An Intersection Inequality Sharper than the Tanimoto Triangle Inequality for Efficiently Searching Large Databases

Pierre Baldi, and Daniel S. Hirschberg*

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

E-mail: pfbaldi@ics.uci.edu

Abstract

Bounds on distances or similarity measures can be useful to help search large databases efficiently. Here we consider the case of large databases of small molecules represented by molecular fingerprint vectors with the Tanimoto similarity measure. We derive a new intersection inequality which provides a bound on the Tanimoto similarity between two fingerprint vectors and show that this bound is considerably sharper than the bound associated with the triangle inequality of the Tanimoto distance. The inequality can be applied to other intersection-based similarity measures. We introduce a new integer representation which relies on partitioning the fingerprint components, for instance by taking components modulo some integer M, and reporting the total number of 1-bits falling in each partition. We show how the intersection inequality can be generalized immediately to these integer representations and used to search large databases of binary fingerprint vectors efficiently.

Introduction

Bounds on distances or similarity measures can be useful for searching large databases.[1–5] In a fairly typical similarity search situation, if item $\mathscr{A}$ is close to item $\mathscr{B}$ and item $\mathscr{C}$ is far from item $\mathscr{A}$, then one would like to infer that $\mathscr{C}$ is also far from $\mathscr{B}$, thereby allowing one to prune the search space and ignore $\mathscr{B}$. In particular, we consider the typical chemoinformatics situation where a large database of molecules represented by binary fingerprint vectors of fixed length N must be searched using the Tanimoto similarity measure.[6–9] Given a molecule $\mathscr{A}$, we denote its fingerprint vector by $\vec{A}$. The Tanimoto similarity measure is given by

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

Here and everywhere else we use A, $A\cap B$, $A\cup B$ … to denote the number of 1-bits in the vectors (or sets) $\vec{A}$, $\vec{A}\cap\vec{B}$, $\vec{A}\cup\vec{B}$ …. It is well known that $D(\vec{A},\vec{B}) = 1 - T(\vec{A},\vec{B})$ is a distance.[10,11] Thus, using the triangle inequality we can bound the distance $D(\vec{B},\vec{C})$ by

$$|D(\vec{A},\vec{B}) - D(\vec{A},\vec{C})| \leq D(\vec{B},\vec{C}) \leq D(\vec{A},\vec{C}) + D(\vec{A},\vec{B}) \tag{2}$$

from which one obtains the following bound on the similarity $T(\vec{B},\vec{C})$

$$T(\vec{B},\vec{C}) \leq 1 - |T(\vec{A},\vec{B}) - T(\vec{A},\vec{C})| \tag{3}$$

Our goal here is to first derive another general bound on $T(\vec{B},\vec{C})$ and then study its relationship to the triangle inequality. We then further generalize this bound and show how it can be applied to efficiently search large databases of chemical fingerprint vectors.

