR* operator capturing whether the total number of 1-bits in a given partition is odd or even (with, for instance, $M = 128$), one obtains the approach described in ref 20 and the corresponding bounds to prune the database.

The approach that we study here can be viewed as a generalization of the first approach where f is still the sum operator but $M > 1$, or a generalization of the second and third approaches where $M > 1$ but f is the sum operator.[21−24]

THE MODULO M HASHING REPRESENTATION AND THE INTERSECTION INEQUALITY

In this work, we thus propose to use and exploit the modulo M-hashing representation and the intersection bound introduced in previous work.[24] In this approach, given any integer $1 \le M \le N$, for each fingerprint vector $\vec{A}$ we construct a new integer vector signature of length M, $\vec{a}^M = (a_i^M) = (a_i)$ with $i = 1, ..., M$, where $a_i^M = a_i$ counts the number of 1-bits in $\vec{A}$ falling on components that are congruent to i modulo M. For instance, if $\vec{A} = (0, 1, 0, 1, 1, 0)$ and $M = 3$, then $\vec{a} = (1, 2, 0)$ (see Figure 1). The superscript M is used only in situations where several values of M are used in combination. Most of the time, in the following derivations, M is fixed and clear from the context; hence we drop it from the notations and write, for instance, $\vec{a} = (a_i)$. Note that for any M, we have $a(M) = a = \sum_{i=1}^{M} a_i^M = A$. In the next sections, we look at the intersection and other related bounds associated with the modulo-M-hashing representations and show how these can be used for efficient pruning of database searches. The level of pruning for different values of M is estimated mathematically and confirmed empirically. We then turn to the problem of optimizing M, combining different values of M, and optimizing the overall strategy for efficiently searching the database and show empirically how the proposed approach leads to a 1 order of magnitude speedup over a linear search, and a 3-fold speedup over the methods described in ref 20.

To derive the intersection inequality, note that the bits that are in common to two fingerprints $\vec{A}$ and $\vec{B}$ can be partitioned into bits that are in common for each set of components congruent to i modulo M, and these in turn can be bounded by $\min(a_i, b_i)$. Thus, for any $1 \le M \le N$, we have the intersection inequality:[20]

$$A \cap B \le \sum_{i=1}^{M} \min(a_i^M, b_i^M) \tag{3}$$

This leads to the following bound on the Jaccard−Tanimoto similarity:

$$T(\vec{A}, \vec{B}) \le \frac{A \cap B}{A \cup B} = \frac{A \cap B}{A + B - A \cap B} \le \frac{\sum_{i=1}^{M} \min(a_i^M, b_i^M)}{\sum_{i=1}^{M} \max(a_i^M, b_i^M)} = T(\vec{a}(M), \vec{b}(M)) \tag{4}$$

Furthermore, note that for any $1 \le M \le N$ and $1 \le k \le N/M$

$$T(\vec{A}, \vec{B}) = T(\vec{a}(N), \vec{b}(N)) \le T(\vec{a}(kM), \vec{b}(kM)) \le T(\vec{a}(M), \vec{b}(M)) \le T(\vec{a}(1), \vec{b}(1)) = \frac{\min(A, B)}{\max(A, B)} \tag{5}$$

APPLICATION TO DATABASE PRUNING

If we are interested in finding molecules with a Jaccard−Tanimoto similarity above some threshold t, then we can remove all the molecules satisfying

$$T(\vec{A}, \vec{B}) \le \frac{\sum_{i=1}^{M} \min(a_i^M, b_i^M)}{\sum_{i=1}^{M} \max(a_i^M, b_i^M)} = \frac{\sum_{i=1}^{M} \min(a_i^M, b_i^M)}{A + B - \sum_{i=1}^{M} \min(a_i^M, b_i^M)} \le t \tag{6}$$

which gives

$$S = \sum_{i=1}^{M} \min(a_i^M, b_i^M) \le \frac{t}{1 + t}(A + B) \tag{7}$$

The Case of $M = 1$. This is exactly the case described in previous work.[18] In this case, eq 7 implies that we can discard all the molecules $\vec{B}$ with $B \le tA$ or $B \ge A/t$. For example, with fingerprints of length $N = 1024$, consider a typical query $\vec{A}$ with $A = 400$ and a similarity threshold of $t = 0.8$. Then, using the triangle inequality, we can immediately discard from the search all the molecules $\vec{B}$ with $B \le 320 = 0.8 \times 400$ or $B \ge 500 = 400/0.8$ and restrict the search to molecules satisfying $320 < B < 500$.

The Case of $M = 2$. The case $M = 2$ subsumes the case $M = 1$ and allows one to prune additional molecules. In this case, we have $\vec{a} = (a_1, a_2)$ and $\vec{b} = (b_1, b_2)$ where, for instance, b_1 represents the number of 1-bits in $\vec{B}$ falling on odd-numbered components, and b_2 the number of 1-bits in $\vec{B}$ falling on even-numbered components, with $b_1 + b_2 = B$. In this case, the intersection bound leads to pruning all the molecules $\vec{B}$ such that

$$\min(a_1, b_1) + \min(a_2, b_2) \leq \frac{t}{1+t}(A + B) \qquad (8)$$

Depending on the realization of each minimum in eq 8, one obtains four exclusion subcases:

(1) $a_1 + a_2 = A \leq t(A + B)/(1 + t)$, which yields $B \geq A/t$ (as in the case of $M = 1$).

(2) $b_1 + b_2 = B \leq t(A + B)/(1 + t)$, which yields $B \leq tA$ (as in the case of $M = 1$).

(3) $a_1 + b_2 \leq t(A + B)/(1 + t)$, which yields $b_2 \leq -a_1 + t(A + B)/(1 + t)$.

(4) $b_1 + a_2 \leq t(A + B)/(1 + t)$, which yields $b_1 \leq -a_2 + t(A + B)/(1 + t)$.

Subcases 1 and 2 correspond to the pruning derived when $M = 1$, and subcases 3 and 4 correspond to additional pruning. From subcases 1 and 2, we can focus the search exclusively on the set of molecules satisfying $At < B < A/t$. And for a fixed B in that set, using subcases 3 and 4, we can focus exclusively on the molecules satisfying $-a_2 + t(A + B)/(1 + t) < b_1 < B + a_1 - t(A + B)/(1 + t)$, since $b_1 = B - b_2$.

In the example above, with a query $\vec{A}$ with $A = 400$ and $t = 0.8$, subcases 3 and 4 allow us to prune additional molecules $\vec{B}$ with $320 < B < 500$. For instance, assume that $\vec{a} = (250, 150)$. Then, even among the molecules $\vec{B}$ containing the same number of 1-bits as the query (i.e., $A = B = 400$), we can eliminate all the ones associated with $\vec{b} = (b_1, b_2)$ for which $b_1 \leq 205$ or $b_2 \leq 105$. Because $b_1 + b_2 = 400$, eliminating $b_1 \leq 205$ and $b_2 \leq 105$ is equivalent to restricting b_1 to be in the range $205 < b_1 < 295$.

The Case of General (Large) M. In the general case, pruning is implemented similarly using the intersection inequality. As M is increased, the amount of pruning can be expected to increase too. However, M cannot be chosen to be arbitrarily large because other important factors must be taken into consideration to optimize a given implementation, including the additional storage and computing resources associated with the signature, as well as the corresponding data structures. In any case, before one can begin addressing these trade-offs, it is essential to be able to understand and estimate analytically, as well as empirically, the amount of pruning as a function of M.

