Similarity by Compression

James L. Melville, Jenna F. Riley, and Jonathan D. Hirst*

School of Chemistry, University of Nottingham, University Park, Nottingham NG7 2RD, United Kingdom

Received September 1, 2006

We present a simple and effective method for similarity searching in virtual high-throughput screening, requiring only a string-based representation of the molecules (e.g., SMILES) and standard compression software, available on all modern desktop computers. This method utilizes the normalized compression distance, an approximation of the normalized information distance, based on the concept of Kolmogorov complexity. On representative data sets, we demonstrate that compression-based similarity searching can outperform standard similarity searching protocols, exemplified by the Tanimoto coefficient combined with a binary fingerprint representation and data fusion. Software to carry out compression-based similarity is available from our Web site at http://comp.chem.nottingham.ac.uk/download/zippity.

INTRODUCTION

Similarity searching is a core part of all informatics disciplines, and the calculation of the similarity of two molecules is the basis of a large number of computational practices in drug design.[1−3] The most obvious of these is similarity searching, where a database of molecules is searched for chemical entities that are similar to known drugs, leads, or druglike molecules.[4] In defining a similarity between two molecules, there are two equally important factors to consider: the representation of the molecules and the means by which these representations are used to calculate the similarity. Representation of molecules normally involves reducing them to a vectorial form. Usually, each element of the vector represents the presence or absence of a structural feature, although topological or other real-valued physicochemical features may be used. While a count of structural features is sometimes kept, it is more usual to represent structures as bit strings. These bit strings can be further subclassified into structural keys and hashed fingerprints. The former uses a predefined "dictionary" of substructures, while the latter generates the dictionary de novo for each set of molecules, using the set of unique paths found in the molecules studied (typically between two to eight atoms in length). As these fingerprints are extremely long, they are normally hashed and folded to a shorter length. There is a plethora of possible functions used to calculate distances between fingerprints, but the Tanimoto coefficient has established itself as the pre-eminent measure of similarity.[4,5]

Despite its widespread use, the deficiencies of the Tanimoto/fingerprint combination are well-known: a size bias, where large molecules are, on average, more similar to each other than small molecules,[6,7] and uncertainty about the connection between the calculated similarities and the probability of bioactivity.[5,8,9] In response to these challenges, recent research into similarity methods can be delineated along the following broad areas. First, more sophisticated representations of structures have been sought, where fingerprint dictionaries contain fewer but more pharmacophorically meaningful structural fragments than hashed fingerprints. Examples are feature trees,[10] reduced graph descriptors,[11] and circular fingerprints.[12] Alternatively, Bajorath and co-workers have developed techniques to reduce the dimensionality of the descriptor spaces that are used for searching.[13−15] Second, when improving the similarity measure, the overriding theme is that of merging the large amounts of complementary approaches into a more cohesive (and hopefully more accurate) whole. One way is by creating hybrids of different similarity coefficients to compensate for their relative weaknesses;[16,17] another is to merge information from multiple measurements on different molecules, normally (but not always)[18] under circumstances where more than one known active is present.[19−21] These approaches are known as either consensus scoring[22] or data fusion,[23] with the latter term more common in similarity searching.

Outside of cheminformatics, an alternative to bit-string fingerprint representations are quite common: alphabetic strings. Here, each element of the bit string is not restricted to a one or zero but may come from an alphabet. Usually, there is no ordering implied in the alphabet, unlike integer counts, for example. When objects are represented by strings, different similarity coefficients can be applied. One intriguing possibility is to use compression, familiar to most users of computers as a way to save disk space: an object is similar to another if their combined string representation can be compressed efficiently. Compression-based similarity methods, the theory of which is based on Kolmogorov complexity,[24] have been used in a wide variety of domains, most famously in the evolution of chain letters,[25] but also in time series analysis,[26,27] authorship attribution,[28,29] language classification,[29,30] plagiarism detection,[31] and musical classification.[32] In the field of biomolecular science, compression techniques have been employed in bioinformatics, where several problem domains lend themselves naturally to string representation, for example, DNA sequences,[33−37] protein sequences,[38] and protein contact maps.[39]

However, despite their popularity in other areas of informatics, no use of the compression-based similarity of

* Corresponding author phone: +44-115-951-3478; fax: +44-115-951-3562; e-mail: jonathan.hirst@nottingham.ac.uk.

10.1021/ci600384z CCC: $37.00
Published on Web 12/09/2006

small molecules has been reported. This is surprising, as string-based representations of molecules are ubiquitous in cheminformatics. In particular, the SMILES format of representing molecules is widely used as an efficient method of transmitting small molecule topologies, of precisely the level of detail used in generating fingerprints.[40] Importantly, SMILES strings are predominantly alphabetic; numeric quantities play only a minor role in conveying the information in a molecule, and this can be minimized when comparing similar molecules by canonicalization. Indeed, SMILES strings are rarely used directly in cheminformatics applications at all. Vidal et al. introduced the LINGO method,[41] where SMILES strings are partitioned into overlapping substrings (without regard to the underlying graph-based structure) and used directly as a replacement for physically meaningful fragment dictionaries in QSPR and virtual screening.[42] Additionally, Filimonov and Poroikov reported an experiment on compressing SMILES strings,[43] but with a view to evaluating the intrinsic complexity of molecules versus scientific text, not for similarity searching. While this paper was undergoing peer review, two other relevant studies were published: Grant et al. extended the LINGO approach by the use of finite state machines to generate very fast similarity searches,[44] and Karwath and De Raedt described a method to apply the rule-mining algorithm IREP to SMILES fragments to produce interpretable rules for classification.[45] Therefore, it seems timely to investigate the possibilities of compression-based similarity searching. Not only does it provide a radically different insight into concepts of molecular similarity and representation than hitherto considered, but given that suitable compressors come preinstalled with most Unix-based operating systems (e.g., Apple Mac OS X and Linux), they can be easily obtained for Windows, and SMILES is one of the most widely used chemical file formats, little more than a computer and a connection to the Internet to access online databases would be required to engage in effective similarity searching, a contrast to the increasing complexity of modern similarity searching. In this paper, we give a brief introduction the normalized compression distance (NCD), investigate the metric properties of the NCD, when combined with SMILES strings, and then evaluate its use in a series of simulated virtual high-throughput screening (VHTS) experiments, comparing its performance to a standard way of carrying out a similarity search: binary fingerprints based on unique paths found in a set of molecules, combined with the Tanimoto coefficient.