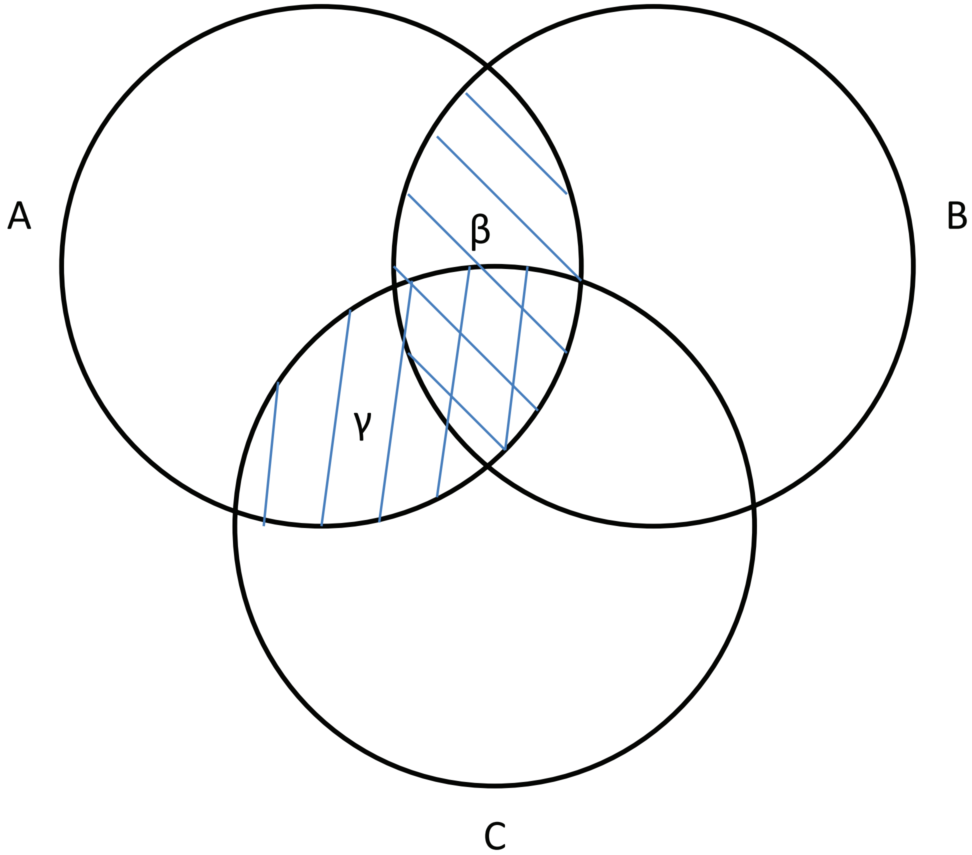
The Intersection Bound

As can be seen in the Venn Diagram of Figure 1, the following relation holds:

$$\vec{B} \cap \vec{C} = (\vec{B} \cap \vec{C} \cap \vec{A}) \cup [(\vec{B} - \vec{A}) \cap (\vec{C} - \vec{A})] \tag{4}$$

By letting $A \cap B = \beta$ and $A \cap C = \gamma$

Figure 1: Venn Diagram. The three sets represent the 1-bit components of three fingerprint vectors associated with three molecules $\mathscr{A}$, $\mathscr{B}$, and $\mathscr{C}$. $\beta = A \cap B$ and $\gamma = A \cap C$. We are interested in estimating or bounding the Tanimoto similarity between $\vec{B}$ and $\vec{C}$, as a function of $\vec{A}$ and its similarity to $\vec{B}$ and $\vec{C}$.



$$|\vec{B} \cap \vec{C} \cap \vec{A}| \leq \min(A \cap B, A \cap C) = \min(\beta, \gamma) \tag{5}$$

and

$$|(\vec{B}-\vec{A})\cap(\vec{C}-\vec{A})| \leq \min[B-(A\cap B),C-(A\cap C)] = \min(B-\beta,C-\gamma) \tag{6}$$

Combining these inequalities, we get

$$B\cap C \leq \min(A\cap B,A\cap C)+\min[C-(A\cap C),B-(A\cap B)] = \min(\beta,\gamma)+\min(B-\beta,C-\gamma) \tag{7}$$

Since $T(\vec{B},\vec{C}) = (B\cap C)/(B\cup C) = (B\cap C)/(B+C-(B\cap C))$,we finally have

$$T(\vec{B},\vec{C}) \leq \frac{\min(\beta,\gamma)+\min(B-\beta,C-\gamma)}{B+C-\min(\beta,\gamma)-\min(B-\beta,C-\gamma)} \tag{8}$$

which we call the intersection bound or intersection inequality.

The Intersection Inequality is Sharper than the Triangle Inequality

In this section, we prove that the intersection inequality is sharper than the triangle inequality. More specifically we have the following theorem.

Theorem: For any fingerprints $\vec{A}$, $\vec{B}$, and $\vec{C}$, we have the inequality

$$T(\vec{B},\vec{C}) \leq \frac{\min(\beta,\gamma)+\min(B-\beta,C-\gamma)}{B+C-\min(\beta,\gamma)-\min(B-\beta,C-\gamma)} \leq 1-\left|\frac{\beta}{A\cup B}-\frac{\gamma}{A\cup C}\right| \tag{9}$$

Furthermore the inequality is strict, except in degenerate cases detailed in the proof where: (1) at least one of the fingerprints is equal to $\vec{0}$; or (2) $\vec{A}\subset\vec{B}$ or $\vec{A}\subset\vec{C}$; or (3) $\vec{B}$ and $\vec{C}$ are symmetric with respect to $\vec{A}$ (i.e. $B=C$ and $\beta=\gamma$). Furthermore, if two of the fingerprints are identical ($\vec{A}=\vec{B}$ or $\vec{B}=\vec{C}$ or $\vec{A}=\vec{C}$) then the two bounds are equal and exact.