ESTIMATING THE AMOUNT OF PRUNING ANALYTICALLY

The Case of $M = 1$. For a given fingerprint query containing A 1-bits, we have seen that we can discard all the molecules with $\vec{B}$ satisfying $B \leq tA$ or $B \geq A/t$. Thus, the fraction of total pruning for $M = 1$ is given by

$$P_1(A) = \int_0^{tA} g(u)\,\mathrm{d}u + \int_{A/t}^{N} g(u)\,\mathrm{d}u = 1 - [G(A/t) - G(tA)] \qquad (9)$$

where we use a continuous notation and let $g(u)$ be the density approximation to the histogram of the number of molecules per number of 1-bits contained in their fingerprints. This is the formula derived in ref 18. In practice, it can be easily verified empirically and shown analytically for some simple probabilistic models of fingerprints that g is well approximated by a Normal density. In other words, in a typical large database containing D small molecules, the number of molecules containing A 1-bits in their fingerprint is approximately given by $D \times g(A) = D \times (2\pi)^{-1/2}\sigma^{-1} \exp[-(A - \mu_D)^2/2\sigma_D^2]$. The mean and variance of the Normal density can easily be computed from a sample of molecules and their values depend on the details of a particular implementation. It can be shown,[18] using standard approximations, that when the threshold t is close to 1, the fraction of total pruning is given approximately by

$$1 - \frac{A(1-t^2)}{t\sigma_D}\frac{1}{\sqrt{2\pi}}\exp[-(A - \mu_D)^2/2\sigma_D^2] \approx 1 - (1-t)\frac{2A}{\sqrt{2\pi}\sigma}\exp[-(A - \mu_D)^2/2\sigma_D^2] \qquad (10)$$

so the fraction of pruning scales at least like $1 - C(1 - t)$ when t is close to 1 for some constant C. Note that the formula given in this section can be integrated over the distribution of the query sizes ($g(A)$) to get the average fraction of pruning across the queries. In particular, if the distribution of the number of 1-bits in the queries is the same as in the database, one can integrate eq 9 with respect to the density g to get the average fraction of pruning across all queries, in the form

$$P_1 = \int_{A=1}^{N} g(A)\,P_1(A) = \int_{A=1}^{N} g(A)[1 - (G(A/t) - G(tA))] \qquad (11)$$

The Case of $M = 2$. When $M = 2$, we have seen that there are two levels of pruning. For a given query with signature (a_1, a_2) and $A = a_1 + a_2$ 1-bits, the first level of pruning is identical to the case of $M = 1$, and we need only to focus on molecules in the database for which $At < B < A/t$. For each B in this range, one can additionally prune all the molecules with signature (b_1, b_2) such that $b_1 \leq -a_2 + t(A + B)/(1 + t)$ or $b_1 \geq B + a_1 - t(A + B)/(1 + t)$. The corresponding proportion of molecules can again be estimated by

$$\begin{aligned} &\int_0^{-a_2+t(A+B)/(1+t)} g_B(u)\,\mathrm{d}u + \\ &\int_{B+a_1-t(A+B)/(1+t)}^{B} g_B(u)\,\mathrm{d}u = \\ &1 - [G_B(B + a_1 - t(A + B)/(1 + t)) - \\ &\quad G_B(-a_2 + t(A + B)/(1 + t))] \end{aligned} \qquad (12)$$

where g_B is now the density approximation to the histogram associated with the values of b_1, within the set of molecules with fingerprints containing B 1-bits. Again, in most practical cases, this density is approximately Normal. [Of course, this approximation can break down for extreme values of B (e.g., B close to 0 or its maximal value), but these are atypical and can be neglected to a first degree of approximation]. Approximate Normality is also shown in the Appendix using a theoretical model from which one can also derive the mean

and standard deviation of this Normal distribution in the form

$$\mu_B = \frac{B}{2} \text{ and } \sigma_B^2 = \frac{B(N-B)}{4(N-1)} \tag{13}$$

Thus, for a query with A 1-bits, the total fraction of pruning is given by

$$P_2(A) = 1 - [G(A/t) - G(tA)] + \sum_{At}^{A/t} g(B)\, g(B)\, (1 - [G_B(B + a_1 - t(A+B)/(1+t)) - G_B(-a_2 + t(A+B)/(1+t))]) \tag{14}$$

This fraction can again be integrated over the distribution of the query sizes ($g(A)$) to get the total average fraction of pruning across all possible queries. If the distribution of the number of 1-bits in the queries is the same as in the database, the average fraction of pruning can again be computed as

$$P_2 = \int_{A=1}^{N} g(A)\, P_2(A) \tag{15}$$

The Case of General (Large) *M*. The estimation of the total fraction of pruning is somewhat more involved, and the mathematical derivation is given in the Appendix. However, the basic idea is easy to understand. From eq 9, the basic quantity of interest is the intersection bound $S = \sum_{i=1}^{M} \min(a_i^M, b_i^M)$. Intuitively, for large values of M, one ought to be able to approximate the distribution of S reasonably well by a Normal distribution by the central limit theorem.[25] This is not a rigorous argument since the individual terms $\min(a_i, b_i)$ are not independent and identically distributed. Under exchangeability assumptions, these terms are identically distributed as discussed in the Appendix. In any case, the results presented in the next sections show that the Normal approximation works well in practice.

DATA

In the simulations, we illustrate the methods using fingerprints that are randomly selected from more than 5 million molecules available in the ChemDB database.[1] While the approach described in this paper can be applied with any kind of fingerprint system, in the simulations we use fingerprints based on labeled paths of lengths up to eight. For this scheme, each vertex is labeled by the element (C, N, O, etc.) associated with the corresponding atom, and each edge is labeled by the type (single, double, triple, aromatic, and amide) of the corresponding bond. This scheme is closely related to the scheme used in many existing chemoinformatics systems, including the Daylight system.[8] In the simulations, we used both uncompressed fingerprints, corresponding also to lossless compressed fingerprints,[12] as well as lossy compressed fingerprints obtained using the standard modulo-OR-compression algorithm to generate fingerprint vectors of length $N = 1024$. Typical simulations are run using a sample of $n = 100$ queries against a background of 100 000 fingerprints randomly sampled from ChemDB.

RESULTS

Pruning Results. The case $M = 1$ has been studied extensively in ref 18.

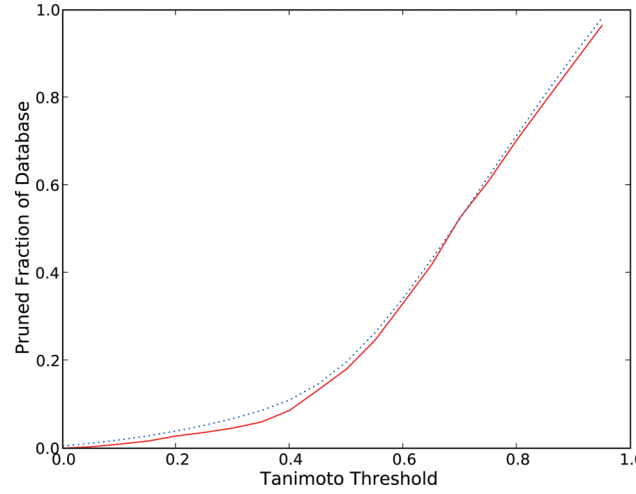

Figure 2. Theoretical curve following the empirical level of pruning as a function of the Tanimoto threshold. The solid curve represents the empirical fraction of pruning measured over different Tanimoto thresholds with the $M = 2$ hashing approach. A sample of 100 query fingerprints with $A = 205$ is selected from ChemDB and run against a random sample of 100 000 fingerprints from ChemDB. The dotted curve shows the analytically predicted fraction of pruning (eq 14) given $A = 205$, the mean (205) and standard deviation (97.9) of the number B of 1-bits for the fingerprints in the background sample, and the threshold t.