THE NORMALIZED COMPRESSION DISTANCE

We provide a brief introduction to Kolmogorov complexity and the normalized compression distance; further details can be found in the work of Vitányi et al.[24,46−48] Intuitively, a compression-based distance can be formulated by considering two objects to be similar to each other, if one can be used to "describe" the other, that is, they share information. Given that compression works by finding repeating blocks of data, then two similar objects should be compressible to a smaller size than that for two dissimilar objects. The mathematics used to develop this theory is based on Kolmogorov complexity, also known as algorithmic entropy, and has its basis in information theory.

The Kolmogorov complexity of a string x, $K(x)$, is defined as the shortest binary program that can compute x on a universal computer. This definition can be extended to include conditional Kolmogorov complexity: the Kolmogorov complexity of a string x, relative to another string y, $K(x|y)$, can be thought of as the shortest program that can generate x, if y is available as an additional input to the program.

The normalized information distance, NID, which takes values between 0.0 and 1.0, can be expressed as[47]

$$\mathrm{NID}(x,y) = \frac{\max\{K(x|y),K(y|x)\}}{\max\{K(x),K(y)\}} \tag{1}$$

Li et al.[47] showed that the NID could be considered to express *all* other distances (e.g., the Hamming, Euclidean, Tanimoto coefficient, etc.), in the sense that, if two objects are deemed to be similar by some effective similarity, then they are also close according to the NID. Therefore, similarity based on Kolmogorov Complexity has also been referred to as a "universal similarity metric",[39] or just "the" similarity metric.[47] Unfortunately, the Kolmogorov complexity is noncomputable. However, it can be approximated by expressing an object as a string of text and then using a real-world compressor, including those commonly found on most desktop computers. In this study, we considered two compressors, gzip, a dictionary-style compressor based on the Lempel−Ziv algorithm,[49] and bzip2, a block-based algorithm, using the Burrows−Wheeler transform.[50] The compression-based approximation to the NID, the normalized compression distance, NCD, can be defined as[48]

$$\mathrm{NCD}(x,y) = \frac{\min\{C(xy),C(yx)\} - \min\{C(x),C(y)\}}{\max\{C(x),C(y)\}} \tag{2}$$

where $C(x)$ is the size of the file containing the string x after compression and $C(xy)$ is the size of the file containing string y concatenated to the end of string x, again after compression. In their study, Cilibrasi and Vitányi approximate the first term of the numerator of eq 2 with $C(xy)$,[48] which is also the approach used in the CompLearn toolkit,[51] as they found only small deviations from symmetry. However, as the objects we intend to compress (SMILES strings) are much smaller than the objects previously studied, we investigated whether this assumption holds true for similarity searching with small molecules, before embarking on a simulated VHTS experiment.

METHODS

Data Sets. We compared our similarity methods using five data sets previously used by Jorissen and Gilson,[52] downloaded from http://gilsonlab.umbi.umd.edu/compounds.html. This consisted of five sets of 50 actives against five different targets: reversible inhibitors of cyclin-dependent kinase 2 (CDK2), cyclooxygenase-2 (COX2), factor Xa (FXA) and phosphodiesterase-5 (PDE5), and reversible antagonists of the α_{1A} adrenoreceptor (A1A) and a decoy set of 1892 inactives, taken from the National Cancer Institute (NCI) data set. The structures were converted to SMILES format and canonicalized using a program written with the open-source Java cheminformatics library JOELib2.[53,54] Details of the canonicalization process, which expresses each atom environment as a string and then uses Java's string comparison methods to determine a unique ordering, have been described previously.[55] During the conversion, six duplicates in the inactives were identified and removed. Additionally, one molecule (NCI048949) could not be parsed by JOELib's aromaticity handling routine, leaving 1885 inactives. Identities of the duplicates are given in the Supporting Information.

Calculation of Descriptors. For the compression-based similarity, no descriptor calculation is necessary; the text strings are used directly. To compare with standard similarity searching techniques, we used an in-house implementation of a binary fingerprint, similar to Daylight fingerprints.[56] To calculate this descriptor, we generated all paths of between two and seven atoms, representing each atom by its element, bond types to any other atom in the path, number of attached hydrogen atoms, number of connections to heavy atoms, and chirality (if any). Each path was then canonicalized and assigned a position in the fingerprint. The fingerprint for each molecule was then made binary by assigning a "1" to each position in the fingerprint where a path was found in that molecule and a "0" otherwise. We did not hash the fingerprint, in order to ensure that the results were as accurate as possible—this removes any possibility of artificially inflated similarities due to different paths being hashed to the same bin in the reduced fingerprint. Recently, Bender and Glen[57] noted that very simple descriptors of atom counts were surprisingly effective in similarity searching and that this may be a more effective benchmark than a comparison with random selection. Therefore, to ensure that our similarity searching methods were valid, we also generated a set of "dumb" descriptors for each data set, consisting of an integer string recording a count of the following types of atoms: boron, carbon, nitrogen, oxygen, fluorine, phosphorus, sulfur, chlorine, bromine, iodine, non-hydrogen, and "any" (i.e., both hydrogen and non-hydrogen). These descriptors were generated using a program written in Java, using the PATTY atom typer in JOELib.

Similarity Measures. For the compression similarity, we used the normalized compression distance, using the definition provided in eq 2. Compression was carried out using the gzip and bzip2 compression programs. These are available preloaded on any Apple Mac running the OS X operating system and on any Unix/Linux distribution. For the fingerprints, we used the Tanimoto coefficient for binary variables. For the "dumb" atom counts, we used the Euclidean distance for continuous variables. Formulas for their calculation are provided in the review by Willett et al.[4]

Evaluation. For each data set, we carried out the following protocol. Five actives were chosen randomly from the set of 50. The remaining 45 are considered "baits" and added to the set of 1885 inactives. The combined baits and inactives are termed "library compounds". Each one of the five actives is then used singly as a "query". The similarity of each query to every library compound is then measured. Each of the library compounds is then ranked according to the degree of similarity with the query. The process is then repeated for all the other queries, yielding five ranked lists of library compounds. A consensus scoring routine is then carried out to yield a single score for each library compound. For this, we used the maximum similarity recorded between the library compound and any of the queries, as studies by Schuffenhauer et al.[58] and Hert et al.[21] indicated that this method is more effective than data fusion schemes based on ranks or averaging. Some studies advocate range-scaling the ranked list of similarities between 0.0 and 1.0 before fusion;[59] however, in this study, we found that this was less effective than using unscaled similarities, so we use the unscaled similarities in all fusion schemes. The effectiveness of the searching is then measured by the ability of the consensus score to the rank the library compounds so that the baits are ranked further up the list than inactives. For this purpose, the "enrichment factor" (EF) is commonly used. This expresses the ratio of active molecules found in the top portion of the ranked list, over those that would be expected to found by a random selection. The size of the top of the ranked list to examine commonly ranges from 1 to 5%, so we present results for the top 1% and 5% of the list, which we abbreviate to EF@1 and EF@5, respectively. This represents assessing the enrichment in approximately the top 20 and 100 molecules, respectively. The selection of a particular cutoff is somewhat arbitrary and, for a small number of actives, can lead to large changes in the enrichment factor as actives fall below the threshold. Additionally, the enrichment factors we present are only approximate because, in the case of ties in the sorted database, we return all molecules below the cutoff that have the same similarity as the molecule at the cutoff. Recently, an alternative to the enrichment factor, the receiver operating characteristic[60] curve, has been advocated, which can be summarized as a single number by calculating the area under the curve (AUC).[61] For binary classifications, Hand and Till provided the following equation for calculating the AUC:[62]

