Published paper

Gan, S., Cosgrove, D.A., Gardiner, E.J. and Gillet, V.J. (2014) *Investigation of the Use of Spectral Clustering for the Analysis of Molecular Data*. Journal of Chemical Information and Modeling, 54 (12). 3302 – 3319

10.1021/ci500480b

JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

Investigation of the Use of Spectral Clustering for the Analysis of Molecular Data

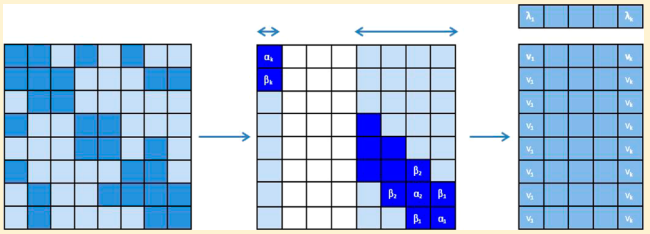
Sonny Gan,[†] David A. Cosgrove,[‡] Eleanor J. Gardiner,[†] and Valerie J. Gillet*[,†]

[†]Information School, University of Sheffield, Regent Court, 211 Portobello Street, Sheffield S1 4DP, United Kingdom

[‡]AstraZeneca, Mereside, Alderley Park, Macclesfield SK10 4TG, United Kingdom

Supporting Information

ABSTRACT: Spectral clustering involves placing objects into clusters based on the eigenvectors and eigenvalues of an associated matrix. The technique was first applied to molecular data by Brewer [*J. Chem. Inf. Model.* **2007**, *47*, 1727−1733] who demonstrated its use on a very small dataset of 125 COX-2 inhibitors. We have determined suitable parameters for spectral clustering using a wide variety of molecular descriptors and several datasets of a few thousand compounds and compared the results of clustering using a nonoverlapping version of Brewer's use of Sarker and Boyer's algorithm with that of Ward's and *k*-means clustering. We then replaced the exact eigendecomposition method with two different approximate methods and concluded that Singular Value Decomposition is the most appropriate method for clustering larger compound collections of up to 100 000 compounds. We have also used spectral clustering with the Tversky coefficient to generate two sets of clusters linked by a common set of eigenvalues and have used this novel approach to cluster sets of fragments such as those used in fragment-based drug design.



INTRODUCTION

Clustering is the division of a collection of objects into sets, *clusters*, such that objects within a cluster are similar and objects taken from different clusters are dissimilar. Compound collections are routinely clustered based on structural or other features of the compounds. A representative can then be selected from within a cluster with the expectation that the compound is typical of those within the cluster. Clustering is routinely used for the analysis of chemical information, in, for example, high-throughput screening[1] and diverse subset selection.[2,3] A review of clustering algorithms used in analyzing chemical datasets is given by Downs and Barnard.[4]

Common clustering methods include sequential agglomerative hierarchical nonoverlapping (SAHN) clustering, of which the most commonly used are probably Ward's method, relocation methods such as K-means clustering, and single-pass clustering.[4,5] Hierarchical methods produce the typical cluster tree diagram where the clustering is produced either by regarding the dataset as a single cluster which is successively partitioned into subclusters or by initially regarding each object as a cluster and successively merging clusters. The most common single-pass algorithm is the leader algorithm.[6] Molecular clustering requires a molecular descriptor, such as a fingerprint, and a metric for quantifying the similarity between descriptors, such as the Tanimoto coefficient The similarity metric is usually a symmetric one (i.e., for molecules a,b, sim(a, b) = sim(b, a)) but asymmetric measures can be used for clustering. For example, the hierarchical clustering method of Tarjan is suitable for use with an asymmetric measure.[7] In this case a single cut of the hierarchy will generate a single clustering. An overview of asymmetric clustering within the chemoinformatics literature is given by MacCuish and MacCuish.[8] Clusters may be *crisp* (nonoverlapping) or *fuzzy* (objects can belong to more than one cluster). In fuzzy clustering the degree of membership of a cluster is usually given by a probability function. An alternative method of forming overlapping clusters is given by Nicolaou et al.,[9] whereby molecules which are equidistant from two clusters are placed in both.

There has recently been significant interest in the use of spectral clustering methods for the analysis of both biological and chemical data. Paccanaro et al. used the eigensolver algorithm of Ng et al.[10] to assign the sequences of the SCOP database[11] to superfamilies which compared extremely well with the manually curated superfamilies in SCOP and with the superfamilies assigned by three other methods.[12] Paccanaro concluded that the success of spectral clustering was due to the global nature of the method which meant it did not require the "hard" cut-offs used by predominately distance-based local methods, which assign cluster membership based on a similarity threshold. Indeed he demonstrated that a hard cutoff could not produce a perfect clustering for SCOP. The method has now been incorporated into the free SCPS (Spectral Clustering of Protein Sequences) software.[13] Spectral clustering has also been used to cluster protein conformations from MD simulations[14] and for cancer class discovery from gene expression profiles.[15]

In 2007, Brewer published the first use of a spectral clustering algorithm for analyzing molecular data, the motivation

Received: August 4, 2014
Published: November 7, 2014

dx.doi.org/10.1021/ci500480b | *J. Chem. Inf. Model.* 2014, 54, 3302−3319

being the selection of representative scaffolds from within a chemical dataset.[16] He used an algorithm published by Sarkar and Boyer[17] which was demonstrated on a set of 125 COX-2 inhibitors. Subsequent work at Evotec has used Brewer's clustering: to cluster 1800 potential *Trypanosoma cruzi* transsialidase inhibitors into 690 clusters;[18] to cluster 2700 compounds into 126 clusters during the development of a model for MCH-1R antagonists;[19] and in fragment-based drug discovery.[20]

Brewer made several suggestions for further investigation: that spectral clustering using different descriptors and similarity metrics be considered; that the method be parametrized in a systematic manner; that a comparison should be made with other clustering techniques; and that a different method for finding eigenvalues, the Lanczos algorithm,[21] be considered for use with larger datasets. We have undertaken some of these investigations and here report our results. We have chosen to compare spectral clustering with Ward's and *k*-means clustering methods since these two are commonly used for the clustering of chemical compound data. In our investigations the Lanczos algorithm was found to be effective if the number of clusters required is small but did not prove to be a suitable algorithm in general. We therefore turned to Singular Value Decomposition (SVD) which became our method of choice for spectral clustering. The remainder of the paper is structured as follows. We first give a brief overview of spectral clustering, followed by an overview of the Lanczos algorithm and of the relationship between SVD and spectral clustering. We then detail our parametrization experiments and compare spectral clustering with Ward's and *k*-means clustering. Next we describe our use of the Lanczos algorithm and show its limitations. Finally, we demonstrate the application of our SVD spectral clustering software on datasets of up to 100 000 compounds as well as on smaller fragment-based data.

METHODS

Introduction to Spectral Clustering. In linear algebra, a matrix represents a function that acts upon a vector, altering its magnitude and/or its direction. Vectors whose directions are left unaltered or inverted by a matrix are known as the eigenvectors of the matrix. More formally, if $\mathbf{x}$ is a nonzero column vector then $\mathbf{x}$ is an *eigenvector* of matrix $\mathbf{A}$ if and only if there is a scalar λ such that

$$\mathbf{Ax} = \lambda\mathbf{x} \tag{1}$$

λ is the *eigenvalue* of $\mathbf{A}$ associated with the eigenvector $\mathbf{x}$.

The eigenvalues and their associated eigenvectors, together known as *eigenpairs*, can be identified using an *eigendecomposition algorithm*, which is the term given to a procedure for identifying eigenpairs from an input matrix. The term *spectral clustering* is used to describe any clustering algorithm that utilizes the eigenvectors of a matrix as the basis for partitioning a dataset.[10] The approach to spectral clustering can vary in several ways, including the type of matrix that is formed from the dataset and the way in which the eigenvectors are used as the basis for the clustering. In general, the eigenvectors of the matrix (which can be a similarity, Laplacian or any other input matrix) constitute a set of weights, which can be used to cluster similar nodes.[17] Brewer made the key step of recognizing that one method to partition a set of N molecules by spectral clustering is to

(1) Form an $N \times N$ similarity matrix $\mathbf{S} = (s_{ij})$, where s_{ij} is the similarity between molecules i and j, where 1 indicates identity and 0 indicates maximally dissimilar; $\mathbf{S}$ is a real symmetric matrix.

(2) Transform the similarity matrix into a matrix, of the form, $\mathbf{A} = (a_{ij})$,

$$a_{ij} = e^{-\gamma(s_{ij}-1)^2} \tag{2}$$

where γ is a scaling parameter. Matrices of this form are sometimes known as *affinity* matrices,[7] although it is more usual for affinity matrices to have the leading diagonal set to zeroes; in Brewer's case the leading diagonal is set to 1. The effect of this Gaussian filtering function is to minimize low similarity scores and emphasize the spread of the higher scores.

(3) The eigenvalues of $\mathbf{A}$ are given by the solution to the matrix equation

$$\mathbf{AX} = \lambda\mathbf{X} \tag{3}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$ is the matrix whose columns are eigenvectors of $\mathbf{A}$ and λ is a diagonal matrix of eigenvalues, $\lambda = \text{diag}(\lambda_1, ..., \lambda_N)$. Since $\mathbf{A}$ is real and symmetric, it is possible to decompose $\mathbf{A}$ into the product

$$\mathbf{A} = \mathbf{X}\lambda\mathbf{X}^{\mathrm{T}} \tag{4}$$

We can choose both the columns (eigenvectors) and the rows of $\mathbf{X}$ to be normalized so that $\Sigma_{j=1}^{N} x^2_{ij} = 1$, and the entries of λ are ordered so that $\lambda_1 \geq .. \geq \lambda_N$.

Eigenvalues and eigenvector elements are not always positive numbers. However, there is no physical meaning to negative values in the context of clustering since cluster membership cannot have a negative value.[14] Thus, the eigenpairs are then subjected to the 95% positive rule.[17] This states that an eigenvector can only be considered to represent a meaningful cluster if two conditions hold:

(i) the associated eigenvalue is positive;

(ii) 95% of the eigenvector's magnitude is contributed by either the squared values of the negative or positive components only.

(4) In matrix $\mathbf{X}$, the rows represent the molecules and the columns (the eigenvectors) represent the clusters. Thus, the ijth entry gives the contribution of molecule i to cluster j. Each column is associated with an eigenvalue, with the size of the eigenvalue being related to the significance of the cluster. Since the eigenvalues are ordered, the first clusters are the most significant. Unless otherwise modified, spectral clustering is an overlapping clustering method. In general there are as many clusters as molecules and all molecules will contribute to all clusters. Clusters obtained using a spectral method are sometimes referred to as *eigenclusters*.

Spectral clustering can be considered a global clustering method since all objects are considered at the same time, and the assignment of an object to a cluster takes into account not only its relationship to every other object but also the relationship between any other pair of objects in the dataset. This leaves the question "how does an eigenvector give an insight into the relationships between chemical compounds". Imagine a theoretical chemical space that contains the molecules. If the eigenspace (the space spanned by the eigenvectors) is superimposed over the chemical space, an eigenvector describes a movement through this space between two points. Looking down the vector gives a view into chemical space from the perspective of that eigenvector, with molecules that provide

dx.doi.org/10.1021/ci500480b | J. Chem. Inf. Model. 2014, 54, 3302−3319

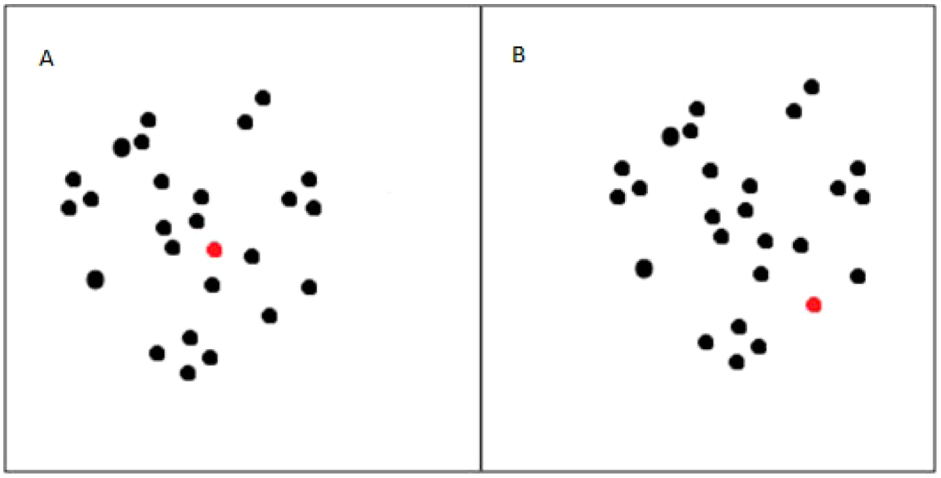


Figure 1. Cohesiveness of eigenclusters. The red spots represent eigenvectors, and the black spots, molecules. Cluster A would have a large eigenvalue, and cluster B, a small eigenvalue.

the largest eigenvector components being located closest to the vector and molecules that make the smallest contribution being located the furthest away. Looking down individual eigenvectors allows the data to be viewed from different perspectives.