The Case of $M = 2$. Figure 2 shows empirical and theoretical results corresponding to the case of $M = 2$. The figure displays the level of pruning as a function of the Tanimoto threshold. The empirical curve is obtained by averaging the pruning levels of 100 query molecules with $A = 205$ used to search a background sample of 100 000 molecules from ChemDB. In this case, the fingerprints are compressed in a lossy fashion using the standard modulo-OR-compression algorithm. The theoretical results are computed using eq 14. $g(B)$ is approximated using a Normal distribution with parameters obtained empirically ($\mu_D = 205$, $\sigma_D = 97.9$). As Figure 2 shows, the theoretical curve follows very closely the empirical level of pruning as a function of the Tanimoto threshold.

The Case of General (Large) M. Here, we consider fingerprints with an associated signature of length M. As discussed in the Appendix, the analytical derivation to predict the levels of pruning starts by studying the distribution of the number a_i^M of 1-bits in a particular class modulo M. Figure 3 shows the empirical distributions of a_i^M at $M = 16$, 32, 64, and 128, with their Normal approximations in solid black lines. The empirical results are derived from a sample of 100 000 molecules from ChemDB. The means and variances of the Normal distributions are obtained using the hypergeometric model described in the Appendix (eqs 33 and 34). As expected, as M increases relative to A, or as the ratio A/M decreases, the distribution of a_i^M transitions from a Normal distribution to a Poisson distribution with mean $\lambda = A/M$, corresponding to more rare bit events. This is visible in the two bottom subfigures of Figure 3 where the Poisson approximations, shown in dashed black lines, provide a better fit for the empirical data.

Next, we study the distribution of $S = \sum_{i=1}^{M} \min(a_i^M, b_i^M)$, corresponding to the intersection bound and the left-hand side of eq 7. The top left subfigure of Figure 4 shows the empirical means and standard deviations (μ, σ) of S in black. The empirical results are obtained again by using 100 query

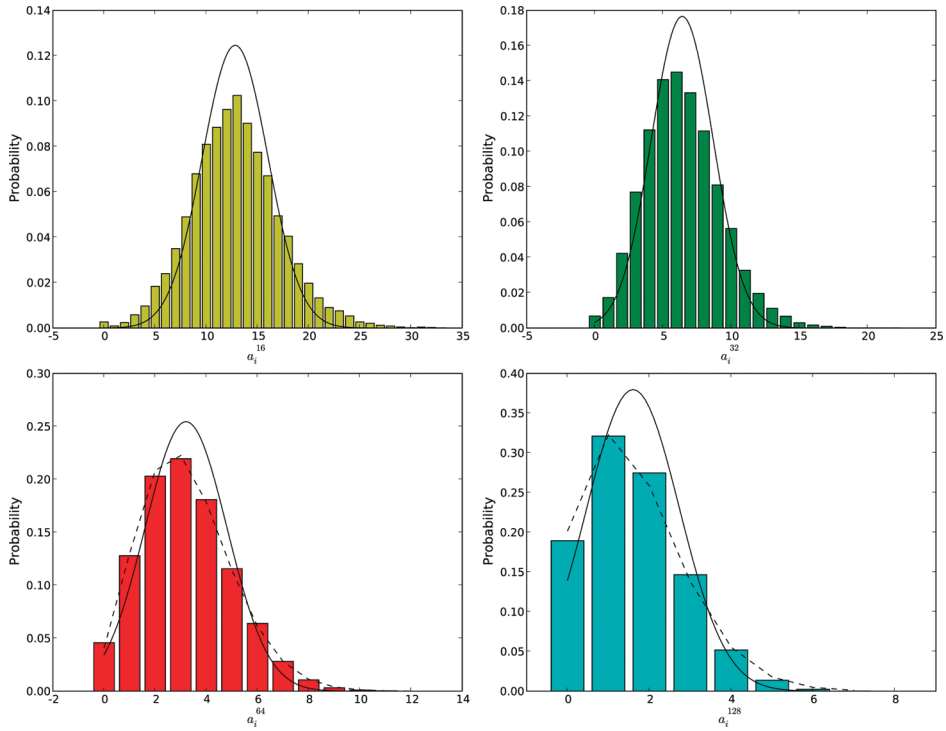

Figure 3. The distribution of the number a_i^M of 1-bits in a generic bin i for $M = 16$, 32, 64, and 128, for ChemDB fingerprints with $A = 205$. The black curve shows the Normal approximation with parameters estimated using the hypergeometric model (eqs 33 and 34). For larger values of M in the two lower figures, the dashed black line shows how the Poisson distribution provides a better approximation of the empirical data.

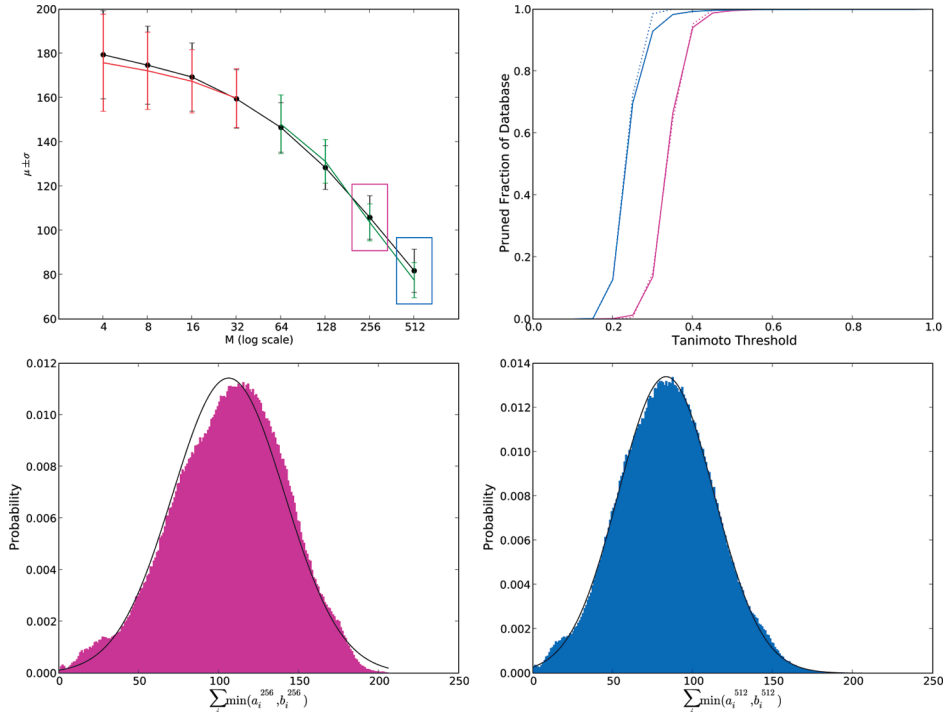

Figure 4. Top left: the empirical means (black dots) and standard deviations (black error bars) of the intersection bound $S = \sum_{i=1}^{M} \min(a_i^M, b_i^M)$ as a function of M. The results correspond to 100 query molecules with $A = 205$ against a sample of 100 000 molecules from ChemDB. The red and green curves show the analytical predictions using the Normal and the Poisson approximations for a_i^M, respectively. Top right: analytical (dotted lines) and empirical amount of pruning for $M = 256$ and $M = 512$. Bottom left: empirical distribution of S for $M = 256$ (magenta) with its Normal approximation. Bottom right: empirical distribution of S for $M = 512$ (blue) with its Normal approximation.