$$\mathrm{AUC} = \frac{S_+ - \frac{1}{2}[n_+(n_+ + 1)]}{n_+ n_-} \qquad (3)$$

where n_+ is the number of positive samples (in this case, actives), n_- is the number of negative examples (inactives), and S_+ is the sum of the ranks of the positive examples. In the case of ties, we assigned the averaged ranks to all tied objects. Further details on the AUC are provided in the tutorial of Brown and Davis.[63] The AUC is not dependent on any cutoff and is therefore also presented, along with enrichment factors, although it could be argued that the enrichment factor is a more relevant measure of success in virtual screening simulations, as only the top portion of a virtually screened list of candidates would be experimentally screened. For each data set, we repeated the entire simulation 10 times, randomly choosing five new queries each time. Average AUC and EF values over the 10 runs are presented.

Similarity searching using the Tanimoto coefficient and Euclidean distance, consensus scoring, and statistics were generated using the object-oriented scripting language Ruby, which also provided a simple wrapper around the command line bzip2 and gzip commands. The simulations were carried out on a Debian cluster running Linux 2.4.22, with bzip2 version 1.0.2 and gzip version 1.3.5.

RESULTS

Is the NCD a Metric? Li et al.[47] provide a theoretical justification that the NCD satisfies the metric inequalities.

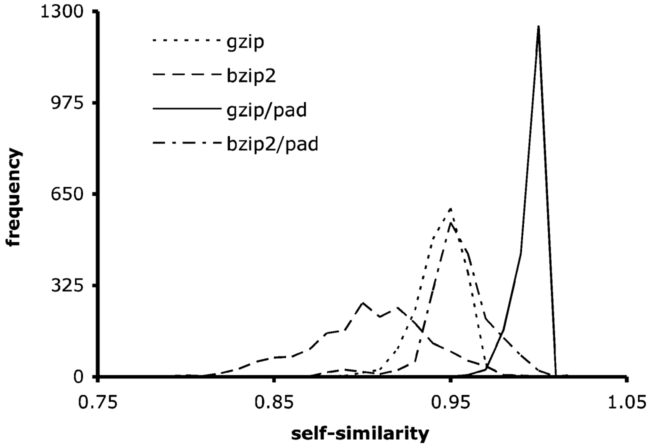

Figure 1. Distribution of self-similarities as measured by the NCD, using gzip and bzip2 as compressors, and including the effect of padding the strings, for 1885 compounds represented as SMILES strings.

In practice, there may be some violations. Therefore, we investigated the metric properties of the NCD when using the bzip2 and gzip compressors, using the set of inactives from the NCI database as a representative sample of molecules. For a distance coefficient to be a metric, it must have four properties.

The first is that distances between any objects *x* and *y*, *D*(*x*,*y*), should be zero or positive. It is easy to see from eq 2 that the form of the NCD makes it impossible for negative distances to occur, as file sizes are always positive and compressing two files together will never make the output smaller than the compressed size of one file. Further, the distance between an object and itself should be zero. For a normalized distance, this means that the “self-similarity” of an object should be 1.0. To test this, we calculated the self-similarities for all 1885 SMILES strings in the inactive data set. The distribution of self-similarities is shown in Figure 1 as the dotted and broken lines for gzip and bzip2, respectively. For gzip, the self-similarities ranged between 0.88 and 0.97, with an average of 0.94; bzip2 self-similarities ranged between 0.78 and 0.99 with an average value of 0.90. Clearly, there is a substantial amount of variability in the self-similarities. It is possible that, given the small size of the strings we are studying (on average, the SMILES strings we used contained 48 characters), the storage overhead associated with the compression (e.g., the dictionary for gzip) leads to a disproportionate amount of “noise” in the file sizes and, hence, the NCD values we obtain. To investigate this further, we looked at padding the strings, by concatenating duplicates to the end of the string a certain number of times. We looked at padding from 1 to 50 times and calculated the average self-similarity for the 1885 strings in the inactives set. The trend in average self-similarities is shown in Figure 2, where the unbroken line represents the behavior of gzip and the broken line that of bzip2. For gzip, there is an obvious spike when the strings are padded with one extra copy of themselves, and then the self-similarity swiftly declines. This is probably connected with the size of the “window” over which gzip looks for duplicated characters. For bzip2, there is no obvious relationship between padding and self-similarity, suggesting that this is not an effective way to increase the self-similarity. However, because padding the gzip string shows an obvious improvement, and the same padding is an improvement over no padding for bzip2, we decided to experiment with a padding value of two in our subsequent experiments with both compressors, as well as using the unpadded SMILES strings. The effect of increasing the padding on the distribution of self-similarities can be seen in Figure 1. For gzip (the unbroken line), a padding value of two results in a marked improvement of the self-similarities, with all 1885 similarities lying between 0.95 and 1.00, with an average value of 0.99. For bzip2 (dotted/broken line), the effect was more erratic; the average increased to 0.95, and the minimum self-similarity increased to 0.85, but for eight comparisons (0.4% of the data set), similarities larger than 1.00 were obtained, with a maximum self-similarity of 1.013. This is in line with previous experiments with the NCD for nonchemical data.[48] An obvious solution to the problem of self-similarities deviating from 1.0 is to rescale all comparisons with respect to the self-similarity. This will ensure that the self-similarity is 1.0. As long as all other similarities with respect to the string are smaller than the self-similarity (vide supra), this will ensure that all similarities are scaled to between 0.0 and 1.0. A second property of a metric is that it should be symmetric, that is, $D(x,y) = D(y,x)$. As mentioned, the form of the NCD commonly used is not strictly symmetric unless concatenating string *x* to the end of string *y* gives the same compressed file size as concatenating string *y* to the end of string *x*. For efficiency purposes, only $C(xy)$ is calculated. Previous studies suggested that only small deviations from symmetry were found, but the data sets used were of significantly different dimensions than those commonly found in cheminformatics: previously, studies have considered small numbers (on the order of tens) of “large” objects, that is, large in terms of the number of characters representing each object. We are more interested in comparing large numbers of small objects. To test whether the symmetry assumption still held, we sampled 1000 pairs of strings from the inactive SMILES strings and calculated the uncorrected $C(xy)$ and $C(yx)$. bzip2 showed perfect symmetry; this is a property of block-coding compressors. However, gzip was asymmetric, with an