The eigenvalue associated with an eigencluster provides a means of quantifying the cluster's cohesiveness. In spectral clustering, cluster cohesiveness defines the number of connections between molecules, and the weights of the connections, such that a set of identical molecules would produce an eigenvalue of $N - 1$ reflecting the presence of a maximum number of connections between the molecules, each weighted with the maximum value of one. In Figure 1 each black point represents a molecule in chemical space and the red point indicates the location where the eigenvector passes through the set of molecules. In cluster A, the eigenvector passes close to the center of the cluster giving a large eigenvalue, as the point where the vector passes through the molecules minimizes the mean distance between the vector and the molecules. Conversely, in cluster B the eigenvector travels through the cluster in a position that is far away from the bulk of the data points giving a small eigenvalue. As this discussion indicates, spectral clustering is closely related to principal components analysis. Given a real symmetric matrix **X**, the covariance matrix of **X**, $\mathbf{C}_\mathbf{X}$ is given by

$$\mathbf{C}_\mathbf{X} = \frac{1}{n}\mathbf{X}\mathbf{X}^\mathrm{T} \tag{5}$$

Then the principal components of **X** are the eigenvectors of $\mathbf{C}_\mathbf{X}$.[22]

The Lanczos Algorithm. Decomposition into the form of eq 3 is referred to as a *full matrix diagonalization* (FMD). Algorithms which approximate the eigenpairs from an incomplete diagonalization of a matrix are often called eigensolvers. The most stable algorithms for identifying eigenpairs from a symmetric matrix are based on a FMD, where *stability* refers to the extent that an algorithm is affected by the presence of roundoff errors.[23] Unfortunately FMD is a time-consuming operation, being $O(N^3)$. However, there are more efficient eigensolvers, such as that by Lanczos,[21] which was suggested by Brewer for further investigation. The Lanczos algorithm is designed for use with sparse matrices. It uses a matrix tridiagonalization procedure, i.e., it reduces the matrix to one where only the diagonal and first off-diagonal elements are nonzero (see Figure 2a). A tridiagonal matrix, **T**, is used as a simple representation of any symmetric matrix, **A**, since it has a number of advantages, including: the eigenpairs of **T** can be elucidated in significantly fewer arithmetic operations than are required for **A**; every **A** can be reduced to **T** in a finite number of elementary orthogonal transformations, whereas (in principle if not in practice), an FMD of **A** can require an infinite number of transformations.

Any real matrix **A** can be written in the form

$$\mathbf{A} = \mathbf{QL} \tag{6}$$

where **Q** is an orthogonal matrix and **L** is a lower triangular matrix (i.e., **L** is zero above the diagonal). This decomposition is known as a **QL** procedure, and where **A** is tridiagonal, it is of $O(N)$.[24] (NB There is, of course, an analogous **QR** procedure.)

Initially the Lanczos algorithm was regarded as a simple way to reduce a matrix to its tridiagonal form but the algorithm proved very susceptible to roundoff error and other issues that occur when using the algorithm for finite precision arithmetic problems.[25] However, Paige showed that despite its mathematical instability, the simple Lanczos algorithm is still an effective tool for the computation of a low number eigenpairs of a matrix,[26] which led to a considerable amount of research into improving the performance of the algorithm.

One of the most common implementations of the symmetric Lanczos algorithm uses an iterative procedure, based on two major steps per iteration, to identify k of the eigenpairs from a matrix **A**. In the initial step, the Lanczos algorithm is applied to **A**, identifying the diagonal elements, α, and the first off-diagonal elements, β, associated with each of the k first-to-converge eigenvalues along with a set of *Lanczos vectors*. It is important to understand that the value of k that is input to the Lanczos method specifies the number of eigenpairs that are calculated by the algorithm and that each of the k eigenpairs can be related to either a positive or a negative eigenvalue depending on which eigenvalues are converged upon first. Hence, k can be divided into two sets: the *pos* set, which are the elements associated with the positive eigenvalues, which form eigenclusters; and the *neg* set, that is related to the negative eigenvalues that do not form eigenclusters. The second step in this process is the use of an iterative solver, such as the **QL** procedure, to identify the k eigenpairs of **A** from the elements of **T** and the Lanczos vectors. One iteration of the Lanczos

dx.doi.org/10.1021/ci500480b | J. Chem. Inf. Model. 2014, 54, 3302−3319

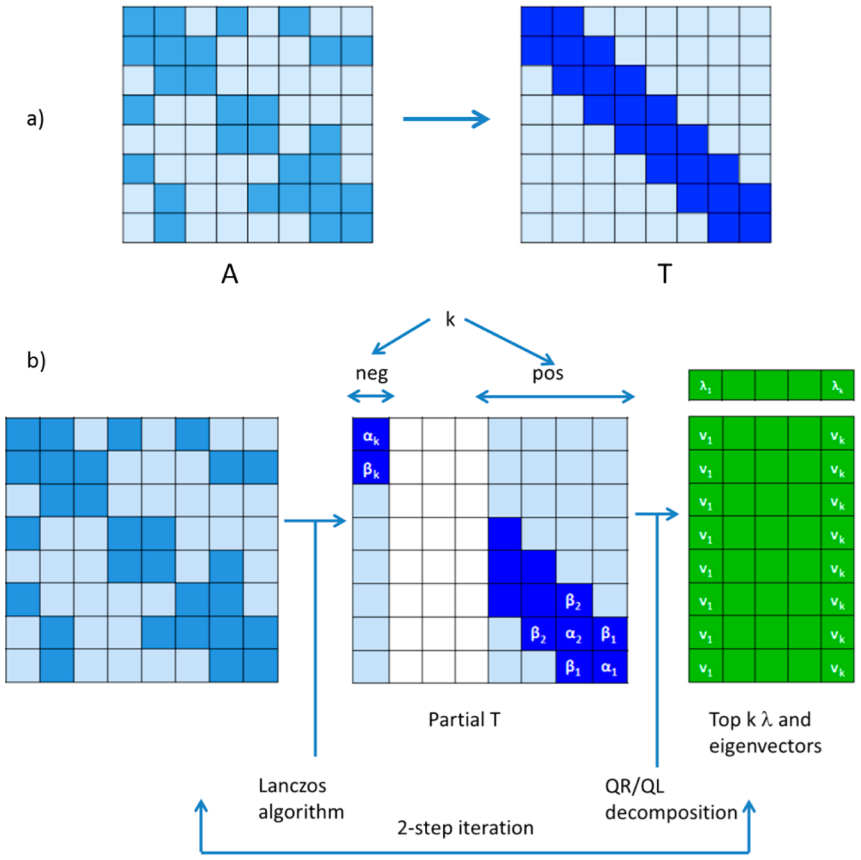


Figure 2. Schematic of eigendecomposition. (a) Idealized tridiagonalization of a sparse symmetric matrix. In matrix **A** the light blue elements represent zero elements, the mid blue elements the nonzero elements before triadiagonalization, and the dark blue elements are the tridiagonal values. (b) Identifying the top *k* eigenpairs of matrix **A**, using repeated iterations of the Lanczos algorithm coupled with a **QL** decomposition. The blue elements have the same meaning as in part a.The elements in white are those for which eigenpairs will not be calculated. The output eigenvalues and vectors are shown in green.

algorithm is illustrated in Figure 2. During subsequent iterations, the eigenvalue/vector approximations already generated are refined and improved. When using the Lanczos algorithm, the eigenvalues tend to be found in order of decreasing absolute magnitude, which is a desirable feature of the algorithm.[26]

When the Lanczos algorithm is applied to a problem in finite precision mathematics, "during its first few iterations, sometimes three, other times as many as 30," the algorithm produces results that are indistinguishable to those calculated by the exact process.[25] This continues, until a new Lanczos vector, *q*, is calculated that is not orthogonal, to working precision, to its predecessors. After a few more steps, the roundoff errors are compounded such that each of the new Lanczos vectors generated is linearly dependent on those that precede it. This in turn leads to the approximation of new incorrect eigenvalues.[23] These problems are compounded as the algorithm begins to recalculate the largest eigenvalues, leading to the calculation of both degenerate eigenvalues and associated eigenvectors that are multiples of previous vectors. The end result is that the Lanczos algorithm in this form is not suitable for use in finding more than the outermost eigenpairs of a matrix.[25] The loss of orthogonality between the Lanczos vectors can be overcome by applying a reorthogonalization procedure to the vectors, for example, using the Gram−Schmidt method.[24]

Eigendecomposition Using Singular Value Decomposition. Other methods exist for approximate eigenvalue/eigenvector calculation, including Singular Value Decomposition (SVD). SVD is a commonly used method of matrix factorization according to the equation:[27]

$$\mathbf{A}_{mn} = \mathbf{U}_{mm}\mathbf{S}_{mn}\mathbf{V}^{\mathrm{T}}_{nn} \tag{7}$$

Where **A** is a matrix of size $m \times n$, **U** is a unitary matrix of size $m \times m$, **S** is a matrix of size $m \times n$ containing the singular values, and $\mathbf{V}^{\mathrm{T}}$ is the conjugate transpose of an $n \times n$ unitary matrix **V**. $\mathbf{S} = \mathrm{diag}(\sigma_1, \ldots \sigma_p)$ where $p = \min(m, n)$. $\sigma_1, \ldots, \sigma_p$ are the singular values of **A**.[28]

Interest in SVD methods stems from their close association with eigendecomposition algorithms, i.e.

- The left singular vectors of **A**, i.e., the columns of **U**, are equal to the eigenvectors of matrix $\mathbf{AA}^{\mathrm{T}}$.
- The right singular vectors of **A**, i.e., the columns of $\mathbf{V}^{\mathrm{T}}$, are equal to the eigenvectors of matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$.
- The nonzero singular values of **A**, the diagonal elements of **S**, are the square roots of the nonzero eigenvalues of both $\mathbf{AA}^{\mathrm{T}}$ and $\mathbf{A}^{\mathrm{T}}\mathbf{A}$.[24] As the eigenvalues are equal to the roots of the singular values, one can select either the positive or negative roots to represent the eigenvalues. In the case of SVD, the positive roots are always selected.

The augmented matrix

$$\mathbf{A}_{\mathrm{aug}} = \begin{pmatrix} 0 & A \\ A^{\mathrm{T}} & 0 \end{pmatrix}$$

dx.doi.org/10.1021/ci500480b | J. Chem. Inf. Model. 2014, 54, 3302−3319

has eigenvalues $\pm\sigma_1$, ..., $\pm\sigma_n$ with corresponding eigenvectors

$$\frac{1}{\sqrt{2}}\begin{pmatrix} u_i \\ \pm v_i \end{pmatrix}$$

Thus, an eigenproblem can be reformulated in the form of a SVD, leading to a solution that produces two distinct sets of eigenvectors linked through a common set of eigenvalues.[28] In the case of symmetric matrices the two sets of eigenvectors, $\mathbf{U}$ and $\mathbf{V}^T$, are equal. One advantage of the SVD method is that since all positive eigenvalues are selected, the number of eigenvalues is the same as the number of clusters.

EXPERIMENTAL SECTION

Programs. We have investigated the performance of the three different eigendecomposition methods described above for spectral clustering of datasets of molecules in three programs developed in-house. For all of the programs, computations were performed using double precision arithmetic.

The first is called NOSC (Non-Overlapping Spectral Clustering) and is based on the FMD eigendecomposition algorithm of Sarker and Boyer[17] which was also used by Brewer. Brewer followed the usual implementation of spectral clustering in which all molecules make a contribution to all clusters. Here, we have implemented a nonoverlapping spectral clustering method to enable comparison with Ward's and *k*-means clustering. This can be achieved by assigning molecules to the cluster to which they make the largest contribution (based on the corresponding eigenvector element). However, initial tests showed that this frequently resulted in molecules being assigned to clusters to which, upon inspection, they did not belong, if their eigenvalue contribution was very small. We therefore introduced a postprocessing *eigenvector threshold*, whereby a molecule whose largest contribution to any cluster was less than the eigenvector threshold was not assigned to any cluster.