molecules with $A = 205$ against a sample of 100 000 molecules from ChemDB. The fingerprints use the standard modulo-OR lossy compression algorithm. Theoretical results, shown by the red curve and error bars for $M < 64$, are obtained by approximating the underlying distribution of a_i^M by a Normal distribution, as described in the Appendix. The green curve and error bars at $M \geq 64$ show the theoretical results when the Poisson approximation is used for a_i^M. As expected, S decreases as M increases, resulting in higher levels of pruning according to eq 7. As shown in the

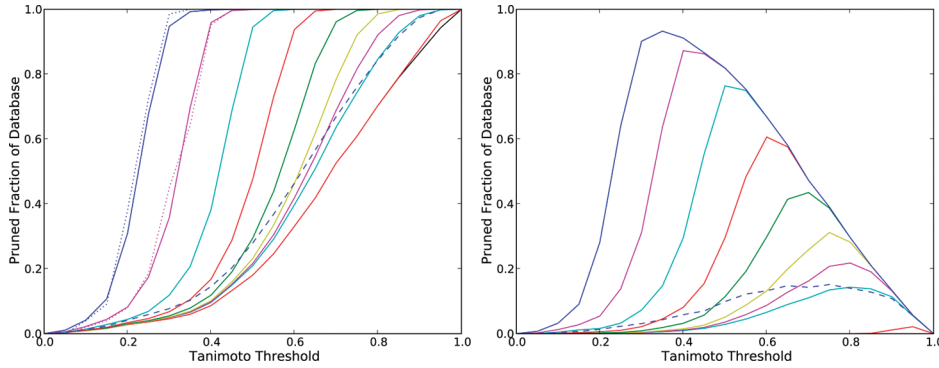

Figure 5. Left: curves showing the average fraction of pruning for different thresholds t and different values of M. The results are obtained by averaging runs of 100 random query molecules against a background of 100 000 molecules from ChemDB. Color legend from right to left: black corresponds to $M = 1$, red corresponds to $M = 2$, dashed curve to recursive hashing (see text), cyan to $M = 4$, magenta to $M = 8$, yellow to $M = 16$, green to $M = 32$, red to $M = 64$, cyan to $M = 128$, magenta to $M = 256$, blue to $M = 512$. Predicted curves obtained using eq 28 are shown with dots of the corresponding color for $M = 256$ and $M = 512$. Right: similar results and colors, but displaying the fraction of molecules eliminated for different values of M, relative to $M = 1$.

Appendix, integrating the distribution of S over regions determined by A, B, and t yields an estimate for the amount of pruning. The analytical approach in the Appendix provides a good approximation for the mean μ and standard deviation σ of S. For demonstration purposes, the bottom row subfigures of Figure 4 show the full distributions at $M = 256$ and 512, encapsulated by the magenta and blue rectangles in the top left subfigure of Figure 4. For these values of A and M, the Normal approximation shown by the black solid line provides a good approximation of the empirical distribution. However, the distribution of S does not always follow closely a Normal distribution for a variety of reasons. Exceptions are found when M is small, or in regimes where the correlations between the terms in the sum are too large.

In the top right subfigure of Figure 4, the amount of pruning at $M = 256$ and $M = 512$ is shown in magenta and blue respectively. The dotted lines show the theoretical levels of pruning computed using eqs 28−32. As expected, the level of pruning increases at higher values of M. The distribution of S, whether obtained empirically or approximated using a Normal distribution (bottom row of Figure 4), leads to a good approximation for the level of pruning.

Figure 5 shows empirical pruning results obtained by averaging runs of 100 random query molecules against a sample of 100 000 molecules from ChemDB. In this experiment, uncompressed fingerprints are used. Cases corresponding to $M = 1$, 2, 4, 8, 16, 32, 64, 128, 256, and 512 are shown in black, red, cyan, magenta, yellow, green, red, cyan, magenta, and blue, respectively. In the left subfigure, each curve corresponds to the fraction of database pruning as a function of the Tanimoto threshold on the horizontal axis. Predicted curves obtained using eq 28 are also shown for $M = 256$ and $M = 512$ and closely agree with the empirical values. Predictions for smaller values of M are less accurate. The right subfigure displays the additional pruning fraction relative to $M = 1$, for various values of M. As expected, the pruning levels increase as M increases. Empirical results show that the case of $M = 2$ only slightly outperforms the case of $M = 1$, primarily for Tanimoto thresholds greater than 0.85. Furthermore, larger values of M show substantial levels of additional pruning, compared to the case where $M = 1$ essentially at all threshold values.

Implementation Optimization and Data Structures: The Choice of *M*. While using larger values of M induces greater pruning, one must carefully consider implementation issues, in particular the data structure used. Without any data structure considerations, for a given query, one must compute the intersection bound for all the molecules in the database in order to decide which molecules to discard and which molecules to keep for computing the Jaccard−Tanimoto similarity. Thus, while there is pruning at the level of the Jaccard−Tanimoto computation, there is no pruning at the level of the intersection bound computation—all the signatures in the database must be considered sequentially. As we shall see, however, this does not have to be so with the choice of proper data structures. Thus, the final search speedup is dependent on the data structure used, as the use of different data structures results in different levels of pruning and overhead.

For the case $M = 1$, it is possible to use a data structure that induces very little overhead while still pruning a significant portion of the database.[18] In the preprocessing step of this approach, the fingerprints $\vec{B}$ in the database are binned according to their size B, and the bins are organized in increasing (or decreasing) order. The intersection bound with $M = 1$ leads immediately to discarding bins with values of B that are either too small or too large relative to the number A of 1-bits in the query. Thus, even in the simple case of $M = 1$, organizing the data into bins with increasing B provides additional savings since the intersection bound does not need to be computed for all the fingerprints found in bins that are excluded. The overhead machinery (tables/pointers) required to navigate the bins is minimal. This efficient organization can be extended for hashing with $M = 2$, simply by organizing all the fingerprints within a bin associated with B 1-bits by increasing (or decreasing) values of b_1. Thus, for instance, within the bin associated with $B = 400$, the (b_1, b_2) signatures are ordered as $(0, 400) < (1, 399) < (2, 398)$ etc. [If several fingerprints have the same (b_1, b_2) signature, their ordering does not matter for the $M = 2$ hashing approach]. During a search, the $M = 1$ intersection bound is used to remove the bins with unfavorable B values. For the remaining bins, the process is repeated using again the intersection inequality to search over b_1. Thus, for $M =$

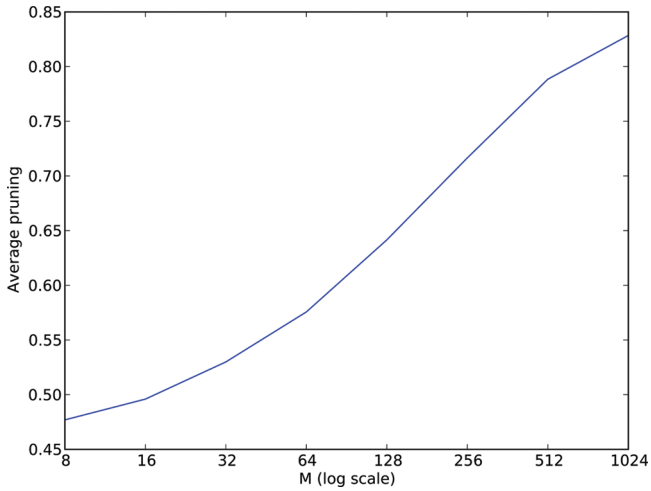

Figure 6. Average amount of pruning across all possible thresholds (t) obtained empirically from 100 random query molecules used to search a random sample of 100 000 ChemDB molecules using fingerprints with lossless compression.

2, a signature of the form (B, b_1) may be equivalent in content, but more efficient, than a signature of the form (b_1, b_2).