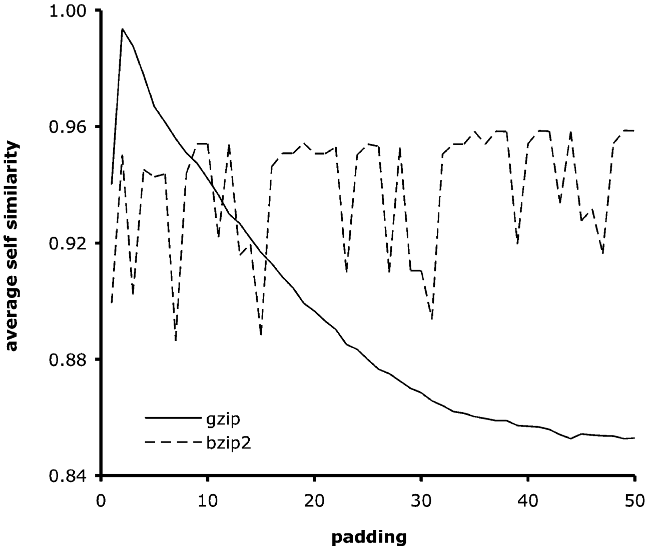

Figure 2. Effect of padding a SMILES string by concatenating duplicates to the end of it on average self-similarity via the NCD, using gzip (unbroken line) and bzip2 (broken line). Values were calculated as averages over 1885 calculations on the inactive data set used in the VHTS simulation.

average difference between $C(xy)$ and $C(yx)$ of 0.016. A total of 63.9% of the distances we measured showed a nonzero asymmetry. Padding did not help; for gzip, the average asymmetry was effectively unchanged at 0.017 with 65.5% of distances showing nonzero asymmetry. bzip2 was unaffected by padding, again showing perfect symmetry. The solution to this nonmetric behavior was to calculate the NCD using eq 2, which explicitly accounts for both $C(xy)$ and $C(yx)$. This does however incur a speed penalty, as an extra compression is carried out per comparison. A third requirement for a metric is that the distances obey the triangle inequality, that is, $D(x,y) \leq D(x,z) + D(z,y)$. To test this requirement, we randomly sampled three strings from the inactive set and measured their mutual similarities, repeating this 10 000 times. In no cases did we detect any violations of the triangle inequality, for any combination of compressor, symmetrizing, or padding. We are therefore confident that violations of the triangle inequality occur negligibly often, if at all. The final requirement of a metric is that nonidentical objects should always have a nonzero distance, that is, $D(x,y) > 0.0$ when $x \neq y$. To test this, we carried out an all-against-all similarity measurement for 1000 SMILES strings from the inactive se and for each string ranked all the similarities in nonincreasing order. If nonidentical objects have a nonzero distance, then the self-similarities for each string should be at the top of each ranked list, and there should be no other string equally similar. For gzip, there were four violations; for bzip2, there were seven. In the majority of cases, there was one other object that was found to be equally similar to the object itself. However, in the case of the bzip2 compressor only, for three objects, a string was found which was *more* similar than the self-similarity. However, in all cases, the similarity did not exceed 1.001. For gzip, the molecules that were considered to be identical differed only in one atom, where a carbon was replaced by a nitrogen or sulfur. For bzip2, the differences were greater, where a common scaffold would be present but two functional groups differed, for example, a methyl replaced with an acetyl group or a secondary amine replaced by sulfone. Under some circumstances, for example, scaffold hopping, this behavior could be useful, but we sought ways to remove it for the purposes of this study. Symmetrizing the NCD had no effect on the gzip compressor but reduced the number of violations for bzip2 to two. For both symmetric and asymmetric forms of the NCD, however, padding the strings by a factor of 2.0 removed all violations.

To conclude this section, the NCD combined with the SMILES string can show distinct nonmetric behavior. In particular, the assumption that the difference between $C(xy)$ and $C(yx)$ is negligible is not valid for SMILES strings. However, it is possible to "metricize" the NCD by appropriate scaling and averaging, at the cost of a slightly increased computation time. Further, a nonmetric can still be practically useful for similarity searching. To validate both the NCDs, we therefore carried out a series of VHTS simulations.

Table 1. Area under the Receiver Operating Characteristic Curve (AUC) for the Five Data Sets Studied[a]

descriptor/coefficient	A1A	CDK2	COX2	FXA	PDE
dumb	0.941	0.734	0.852	0.893	0.754
Tanimoto	0.942	0.836	0.912	0.961	0.911
Gz	0.951	0.883	0.907	0.950	0.917
Bz	*0.860*	*0.689*	0.873	*0.880*	*0.721*
Gz-p	0.960	0.893	0.933	0.970	0.933
Bz-p	*0.868*	0.777	0.898	*0.827*	*0.717*
Gz-s	0.953	0.882	0.914	0.957	0.919
Bz-s	*0.889*	0.750	0.897	0.904	0.760
Gz-sp	**0.964**	**0.894**	**0.936**	**0.975**	**0.934**
Bz-sp	*0.877*	0.775	0.894	*0.833*	*0.722*

[a] Key: dumb − simple atom count descriptor + Euclidean distance; Tanimoto − binary fingerprint with Tanimoto coefficient; Gz − NCD with the gzip compressor; Bz − NCD with the bzip2 compressor; s − NCD was symmetrized using eq 2; p − the SMILES string was padded by concatenating the string to the end of itself. Values quoted are averaged over 10 runs. The best performing similarity measure for a given data set is shown in bold; similarity measures that do worse than the dumb atom counts are shown in italics.

Table 2. Enrichment Factor after the Top 1% of the Database Has Been Screened, for the Five Data Sets Studied[a]

descriptor/coefficient	A1A	CDK2	COX2	FXA	PDE
dumb	15.4	5.0	7.3	13.8	8.8
Tanimoto	**38.6**	32.5	**35.7**	38.0	**37.2**
Gz	33.6	28.3	31.9	30.2	26.2
Bz	*14.9*	14.7	18.1	20.1	17.3
Gz-p	36.1	32.7	35.5	35.4	31.5
Bz-p	23.9	24.8	27.7	26.1	17.6
Gz-s	33.5	29.8	33.7	33.5	29.6
Bz-s	19.0	17.9	19.9	20.4	19.7
Gz-sp	36.1	**34.2**	35.6	**38.1**	31.6
Bz-sp	24.6	25.2	30.6	28.4	19.9

[a] Key: dumb − simple atom count descriptor + Euclidean distance; Tanimoto − binary fingerprint with Tanimoto coefficient; Gz − NCD with the gzip compressor; Bz − NCD with the bzip2 compressor; s − NCD was symmetrized using eq 2; p − the SMILES string was padded by concatenating the string to the end of itself. The best performing similarity measure for a given data set is shown in bold; similarity measures that do worse than the dumb atom counts are shown in italics. The maximum achievable enrichment factor at 1% was 41.2.