Our Lanczos-based spectral clustering method is called L_NOSC and is based on the Lanczos eigendecomposition algorithm implemented in the COLT matrix package[29] along with a full reorthogonalization of the Lanczos vectors based on the Gram−Schmidt method.[24]

Our SVD-based spectral clustering program uses the SVBLIBC library, a C/C++ library, based on SVDPACKC.[30] SVDLIBC is designed solely for application in large sparse matrix problems and hence employs a single SVD algorithm, las2, to carry out operations. las2 has been shown to be consistently the fastest algorithm available for the identification of singular values from large sparse matrices. We have implemented both a nonoverlapping spectral clustering program, SVD-NOSC and an overlapping method, SVD-OSC. The SVD spectral clustering algorithms, together with an implementation of *k*-means clustering, are included in a freely available clustering software program, svdclus, which is available via the Supporting Information.

Data. Brewer used one very small dataset, containing only 125 COX-2 inhibitors. Most of the subsequent reports of the use of spectral clustering involve datasets containing 2000−3000 compounds[18,19] which is a typical size for a diverse set of presumed active compounds, such as those obtained from postfiltering in silico docking results. We therefore selected four activity classes (5HT1A antagonists, Matrix Metalloprotease inhibitors, Renin inhibitors, and Substance P antagonists), each containing between 2000 and 4000 compounds, which were extracted from the ChEMBL database[32] using the Pipeline Pilot software.[33] Each dataset was cleaned by deleting any duplicate molecules, removing all counterions from salts and neutralizing the remaining cations/anions using Pipeline Pilot. A molecule was classified as active if the corresponding $IC_{50} \leq 10\,000$ nM or $-\log(IC_{50}) \geq 5$, otherwise it was classed as inactive. The homogeneity of each of these activity classes was characterized using the mean pairwise similarity among the molecules calculated using Unity fingerprints and the Tanimoto similarity coefficient. Data sets that have a high mean pairwise similarity (larger than 0.5) are described as homogeneous, whereas a mean pairwise similarity of less than 0.5 indicates a dataset is heterogeneous. Table 1 provides further information on the datasets.

Table 1. Datasets

activity class	abbreviation	number of molecules	percentage of actives	mean pairwise similarity
matrix metalloprotease inhibitors	MMP1	3482	51	0.381
5HT1A antagonists	5HT1A	2784	11	0.354
Renin inhibitors	Renin	2166	81	0.520
substance P antagonists	SubP	2760	54	0.411

Molecular Representation. The five fingerprints examined were Unity, BCI, Daylight, ECFP4, and MDL public keys. Brewer used Unity fingerprints, the remainder were chosen to represent a good selection of those available, from very simple (MDL public keys) to highly selective (ECFP4). Although MDL public keys use only 166 bits they have been shown, in some cases, to be more discriminating than structural key fingerprints using many more features,[34,35] while extended connectivity fingerprints have shown the best performance in recent comparative tests including virtual screening,[36] scaffold-hopping,[37] and clustering.[38] BCI fingerprints were generated within the BCI software and are represented by a bitstring of 1052 binary variables derived using a fragment dictionary. Each bit represents a different structural fragment selected for its ability to discriminate between molecules; the presence/absence of a fragment within a molecule is denoted by 1/0, respectively.[39] The default settings were used within the Daylight toolkit to calculate all sequences to a maximum path length of 7, generating Daylight fingerprints of length 2048.[40] Extended connectivity fingerprints, ECFPs, are generated using a circular substructure approach to encoding molecules—the number that is appended to the name refers to the number of bonds that the circular substructures span.[41] The ECFP4 fingerprints were generated using the Pipeline Pilot software and folded to give 1024-element fingerprints. MDL public keys are small two-dimensional fingerprints based upon the use of a structural key, with a one-to-one mapping between predefined chemical features and bits set. These were generated within the Pipeline Pilot software and are represented by a 166 binary bitstring.[42] The Unity fingerprint system uses a combination of both a structural key and a hashed fingerprint system. Unity fingerprints are represented by a bitstring of 992 binary variables and were generated within the Sybyl software.[43]

Parameters Investigated. We consider the following: homogeneity of the dataset; molecular representation;

the parameter γ in the Gaussian filtering function; a threshold on the size of the eigenvector elements; and the size of the dataset. A Gaussian filtering function, which depends on the value of a single tunable parameter, γ, is used to minimize low similarity scores and emphasize the spread of the higher scores. Brewer used a value of 25 for γ showing that it was satisfactory for the small set of COX-2 inhibitors, represented by Unity fingerprints, which formed his test data. Clearly there may not be a single optimum value for γ. It may depend upon the dataset composition or size, or the molecular representation (or other variables). Figure 3a shows the distribution of similarity

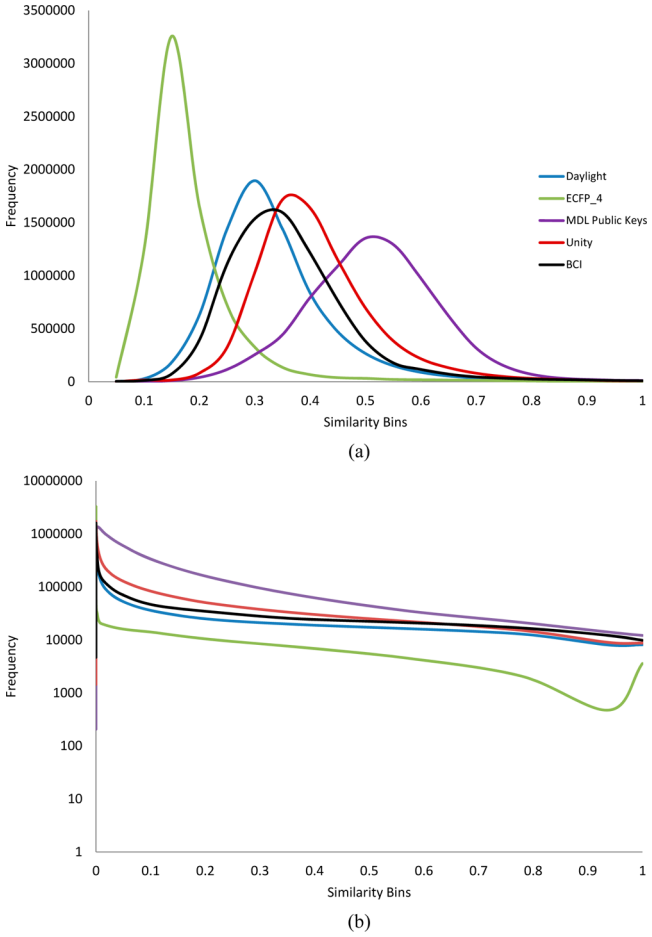


Figure 3. Effect of the Gaussian filtering function. (a) Distribution of similarity scores in the SubP dataset. (b) Distribution of the affinity scores in the SubP dataset, when γ = 25.

scores for the SubP dataset for each fingerprint type, and Figure 3b shows the distribution of the scores when transformed into affinity scores using a value of γ = 25. The skewing of the data toward very low values is shown by the use of the log scale on the vertical axis in Figure 3b. Using γ = 25, a similarity score of 0.3 becomes an affinity score of 4.785×10^{-6}. Table 2 shows the percentage of similarity scores which are below 0.3 for each of the fingerprint types for the each of the four datasets. So, for example, in the SubP dataset, 95% of the ECFP4 affinity scores are lower than 4.785×10^{-6}. Figure 3a and Table 2 show that the ECFP4 fingerprints are the most skewed toward small values in the untransformed state. This poses the question: "Do ECFP4 values need transforming at all?" We initially attempted spectral clustering using the raw ECFP4 scores and, as did Brewer when using Unity

Table 2. Percentage of Similarity Scores Less than 0.3

	5HT1A	MMP1	Renin	SubP
BCI	50	54	19	41
Daylight	74	76	35	55
ECFP4	97	94	71	95
MDL	3	7	3	6
Unity	49	35	9	19

fingerprints, we found that all molecules belonged to all clusters with approximately equal contribution, and thus concluded that a filtering function was still necessary for the ECFP4 measure.

Figure 4 shows the effect on the ECFP4 and MDL scores of the SubP dataset for various values of γ and demonstrates that

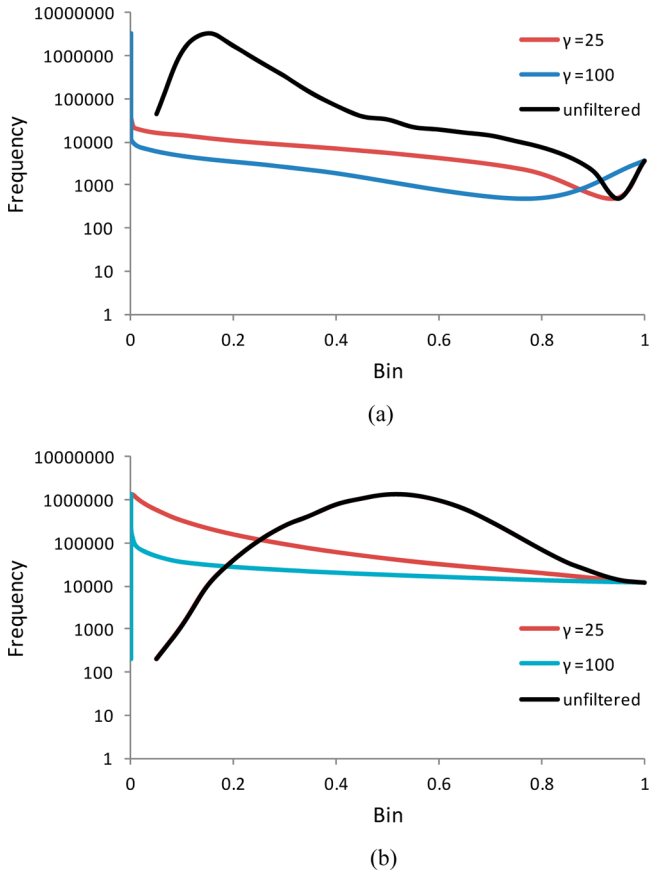


Figure 4. Effect of varying γ on the distribution of scores of the SubP dataset. The x-axis represents binned affinity scores for different values of γ in the Gaussian filtering function and represents binned similarity scores in the unfiltered case, which is why there is no frequency for unfiltered similarity scores below 0.05. (a) ECPF4. (b) MDL.

the main effect of increasing γ is to make already low scores lower, while having relatively little effect on the higher scores.

As well as the homogeneity of a dataset we expect dataset size to be a significant factor in the ability of a spectral clustering algorithm to cluster the data in a reasonable amount of time, since full matrix diagonalization requires $O(N^3)$ operations.

Measuring Clustering Success. The clustering of each dataset was evaluated using the *quality clustering index* measure, QCI. The QCI measure was developed by Varin et al.[38,44] and evaluates the performance of a clustering algorithm in terms of

dx.doi.org/10.1021/ci500480b | J. Chem. Inf. Model. 2014, 54, 3302−3319

Table 3. Analysis of Cluster Types Produced Using ECFP4 Fingerprints

		$\gamma = 25$			$\gamma = 75$		
	eig. thr.	clusters[a]	singletons	unclassified	clusters[a]	singletons	unclassified
5HT1A	0.1	343	30	892	724	120	356
	0.01	353	20	65	731	113	22
	0.001	353	20	3	731	113	3
	1.00×10^{-04}	353	20	0	731	113	2
	1.00×10^{-05}	353	20	0	731	113	1
	1.00×10^{-06}	353	20	0	731	113	1
MMP1	0.1	273	18	1365	802	168	544
	0.01	279	12	157	825	145	43
	0.001	279	12	2	827	143	2
	1.00×10^{-04}	279	12	0	827	143	0
	1.00×10^{-05}	279	12	0	827	143	0
	1.00×10^{-06}	279	12	0	827	143	0
Renin	0.1	81	3	1305	344	56	637
	0.01	81	3	269	360	40	57
	0.001	81	3	12	360	40	1
	1.00×10^{-04}	81	3	0	360	40	0
	1.00×10^{-05}	81	3	0	360	40	0
	1.00×10^{-06}	81	3	0	360	40	0
SubP	0.1	217	17	990	571	104	523
	0.01	223	11	107	587	88	53
	0.001	224	10	11	588	87	15
	1.00×10^{-04}	224	10	0	588	87	10
	1.00×10^{-05}	224	10	0	588	87	8
	1.00×10^{-06}	224	10	0	589	86	4

[a]Clusters is the total number of clusters including singletons.

its ability to separate active and inactive molecules within a dataset, using the equation:

$$\mathrm{QCI} = \frac{p}{p+q+r+s} \times 100 \tag{8}$$

Where, p is the number of active molecules in active clusters. q is the number of inactive molecules in active clusters. r is the number of active molecules in inactive clusters. s is the number of active singleton clusters.

An *active cluster* is defined as a cluster containing a greater percentage of active molecules than the dataset as a whole.