For values of $M > 2$, there are different possible strategies. For simplicity, here we consider the case where we first implement the $M = 2$ pruning, since the corresponding overhead is small, followed by pruning at a fixed level $M > 2$. In this framework, what is the optimal value of $M > 2$ that one ought to choose? To answer this question, an estimate of the search time can be given by

$$\text{search time} \approx [D \times (1 - P_2) \times T_M] + [D \times (1 - P_M) \times T_{\text{tanimoto}}] + \phi \quad (16)$$

where D is the database size, P_M is the average fraction of pruning at a given M averaged over all Tanimoto thresholds, T_M represents the average time required for a single computation of $S = \sum_{i=1}^{M} \min(a_i^M, b_i^M)$, and T_{Tanimoto} represents the average time required to compute the Jaccard−Tanimoto similarity between two uncompressed fingerprints. ϕ denotes constant overhead calculations independent of M. Thus, the first term in eq 16 represents the time spent on computing the intersection inequality on the signatures that have passed the $M = 2$ bound, and the second term represents the time spent on computing the Jaccard−Tanimoto similarity measure on the fingerprints that have passed the $M = 2$ and $M > 2$ intersection bounds. Here, we use a uniform distribution over all possible thresholds, but other distributions can easily be used instead. The average fraction of pruning for various values of M is shown in Figure 6. While the figure shows empirical results of pruning, this work also provides a framework to predict these values analytically. T_M and T_{Tanimoto} are given in Table 1.

To select the value of $M > 2$ that minimizes the search time, we assume that the overhead is roughly constant or negligible. Thus, the goal is to find $M > 2$ to minimize the quantity

$$Q = [(1 - P_2) \times T_M + (1 - P_M) \times T_{\text{tanimoto}}] \quad (17)$$

The left subfigure of Figure 7 plots Q as a function of M. Here, the quantities P_2, P_M (Figure 6), T_M, and T_{Tanimoto} (Table 1) are estimated from Monte Carlo runs. The approximation is consistent with the empirical results shown in the right subfigure of Figure 7. The red curve corresponds to empirical timing results obtained by searching 100 random molecules against a 100 000 database sample of fingerprints with lossless compression, while the blue curve shows the approximation given by eq 16 with the overhead fitted to empirical data ($\phi = 0.32$). The plots shows that for this particular implementation, using $M = 2$ hashing followed by $M = 256$ hashing minimizes the search time over all Tanimoto thresholds. And this is the strategy that we use to run the timing tests.

Table 1. Averages of 10^6 Computations at Various Values of M and the Average of 10^6 Tanimoto Computations on Full Fingerprints with Lossless Compression

computation	time (μs)
$T_{M=4}$	6.51 ± 1.09
$T_{M=8}$	6.35 ± 0.61
$T_{M=16}$	6.47 ± 0.57
$T_{M=32}$	6.48 ± 0.78
$T_{M=64}$	6.59 ± 0.79
$T_{M=128}$	6.73 ± 0.70
$T_{M=256}$	7.03 ± 0.52
$T_{M=512}$	7.97 ± 0.91
$T_{M=1024}$	9.68 ± 1.43
T_{Tanimoto}	48.8 ± 20.4

Timing Results. Search time is measured by the time it takes to search a query molecule against a database of molecules. Time that is not directly related to searching, such as preprocessing, is thus not considered. In a search, the most computationally expensive operation is the computation of the Jaccard−Tanimoto similarity between two fingerprint vectors. Database pruning speeds up the search by limiting the number of pairwise similarity computations to only a fraction of the database. The speedup is dependent on the data structure as different data structures bring different levels of pruning and overhead. In the hashing approach described here, the main overhead that contributes to the overall running time is the computation of the intersection bound $S = \sum_{i=0}^{M} \min(a_i^M, b_i^M)$.

The timing experiments are run using a dual-core AMD Opteron 280 processor, with a 2.4 GHz CPU, 1 megabyte of cache, and 4 gigabytes of RAM. In these experiments, we average the timing results obtained using 100 random queries extracted from ChemDB against a random background of 100 000 molecules. For this experiment, we use uncompressed fingerprints. Figure 8 compares the best speed results, obtained by using hashing at $M = 2$ followed by hashing at $M = 256$, with our previous best method obtained by combining hashing at $M = 128$ with an XOR-folding signature approach.[20] The red curve, corresponding to $M = 2$ hashing followed by $M = 256$ hashing, shows a remarkable speedup over the previous approach represented by the blue curve. For instance, at $t = 0.6$ the red curve is about 10 times faster than the linear search and about 5 times faster than the blue curve; at $t = 0.8$, the red curve is about 20 times faster than the linear search and about 3 times faster than the blue curve. Thus, the hashing framework allows one to gain speedups of at least 1 order of magnitude over a linear search and at least 3-fold over a previous state-of-the-art approach.

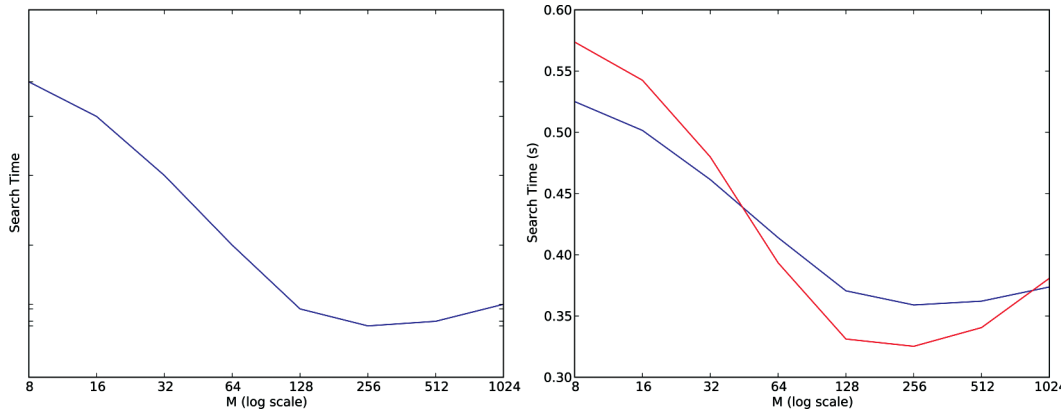

Figure 7. Left: quantity $Q = (1 - P_2) \times T_M + (1 - P_M) \times T_{\text{Tanimoto}}$ plotted as a function of M to examine at which M the minimum occurs. Right: the plot compares empirical search times (red) to the approximation formula of eq 16 with $\phi = 0.32$ (blue). The empirical results are obtained by averaging search times across all thresholds t for 100 random query molecules searched against a random sample of 100 000 ChemDB fingerprints encoded with lossless compression.

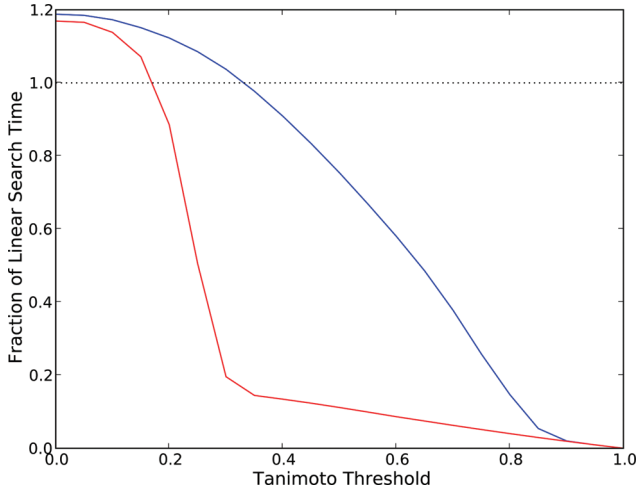

Figure 8. Curves showing the improvement in time over a linear search of the database. The timing results correspond to the average of 100 random ChemDB molecules queried against a random background of 100 000 molecules using a combination of $M = 1$ hashing with the XOR approach[20] (blue) and the hashing approach with $M = 2$ followed by $M = 256$ (red). Note that for very small thresholds ($t \leq 0.2$) of little general interest, the search time is worse than linear (which corresponds to the dotted line). This is because the amount of pruning is negligible, and therefore the extra computational cost associated with the signatures becomes significant.