Proof: First note that B and C play entirely symmetric roles and this symmetry can be used to reduce the number of cases that need to be examined. Second consider situations where some of the denominators in Equation 9 are equal to 0. If $A \cup B = 0$, then $A = B = 0$ and similarly for $A \cup C = 0$. The denominator on the left hand side is 0 if and only if $B = 0$ or $C = 0$. Cases where $A = 0$, or $B = 0$, or $C = 0$ can be examined directly and are summarized in Table 1 and Table 2.

Table 1: Table of possibilities with $A = 0$. In each case, the left entry is the value of the bound given by the left-hand side of Equation 9 and the right-hand side is the bound resulting from the triangle inequality (right-hand side of Equation 9). Note that when $B \neq 0$ and $C \neq 0$, the two bounds are identical if and only if $B = C = 0$. Otherwise, if $B \neq C$, then the intersection bound is strictly better than the triangle inequality.

	$B = 0$	$B \neq 0$
$C = 0$	$(1,1)$	$(0,0)$
$C \neq 0$	$(0,0)$	$(\frac{\min(B,C)}{\max(B,C)}, 1)$

Table 2: Table of possibilities with $A \neq 0$. In each case, the left entry is the value of the bound given by the left-hand side of Equation 9 and the right-hand side is the bound resulting from the triangle inequality (right-hand side of Equation 9). Note that when $C = 0$ and $B \neq 0$, the bounds are identical if and only if $\vec{A} = \vec{B}$. Otherwise, if $\vec{A} \neq \vec{B}$, then the intersection bound is strictly better than the triangle inequality. Similar considerations hold for the symmetric case ($B = 0$ and $C \neq 0$).

	$B = 0$	$B \neq 0$
$C = 0$	$(1,1)$	$(0, 1 - \frac{\beta}{A \cup B})$
$C \neq 0$	$(0, 1 - \frac{\gamma}{A \cup C})$	main case

Thus in the rest of the proof we can assume that $A \neq 0$, $B \neq 0$, and $C \neq 0$ and that none of the denominators in Equation 9 is equal to 0. To aggregate the fractions in the right-hand side of Equation 9, it is convenient to parameterize the problem by assuming that $A \cup B = k(A \cup C)$ where k is a strictly positive real number. Without any loss of generality, we can assume that $k \geq 1$. By the definition of k, we have $A + B - \beta = k(A + C - \gamma)$. This can be rewritten as:

$$B - C = (k-1)(A+C) + \beta - k\gamma \tag{10}$$

and also as:

$$B - \beta = (k-1)(A \cup C) + C - \gamma \tag{11}$$

Equation 11 shows that under our assumptions $B - \beta \geq C - \gamma$, with equality possible only if and only if $k = 1$. By noticing that $A + C \geq 2\gamma$, we also have

$$B - C \geq k\gamma - 2\gamma + \beta \tag{12}$$

Now we can write Equation 9 as:

$$\frac{\min(\beta,\gamma) + \min(B-\beta, C-\gamma)}{B + C - \min(\beta,\gamma) - \min(B-\beta, C-\gamma)} \leq 1 - \frac{|\beta - k\gamma|}{A \cup B} = 1 - \frac{\max(\beta, k\gamma) - \min(\beta, k\gamma)}{A \cup B} \tag{13}$$

By assumption here none of the denominators is equal to 0, and we have seen that $\min(B-\beta, C-\gamma) = C - \gamma$. Thus after reducing to common denominators and rearranging the terms, we need to prove the equivalent relation given by:

$$[B - \min(\beta,\gamma) + \gamma][\max(\beta,k\gamma) - \min(\beta,k\gamma)] \leq [A \cup B][B - C + 2\gamma - 2\min(\beta,\gamma)] \tag{14}$$

Three cases can be distinguished depending on the size of β relative to γ and $k\gamma$. It is convenient to treat the special symmetric case $k = 1$ separately, and we begin with that.

Case 1: $k = 1$**.** In this case, Equation 10 yields $B - \beta = C - \gamma$. Two subcases can be considered.

- In the most symmetric subcase, $\gamma = \beta$ which implies that $B = C$, and both bounds are equal to 1.
- Alternatively, without any loss of generality, assume that $\gamma < \beta$. Then Equation 14 becomes $B(\beta - \gamma) \leq (A \cup B)(B - C)$. Since here $B - C = \beta - \gamma$, the inequality simplifies to $B \leq A \cup B$ which is always true. Furthermore, equality is achieved only when $\vec{A} \subset \vec{B}$ in which case both

bounds are equal to C/B and are not necessarily exact.