The value of the QCI measure depends on both the dataset and the quality of the clustering produced. Since it rates a clustering by considering the position of active and inactive molecules, it varies depending on the proportion of active molecules. In the extreme case where all molecules are active then the QCI score is inversely proportional to the number of singleton clusters. We obtained estimates for the mean and variance of the QCI score for the MMP1 dataset as follows: a set of 2000 MMP1 compounds were drawn at random from the dataset and 51% (the percentage of active compounds in the dataset) of these were selected at random and labeled as active. These compounds were clustered using NOSC and the QCI score calculated. This was repeated 50 times (with new selections of 2000 compounds) and a mean QCI score of 48 with variance of 5.5 was obtained. The same procedure was used for the other three datasets giving: a mean QCI score of 55 with variance 7.4 for the Renin dataset, which contains 80% actives; a mean QCI score of 49 with variance 4.8 for the SubP dataset which has 54% actives; and a mean QCI score of 22 of with variance 1.1 for the 5HT1A dataset which has 11% active compounds.

RESULTS AND DISCUSSION

Parameterization of FMD Spectral Clustering using NOSC. Each of the four data sets was represented using each descriptor and similarities were calculated using the Tanimoto coefficient. In the Gaussian filtering function, γ was varied from 25 to 400 in increments of 25, while the eigenvector threshold used for assigning molecules to clusters was set at values of 0.1, 0.001, ..., 10^{-6}. As γ increases, more emphasis is placed on the biggest similarity scores. In practice this means that the most significant eigenvalues become larger, and the next eigenvalues are relatively small. If these differences are too big then artificial clusters are created. We are therefore looking for the smallest value of γ for which a "reasonable" number of nonsingleton clusters are produced.

Table 3 shows the variation in the number of clusters and singletons for all four datasets, represented by ECFP4 fingerprints, at all eigenvector thresholds for $\gamma = 25$ and $\gamma = 75$. We see that, at all values of the eigenvector threshold, using $\gamma = 75$ gave many more clusters and a decrease in the number of unclassified compounds, but at the expense of the creation of more singleton clusters. Decreasing the eigenvector threshold from 0.1 to 0.0001 clearly has the effect of allowing molecules to be placed in clusters based on lower eigenvector contributions, while lowering the threshold further has no effect since all molecules have already been placed in clusters. The only exception is for SubP where, for $\gamma = 75$, a very few molecules remain unclassified. For these four datasets, represented by ECFP4 fingerprints, $\gamma = 25$, 50, or 75 seems a reasonable choice, depending on the number of clusters required. An eigenvector threshold of 0.001 or 1.00×10^{-04} is sufficient to allow most or all compounds to be clustered, for both values of γ.

dx.doi.org/10.1021/ci500480b | J. Chem. Inf. Model. 2014, 54, 3302−3319

Table 4. Parameter Values for the NOSC Algorithm

	5HT1A		MMP1		Renin		SubP	
	γ	threshold	γ	threshold	γ	threshold	γ	threshold
BCI	75	0.001	75	0.001	175	1×10^{-6}	75	0.001
Daylight	75	0.001	50	0.001	175	1×10^{-6}	50	0.001
ECFP4	50	0.001	50	0.001	75	1×10^{-4}	50	0.001
MDL	100	0.001	100	0.001	200	1×10^{-6}	100	0.001
Unity	75	0.001	75	0.001	175	1×10^{-6}	75	0.001

The outcome of the parametrization experiments is given in Table 4. The results show that the different molecular representations require different sets of parameters. Generally, NOSC was able to cluster the heterogeneous datasets effectively using a low γ value for more specific fingerprints, such as ECFP4 and Daylight (e.g., γ = 25 or 50), while BCI or Unity required a higher value (e.g. γ = 75) and MDL Public keys a very high value (e.g. γ = 100). The required value of γ reflects the distribution of the similarity scores in Figure 3a. For ECFP4, Figure 4a shows that a value of γ = 25 means that only a few thousand affinity scores are above 0.6 and 97% of scores are below 10^{-4}. At the opposite extreme are the MDL public keys. Figure 4b shows that, even when γ = 100, many more of the scores are above 0.6 and only 11% of the scores are below 10^{-4}. The homogeneity of the dataset also plays a part, with the homogeneous Renin data set generally needing a higher γ value (e.g., γ = 75 for ECFP4) and also needing a lower value for the eigenvector threshold. Values of 10^{-3} or 10^{-4} were appropriate for the eigenvector threshold. However, these parameter values are likely to be dataset dependent, and we recommend that parametrization be carried out before using spectral clustering methods with any new dataset.

Table 5 shows the number of clusters, number of singletons, and QCI values when each of the data sets was clustered using the parameter values taken from Table 4. It is clear that using ECFP4 with the NOSC algorithm produced the best clustering of each of the datasets, at least as measured using the QCI value. ECFP4 always gives more clusters and, although there are more singletons, the majority of the additional clusters are nonsingletons.

Table 5. Results of Clustering Using NOSC[a]

dataset	fingerprint type	clusters	singletons	QCI
5HT1A	BCI	391	67	34.1
	Daylight	460	86	35.2
	ECFP4	604	82	57.4
	MDL	181	15	31.1
	Unity	401	61	32.3
MMP1	BCI	308	51	71.0
	Daylight	369	49	72.8
	ECFP4	631	76	81.7
	MDL	143	14	66.2
	Unity	280	43	73.4
Renin	BCI	228	24	60.7
	Daylight	325	69	61.0
	ECFP4	360	40	80.9
	MDL	131	16	72.7
	Unity	220	36	56.7
SubP	BCI	262	30	59.0
	Daylight	237	28	61.8
	ECFP4	450	53	65.8
	MDL	166	14	46.3
	Unity	233	26	57.0

[a]NOSC was run using the parameters detailed in Table 4.

Table 6. Time Taken to Perform an FMD Using the Optimal Parameters Given in Table 4[a]

dataset	fingerprint	mean time (s)
5HT1A	BCI	1627
	Daylight	1637
	ECFP4	1524
	MDL	1502
	Unity	1592
MMP1	BCI	3299
	Daylight	3384
	ECFP4	3204
	MDL	3305
	Unity	3392
Renin	BCI	687
	Daylight	650
	ECFP4	634
	MDL	652
	Unity	679
SubP	BCI	1513
	Daylight	1497
	ECFP4	1530
	MDL	1465
	Unity	1593

[a]The timings are the mean of five runs, with γ = 25.

Table 6 shows the time required for a FMD for each of the datasets, for each fingerprint type when γ = 25. The FMDs were run on an Intel Core2 with 3 GHz processor and 4 Gb memory running Linux. There is some variation between the different descriptors, but the size of the data set has a much greater effect on the time required for the FMD, with the time varying from about 10 min for Renin, to up to an hour for MMP1. The number of operations required to decompose the Renin similarity matrices is $\approx 1.06 \times 10^{10}$ while $\approx 4.22 \times 10^{10}$ operations are needed for the MMP1 similarity matrices. Note that the same value of γ = 25 was used in all these runs, since altering γ has little effect on the timing of FMD for these relatively small matrices.

NOSC was then compared with more conventional clustering methods as follows. The Ward's and *k*-means clustering algorithms were applied to each of the activity classes using the relevant programs from Digital Chemistry.[39] Both Ward's and *k*-means require the user to specify the number of clusters required. Therefore, for this investigation each of the activity classes was clustered twice, at two different choices of clustering level. The OPTCLUS program from Digital Chemistry uses the Kelley measure[45] to identify the optimal level of hierarchy for the clustering of a dataset. The Kelley measure was initially

dx.doi.org/10.1021/ci500480b | J. Chem. Inf. Model. 2014, 54, 3302−3319

derived to select optimal clusters of protein NMR ensembles. It uses the following equation:

$$\mathrm{KELLEY}_l = (n-2)\left(\frac{\bar{d}_{wl} - \min(\bar{d}_w)}{\max(\bar{d}_w) - \min(\bar{d}_w)}\right) + 1 + k_l$$

Where: N is the number of objects; $\bar{d}_{wl}$ is the mean of distances between points in the same cluster at level l; $\max(\bar{d}_w)$ is the maximum distance value across all cluster levels; $\min(\bar{d}_w)$ is the minimum distance value across all cluster levels; k_l is a user defined value which is used to penalize cluster levels which contain a large number of singletons.

The Kelley measure requires the calculation of the Kelley score for each clustering level and subsequent identification of the level which has the maximum score. Initially the OPTCLUS program was used to select an optimal number of clusters for each combination of activity class and fingerprint type when using the Ward's algorithm. The *k*-means algorithm was then configured to generate the same number of clusters. The optimal hierarchy level selected by OPTCLUS depends on the clustering algorithm being used and generally gave a different number of clusters from that produced by the NOSC algorithm. Thus, to ensure that the difference in the QCI scores between clustering methods was not solely due to the difference in the number of clusters produced by the respective algorithms, both the Ward's and *k*-means methods were also configured to generate the same number of clusters as produced by the NOSC algorithm. The scores for the NOSC number of clusters are given in Table 7, and the QCI scores for the OPTCLUS number of clusters are given in Table 8.

Table 7. Results of Clustering Using NOSC, Ward's, and *k*-Means[a]

			NOSC	Ward's	*k*-means
dataset	fingerprint type	clusters	QCI	QCI	QCI
5HT1A	BCI	391	34.1	33.3	34.3
	Daylight	460	35.2	31.0	31.9
	ECFP4	604	**57.4**	35.6	34.6
	MDL	181	31.1	26.1	22.7
	Unity	401	32.3	32.2	30.8
MMP1	BCI	308	71.0	69.9	73.3
	Daylight	369	72.8	70.2	74.5
	ECFP4	631	81.7	76.8	80.6
	MDL	143	66.2	73.3	70.6
	Unity	280	73.4	76.9	75.4
Renin	BCI	228	60.7	60.4	60.0
	Daylight	325	61.0	60.8	61.6
	ECFP4	360	**80.9**	69.7	66.8
	MDL	131	**72.7**	60.3	62.7
	Unity	220	56.7	63.0	62.3
SubP	BCI	262	59.0	63.7	62.4
	Daylight	237	61.8	61.7	61.3
	ECFP4	450	65.8	65.6	65.4
	MDL	166	46.3	58.9	55.6
	Unity	233	57.0	60.0	60.0

[a]NOSC was run using the parameters detailed in Table 4, giving the number of clusters in the clusters column. Ward's and *k*-means were run to give the same number of clusters. Values in bold indicate that NOSC noticeably outperformed Ward's and *k*-means.

Considering first Table 7, all fingerprints and all clustering methods gave QCI scores significantly better than random for the MMP1 and SubP datasets. For the 5HT1A dataset, MDL fingerprints gave QCI scores which were no better than randomly assigning compounds to be active no matter which clustering method was used but all other fingerprint/clustering method combinations were significantly better than random. For the Renin dataset, clustering using ECFP4 fingerprints gave better than random QCI scores for all methods and using NOSC with MDL fingerprints was also better than random but all other fingerprint/clustering method combinations gave QCI values within random variation of the estimated mean of 55. In Table 8 where the OPTCLUS number of clusters is used for Ward's and *k*-means, the performance of these two methods is in some cases better than in Table 7. So for the 5HT1A dataset, the MDL fingerprints now give significant QCI scores while for the Renin dataset only the BCI fingerprints now give QCI scores no better than random.

Table 8. Results of Clustering to Generate the Number of Clusters Determined Using the Kelley Measure

		OPTCLUS	Ward's	k-means
dataset	fingerprint type	clusters	QCI	QCI
5HT1A	BCI	359	29.7	30.6
	Daylight	375	30.9	27.4
	ECFP4	396	33.5	31.8
	MDL	361	31.1	29.5
	Unity	396	31.7	29.1
MMP1	BCI	329	74.5	72.8
	Daylight	359	74.2	72.1
	ECFP4	413	82.7	81.7
	MDL	362	78.5	77.5
	Unity	362	79.3	77.3
Renin	BCI	194	59.0	57.5
	Daylight	224	63.9	59.6
	ECFP4	255	64.8	65.7
	MDL	208	64.6	63.0
	Unity	237	64.8	64.0
SubP	BCI	278	64.4	62.9
	Daylight	279	62.5	61.3
	ECFP4	323	65.5	63.7
	MDL	312	61.8	59.4
	Unity	269	62.0	60.6

Table 7 shows that when the NOSC algorithm "chooses" the number of clusters, the spectral clustering NOSC method consistently gave the highest QCI score of the three methods. ECFP4 gave the best results of all five fingerprints and in some cases was much better than all other fingerprints. For example, in the case of the 5HT1A dataset, the ECFP4 QCI score was 57 and the next best score was 35. When the OPTCLUS number of clusters was used there is a single instance, for MMP1, where both Ward's and *k*-means score slightly higher than NOSC, but in general the NOSC method is at least as good as the other two methods and is often significantly better. ECFP4 again gave the best QCI values for each dataset, for both Ward's and *k*-means clustering, and we therefore only report results using ECPF4 in the remainder of the paper.