DISCUSSION OF POSSIBLE EXTENSIONS

We describe some possible extensions of this work.

Adaptive Hashing. The selection of M that minimizes the search time can be derived per Tanimoto threshold, t, and/or per query size, A. Thus, in principle, different values of M could be used for different queries. The quantities P_2 and P_M in eq 16 can be derived using analytical formulas (or tabled from empirical results) as a function of t and A (see Figures 2 and 5). During a search, t and A are used to derive P_2 and P_M in order to select the value of M that minimizes the search time in eq 17. At the expense of memory storage, this technique requires multiple M-hashed signatures for each database fingerprint to be computed offline. For instance, after having preprocessed all signatures $\vec{b}^M$ for each database fingerprint, B, consider a query fingerprint $\vec{A}$ with $A = 205$. At Tanimoto threshold $t = 0.5$, $M = 128$ minimizes eq 16. This leads to obtaining $\vec{a}^{128}$ from $\vec{A}$ and computing $\sum_{i=0}^{128} \min(a_i^{128}, b_i^{128})$ to prune the database. At $t = 0.8$, $M = 16$ minimizes eq 16, and the database is thus pruned with the computation of $\sum_{i=0}^{16} \min(a_i^{16}, b_i^{16})$.

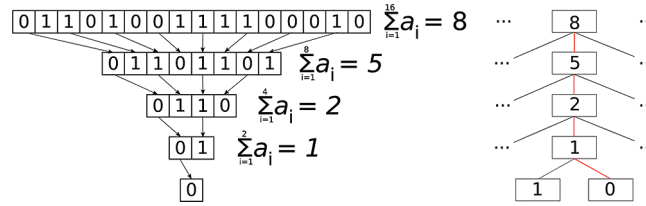

Figure 9. Left: an example of recursive hashing of a small 16-bit fingerprint. Right: the location in the tree data structure where the fingerprint is stored.

Recursive Hashing. Another approach is to use recursive hashing of fingerprints to build a tree data structure. At the top level, the fingerprints are binned as in the case of $M = 1$. Within each bin, the fingerprints are binned, as in the case of $M = 2$, according to the number of bits in half of the fingerprint components (example: odd-numbered components). This is repeated at the next level by binning according to the number of bits in half of the bits of the previous level. Figure 9 demonstrates this approach on a small 16-bit fingerprint. During a search, the query fingerprint is hashed, and the tree is traversed accordingly. At each level, the bins satisfying eq 7 and their subtrees are pruned. While this approach has the advantage of little overhead, the level of pruning achieved does not compare to approaches described above for $M > 4$. The level of pruning achieved by this approach is demonstrated by the dashed blue curve in Figure 5.

Alternative Partitions. Another approach is to partition the fingerprint components in a different way, allowing also for uneven partitions. Partitioning the components evenly by using their position modulo M is an approach that is consistent with the exchangeability assumption. However, the components of real fingerprints are not exchangeable. First, the bit probabilities of each component are far from identical and tend to follow a power-law distribution.[26] Second, there are both positive and negative correlations between the components. Thus, it may be possible to exploit these features to come up with even faster algorithms, for instance by using different ways of partitioning the components. Note that even with a different partition system, the intersection inequality can still be applied.

Alternative Signatures. In this paper, we have associated a signature of length M with each fingerprint by counting the number of 1-bits in each partition modulo M. An alternative approach is to organize the data into clusters offline with a representative center for each cluster and use signatures such that if the signature of the query and the signature of the center of a cluster lead to an unfavorable bound, the entire cluster can be eliminated from the search. This approach may also be useful to speed up existing techniques for queries consisting of a cluster of related molecules[27] by associating the set of query fingerprints with one center.

Extension to Other Similarity Measures. We used the Jaccard−Tanimoto similarity measure because it has been shown to be a good measure for molecular fingerprints and is the most widely used similarity measure in chemoinformatics systems. However, the methods described here can be extended to other similarity measures, as most similarity measures can also be expressed in terms of $A \cap B$ and $A \cup B$, as well as obvious terms such as A, B, and N.

CONCLUSION

In summary, a new approach has been developed for fingerprint-based chemical searches, which relies on computing for each fingerprint small signature vectors containing primarily the sums of hashed subsets of bits. The intersection inequality applied to these signatures provides efficient bounds on the intersection of two fingerprints, hence also on their Jaccard−Tanimoto similarity. During a database search, whenever the bound is unfavorable, the corresponding fingerprint can be pruned from the search. We have implemented and tested this approach using large sets of molecules and shown that there is good agreement between the theory and its predictions and the empirical data. With the proper data structure organization, this approach leads to speedups of 1 order of magnitude over a linear search.

ACKNOWLEDGMENT

Work supported by NIH Biomedical Informatics Training grant (LM-07443-01), NSF MRI grant (EIA-0321390), NSF grant 0513376, and a Microsoft Research Award to P.B. We would like to acknowledge discussions with Francesco Napolitano and Mingguo Li while they were visiting the Baldi laboratory. We would like to also acknowledge the OpenBabel open source project and OpenEye Scientific Software for its free software academic license.

APPENDIX

In this appendix, we derive the analytical formula for certain relevant distributions and for the amount of pruning under certain assumptions. In particular, certain calculations become more tractable under the assumption that the fingerprint components are exchangeable. Exchangeable is a weaker concept than independent—it basically means that the validity of any formula should remain unchanged under any permutation of the fingerprint components. It is clear that in most cases real fingerprint components are not exchangeable. In fact the different components do not even have similar distributions—some features may be very common, others very rare. However, in spite of the deviations from exchangeability, previous work[28] has shown that the exchangeability assumption leads to good global estimates of bulk properties, such as the distribution of the number B of 1-bits across all the molecules in a large database.

Estimating Distributions and Normal Approximations when $M = 2$. Here, we consider the case $M = 2$, with a query molecule A with signature (a_1, a_2) and $A = a_1 + a_2$. We have seen that for a given threshold t, the $M = 1$ filtering stage allows one to limit the search only to molecules of size B, with $At < B < A/t$. Under the exchangeability assumption, given a molecule with B 1-bits, the probability of having b_1 of these 1-bits associated with odd components is

$$P(b_1|N,B) = \frac{\binom{\lceil N/2 \rceil}{b_1}\binom{\lfloor N/2 \rfloor}{B-b_1}}{\binom{N}{B}} \tag{18}$$

To simplify notations, from now on we assume that N is even (as in most chemoinformatics systems), $N = 2P$, so that

$$P(b_1|N,B) = \frac{\binom{P}{b_1}\binom{P}{B-b_1}}{\binom{N}{B}} = \frac{\binom{P}{b_1}\binom{P}{b_2}}{\binom{N}{B}} \tag{19}$$