VHTS Simulations. For each data set and similarity coefficient/descriptor combination, we chose five active molecules to act as references, against which we screened all library compound molecules (baits + inactives). The max-score data-fusion technique was used to derive a consensus score for each library compound, and the ranked database was evaluated using enrichment factors and AUC. We repeated the entire process 10 times, using a new set of five references, randomly chosen, for each run. Averages over the 10 runs are presented in Tables 1−3. The row marked "dumb" gives the results of the simple atom count descriptors and is used as a baseline; search techniques that do worse than this are probably not worth the effort of implementing. The row marked "Tanimoto" represents results obtained using a fingerprint with the Tanimoto coefficient, a widely used standard for similarity searching. It can be seen that, in all cases, the Tanimoto coefficient outperforms the atom count descriptors, although the difference in performance is negligible by the AUC statistic for the A1A data set. Across all statistics, there is a general agreement that the Tanimoto coefficient shows a substantially better performance than the dumb descriptors on the CDK2, COX2, and PDE data sets. However, for the A1A and FXA data sets, the dumb descriptor enrichment factors are twice as high. Therefore, we might consider those two data sets as being slightly "easier" than the other three. Turning to the compressor results, we see that gzip compression performs very well— well above the performance of the dumb descriptors and nearly as well as the Tanimoto coefficient. Indeed, for some

Table 3. Enrichment Factor after the Top 5% of the Database Has Been Screened, for the Five Data Sets Studied[a]

descriptor/coefficient	A1A	CDK2	COX2	FXA	PDE
dumb	10.2	4.0	5.5	9.2	4.5
Tanimoto	13.1	8.9	12.7	15.0	**13.1**
Gz	14.3	10.3	11.8	13.3	10.7
Bz	*7.1*	4.9	8.9	9.6	5.9
Gz-p	14.5	10.2	12.5	14.8	11.3
Bz-p	*9.7*	6.42	10.5	10.4	6.5
Gz-s	14.5	10.2	12.5	14.8	11.3
Bz-s	*9.7*	6.4	10.5	10.4	6.5
Gz-sp	**15.2**	**11.9**	**13.2**	**16.4**	12.6
Bz-sp	*9.8*	8.6	11.2	10.2	7.1

[a] Key: dumb − simple atom count descriptor + Euclidean distance; Tanimoto − binary fingerprint with Tanimoto coefficient; Gz − NCD with the gzip compressor; Bz − NCD with the bzip2 compressor; s − NCD was symmetrized using eq 2; p − the SMILES string was padded by concatenating the string to the end of itself. The best performing similarity measure for a given data set is shown in bold; similarity measures that do worse than the dumb atom counts are shown in italics. The maximum achievable enrichment factor at 5% was 19.3.

statistics, the gzip compressor outperforms the Tanimoto coefficient on the A1A and CDK2 data sets. Conversely, the bzip2 compressor performs rather poorly. By the AUC and enrichment at 5% statistics, it gives a comparable performance to that of the dumb descriptors for all data sets except COX2. Given that padding the string seemed to improve the metric properties of the NCD, we were curious to see whether this would have any effect on the similarity searching. As padding the string to twice its length showed a noticeable improvement in the self-similarity measurements for both compressors, we repeated the similarity searches with this level of padding. Results are shown in Tables 1−3. Padding the string gave further improvements to the performance of the gzip compressor. Indeed, by the AUC statistic, it was now rated to be more effective than the Tanimoto coefficient across all data sets; by the EF@1 it was more effective than the Tanimoto coefficient on the CDK2 data set, and by the EF@5 statistic, it was superior in all data sets except the PDE data set. Improvement for the bzip2 compressor was also observed for most cases, with its performance becoming superior to the dumb descriptors overall. From these results it appeared that the gzip compressor was an effective tool in similarity searching. The bzip2 compressor, however, was uniformly inferior across all data sets and statistics and struggled to outperform the dumb descriptor baseline.

Symmetrized Searching. Having established that the NCD with SMILES showed asymmetric behavior, but that it could be symmetrized by appropriate averaging, we were interested to see whether this could translate to superior similarity search performance. In particular, given the rather weak performance of bzip2, this might provide a way to improve its accuracy. We therefore repeated the VHTS simulation above, but this time using the symmetrized form of the NCD. Results are shown in Tables 1−3. In all cases except the AUC for A1A, statistics are uniformly improved when compared to the “asymmetric” results, with the improvement more marked for the bzip2 compressor. However, the improvement is not as great as was seen when padding the strings. Therefore, the obvious course of action was to see if padding combined with symmetrized similarity could yield further improvements still. For gzip, this is the case, where the best result is obtained for all data sets and statistics, except the EF@1 for A1A. For bzip2, the improvement is again uncertain, where padding improves the enrichment factors, but the AUC is somewhat reduced. The results of the symmetrized similarity search suggest that it is more effective than the asymmetric compression results, and again the gzip compression performs well. The improvement over the padded asymmetric results is fairly small, however.

Table 4. Average Number of Actives, Rounded to the Nearest Integer, Returned by the Tanimoto Coefficient and Fingerprints, and the Symmetrized NCD with the gzip Compressor and Padding, at the Top 1% of the Database

	A1A	CDK2	COX2	FXA	PDE	average
num Tani	17	14	16	17	17	16
num NCD	16	16	16	17	14	16
union	23	17	21	23	20	21
intersection	10	13	11	11	11	11
unique Tani (%)[a]	7 (29)	2 (9)	5 (23)	6 (25)	6 (28)	5 (22)
unique NCD (%)[a]	6 (27)	3 (16)	5 (25)	6 (25)	4 (16)	5 (22)

[a] Number in parentheses expresses the number of unique actives found by the searching method as a percentage of the total number of actives found by either method.