Lanczos-Based Spectral Clustering. The main limitation of the NOSC algorithm is that it does not scale well to larger datasets, being of $O(N^3)$ in the number of compounds, N. In order to move to a more efficient algorithm, such as the Lanczos, it is necessary to use a sparse input matrix. Sparsity is defined as the percentage of zero entries. There were no (off-diagonal) zero entries in the affinity matrices thus the initial

dx.doi.org/10.1021/ci500480b | J. Chem. Inf. Model. 2014, 54, 3302−3319

sparsity was 0%, although at high values of γ many values are extremely small. We therefore applied an *affinity threshold*, t, to each affinity matrix, setting affinity values less than t to 0. The affinity threshold required depends on the value of γ. For $\gamma = 50$ or greater, preliminary experiments showed that, when represented by ECFP4 fingerprints, an affinity threshold of 1.0×10^{-6} removed sufficient entries that the matrices were at least 95% sparse for the more heterogeneous 5HT1A, MMP1, and SubP datasets. For the Renin dataset, a threshold of 0.001 was required to make the affinity matrix 95% sparse. When $\gamma = 25$, a higher threshold of 1.0×10^{-4} was required for 95% sparsity, except for the Renin dataset which required a threshold of 0.01. Thus, a threshold of 0.001 ensured that all matrices were at least 95% sparse for $\gamma \geq 50$. When the NOSC algorithm was rerun with the affinity threshold set at 0.001, using the optimal parameter settings of Table 4, very similar QCI scores were obtained, demonstrating that the affinity threshold does not have a detrimental effect on the NOSC algorithm.

When using the Lanczos algorithm (L-NOSC) for clustering purposes, it is necessary to specify the number of clusters required. This is a difficult problem in clustering in general, but with the Lanczos algorithm there is an added complication. As previously mentioned, eigenpairs that are calculated by L-NOSC can include both positive and negative eigenvalues. However, clusters are only generated from eigenvectors with positive eigenvalues.[17] Therefore, the relationship between the number of eigenvalues found, k, and the number of positive eigenvalues, p, was investigated. This relationship is not trivial, and the ratio of positive to negative eigenvalues varies based on the characteristics of the dataset being clustered.

Affinity matrices for each of the datasets were generated, using values of the Gaussian filtering parameter $\gamma = 25$, 50, 75, and 100. These affinity matrices were then made sparse using different affinity thresholds, and the L-NOSC algorithm was applied for different values of k and the percentage of positive eigenvalues recorded. Results are given in Table 9. In most cases the percentage of positive eigenvalues increases with the number of eigenvalues sought and also increases as γ is increased. Thus, it was decided that if p clusters, corresponding to p positive eigenvalues, were required, then k should be set at 120% of p. This ensures that approximately p clusters are found.

Another important consideration is how to decide on an appropriate value of p. Unfortunately there is no hard and fast rule for this and instead a balance must be struck between finding enough eigenpairs to give a good separation of the data and avoiding the identification of too many eigenpairs due to the additional computational cost. Factors to consider are the magnitudes of the eigenvalues generated and also the characteristics of the clusters that are generated.

In the L-NOSC algorithm, the eigenvalues are generally found in decreasing order. This is a desirable feature, as the eigenpairs that contain the "most" information will be found first, as in principal component analysis where the first few principal components contain the greatest share of the information. Figure 5 is a plot of the positive eigenvalues obtained for the SubP dataset, with $\gamma = 50$, which shows that the slope of the eigenvalue distribution decreases to a plateau where the eigenvalues are close to 1, before the eigenvalues begin to decrease again at about the 1500th eigenvalue.

The second factor to consider is the nature of the clusters generated. Four different types of clusters are formed during spectral clustering:

Table 9. Percentage of Positive Eigenvalues for Different Values of γ and Number of Eigenvalues, k[a]

		MMP1	Renin	SubP	5HT1A
γ	k	%p	%p	%p	%p
25	100	83	89	84	81
	200	82.5	86	81	81.5
	300	80.7	80.7	79.3	83
	400	84	78.5	80.7	87
50	100	85	78	81	87
	200	89.5	83.5	83.5	91.5
	300	93	88.7	88.3	94.3
	400	93.5	91.5	91.2	95
75	100	90	84	88	91
	200	94	91	92.5	97.5
	300	95.7	95	95	96
	400	96.5	95.2	94.7	96.7
100	100	95	91	91	97
	200	98	94	95.5	97.5
	300	97.6	96.3	95	97
	400	98.2	96.5	96.7	97.2

[a]The affinity threshold is 0.001.

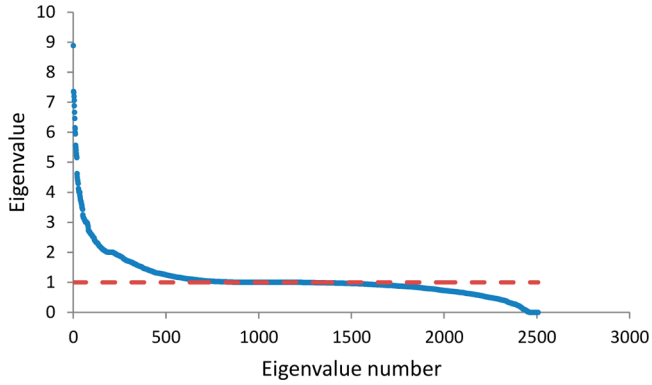


Figure 5. Eigenvalues in decreasing order. The eigenvalues were obtained from the SubP data set using $\gamma = 50$. The dashed red line represents eigenvalue = 1.0.

- A *positive cluster* is an eigencluster that contains at least two compounds.
- An *empty cluster* is formed when the requirements are met to generate an eigencluster but no compounds are assigned to it based upon their eigenvector score.
- A *true singleton* is a singleton cluster that is naturally formed, i.e., formed as a result of one compound having a chemical scaffold significantly different to all others which produces an eigenvector dominated by a single molecule.
- A *forced singleton*/unclassified molecule is formed when a molecule does not make a contribution large enough to be assigned to an eigencluster and, as a result, is forced into a cluster on its own.

Figure 6 shows the distribution of the different types of clusters formed for the SubP data set as the number of positive eigenvalues increases. As the number of eigenvalues reaches 850, there is an obvious change in the gradient of the curve representing the number of positive clusters. This represents the point at which finding further eigenvalues becomes undesirable as the quality of the clustering begins to decrease rapidly, which is shown by the lack of new clusters containing

dx.doi.org/10.1021/ci500480b | J. Chem. Inf. Model. 2014, 54, 3302−3319

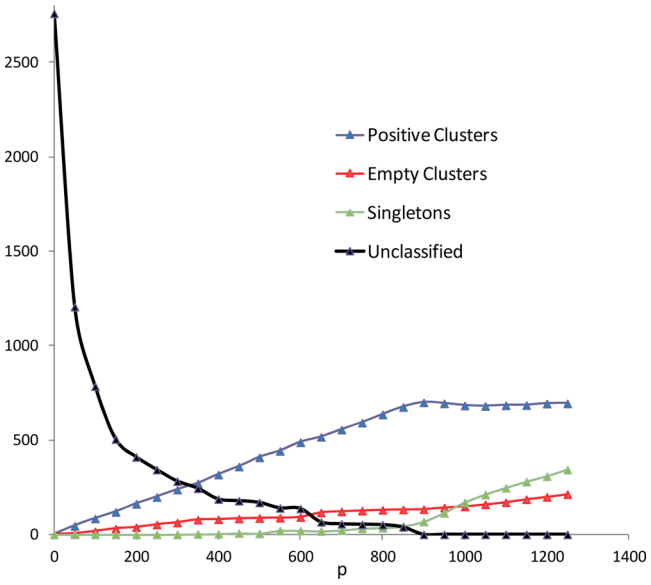


Figure 6. Distribution of different cluster types for the SubP activity class.

more than a single molecule, and the rapid increase in the number of singletons formed. The eigenvalues associated with clusters beyond this point have magnitudes of 1.01 or less. This correlates with the leveling off of the slope in Figure 5. This trend was also found in the other datasets, with the number of singleton clusters increasing rapidly when more than 750 eigenvalues are found for Renin, and 1000 eigenvalues for both MMP1 and 5HT1A, corresponding to eigenvalues of 1.0, 1.02, and 1.0, respectively. This allows us to conclude that clusters related to eigenvalues ≤ 1 can be discounted for these datasets. However, it is not clear that all the clusters associated with positive eigenvalues greater than 1 are meaningful clusters. We note that in the reported uses of spectral clustering, typically only a few hundred clusters are required,[18,19] in order that a representative set of compounds may be selected.

As noted in the Methods section, the basic Lanczos algorithm is a very good algorithm for finding a few eigenvalues of a large sparse matrix.[46] However, the need for reorthogonalization when more eigenpairs are required means that the time (and space) requirements of the algorithm can increase significantly. To investigate the time requirements of the algorithm, affinity matrices were generated for each of the four datasets, represented by each of the descriptors, using a value of $\gamma = 25$. These matrices were then made sparse through the application of an affinity threshold of 0.001. The L-NOSC algorithm was applied to each of these sparse affinity matrices for different values of k, and the time required to identify the top k first-to-converge eigenpairs recorded. These times were then compared against the time taken to carry out a FMD of the activity classes. The aim was to identify the point at which the Lanczos based approach becomes more time-consuming than using the full eigenvalue decomposition.

Table 10 shows that, for the ECFP4 fingerprints, the Lanczos algorithm is faster than a FMD for only up to between 300 and 400 eigenvalues, which equates to about 300 clusters (since to find 300 clusters requires approximately 350 eigenvalues). Thus, for datasets of up to 4000 compounds (the size of the MMP1 dataset) where more than 300 clusters are required, then a FMD is a better option than the use of the approximate Lanczos algorithm. Interestingly, the time required to calculate k eigenpairs for ECFP4 fingerprints was found to be considerably longer than for any other fingerprint type. This result was unexpected as in general ECFP4 fingerprints produce the most sparsely populated input matrices, and therefore, we would expect their decomposition to take significantly less time than other fingerprint types which produce more densely populated input matrices. The reason seems to be the problems encountered by the Lanczos algorithm when it is applied to finite precision mathematical problems. These issues mean that the generation of eigenvectors from ECFP4 similarity matrices, which produce eigenvectors containing a large number of extremely small elements, leads to the need for more operations per iteration to be carried out in order to elucidate and optimize the calculated eigenvectors, significantly increasing the execution time of the algorithm.

Table 10. Comparison of Execution Times (Seconds) for the Lanczos and FMD Algorithms

	time taken for FMD	largest k value for which L-NOSC is faster
5HT1A	1616.19	350
MMP1	3584.63	400
Renin	758.82	300
SubP	1645.95	350

Although FMD appears better suited than the Lanczos algorithm to finding more than about 300 clusters from sets of 2000−3000 molecules, FMD is unsuitable for larger datasets. Hence, experiments were carried out aimed at identifying, first, if it was possible to cluster datasets up to 10 000 molecules using L-NOSC and, second, the times required to calculate the top k eigenpairs of each dataset. In order to generate data for these experiments the four datasets (5HT1A, MMP1, Renin, and SubP which contain 2784, 3482, 2166, and 2760 molecules, respectively) were agglomerated, and random selections of between 1000−10 000 molecules identified. Up to 600 eigenpairs, giving up to ∼500 clusters, were identified using the Lanczos algorithm.

The timings for the Lanczos algorithm are given in Table 11. For comparison, 5000 molecules took about 9150 seconds using FMD and 10 000 compounds took about 46 500 seconds (∼13 hours). Table 11 shows that if 500 or fewer clusters are required then the Lanczos algorithm is faster for datasets of 5000 or more molecules. However, it is still not fast enough for larger scale clustering since it took about 5 hours to cluster

Table 11. Increase in Time with Dataset Size[a]

	k					
dataset size	100	200	300	400	500	600
1000	4	16	49	159	159	509
2000	11	70	135	579	1007	1742
3000	20	93	588	905	1743	3476
4000	35	207	811	1981	2964	4791
5000	59	335	853	3631	7101	8746
6000	73	261	1213	3275	6997	15181
7000	92	374	1305	4153	8448	15776
8000	124	430	1547	5532	12614	18518
9000	168	521	1875	6494	15425	23289
10000	221	548	2051	7013	18909	27695

[a]Time (seconds) required to identify the top k eigenpairs using L-NOSC. Molecules represented by ECFP4 fingerprints using a γ value of 25.

dx.doi.org/10.1021/ci500480b | J. Chem. Inf. Model. 2014, 54, 3302−3319

10 000 molecules into 500 clusters. Brewer recommended the Lanczos algorithm, based on the work of Shi and Malik, but the circumstances in which they use Lanczos are very different[47] to the molecular clustering being considered here.