Thus in the rest of the proof, we can assume that $k > 1$ (in addition to $A \neq 0$, $B \neq 0$, and $C \neq 0$).

Case 2: $k\gamma \leq \beta$. In this case, Equation 14 yields:

$$B[\beta - k\gamma] \leq [A \cup B][B - C] \tag{15}$$

This inequality is true because $B \leq A \cup B$, and $\beta - k\gamma \leq B - C$ by Equation 10. Under the current assumptions, equality in Equation 15 cannot be achieved since equality could occur only in the following two situations:

- $\beta = k\gamma$ and $B = C$. But then Equation 10 would imply $k = 1$, contradicting the assumption $k > 1$.

- $B = A \cup B$ and $\beta - k\gamma = B - C$, which also implies $k = 1$.

Case 3: $\gamma \leq \beta \leq k\gamma$. In this case, Equation 14 yields:

$$B[k\gamma - \beta] \leq [A \cup B][B - C] \tag{16}$$

Again this inequality is true because $B \leq A \cup B$, and $k\gamma - \beta \leq B - C$ by Equations 10 and 12. Equality between the bounds in Equation 16 can only be achieved when $\vec{A} = \vec{C} \subset \vec{B}$. In this case, both bounds are exact and equal to C/B. To see this, under the current assumptions, equality could occur only in the following two situations:

- $\beta = k\gamma$ and $B = C$. But then Equation 10 would imply $k = 1$, contradicting the assumption $k > 1$.

- $B = A \cup B$ and $k\gamma - \beta = B - C$. But then $\vec{A} \subset \vec{B}$ and Equation 12 would imply $\gamma \geq \beta$, hence $\beta = \gamma = A$. In turn, this would yield $B = A \cup B = k(A \cup C) = kC$ which combined with $k\gamma - \beta = B - C$ forces $C = \gamma$ and therefore $\vec{C} = \vec{A} \subset \vec{B}$. In this case, we have $T(\vec{B}, \vec{C}) = C/B$ and this is the value assumed by both the intersection inequality and the triangle inequality and they are both exact.

Case 4: $\beta \leq \gamma$. In this case, Equation 14 yields:

$$[B-\beta+\gamma][k\gamma-\beta] \leq [A\cup B][B-C+2\gamma-2\beta] \tag{17}$$

Again this inequality is true because $B-\beta+\gamma \leq A\cup B$, and $k\gamma-\beta \leq B-C+2\gamma-2\beta$ by Equations 10 and 12. Equality between the bounds in Equation 16 can only be achieved when $\vec{A}=\vec{C}$. In this case, both bounds are exact and equal to $T(\vec{B},\vec{C}) = T(\vec{A},\vec{B}) = \beta/(A\cup B) = \beta/(A\cup C)$. To see this, under the current assumptions, equality could occur only in the following two situations:

- $B-C=2\beta-2\gamma$. Using Equation 12, this would imply $\beta=k\gamma$ which is possible here only if $\beta=\gamma=0$ and $B=C$. But this implies $A\cup B = A\cup C$ which contradicts $k>1$
- $A\cup B = B-\beta+\gamma$ and $k\gamma-\beta = B-C+2\gamma-2\beta$. The first equality yields $A=\gamma$, hence $\vec{A}\subset\vec{C}$. Combining this with $A\cup B = k(A\cup C)$ gives $C=\gamma$, and so $\vec{A}=\vec{C}$. When $\vec{A}=\vec{C}$, the intersection and triangle inequality bounds are identical and exact and equal to $\beta/(A\cup B)$.

Additional Properties and Generalization of the Intersection Inequality

The following properties can be derived immediately.

Property 1. The intersection bound does not depend on any bits in $\vec{A}$ that are not in common with $\vec{B}$ or $\vec{C}$.

Property 2. The intersection bound can be applied to many other similarity measures $S(\vec{A},\vec{B})$ for binary fingerprints. Although the Tanimoto measure is by far the most widely used measure, two dozen or so other measures have been described in the literature.[12] All these measures involve algebraic expressions that explicitly depend on the intersection $A\cap B$. Thus, with the proper adjustments, bounds on the intersection naturally lead to bounds on these other similarity measures.