Table 5. Average Number of Actives, Rounded to the Nearest Integer, Returned by the Tanimoto Coefficient and Fingerprints, and the Symmetrized NCD with the gzip Compressor and Padding, at the Top 5% of the Database

	A1A	CDK2	COX2	FXA	PDE	average
num Tani	29	20	29	34	29	28
num NCD	35	27	31	38	30	32
union	38	28	33	41	36	35
intersection	26	19	27	31	23	25
unique Tani (%)[a]	3 (7)	1 (3)	2 (6)	3 (6)	6 (17)	3 (8)
unique NCD (%)[a]	9 (22)	8 (28)	4 (12)	7 (17)	7 (20)	7 (20)

[a] Number in parentheses expresses the number of unique actives found by the searching method as a percentage of the total number of actives found by either method.

Tanimoto + Compression Fusion. The results given in Tables 1−3 indicate that the NCD and the Tanimoto coefficient perform very similarly in terms of enrichment factors, but an obvious question is whether they retrieve the *same* actives. Tables 4 and 5 report some statistics related to this, comparing the number of actives returned by the Tanimoto coefficient and the symmetrized NCD with gzip and padding. We report the average number of actives retrieved for each data set, the union of the actives returned by both methods, and the number of unique actives returned by each method. The final column gives the average values across all five data sets. It can be seen that, at 1% of the database, approximately 21 actives are retrieved. Of these, the Tanimoto and NCD contribute five unique actives each. Put another way, 22% of the combined actives are unique to one of the methods. At 5% of the database, 35 unique actives are returned, and the NCD is responsible for seven unique actives (20%), while the Tanimoto contributes three unique actives (8%). Therefore, it seems that the NCD and Tanimoto coefficient display a substantial complementarity in the actives they retrieve. As a result, we decided to experiment with further data fusion, by combining the results returned by the NCD and Tanimoto coefficient. Here, we experimented with fusing ranks and scores, and using the maximum or average of these values. Results are shown in

Table 6. Area Under the Receiver Operating Characteristic Curve (AUC) Results Averaged over 10 Runs, for Data Fusion Experiments Combining the Tanimoto Coefficient and Fingerprints with the Symmetrized NCD with gzip and Padding[a]

	A1A	CDK2	COX2	FXA	PDE
best	0.964	**0.894**	0.936	0.975	0.934
max score	0.964	**0.894**	0.936	0.975	0.935
max rank	0.960	0.872	0.934	0.976	0.948
mean score	**0.972**	0.890	**0.941**	**0.986**	**0.958**
mean rank	0.966	0.879	0.938	0.981	0.952

[a] "Best" indicates the best result obtained from any previous experiment in Tables 1−3. Numbers in bold indicate the highest value obtained for a given data set.

Table 7. Enrichment Factor at 1% Results Averaged over 10 Runs, for Data Fusion Experiments Combining the Tanimoto Coefficient and Fingerprints with the Symmetrized NCD with gzip and Padding[a]

	A1A	CDK2	COX2	FXA	PDE
best	38.6	34.2	35.7	38.1	37.2
max score	36.1	34.2	35.6	38.4	31.6
max rank	38.2	34.6	37.7	40.2	35.6
mean score	**41.3**	35.0	**39.3**	**42.7**	**39.5**
mean rank	39.7	**35.4**	39.1	41.5	38.4

[a] "Best" indicates the best result obtained from any previous experiment in Tables 1−3. Numbers in bold indicate the highest value obtained for a given data set.

Table 8. Enrichment Factor at 5% Results Averaged over 10 Runs, for Data Fusion Experiments Combining the Tanimoto Coefficient and Fingerprints with the Symmetrized NCD with gzip and Padding[a]

	A1A	CDK2	COX2	FXA	PDE
best	15.2	**11.9**	13.2	16.4	13.1
max score	15.2	**11.9**	13.2	16.4	12.8
max rank	15.1	10.9	13.6	16.7	14.8
mean score	**16.8**	11.4	**14.0**	**17.9**	**15.2**
mean rank	16.0	11.4	13.8	17.5	15.1

[a] "Best" indicates the best result obtained from any previous experiment in Tables 1−3. Numbers in bold indicate the highest value obtained for a given data set.

Tables 6−8, along with the best result obtained in any previous experiment, as a comparison. The mean-score method is clearly the most effective fusion method, successfully increasing the value of all statistics for four out of five data sets, and increasing the EF@5% value for the CDK2 data set. Therefore, it seems that the Tanimoto coefficient can be profitably combined with the NCD to improve similarity searching results.

CONCLUSIONS

Combining canonicalized SMILES strings and the NCD using the gzip compressor has proved to be a very (perhaps surprisingly) successful combination: nearly always performing an order of magnitude more successfully than random selection, always better than the use of "dumb" atom count descriptors, and for some data sets, providing performance close to, and for some statistics even better than, that of traditional searching methods, as exemplified by the use of fingerprints combined with the Tanimoto coefficient. Additionally, the NCD returns a substantial number of actives not recovered by the Tanimoto coefficient; therefore, the NCD is not merely competitive with but complementary to established similarity searching techniques. bzip2 underperformed gzip considerably; therefore, it seems that block-based compression algorithms may be less effective for similarity searching. Padding the string to twice its length had a notable effect on the performance of the similarity searching; likewise, symmetrizing the NCD by averaging $C(xy)$ and $C(yx)$ further improved the performance, albeit to a smaller extent, and at a slightly increased cost in speed. However, as this ensures that the NCD acts as a metric when used with SMILES strings, it may be preferable to use it in this mode to give more confidence in its behavior. Clearly, further optimization is possible. Nonetheless, we have sought to minimize any parameter tuning, in order to create an "off-the-shelf" similarity searching protocol that can be used without access to any sophisticated cheminformatics algorithms. The requirement for canonicalized SMILES strings may seem to be a mark against this ease of use, particularly given that the full SMILES canonicalization algorithm has never been published (an algorithm that ignores stereoisomerism and isotopes was provided by Weininger at al.),[64] but an increasing number of sources, for example, PubChem,[65] provide SMILES strings that are already in canonicalized form, thus reducing any necessary preprocessing steps.

There are several directions for further work. Most obviously, other, better (albeit slower) compressors exist, for example, those based on the prediction by partial matching method.[66] Some compressors produce a dictionary based on the repeating motifs found in a string, which is of intrinsic interest;[67] in the context of similarity searching, it may provide information on important substructures or pharmacophores.

Also, it may be profitable to design alternative string-based representations of molecules, to aid in activities such as scaffold hopping. Finally, the concept of similarity is used in several other core cheminformatics practices, such as library design and clustering, and is an essential part of kernel-based learners, such as support vector machines. These are all areas that can benefit from the use of compression-based distances.

To conclude, we have demonstrated that SMILES strings and compression programs are a simple, yet powerful method for similarity searching, competitive with state-of-the-art techniques. The Ruby scripts used to carry out the experiments described in this paper are available for download from http://comp.chem.nottingham.ac.uk/download/zippity.