Singular Value Decomposition. While the performance of the Lanczos algorithm is acceptable for clustering up to 10 000 molecules, it is disappointing to conclude that it is not really suitable for replacement of an FMD procedure since it also scales badly with dataset size. Additionally, the Lanczos algorithm is slower when more clusters are required, which is not a problem for FMD. However, SVD offers an alternative approximate eigendecomposition.

We first wanted to check that SVD-NOSC was a direct replacement for the NOSC algorithm. We therefore used SVD-NOSC to cluster the 125-compound COX-2 dataset used by Brewer. As expected, this gave identical results, in terms both of cluster membership and of eigenvector contribution to each cluster, to those obtained by Brewer. In order to assess the scalability of SVD-NOSC to large datasets we used subsets of compounds extracted from the MDL Drug Data Report database (MDDR).[48] The database contained 102 513 biologically relevant molecules and their structural analogues. Random subsets of N, in 10 000 increments, molecules were extracted from the MDDR database using Pipeline Pilot. The compounds were represented using RDKit circular fingerprints[31] (an open source implementation of the ECFP fingerprint) and similarity was calculated using the Tanimoto coefficient. Each subset was clustered at two different values of k, 100 and 1000, using SVD-NOSC. The time required to cluster each dataset when $\gamma = 100$ and both the similarity and eigenvector thresholds were set to 1×10^{-6} was recorded. Figure 7 shows the increase

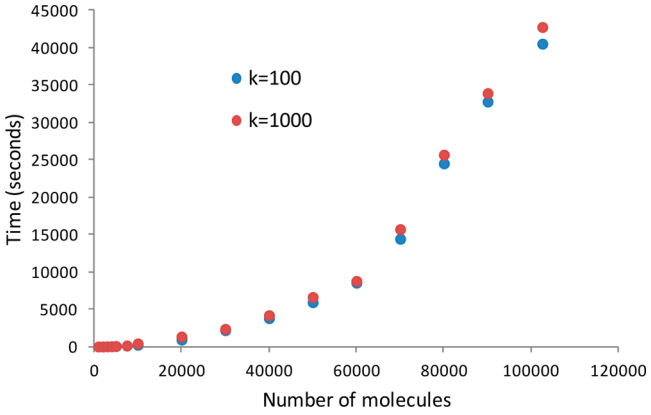


Figure 7. Clustering datasets of increasing size extracted from MDDR using RDKit circular fingerprints and SVD-NOSC. $\gamma = 100$.

in time with subset size. Above about 60 000 compounds the increase in time was approximately linear and there is not much increase in time in going from $k = 100$ to 1000. It took about 12 hours to cluster the full set of 100 000 compounds, demonstrating that SVD-based spectral clustering is indeed applicable to larger datasets.

The value of $\gamma = 100$ was chosen for this experiment since large datasets were under consideration and a consistent value for γ was needed for a fair time comparison. However, the earlier discussion showed the need for different values of γ depending on the dataset size and this is borne out by a closer analysis of the clusters produced during the timing experiments. Examination of the 100 clusters produced for the 5000 molecule dataset showed that the molecules were placed into a set of clusters mainly comprised of singletons. This occurred since the relatively high value for γ led to the formation of a matrix which was too sparse, and therefore did not contain sufficient information, leaving a minimal number of eigenvector contributions above the eigenvector threshold which is used to place molecules into clusters. By reducing the value of γ to 10, the clusters produced using SVD-NOSC significantly improved with compounds based on similar scaffolds being clustered together. The superior clustering obtained by decreasing the value of γ from 100 to 10 comes with a time penalty, with the time taken increasing from 58 seconds when $\gamma = 100$ to 328 seconds when $\gamma = 10$ for 5000 compounds and 100 clusters. Examination of the clusters produced for datasets of increasing size leads us to conclude that for sets of 10 000 to 50 000 molecules a value of $\gamma = 100$ is appropriate while for 50 000 or more molecules a value of $\gamma = 200$ is more appropriate.

Asymmetric Clustering using SVD. A potential advantage of using an SVD for clustering is that two sets of eigenvectors are produced, linked by a common set of eigenvalues. This allows the possibility of basing clustering on an asymmetric similarity measure, which is generally not possible with more conventional clustering algorithms. The Tversky Index is an asymmetric similarity measure calculated using the formula

$$S_{AB} = \frac{c}{\alpha(a - c) + \beta(b - c) + c}$$

where a is the number of bits set in molecule A and not molecule B, b is the number of bits set in molecule B and not molecule A, and c is the number of bits in common. The two weighting functions, α and β, determine the relationship between the three variables a, b, and c. When both α and β are set to 0.5, the similarity scores are equal to those obtained when using the Dice coefficient, and when both are set to 1, the similarity scores are equal to those of the Tanimoto coefficient.[49] Setting α and β to be unequal produces similarity scores where the similarity of molecule A to B does not equal that of B to A. If α is 1.0 and β is 0.0, a high value of S_{AB} means that A is a superstructure of B. Conversely, if α is 0.0 and β is 1.0, a high value of S_{AB} means that A is a substructure of B. The Tanimoto score is dependent on the number of bits set which renders it a poor measure of similarity when used with small molecules, such as fragments,[50] and it has been suggested that the substructure/superstructure possibilities of the Tversky index might be more appropriate for calculating the similarities of chemical fragments. For example, the Tversky index can be used to select larger compounds that contain substructures similar to a fragment of interest. On the other hand, if one wishes to find smaller fragments that might show similar activity to a large molecule hit, the Tversky index can be used in "superstructure mode". The Tversky index with $\alpha > 0.5$ has also been reported to be a better measure of similarity than the Tanimoto coefficient in a recent large scale study.[51]

The availability of SVD-based clustering therefore led us to investigate the use of the Tversky index in clustering a set of fragments such as those used in fragment-based drug discovery (FBDD).[52] A fragment in FBDD is usually considered to be a molecule containing 12 or fewer non-hydrogen atoms. We wanted to see if the SVD-based methods could produce meaningful clusters using the Tversky index and chose to investigate this with a set of chemical fragments which we correctly anticipated would be poorly clustered by the Tanimoto index. The hope was that one of the **U** or **V** clusters would contain sets of molecules linked by superstructure, and the other linked

dx.doi.org/10.1021/ci500480b | J. Chem. Inf. Model. 2014, 54, 3302−3319

by substructure. The two sets of eigenvectors produced by SVD are commonly denoted **U** and **V**. If a symmetric similarity measure, such as the Tanimoto coefficient, is used then **U** = **V** and a single set of clusters is found. If, however, the affinity matrix is asymmetric **U** ≠ **V**, and this gives two distinct sets of clusters which we term the **U** and **V** clusters. As far as we are aware, this use of the **U** and **V** clusters represents a novel approach to clustering.

A set of 100 chemical fragments, here called the DC100 dataset, was extracted from FBDD screens carried out at AstraZeneca. The DC100 dataset is shown in the Supporting Information. This fragment set was manually clustered into 14 clusters. The fragments were represented by RDKit circular fingerprints[31] and clustered into 14 clusters using the SVD-NOSC algorithm using different values of α and β in the Tversky similarity coefficient. The value of γ =10 was used in the Gaussian filtering function since the dataset was very small, and both the affinity and eigenvector thresholds were set to 0.0001. The Jaccard coefficient, was used to compare the clusters produced with the ideal set, where for cluster methods C_1 and C_2:

$$\text{Jaccard}(C_1C_2) = \frac{a}{a+b+c}$$

a is the number of pairs of molecules that are clustered together in both C_1 and C_2; b is the number of pairs of molecules that are clustered together in C_1 but not in C_2; and c is the number of pairs of molecules that are clustered together in C_2 but not in C_1. The Jaccard scores are given in Table 12.

Table 12. Comparison between the Ideal and SVD-NOSC Clustering of the DC100 Fragment Set[a]

	clustering 2		Jaccard	
clustering 1	α	β	**U**	**V**
ideal	1	1	0.604	0.604
	0.9	0.1	0.655	0.627
	0.8	0.2	0.627	0.627
	0.7	0.3	0.627	0.627
	0.6	0.4	0.627	0.612
	0.5	0.5	0.645	0.645

[a] α and β are the values used in the Tversky measure for the SVD-NOSC clustering. Where $\alpha = \beta$, the **U** and **V** clusters are the same.

Table 12 shows that all methods give a reasonable approximation of the manual clustering since all Jaccard scores are at least 0.6. However, use of the Tanimoto coefficient, with SVD-NOSC ($\alpha = \beta = 1$), gives the worst agreement with the manual clustering. Using SVD-NOSC with α = 0.9, β = 0.1, and with $\alpha = \beta$ = 0.5 (the Dice coefficient) gave the highest Jaccard scores. In fact use of the SVD-NOSC algorithm with the Tanimoto coefficient places most molecules (64 of 100) into a single large cluster, which is clearly undesirable and also produces 2 singleton clusters. In contrast, use of other Tversky combinations produce sets of clusters varying in size from 5−13, with no singletons. Visual inspection of the "manual" clusters shows that they are indeed clustered such that most members of each cluster show a large substructure in common. In cluster 1, for example, all the molecules are phenyl sulfonamides and most also have an aniline group. Cluster 3 contains quinolines and isoquinolines, cluster 6 is benzoic acids, etc. This gives initial reassurance that the asymmetric clustering is behaving as hoped. The similarly improved behavior of the Dice coefficient is interesting. It might also be viewed as favoring compounds with a common substructure as it gives twice as much weight to features in common between the two compounds than those in one and not the other.

A larger set of 741 fragments screened against a single biological target at AstraZeneca (referred to as Target X for anonymity) was then selected. 215 of them had measured activity less than 300 in a scaled, arbitrary unit of activity related to percent inhibition—these fragments were deemed active, and the remainder, inactive. The Target X dataset, represented by RDKit circular fingerprints, was clustered using the SVD-NOSC algorithm (similarities calculated using the Tanimoto coefficient) to obtain sets of clusters. In order to determine how many clusters should be specified the eigenvalues were plotted as shown in Figure 8. This plot was obtained with

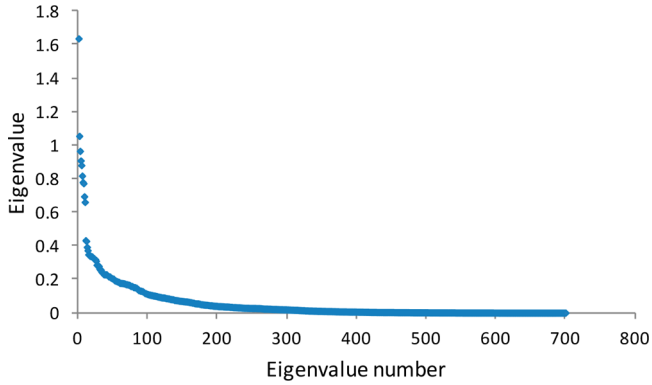


Figure 8. Eigenvalues of the Target X dataset.

γ = 15 and the similarity threshold set to 0.001. We observe the same leveling off of the curve as in Figure 5, but this time the leveling corresponds to much smaller eigenvalues. The precise point at which the difference between successive eigenvalues becomes insignificant is hard to assess but is in the region of the 250th eigenvalue. Similar plots were obtained for other values of γ and the similarity threshold, although the leveling off point varied between the 200th and 400th eigenvalue. If 200 clusters are selected then 431 molecules are placed into 178 clusters, of which 61 are singletons.