This is a hypergeometric distribution with mean $\mu_B = PB/N = B/2$, mode $(B+1)(P+1)/(N+2)$, and variance $\sigma_B^2 = [B(P/N)(1-P/N)(N-B)]/(N-1) = [B(N-B)]/[4(N-1)]$. Furthermore, it is well-known that if B is large, and N is large compared to B, then this hypergeometric distribution can be well approximated by a corresponding Normal distribution with mean μ_B and variance σ_B^2. If there are D fingerprints in the database, then we have seen that the number of fingerprints of size B is typically given by $Dg(B)$, where g is Normal. The expected number of fingerprints of type b_1, b_2 with $B = b_1 + b_2$ is then given by

$$Dg(B)\frac{\binom{P}{b_1}\binom{P}{B-b_1}}{\binom{N}{B}} = Dg(B)\frac{\binom{P}{b_1}\binom{P}{b_2}}{\binom{N}{B}} \tag{20}$$

By the intersection inequality, we have seen that for a given B such that $At < B < A/t$, we need only to retain molecules with signature (b_1, b_2), satisfying

$$a_1 - \frac{A - tB}{1+t} < b_1 < a_1 + \frac{B - tA}{1+t} \tag{21}$$

Thus for all molecules B with B 1-bits that pass the first test (i.e., $tA < B < A/t$), we can discard a fraction of them that is approximately given by

$$1 - (\Phi(X_2) - \Phi(X_1)) \tag{22}$$

where Φ is the distribution function of the normalized Normal distribution with mean 0 and variance 1:

$$X_1 = \left(\left[a_1 - \frac{A - tB}{1 + t}\right] - \frac{B}{2}\right)/\sqrt{B(N - B)/4(N - 1)} \approx \left(\left[a_1 - \frac{A - tB}{1 + t}\right] - \frac{B}{2}\right)/\sqrt{B/4} \quad (23)$$

and

$$X_2 = \left(\left[a_1 + \frac{B - tA}{1 + t}\right] - \frac{B}{2}\right)/\sqrt{B(N - B)/4(N - 1)} \approx \left(\left[a_1 + \frac{B - tA}{1 + t}\right] - \frac{B}{2}\right)/\sqrt{B/4} \quad (24)$$

Of course, this is just another way of writing eq 12. As $x \rightarrow \infty$, the distribution of the normalized Normal can be approximated by

$$1 - \Phi(x) \approx \frac{1}{x}\phi(x) = \frac{1}{\sqrt{2\pi}}\frac{1}{x}e^{-x^2/2} \quad (25)$$

Thus, in the regime where X_1 is small enough and X_2 is large enough, the fraction of discarded molecules can be approximated by

$$\frac{1}{\sqrt{2\pi}}\left[\frac{1}{X_1}e^{-X_1^2/2} + \frac{1}{X_2}e^{-X_2^2/2}\right] \quad (26)$$

Equations 22−26 show the derivation of the fraction of molecules to be pruned for a given B satisfying $At < B < A/t$. One can compute the expression above for all values of B and then weigh the results by the proportion of molecules with B 1-bits in the database. As already mentioned, this proportion is well approximated by a Normal distribution g with mean μ_D and variance σ_D^2. Thus, combining the two filtering stages, the total fraction of pruning can be approximated by

$$\int_{At}^{A/t} g(u)\,du + \int_{AT}^{A/t} g(u)\frac{1}{\sqrt{2\pi}}\left[\frac{1}{X_1}e^{-X_1^2/2} + \frac{1}{X_2}e^{-X_2^2/2}\right] \quad (27)$$

Of course this equation is similar to eq 14.

Estimating the Amount of Pruning in the General Case: Estimation Using Independence. In this section, we assume that the components of the signatures are generated independently of each other. In other words, for each class modulo M, we sample independently from the same fixed distribution, typically a Normal or Poisson distribution, to determine the number of 1-bits in a_i^M or b_i^M for the query or the molecules in the database. The 1-bits are then assigned randomly and uniformly to the fingerprint positions associated with that class. The parameters of the distribution used to generate the number of 1-bits in a given class can be tuned in order to model all the fingerprints in the database, or only the fingerprints with a particular size (total number of 1-bits). Given a typical query $\vec{A}$ with A 1-bits, one would like to understand what fraction of the database can be pruned from the Tanimoto similarity search at a given similarity threshold t. As an approximation, assume that the signature of $\vec{A}$ is generated by the process above using a Normal with mean A/M and variance $\sigma_{A/M}^2$. For cases of $A/M \leq 3$, a Poisson distribution with $\lambda = A/M$ gives a better approximation to the data. Note that the resulting vector does not necessarily have exactly A 1-bits but in general will be close enough to cause only minor fluctuations in the formula. Consider now another molecule $\vec{B}$. Assume that it was generated by a similar process using a Normal with mean B/M and variance $\sigma_{B/M}^2$. At a given threshold t, we know that B can be pruned off the search if $S = \sum_i \min(a_i, b_i) \leq t(A + B)/(1 + t)$. Thus, to estimate how many molecules with approximately B 1-bits are eliminated, we need to estimate the corresponding probability. In the current framework, the sum $S = \sum_i \min(a_i, b_i)$ can be viewed as the sum of M i.i.d. random variables, and therefore, by the central limit theorem, it approaches a Normal distribution when M is large. In what follows, we use g to denote (Normal) density functions and G to denote the corresponding distributions. In particular, we have

$$P\left(\sum_i \min(a_i, b_i) \leq \frac{t(A + B)}{1 + t}\right) \approx \int_{-\infty}^{t(A+B)/1+t} g_{\mu\sigma^2}(u)\,du \quad (28)$$

where the mean μ and the variance σ^2 depend on A, B, M, $\sigma_{A/M}^2$, and $\sigma_{B/M}^2$.

Because $S \geq 0$, the lower bound of the integral can be set to 0. This should not affect the results significantly, especially for large values of M. Equation 28 can then be integrated over B to get the fraction of pruning p_A associated with A. Then, p_A can be integrated over A to get the average fraction of pruning across queries.

If the sizes of the query molecules have the same distribution as the sizes of the molecules in the database, then we can take $A/M = B/M = \mu_D/M$, where μ is the average size in the entire database, and $\sigma_{A/M}^2 = \sigma_D^2/M$, where σ_D^2 is the variance of the size in the entire database, and use eq 28 to estimate the probability of pruning for a random query versus a random molecule, which ought to give also an estimate of the average total amount of pruning.

Finally, to estimate the values of μ and σ^2 in eq 28, we use the fact that the random variables $\min_i(a_i, b_i)$ are i.i.d. to obtain

$$\mu = ME[\min(a_i, b_i)] \text{ and } \sigma^2 = MVar[\min(a_i, b_i)] \quad (29)$$

Given that in this approximation a_i and b_i are Normal random variables, it is easy to derive expressions for the density and cumulative function of the minimum and compute its mean and variance. Let g_a and G_a here denote the Normal (or Poisson) density and cumulative function for a_i, and similarly for b_i. Again, the means and standard deviations of these Normal densities depend on whether one wants to fix A and let B vary, or vice versa, or integrate over all queries and all molecules. But in all cases they can be estimated, and in each case, the density of the minimum is given by

$$f_{\min}(x) = g_a(x)(1 - G_b(x)) + g_b(x)(1 - G_a(x)) \quad (30)$$

$$F_{\min}(x) = P(\min(a_i, b_i) \leq x) = \int_{-\infty}^{x} f_{\min}(u)\,du = 1 - (1 - G_a(u))(1 - G_b(u)) \quad (31)$$

$$= G_a(u) + G_b(u) - G_a(u)G_b(u) \quad (32)$$

From which one can get analytic expressions for $E[\min(a_i, b_i)] = \int_{-\infty}^{+\infty} x f_{\min}(x)\,dx$ and $Var[\min(a_i, b_i)] = \int_{-\infty}^{+\infty}(x - E[\min(a_i, b_i)])^2 f_{\min}(x)\,dx$.