ACKNOWLEDGMENT

We acknowledge the Nuffield Foundation for Undergraduate Research Bursary URB/32975 for J.F.R. and the EPSRC for support (GR/S75765/01). We thank Natalio Krasnogor and Daniel Barthel of the School of Computer Science and I.T. at the University of Nottingham for helpful discussions. We are also grateful for the use of the High Performance Computing Facility at the University of Nottingham.

Supporting Information Available: Names of the duplicate molecules in the NCI inactives set. This material is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Johnson, M. A.; Maggiora, G. M. *Concepts and Applications of Molecular Similarity*; John Wiley & Sons: New York, 1990.

(2) Dean, P. M. *Molecular Similarity in Drug Design*; Blackie Academic & Professional: London, 1995.

(3) Bender, A.; Glen, R. C. Molecular Similarity: A Key Technique in Molecular Informatics. *Org. Biomol. Chem.* **2004**, *2*, 3204–3218.

(4) Willett, P.; Barnard, J. M.; Downs, G. M. Chemical Similarity Searching. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 983–996.

(5) Martin, Y. C.; Kofron, J. L.; Traphagen, L. M. Do Structurally Similar Molecules Have Similar Biological Activity? *J. Med. Chem.* **2002**, *45*, 4350–4358.

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

(7) Dixon, S. L.; Koehler, R. T. The Hidden Component of Size in Two-Dimensional Fragment Descriptors: Side Effects on Sampling in Bioactive Libraries. *J. Med. Chem.* **1999**, *42*, 2887–2900.

(8) Kubinyi, H. Similarity and Dissimilarity: A Medicinal Chemist's View. *Perspect. Drug Discovery Des.* **1998**, *9*–*11*, 225–252.

(9) Roth, H. J. There Is no Such Thing as 'Diversity'! *Curr. Opin. Chem. Biol.* **2005**, *9*, 293–295.

(10) Rarey, M.; Dixon, J. S. Feature Trees: A New Molecular Similarity Measure Based on Tree Matching. *J. Comput.-Aided Mol. Des.* **1998**, *12*, 471–490.

(11) Gillet, V. J.; Willett, P.; Bradshaw, J. Similarity Searching Using Reduced Graphs. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 338–345.

(12) Bender, A.; Mussa, H. Y.; Glen, R. C.; Reiling, S. Molecular Similarity Searching Using Atom Environments, Information-Based Feature Selection, and a Naive Bayesian Classifier. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 170–178.

(13) Godden, J. W.; Furr, J. R.; Xue, L.; Stahura, F. L.; Bajorath, J. Molecular Similarity Analysis and Virtual Screening by Mapping of Consensus Positions in Binary-Transformed Chemical Descriptor Spaces with Variable Dimensionality. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 21–29.

(14) Eckert, H.; Bajorath, J. Determination and Mapping of Activity-Specific Descriptor Value Ranges for the Identification of Active Compounds. *J. Med. Chem.* **2006**, *49*, 2284–2293.

(15) Eckert, H.; Vogt, I.; Bajorath, J. Mapping Algorithms for Molecular Similarity Analysis and Ligand-Based Virtual Screening: Design of DynaMAD and Comparison with MAD and DMC. *J. Chem. Inf. Model.* **2006**, *46*, 1623–1634.

(16) 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.

(17) Salim, N.; Holliday, J.; Willett, P. Combination of Fingerprint-Based Similarity Coefficients Using Data Fusion. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 435–442.

(18) Hert, J.; Willett, P.; Wilton, D. J.; Acklin, P.; Azzaoui, K.; Jacoby, E.; Schuffenhauer, A. Enhancing the Effectiveness of Similarity-Based Virtual Screening Using Nearest-Neighbor Information. *J. Med. Chem.* **2005**, *48*, 7049–7054.

(19) Wilton, D.; Willett, P.; Lawson, K.; Mullier, G. Comparison of Ranking Methods for Virtual Screening in Lead-Discovery Programs. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 469–474.

(20) 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.

(21) Hert, J.; Willett, P.; Wilton, D. J. Comparison of Fingerprint-Based Methods for Virtual Screening Using Multiple Bioactive Reference Structures. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1177–1185.

(22) Charifson, P. S.; Corkery, J. J.; Murcko, M. A.; Walters, W. P. Consensus Scoring: A Method for Obtaining Improved Hit Rates from Docking Databases of Three-Dimensional Structures into Proteins. *J. Med. Chem.* **1999**, *42*, 5100–5109.

(23) Ginn, C. M. R.; Willett, P.; Bradshaw, J. Combination of Molecular Similarity Measures Using Data Fusion. *Perspect. Drug Discovery Des.* **2000**, *20*, 1–16.

(24) Li, M.; Vitanyi, P. M. B. *An Introduction to Kolmogorov Complexity and its Applications*, 2nd ed.; Springer-Verlag: New York, 1997.

(25) Bennett, C. H.; Li, M.; Ma, B. Chain Letters & Evolutionary Histories. *Sci. Am.* **2003**, *288*, 76–81.

(26) Keogh, E.; Lonardi, S.; Ratanamahatana, C. A. Towards Parameter-Free Data Mining. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM Press: Seattle, WA, 2004.

(27) Puglisi, A.; Benedetto, D.; Caglioti, E.; Loreto, V.; Vulpiani, A. Data Compression and Learning in Time Sequences Analysis. *Physica D* **2003**, *180*, 92–107.

(28) Kukushkina, O. V.; Polikarpov, A. A.; Khmelev, D. V. Using Literal and Grammatical Statistics for Authorship Attribution. *Probl. Inf. Transm. (Engl. Transl.)* **2001**, *37*, 172–184.

(29) Baronchelli, A.; Caglioti, E.; Loreto, V. Artificial Sequences and Complexity Measures. *J. Stat. Mech. Theory Exp.* **2005**, article no. P04002.

(30) Benedetto, D.; Caglioti, E.; Loreto, V. Language Trees and Zipping. *Phys. Rev. Lett.* **2002**, 88.

(31) Chen, X.; Francia, B.; Li, M.; McKinnon, B.; Seker, A. Shared Information and Program Plagiarism Detection. *IEEE Trans. Inf. Theory* **2004**, *50*, 1545–1551.

(32) Cilibrasi, R.; Vitanyi, P.; de Wolf, R. Algorithmic Clustering of Music Based on String Compression. *Comput. Music J.* **2004**, *28*, 49–67.

(33) Chen, X.; Kwong, S.; Li, M. A Compression Algorithm for DNA Sequences. *IEEE Eng. Med. Biol.* **2001**, *20*, 61–66.