In Table 13 we give the results of specifying 200 non-overlapping clusters using γ = 15 and several different values of both α and β in the Tversky index. The clusters calculated for sets of Tversky parameters were analyzed using the QCI measure. There are some obvious conclusions. First, when using asymmetric values for Tversky, all the molecules are clustered, whereas using the symmetric Tanimoto and Dice coefficients, significant numbers of molecules were not placed in clusters. Second, the lower values of the QCI scores for the asymmetric clustering probably reflect the fact that more molecules are clustered, since omitting molecules which are hard to place certainly improves a clustering method's ability to cluster correctly. Third, when using asymmetric Tversky similarities, there are fewer singletons than obtained using the symmetric measures. This raises the possibility of the asymmetric measures producing a "better" clustering. However, inspection of the clusters produced did not always bear out this conclusion. An example is given in Figure 9a for α = 0.9 and β = 0.1. This shows the U clusters associated with eigenvalues 7−9. U-cluster 8 is a mixture of mostly phthalazine and indazole derivatives. The V clusters associated with the same eigenvalues are also

dx.doi.org/10.1021/ci500480b | J. Chem. Inf. Model. 2014, 54, 3302−3319

Table 13. Clustering of Target X Fragment Set into 200 Clusters Using the Tversky Index[a]

	α	β	QCI	n clus	n sing	largest	in clusters
	1	1	53.4	178	61	13	431
U	0.9	0.1	44.3	188	16	14	741
V	0.9	0.1	41.7	192	22	12	740
U	0.8	0.2	44.2	192	21	11	741
V	0.8	0.2	43.5	188	18	11	740
U	0.7	0.3	43.4	192	19	11	741
V	0.7	0.3	42.5	188	18	11	740
U	0.6	0.4	41.8	184	12	10	741
V	0.6	0.4	43.5	190	22	13	740
	0.5	0.5	45.7	175	46	12	523

[a] γ = 15 and α and β are the values used in the Tversky measure for the SVD-NOSC clustering. "n clus" is the number of clusters; "n sing" is the number of singletons; "largest" is the size of the largest cluster; and "in clusters" is the number of compounds clustered. Where $\alpha = \beta$, the **U** and **V** clusters are the same.

shown. V clusters 7 and 8 look coherent, while the fragments in V cluster 9 have a mixture of 3 main chemotypes; phthalazinediones, indazolones and benzimidazolones. However, it does seem that the molecules in each cluster in both sets contain a large similar substructure.

Overlapping Clustering. Our discussion so far has concentrated on the nonoverlapping NOSC. The main reason for this is that it is more straightforward to evaluate the clusters produced in a quantitative manner. However, one of the main advantages of spectral clustering is that it naturally provides a method for assigning molecules to more than one cluster. We refer to our overlapping spectral clustering using singular value decomposition as SVD-OSC. The Target X fragment set was clustered into overlapping clusters using SVD-OSC with values of γ = 15, affinity threshold = 0.001, and cluster threshold = 0.01 as before and using the Tanimoto coefficient. The first thing to report is that, even when overlapping clusters are allowed, not all molecules can be placed into sensible clusters. Specifying more clusters results in more molecules being "clustered", but clusters which are essentially combinations of existing clusters are formed with the addition of only a few more molecules. Thus, specifying 100 clusters resulted in 332 molecules being clustered whereas specifying 200 clusters resulted in 431 molecules being clustered, i.e. no more additional molecules than additional clusters. The best sets of clusters, determined by visual inspection, resulted in only half or fewer of the molecules being placed into clusters. In order to cluster most (90%) of the molecules 500 clusters were required. This resulted in a set of very large clusters, with the mean number of molecules per cluster being 53 and a molecule belonging, on average, to 40 clusters. However, if all molecules are clustered, the chemist can browse the clusters intelligently since the eigenvalues and eigenvector contributions give a

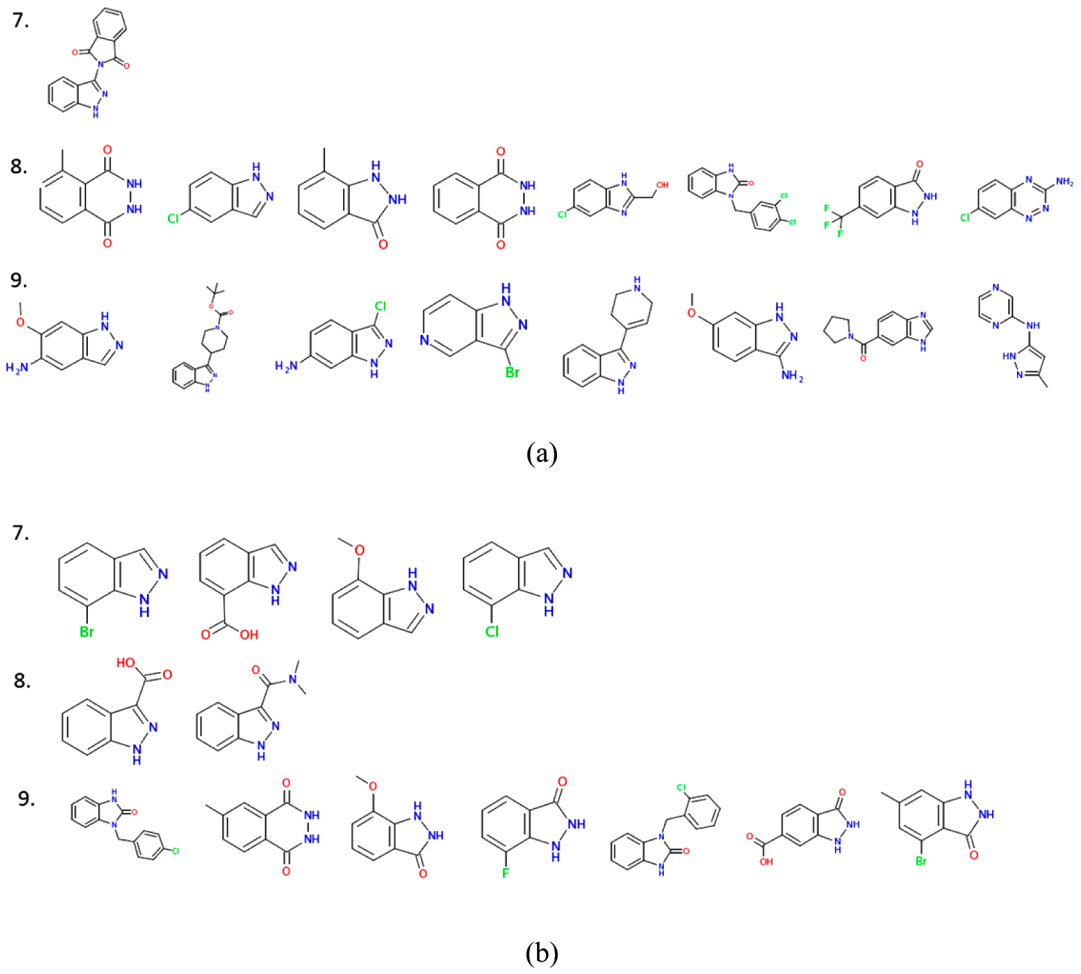


Figure 9. Clusters associated with eigenvalues 7–9. (a) **U** clusters. (b) **V** clusters.

dx.doi.org/10.1021/ci500480b | J. Chem. Inf. Model. 2014, 54, 3302–3319

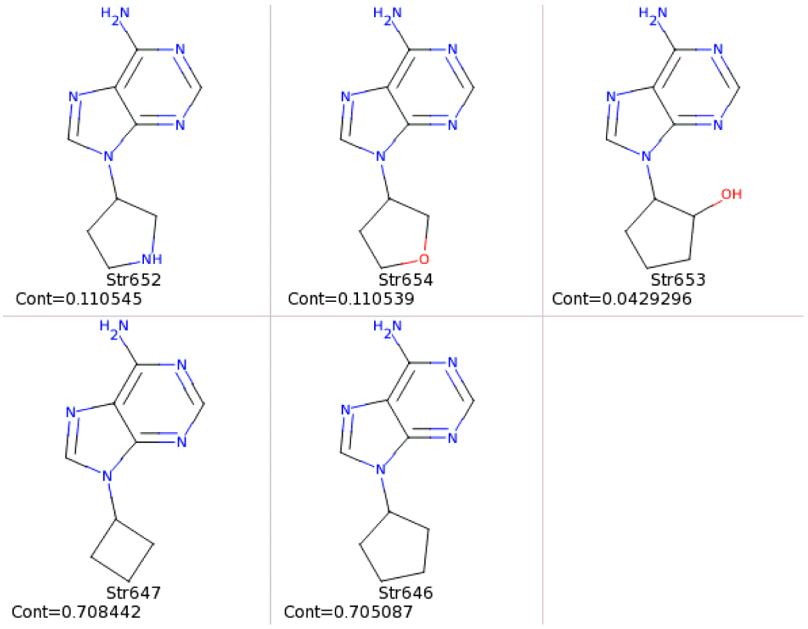


Figure 10. Comparison of overlapping and nonoverlapping clustering. Each row represents a cluster produced by the NOSC method, whereas the overlapping method placed all molecules in the same cluster. Cont is the eigenvector contribution to the cluster.

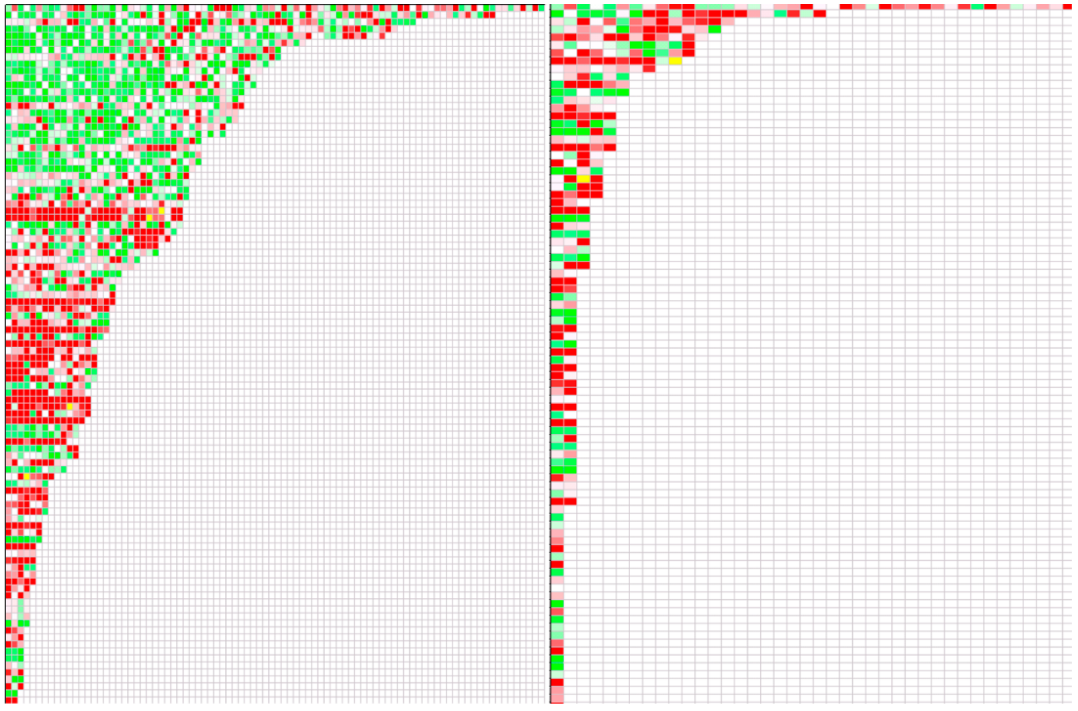


Figure 11. Heat map representation of clusters. The map on the left shows overlapping clusters, on the right are nonoverlapping clusters. Green cells represent active fragments, and red, inactive.

natural order both for cluster exploration and for molecule consideration within a cluster. For the Target X fragment set interpretable results were obtained when 100 clusters were selected. This resulted in only 43% of molecules being clustered with an average of 22 molecules per cluster and each molecule contributing to an average of 7 clusters.

Comparing overlapping and nonoverlapping clustering, recall that our implementation of NOSC involves placing molecules in the cluster to which they make the largest eigenvector contribution. However, the mode of calculation involves a normalization of this quantity, meaning that molecules make apparently larger contributions to smaller clusters. An example of this can be seen in Figure 10. These five molecules from the Target X set are placed into two clusters as shown, when NOSC is used, but when overlapping spectral clustering (OSC) is used (with the same values for all parameters), they are all placed into a single cluster (with no other molecules). This is an advantage of the OSC method. Of course the disadvantage is that the molecules are also placed into other clusters, with always some subset or superset of these compounds being found.

dx.doi.org/10.1021/ci500480b | J. Chem. Inf. Model. 2014, 54, 3302−3319

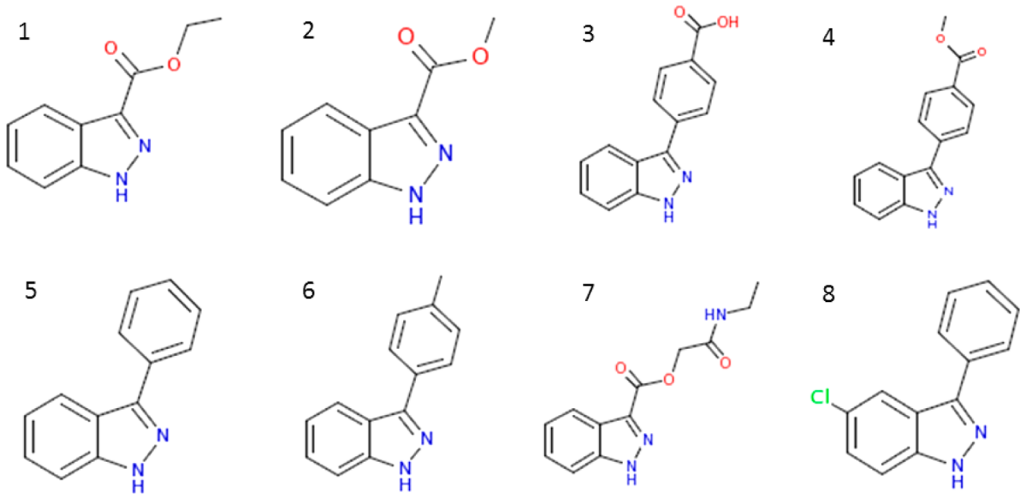


Figure 12. Cluster showing potential for scaffold hopping. Molecules 1−6 and 8 are active; molecule 7 is inactive.