Property 3. In general, the intersection inequality is considerably sharper than the triangle inequality. We have seen in the theorem above that the intersection bound is always better than the triangle inequality, except in some degenerate cases. But is it much better and worth using? The answer to this question is clearly yes as illustrated by the following two examples.

Example 1: Consider the case where the three sets $\vec{A}$, $\vec{B}$, and $\vec{C}$ are disjoint, and $A \neq 0$. Then Equation 9 gives

$$T(\vec{B},\vec{C}) \leq \frac{\min(B,C)}{\max(B,C)} \leq 1 \tag{18}$$

In other words, in this case the triangle inequality bound is useless and equal to 1. The intersection bound is much sharper and can diverge maximally from the triangle inequality bound by being as small as 0 when $\min(B,C) = 0$ and $\max(B,C) \neq 0$.

Example 2: Consider now a typical case (e.g. Daylight fingerprint system with OR-folding lossy compression) where fingerprints of length $N = 512$ are used. Depending on various implementation details, the average number of 1-bits per fingerprint vector could be something like 250. Consider then two fairly typical fingerprints with $B = 150$ and $C = 350$. Note that to derive a bound on $T(\vec{B},\vec{C})$, we are free to choose the vector $\vec{A}$ (and only the vectors of all 0-bits or all 1-bits can be used to compute bounds, without making additional assumptions on $\vec{B}$ and $\vec{C}$). Consider the choice $\vec{A} = \vec{0}$. Then Equation 9 yields

$$T(\vec{B},\vec{C}) \leq \frac{150}{350} \leq 1 \tag{19}$$

Thus in this fairly typical case the triangle inequality gives a useless bound equal to 1, whereas the intersection inequality gives a much sharper bound equal to 3/7. If, for instance, B was the query and we were interested in finding molecules with a Tanimoto similarity of at least 0.5, the intersection inequality would allow us to immediately discard the molecule $\vec{C}$ from the search, without having to compute precisely the value of $T(\vec{B},\vec{C})$. This is of course also the basic idea used in[5] and generalized below in the Section on Applications to Efficient Database Searches.

Generalized Triangle Inequality

Let us use the notation $\vec{A^C}$ to denote the complement of $\vec{A}$. i.e. $\vec{A^C}$ is the vector obtained by replacing the 0-bits of $\vec{A}$ with 1-bits, and the 1-bits of $\vec{A}$ with 0-bits. Then, with the usual notation, Equation 7 can also be written as

$$B \cap C \leq \min(A \cap B, A \cap C) + \min(A^C \cap B, A^C \cap C) \tag{20}$$

More generally, if $A_1, \ldots, A_M$ denotes any family of disjoint subsets of components ($A_i \cap A_j = 0$ for any i and j) then we have immediately the generalized triangle inequality

$$B \cap C \leq \sum_{i=1}^{M} \min(A_i \cap B, A_i \cap C) \tag{21}$$

In particular, this is true for any partition $A_1, \ldots, A_M$ of the entire set of components. Thus one important question is how to choose the partition in order to get good bounds. Below we further investigate the case where the partition is obtained by considering the fingerprint components modulo some integer M.

Applications to Efficient Database Searches: the Modulo Hashing Approach

Let M be any integer between 1 and N and assume that to any molecule $\mathscr{A}$ and binary fingerprint $\vec{A}$, we also attach a new integer vector of counts $\vec{a} = (a_1, \ldots, a_M)$ of length M. The count a_i represents how many 1-bits in $\vec{A}$ fall on components that are congruent to i modulo M. For instance, if $\vec{A} = (1,0,1,1,0,0)$ and $M = 3$, then $\vec{a} = (2,0,1)$ because there are two 1-bits in positions congruent to 1 modulo 3 (1st and 4th), zero 1-bits in positions congruent to 2 modulo 3 (2nd and 5th), and one 1-bit in positions congruent to 3 modulo 3 (3rd and 6th).