(34) Li, M.; Badger, J. H.; Chen, X.; Kwong, S.; Kearney, P.; Zhang, H. Y. An Information-Based Sequence Distance and Its Application to Whole Mitochondrial Genome Phylogeny. *Bioinformatics* **2001**, *17*, 149–154.

(35) Chen, X.; Li, M.; Ma, B.; Tromp, J. DNACompress: Fast and Effective DNA Sequence Compression. *Bioinformatics* **2002**, *18*, 1696–1698.

(36) Otu, H. H.; Sayood, K. A New Sequence Distance Measure for Phylogenetic Tree Construction. *Bioinformatics* **2003**, *19*, 2122–2130.

(37) Ane, C.; Sanderson, M. J. Missing the Forest for the Trees: Phylogenetic Compression and Its Implications for Inferring Complex Evolutionary Histories. *Syst. Biol.* **2005**, *54*, 146–157.

(38) Kocsor, A.; Kertesz-Farkas, A.; Kajan, L.; Pongor, S. Application of Compression-Based Distance Measures to Protein Sequence Classification: A Methodological Study. *Bioinformatics* **2006**, *22*, 407–412.

(39) Krasnogor, N.; Pelta, D. A. Measuring the Similarity of Protein Structures by Means of the Universal Similarity Metric. *Bioinformatics* **2004**, *20*, 1015–1021.

(40) Weininger, D. Smiles, a Chemical Language and Information-System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.

(41) Vidal, D.; Thormann, M.; Pons, M. LINGO, an Efficient Holographic Text Based Method To Calculate Biophysical Properties and Intermolecular Similarities. *J. Chem. Inf. Model.* **2005**, *45*, 386–393.

(42) Vidal, D.; Thormann, M.; Pons, M. A Novel Search Engine for Virtual Screening of Very Large Databases. *J. Chem. Inf. Model.* **2006**, *46*, 836–843.

(43) Filimonov, D.; Poroikov, V. Why Relevant Chemical Information Cannot Be Exchanged without Disclosing Structures. *J. Comput.-Aided Mol. Des.* **2005**, *19*, 705–713.

(44) Grant, J. A.; Haigh, J. A.; Pickup, B. T.; Nicholls, A.; Sayle, R. A. Lingos, Finite State Machines, and Fast Similarity Searching. *J. Chem. Inf. Model.* **2006**, *46*, 1912–1918.

(45) Karwath, A.; DeRaedt, L. SMIREP: Predicting Chemical Activity from SMILES. *J. Chem. Inf. Model.* **2006**, *46*, 6, 2432–2444.

(46) Bennett, C. H.; Gacs, P.; Li, M.; Vitanyi, F. M. B.; Zurek, W. H. Information Distance. *IEEE Trans. Inf. Theory* **1998**, *44*, 1407–1423.

(47) Li, M.; Chen, X.; Li, X.; Ma, B.; Vitanyi, P. M. B. The Similarity Metric. *IEEE Trans. Inf. Theory* **2004**, *50*, 3250–3264.

(48) Cilibrasi, R.; Vitanyi, P. M. B. Clustering by Compression. *IEEE Trans. Inf. Theory* **2005**, *51*, 1523–1545.

(49) Ziv, J.; Lempel, A. A Universal Algorithm for Sequential Data Compression. *IEEE Trans. Inf. Theory* **1977**, *23*, 337–343.

(50) Burrows, M.; Wheeler, D. J. A Block-sorting Lossless Data Compression Algorithm; Digital Equipment Corporation: 1994; p 124.

(51) CompLearn. http://www.complearn.org/ (accessed Aug 31, 2006).

(52) Jorissen, R. N.; Gilson, M. K. Virtual Screening of Molecular Databases Using a Support Vector Machine. *J. Chem. Inf. Model.* **2005**, *45*, 549–561.

(53) Guha, R.; Howard, M. T.; Hutchison, G. R.; Murray-Rust, P.; Rzepa, H.; Steinbeck, C.; Wegner, J.; Willighagen, E. L. The Blue Obelisk–Interoperability in Chemical Informatics. *J. Chem. Inf. Model.* **2006**, *46*, 991–998.

(54) JOELib. http://sourceforge.net/projects/joelib/ (accessed Aug 31, 2006).

(55) Melville, J. L.; Hirst, J. D. TMACC: Interpretable Correlation Descriptors for Quantitative Structure–Activity Relationships. Submitted for publication.

(56) Daylight Theory: Fingerprints. http://www.daylight.com/dayhtml/doc/theory/theory.finger.html (accessed Aug 31, 2006).

(57) Bender, A.; Glen, R. C. A Discussion of Measures of Enrichment in Virtual Screening: Comparing the Information Content of Descriptors with Increasing Levels of Sophistication. *J. Chem. Inf. Model.* **2005**, *45*, 1369–1375.

(58) Schuffenhauer, A.; Floersheim, P.; Acklin, P.; Jacoby, E. Similarity Metrics for Ligands Reflecting the Similarity of the Target Proteins. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 391–405.

(59) Whittle, M.; Gillet, V. J.; Willett, P.; Alex, A.; Loesel, J. Enhancing the Effectiveness of Virtual Screening by Fusing Nearest Neighbor Lists: A Comparison of Similarity Coefficients. *J. Chem. Inf. Model.* **2004**, *44*, 1840–1848.

(60) Triballeau, N.; Acher, F.; Brabet, I.; Pin, J. P.; Bertrand, H. O. Virtual Screening Workflow Development Guided by the "Receiver Operating Characteristic" Curve Approach. Application to High-Throughput

Docking on Metabotropic Glutamate Receptor Subtype 4. *J. Med. Chem.* **2005**, *48*, 2534−2547.

(61) Huang, J.; Ling, C. X. Using AUC and Accuracy in Evaluating Learning Algorithms. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 299−310.

(62) Hand, D. J.; Till, R. J. A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. *Mach. Learn.* **2001**, *45*, 171−186.

(63) Brown, C. D.; Davis, H. T. Receiver Operating Characteristics Curves and Related Decision Measures: A Tutorial. *Chemom. Intell. Lab. Syst.* **2006**, *80*, 24−38.

(64) Weininger, D.; Weininger, A.; Weininger, J. L. Smiles. 2. Algorithm for Generation of Unique Smiles Notation. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 97−101.

(65) PubChem. http://pubchem.ncbi.nlm.nih.gov (accessed Aug 31, 2006).

(66) Cleary, J. G.; Witten, I. H. Data-Compression Using Adaptive Coding and Partial String Matching. *IEEE Trans. Commun.* **1984**, *32*, 396−402.

(67) Baronchelli, A.; Caglioti, E.; Loreto, V.; Pizzi, E. Dictionary-Based Methods for Information Extraction. *Physica A* **2004**, *342*, 294−300.

CI600384Z