Estimation Using Exchangeability. Modeling fingerprint signatures associated with fingerprints containing A 1-bits with M independent Normal distributions, each with mean A/M, is not quite accurate since (1) it ignores dependencies between the signature components (or classes modulo M) and (2) in general it produces fingerprints that do not have exactly A 1-bits, when used in generative mode. Thus, a more precise approach is to use a multivariate hypergeometric distribution where A bits (or balls) are put into M classes (or boxes). The mean, variance, and covariance of each resulting variable a_i are known and given by

$$E(a_i) = \frac{A}{M} \tag{33}$$

$$Var(a_i) = \frac{A}{M}\left(1 - \frac{1}{M}\right)\frac{N - A}{N - 1} \tag{34}$$

$$Cov(a_i, a_j) = -\frac{A}{M^2}\frac{N - A}{N - 1} \tag{35}$$

Thus, to get a better estimate of the variance σ^2 in eq 29 in the fixed-A case, we incorporate the covariance between the pairwise $(\min(a_i, b_i), \min(a_j, b_j))$:

$$Var\left(\sum_i \min(a_i, b_i)\right) = \sum_i Var(\min(a_i, b_i)) + 2\sum_{i<j} Cov(\min(a_i, b_i)\min(a_j, b_j)) \tag{36}$$

Using the exchangeability gives

$$Var\left(\sum_i \min(a_i, b_i)\right) = \sum_i Var(\min(a_i, b_i)) + \frac{M(M-1)}{2} Cov(\min(a_i, b_i)\min(a_j, b_j)) \tag{37}$$

The covariance term in eq 37 can be estimated empirically, and possibly analytically under a Normal approximation, where a_i and a_j are assumed to have identical, but correlated, Normal distributions, and similarly for b_i and b_j.

REFERENCES AND NOTES

(1) Chen, J.; Swamidass, S. J.; Dou, Y.; Bruand, J.; Baldi, P. ChemDB: a public database of small molecules and related chemoinformatics resources. *Bioinformatics* **2005**, *21*, 4133–4139.

(2) Irwin, J. J.; Shoichet, B. K. ZINC--A Free Database of Commercially Available Compounds for Virtual Screening. *J. Chem. Inf. Comput. Sci.* **2005**, *45*, 177–182.

(3) Chen, J.; Linstead, E.; Swamidass, S. J.; Wang, D.; Baldi, P. ChemDB Update--Full Text Search and Virtual Chemical Space. *Bioinformatics* **2007**, *23*, 2348–2351.

(4) Wang, Y.; Xiao, J.; Suzek, T.; Zhang, J.; Wang, J.; Bryant, S. PubChem: a public information system for analyzing bioactivities of small molecules. *Nucleic Acids Res.* **2009**, *37*, W623–W633.

(5) Sayers, E.; Barrett, T.; Benson, D.; Bolton, E.; Bryant, S.; Canese, K.; Chetvernin, V.; Church, D.; DiCuccio, M.; Federhen, S.; et al. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* **2010**, *38*, D5–D16.

(6) Fligner, M. A.; Verducci, J. S.; Blower, P. E. A Modification of the Jaccard/Tanimoto Similarity Index for Diverse Selection of Chemical Compounds Using Binary Strings. *Technometrics* **2002**, *44*, 110–119.

(7) Flower, D. R. On the Properties of Bit String-Based Measures of Chemical Similarity. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 379–386.

(8) James, C. A.; Weininger, D.; Delany, J. *Daylight Theory Manual*; Daylight Chemical Information Systems, Inc.: Aliso Viejo, CA, 2004. Available at http://www.daylight.com/dayhtml/doc/theory/ (accessed June 10, 2010).

(9) Xue, L.; Godden, J. F.; Stahura, F. L.; Bajorath, J. Profile scaling increases the similarity search performance of molecular fingerprints containing numerical descriptors and structural keys. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1218–1225.

(10) Xue, L.; Stahura, F. L.; Bajorath, J. Similarity search profiling reveals effects of fingerprint scaling in virtual screening. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 2032–2039.

(11) Leach, A. R.; Gillet, V. J. *An Introduction to Chemoinformatics*; Springer: Dordrecht, The Netherlands, 2005.

(12) Baldi, P.; Benz, R. W.; Hirschberg, D.; Swamidass, S. Lossless Compression of Chemical Fingerprints Using Integer Entropy Codes Improves Storage and Retrieval. *J. Chem. Inf. Model.* **2007**, *47*, 2098–2109.

(13) Hert, J.; Willett, P.; Wilton, D. J.; Acklin, P.; Azzaoui, K.; Jacoby, E.; Schuffenhauer, A. Comparison of topological descriptors for similarity-based virtual screening using multiple bioactive reference structures. *Org. Biomol. Chem.* **2004**, *2*, 3256–3266.

(14) Bender, A.; Mussa, H.; Glen, R.; Reiling, S. Similarity Searching of Chemical Databases Using Atom Environment Descriptors (MOLPRINT 2D): Evaluation of Performance. *J. Chem. Inf. Model.* **2004**, *44*, 1708–1718.

(15) Hassan, M.; Brown, R. D.; Varma-O'Brien, S.; Rogers, D. Cheminformatics analysis and learning in a data pipelining environment. *Mol. Diversity* **2006**, *10*, 283–299.

(16) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742–754.

(17) Holliday, J. D.; Hu, C. Y.; Willett, P. Grouping of coefficients for the calculation of inter-molecular similarity and dissimilarity using 2D fragment bit-strings. *Comb. Chem. High Throughput Screen.* **2002**, *5*, 155–166.

(18) Swamidass, S.; Baldi, P. Bounds and Algorithms for Exact Searches of Chemical Fingerprints in Linear and Sub-Linear Time. *J. Chem. Inf. Model.* **2007**, *47*, 302–317.

(19) Swamidass, S.; Baldi, P. A Mathematical Correction for Fingerprint Similarity Measures to Improve Chemical Retrieval. *J. Chem. Inf. Model.* **2007**, *47*, 952–964.

(20) Baldi, P.; Hirschberg, D. S.; Nasr, R. J. Speeding Up Chemical Database Searches Using a Proximity Filter Based on the Logical Exclusive-OR. *J. Chem. Inf. Model* **2008**, *48*, 1367–1378.

(21) Burkhard, W.; Keller, R. Some approaches to best-match file searching. *Commun. ACM* **1973**, *16*, 230–236.

(22) Shapiro, M. The choice of reference points in best-match file searching. *Commun. ACM* **1977**, *20*, 339–343.

(23) Shasha, D.; Wang, T. New techniques for best-match retrieval. *ACM Trans. Inf. Syst.* **1990**, *8*, 140–158.

(24) Baldi, P.; Hirschberg, D. An Intersection Inequality Sharper than the Tanimoto Triangle Inequality for Efficiently Searching Large Databases. *J. Chem. Inf. Model.* **2009**, *49*, 1866–1870.

(25) Feller, W. *An Introduction to Probability Theory and its Applications*, 3rd ed.; John Wiley & Sons: New York, 1968; Vol. 1.

(26) Benz, R. W.; Swamidass, S. J.; Baldi, P. Discovery of Power-Laws in Chemical Space. *J. Chem. Inf. Model.* **2008**, *48*, 1138–1151.

(27) Nasr, R.; Swamidass, S. J.; Baldi, P. Large scale study of multiple-molecule queries. *J. Cheminf.* **2009**, *1*, 7.

(28) Baldi, P.; Nasr, R. When is Chemical Similarity Significant? The Statistical Distribution of Chemical Similarity Scores and Its Extreme Values. *J. Chem. Inf. Model.* **2010**, *50*, 1205−1222.

CI100132G