If we consider two fingerprint vectors $\vec{B}$ and $\vec{C}$, with their respective count vectors $\vec{b}$ and $\vec{c}$, then $\vec{B}$ and $\vec{C}$ can have at most $\min(b_i, c_i)$ bits in common at positions that are congruent with i modulo M. Thus in this case the intersection inequality of Equation 21 can be written

$$B \cap C \leq \sum_{i=1}^{M} \min(b_i, c_i) \tag{22}$$

resulting in the following bound on the Tanimoto similarity

$$T(\vec{B}, \vec{C}) \leq \frac{\sum_{i=1}^{M} \min(b_i, c_i)}{B + C - \sum_{i=1}^{M} \min(b_i, c_i)} \tag{23}$$

Suppose that $\vec{B}$ is the query fingerprint and that we are interested in retrieving from the database all the molecules $\vec{C}$ satisfying $T(\vec{B}, \vec{C}) > t$, i.e. with a Tanimoto similarity of at least t to the query. Then, using the intersection bound, we can discard from the search all the molecules $\vec{C}$ satisfying

$$\frac{\sum_{i=1}^{M} \min(b_i, c_i)}{B + C - \sum_{i=1}^{M} \min(b_i, c_i)} \leq t \tag{24}$$

or equivalently

$$\sum_{i=1}^{M} \min(b_i, c_i) \leq \frac{t}{1+t}(B + C) \tag{25}$$

Thus, in summary, if we are searching a database of molecules with a query $\vec{B}$ and a Tanimoto similarity threshold t, we can prune the search by removing all the molecules that satisfy Equation 25 and are thereby guaranteed to have a degree of similarity to $\vec{B}$ that is below the threshold t. To show that this corresponds to a very significant amount of pruning, we consider in more details the cases of $M = 1$ and $M = 2$.

Case M=1: This is exactly the case described and used in.[5] In this case, Equation 25 implies that we can discard all the molecules $\vec{C}$ with

$$C \leq tB \quad \text{or} \quad C \geq B/t \tag{26}$$

Using the same example as above with fingerprints of length $N = 512$, consider a typical query

$\vec{B}$ with $B = 300$ and a similarity threshold of $t = 0.8$. Then, using the intersection inequality, we can immediately discard from the search all the molecules $\vec{C}$ with $C \leq 240 = 0.8 \times 300$ or $C \geq 375 = 300/0.8$.

More generally, Figure 2 shows the significant amount of database pruning achieved by the intersection inequality with $M = 1$ in a practical setting, for different Tanimoto similarity thresholds t and different queries. The curves are derived using a random subset of 50,000 molecules from the ChemDB database[8,13] using Daylight-style fingerprints of length 512 (with folded-OR lossy compression). For example, for a similarity threshold of $t = 0.8$ and a query containing $B = 200$ 1-bits, on average 80% of the database can be pruned from the search.

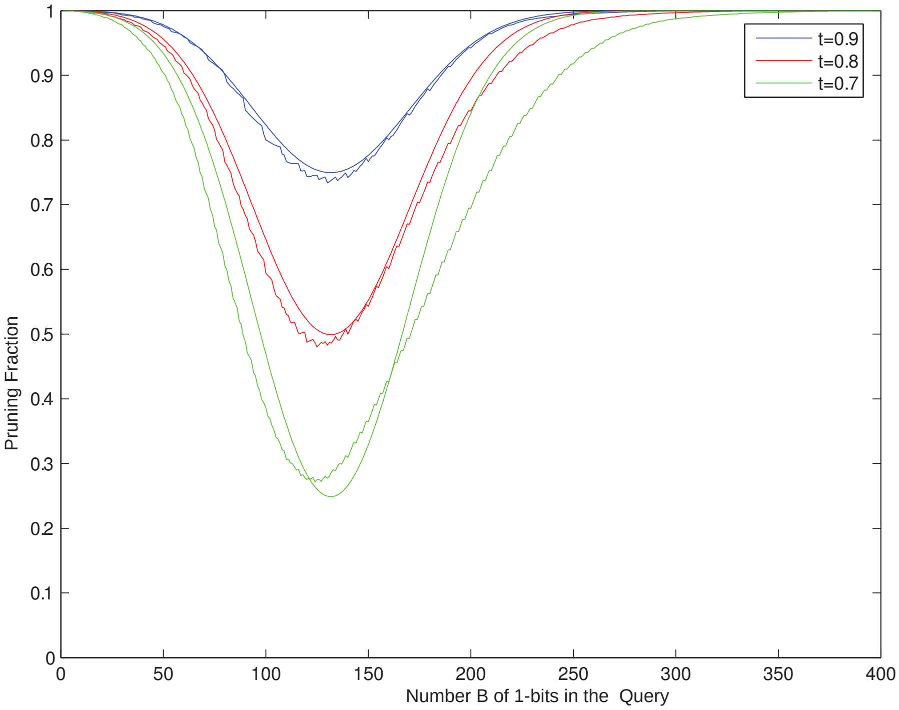


Figure 2: Average database pruning for the case of $M = 1$. The y-axis represents the fraction of the database being pruned. The x-axis represents the number B of bits in the query molecule. Curves correspond to three different Tanimoto similarity threshold. Blue corresponds to $t = 0.9$, red to $t = 0.8$, and green to $t = 0.7$. Results are computed using binary fingerprints of length $N = 512$ using lossy OR-folding compression and a subset of 50,000 molecules extracted randomly from the ChemDB database. Smooth curves correspond to theoretical predictions. Rough curves correspond to simulations.

Case M=2: The case $M = 2$ subsumes the case $M = 1$ and allows one to prune additional molecules.

In this case, we have $\vec{b} = (b_1, b_2)$ and $\vec{c} = (c_1, c_2)$. Here b_1 represents the number of 1-bits in $\vec{B}$ falling on odd-numbered components, b_2 the number of 1-bits in $\vec{B}$ falling on even-numbered components, and similarly for $\vec{c}$. Obviously, $b_1 + b_2 = B$ and $c_1 + c_2 = C$. When $M = 2$, the intersection bound allows one to prune all the molecules $\vec{C}$ such that

$$\min(b_1, c_1) + \min(b_2, c_2) \leq \frac{t}{1+t}(B+C) \tag{27}$$

This yields four subcases:

1. $b_1 + b_2 = B \leq t(B+C)/(1+t)$ which yields $C \geq B/t$ (as in the case of $M = 1$).

2. $c_1 + c_2 = C \leq t(B+C)/(1+t)$ which yields $C \leq tB$ (as in the case of $M = 1$).

3. $b_1 + c_2 \leq t(B+C)/(1+t)$ which yields $c_2 \leq -b_1 + t(B+C)/(1+t)$.

4. $c_1 + b_2 \leq t(B+C)/(1+t)$ which yields $c_1 \leq -b_2 + t(B+C)/(1+t)$.

The first two subcases correspond to the pruning derived when $M = 1$. The last two subcases correspond to new additional pruning.

To see this, consider the example given above in the case of $M = 1$ with a query $\vec{B}$ with $B = 300$ and a similarity threshold of $t = 0.8$. By using the $M = 1$ level corresponding to the first two subcases, one needs to focus only on candidate molecules satisfying $240 < C < 375$. By using $M = 2$ and the last two subcases, one can prune additional molecules that fall within this interval. For instance, assume that $\vec{b} = (200, 100)$. Then even among the molecules $\vec{C}$ containing the same number of 1-bits as the query (i.e. $C = B = 300$), we can eliminate all of those associated with $\vec{c} = (c_1, c_2)$ for which $c_1 \leq 166$ (subcase 4) or $c_2 \leq 66$ (subcase 3).

In short, it should be clear that the greater the value of M, the greater the amount of pruning associated with the intersection bound and Equation 25. While the amount of pruning is an essential consideration, other equally important factors must be taken into account in an actual database implementation. For instance, larger values of M may require additional storage space and computing time. Furthermore, the data structure required to implement the pruning algorithm is also an important consideration. While a complete analysis of these issues is beyond the scope of this paper, it is worth noting that the cases of $M = 1$ and $M = 2$ suggest a natural and efficient data structure. First, organize the fingerprints $\vec{C}$ in the database into bins containing all the fingerprints that have the same number C of 1-bits. Organize the bins by increasing values of C. Second, within any such bin, organize the fingerprints by increasing values of, for instance, c_1 corresponding to the number of bits falling on odd components. For a given query $\vec{B}$ and threshold t, one can first use the intersection inequality with $M = 1$ to discard all the bins associated with values of C that are too large or too small and focus on bins with C in a limited range. Then, for these remaining bins, one can use the intersection inequality with $M = 2$ to discard additional molecules and focus on sub-bins of molecules with c_1 in a limited range. This procedure can be further repeated using, for instance, $M = 4$ raising again interesting questions of optimality and tradeoffs.

Formulae for estimating the amount of pruning for different values of M and different thresholds t have been derived, but will be given elsewhere, together with an analysis of the corresponding tradeoffs. Yet, even without those analyses, it is clear that the intersection bound is a simple but powerful tool for organizing and searching large databases of fingerprint vectors.

Acknowledgement

Work supported by NIH Biomedical Informatics Training grant (LM-07443-01), NSF MRI grant (EIA- 0321390), and NSF grant 0513376 to PB.

References

(1) Burkhard, W.; Keller, R. Some Approaches to Best-Match File Searching. *Communications of the ACM* **1973**, *16*, 230–236.

(2) Shapiro, M. The Choice of Reference Points in Best-Match File Searching. *Communications of the ACM* **1977**, *20*, 339–343.

(3) Shasha, D.; Wang, T.-L. New Techniques for Best-Match Retrieval. *ACM Trans. Inf. Syst.* **1990**, *8*, 140–158.

(4) Downs, G. M.; Willett, P. Similarity Searching and Clustering of Chemical-Structure Databases Using Molecular Property Data. *J. Chem. Info. Comp. Sci.* **1994**, *34*, 1094–1102.

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

(6) Willett, P.; Barnard, J.; Downs, G. Chemical Similarity Searching. *J. Chem. Info. Comp. Sci.* **1998**, *38*, 983–996.

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

(8) 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.

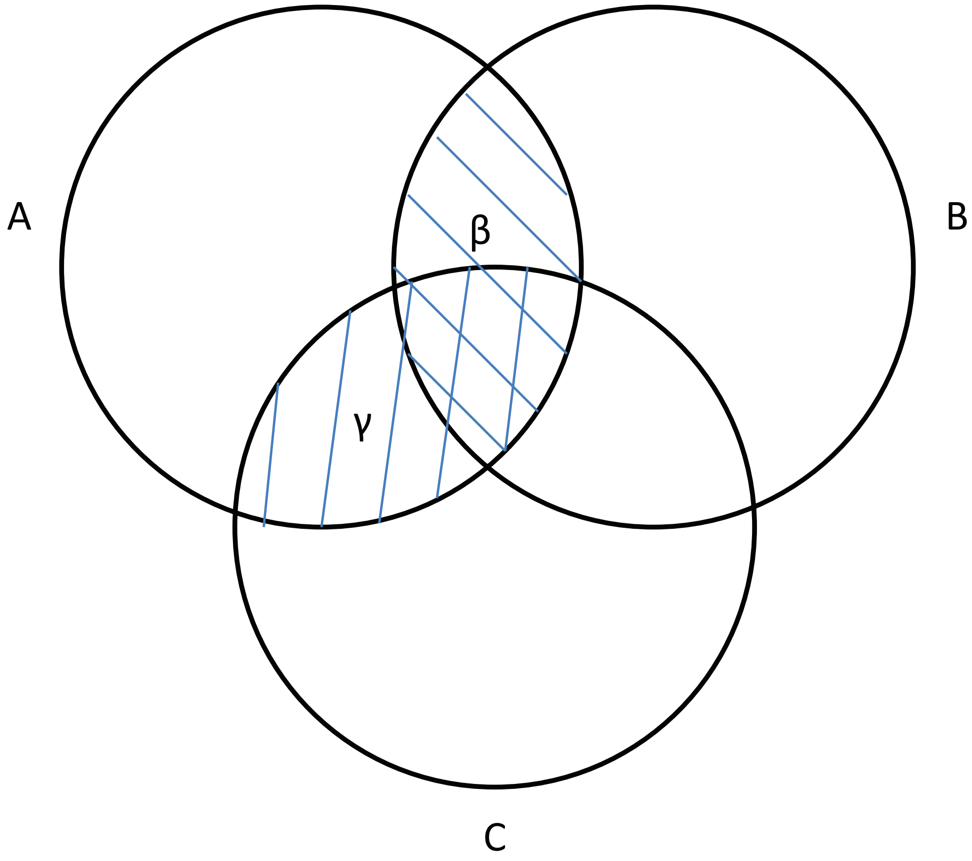
(9) Baldi, P.; Hirschberg, D. S.; Nasr, R. J. Speeding Up Chemical Database Searches Using a Proximity Filter Based on the Logical Exclusive-OR. *J. Chem. Info. Comp. Model.* **2008**, *48*, 1367–1378.

(10) Spath, H. *Cluster Analysis Algorithms*; Ellis Horwood, 1980.

(11) Lipkus, A. A Proof of the Triangle Inequality for the Tanimoto Distance. *J. Math. Chem.* **1999**, *26*, 263–265.

(12) 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.

(13) 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.


A
B
β
γ
C