In fact, overlapping clustering produced better clustering, as measured by QCI, than nonoverlapping. For example, when 100 clusters were selected, 332 molecules were placed into clusters using both methods, giving QCI scores of 56 for OSC and 47 for NOSC. The clusters were represented as heat maps and are shown in Figure 11. Rows represent clusters, with cells representing individual molecules. Green cells are active and red inactive. The cells are ordered left to right by decreasing normalized eigenvector contribution to the cluster, so from a clustering perspective, it is preferable to see green cells at the left of a mixed row.

As was noted by Brewer, another advantage of overlapping clusters is the potential for scaffold-hopping. As can be seen from the heat map, there are many clusters which are predominantly active or inactive. Moreover, looking along the rows, the actives and inactives are also grouped together by contribution within a cluster, so that even a cluster composed of both actives and inactives has, in many cases, the actives grouped together within the cluster. The large size of many of the clusters means that there is a potential for containing different scaffolds within an active (or inactive) cluster. Part of one such cluster is shown in Figure 12 which shows the first eight fragments (of 32 in the cluster), all but one of which are active. The only inactive is fragment 7.

CONCLUSIONS

We have presented a nonoverlapping version, NOSC, of the spectral clustering method proposed by Brewer[16] and performed a systematic investigation into the appropriate parameter values required for the optimum performance of the method. We conclude that, of the molecular representations under consideration, ECFP gave superior performance. We found optimal sets of parameters for the compound classes we considered but concluded that some parameter values were really dataset-dependent and therefore recommend that parametrization experiments are carried out before using spectral clustering. We compared the performance of NOSC with that of both Ward's and *k*-means clustering using the Quality Clustering Index and demonstrated its superiority over these methods on our datasets. Given the computational cost of the full matrix diagonalization used in NOSC, we then investigated the replacement of this step with an approximate diagonalization using the Lanczos algorithm, as recommended by Brewer and others. We showed that, although this increased the size of the dataset which could be clustered by a factor of 2, it was not suitable for use on large datasets of more than 10 000 compounds when more than a few clusters were required. We therefore moved to the use of a SVD-based spectral clustering approach and demonstrated that this was able to partition datasets of up to 100 000 molecules in moderate time.

We compared the performance of spectral clustering with Ward's and *k*-means since these are the methods most commonly used in the chemoinformatics community. However, Ward's and *k*-means both return clusterings that are typically hyper-spherical or elliptical in nature with only modest chaining at best. Spectral clustering, however, has been shown to be very good at finding convoluted and chained clusters[10] which may account for the good performance of the NOSC algorithm in our tests. In the light of this performance it would be of interest to compare the use of SVD clustering for molecular data both with other spectral clustering algorithms such as that of Ng et al.[10] and with other algorithms known to perform well on nonconvex data, such as DBSCAN.[53] The SVD-OSC scaled reasonably well with the number of molecules, However, there is always scope for improvement. There are now GPU implementations of SVD[54] which would enable much larger datasets to be clustered.

One of the main advantages of spectral clustering over more conventional crisp clustering methods is that is provides a way of assigning molecules to more than one cluster. Therefore, we also implemented overlapping spectral clustering using our SVD method, SVD-OSC, and showed some of its advantages and disadvantages vis-à-vis the nonoverlapping version. We have also investigated the ability of spectral clustering to cluster based on an asymmetric similarity index. There was some evidence that for small molecules ("molecular fragments") this produced a helpful way of visualizing the contents of a dataset.

As a general rule, it is difficult to form a quantitative assessment of the superiority of one clustering method over another. We certainly cannot claim that spectral clustering, in either its overlapping or nonoverlapping form, is better than other, less expensive, clustering methods. However, we believe that it does produce a different insight into the contents of a set of compounds that might be useful in some cases.

dx.doi.org/10.1021/ci500480b | J. Chem. Inf. Model. 2014, 54, 3302−3319

■ ASSOCIATED CONTENT

Supporting Information

DC100 fragment set. This material is available free of charge via the Internet at http://pubs.acs.org. Both the SVD spectral clustering algorithm and the k-means algorithm have been included in a free clustering software program, svdclus. The software can also use activity data and evaluate the clusters produced using both the QCI measure and the Silhouette score.[55] The code can be downloaded from https://github.com/DavidACosgrove/SVDClus.git.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail v.gillet@sheffield.ac.uk.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We thank the referees for their very helpful comments. We are very grateful to Professor Barry Pickup and the late Andrew Grant (of AstraZeneca) for their helpful advice. We thank the EPSRC and AstraZeneca for funding. We also thank Accelrys Software Inc., Daylight Chemical Information Systems Inc., Digital Chemistry Ltd., and Tripos Inc. for the provision of software and data. Loredana Spadola kindly provided the fragment datasets and performed the manual clustering of DC100.

■ REFERENCES

(1) Duffy, B. C.; Zhu, L.; Decornez, H.; Kitchen, D. B. Early phase drug discovery: Cheminformatics and computational techniques in identifying lead series. *Bioorg. Med. Chem.* **2012**, *20*, 5324−5342.

(2) Bayada, D. M.; Hamersma, H.; van Geerestein, V. J. Molecular diversity and representativity in chemical databases. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 1−10.

(3) Schuffenhauer, A.; Brown, N. Chemical diversity and biological activity. *Drug Discovery Today: Technologies* **2006**, *3*, 387−395.

(4) Downs, G. M.; Barnard, J. M. Clustering methods and their uses in computational chemistry. In *Reviews in Computational Chemistry*, Lipkowitz, K. B., Boyd, D. B., Eds.; 2002; Vol. *18*, pp 1−40.

(5) Ward, J. H. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.* **1963**, *58*, 236.

(6) Hartigan, J. A. *Clustering algorithms*; Wiley: New York, 1975.

(7) Tarjan, R. E. An improved algorithm for hierarchical-clustering using strong components. *Inform. Process. Lett.* **1983**, *17*, 37−41.

(8) MacCuish, N. E.; MacCuish, J. D. Clustering compound data: Asymmetric clustering of chemical datasets. In *Chemometrics and Chemoinformatics*, Lavine, B. K., Ed.; 2005; Vol. *894*, pp 157−171.

(9) Nicolaou, C. A.; MacCuish, J. D.; Tamura, S. Y. A new multi-domain clustering algorithm for lead discovery that exploits ties in proximities. In *Rational approaches to drug design, Proceedings of the 13th European Symposium on Quantitative Structure−Activity Relationships*, Dusseldorf, Germany, Aug 27−Sep 1, 2000; Holtje, H. D., Sippl, W., Eds; 2001; p 486−495.

(10) Ng, A. Y.; Jordan, M. I.; Weiss, Y. On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems*; Dietterich, T. G., Becker, S., Ghahramani, Z., Eds.; MIT Press: 2002; Vol. *14*, pp 849−856.

(11) Murzin, A. G.; Brenner, S. E.; Hubbard, T.; Chothia, C. SCOP - A structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* **1995**, *247*, 536−540.

(12) Paccanaro, A.; Casbon, J. A.; Saqi, M. A. S. Spectral clustering of protein sequences. *Nucleic Acids Res.* **2006**, *34*, 1571−1580.

(13) Nepusz, T.; Sasidharan, R.; Paccanaro, A. SCPS: a fast implementation of a spectral method for detecting protein families on a genome-wide scale. *BMC Bioinformatics* **2010**, *11*, 120.

(14) Sgourakis, N. G.; Merced-Serrano, M.; Boutsidis, C.; Drineas, P.; Du, Z.; Wang, C.; Garcia, A. E. Atomic-level characterization of the ensemble of the Aβ(1−42) monomer in water using unbiased molecular dynamics simulations and spectral algorithms. *J. Mol. Biol.* **2011**, *405*, 570−583.

(15) Zhiwen, Y.; Le, L.; You, J.; Hau-San, W.; Guoqiang, H. SC(3): Triple spectral clustering-based consensus clustering framework for class discovery from cancer gene expression profiles. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2012**, *9*, 1751−1765.

(16) Brewer, M. L. Development of a spectral clustering method for the analysis of molecular data sets. *J. Chem. Inf. Model.* **2007**, *47*, 1727−1733.

(17) Sarkar, S.; Boyer, K. L. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. *Comput. Vis.Image Und.* **1998**, *71*, 110−136.

(18) Neres, J.; Brewer, M. L.; Ratier, L.; Botti, H.; Buschiazzo, A.; Edwards, P. N.; Mortenson, P. N.; Charlton, M. H.; Alzari, P. M.; Frasch, A. C.; Bryce, R. A.; Douglas, K. T. Discovery of novel inhibitors of Trypanosoma cruzi trans-sialidase from in silico screening. *Bioorg. Med. Chem. Lett.* **2009**, *19*, 589−596.

(19) Heifetz, A.; Barker, O.; Verquin, G.; Wimmer, N.; Meutermans, W.; Pal, S.; Law, R. J.; Whittaker, M. Fighting obesity with a sugar-based library: discovery of novel MCH-1R antagonists by a new computational−VAST approach for exploration of GPCR binding sites. *J. Chem. Inf. Model.* **2013**, *53*, 1084−1099.

(20) Whittaker, M. Picking up the pieces with FBDD or FADD: invest early for future success. *Drug Discovery Today* **2009**, *14*, 623−624.

(21) Lanczos, C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Natl. Bur. Stand.* **1950**, *45*, 255−282.

(22) Shlens, J. *A tutorial on principal component analysis: Derivation, Discussion and Singular Value Decomposition*. http://www.cs.princeton.edu/picasso/mats/PCA-Tutorial-Intuition_jp.pdf (accessed June 19, 2013).

(23) Golub, G.; Van Loan, C. *Matrix computations*; Johns Hopkins University Press: Baltimore, 1996.

(24) Press, W.; Teukolsky, S.; Vetterling, W.; Flannery, B. *Numerical recipes: the art of scientific computing*; Cambridge University Press: Cambridge, MA, 2007.

(25) Parlett, B. N. *The symmetric eigenvalue problem*; Society for Industrial and Applied Mathematics: Philadelphia, PA, 1998.

(26) Paige, C. C. The computation of eigenvalues and eigenvectors of very large sparse matrices. Ph.D. thesis, London, 1971.

(27) Strang, G. *Introduction to Linear Algebra*, 3rd ed.; Wellesley-Cambridge Press: Wellesley MA, 2003.

(28) Kontoghiorghes, E. J. *Handbook of Parallel Computing and Statistics. Statistics: Textbooks and Monograph Series*; Marcel Dekker, Inc.: 2005; Vol. *184*.

(29) *CERN COLT Matrix Package*. http://acs.lbl.gov/software/colt (accessed Sep 30, 2014).

(30) Berry, M.; Do, T.; O'Brien, G.; Krishna, V.; Varadhan, S. *SVDPACKC user's guide*; University of Tennessee, 1993; CS-93-194.

(31) Landrum, G. *RDKit: Open-source cheminformatics*; 2006 http://www.rdk.org (accessed Sep 30, 2014).

(32) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **2012**, *40*, D1100−D1107.

(33) *Pipeline Pilot*, version 7.5; Accelrys: San Diego, CA, 2010.

(34) Durant, J. L.; Leland, B. A.; Henry, D. R.; Nourse, J. G. Reoptimization of MDL keys for use in drug discovery. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1273−1280.

(35) Yu, P.; Wild, D. J. Fast rule-based bioactivity prediction using associative classification mining. *J. Cheminformatics* **2012**, *4*, No. 29.

(36) 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.
(37) Gardiner, E. J.; Holliday, J. D.; O'Dowd, C.; Willett, P. Effectiveness of 2D fingerprints for scaffold hopping. *Future Med. Chem.* **2011**, *3*, 405−414.
(38) Varin, T.; Bureau, R.; Mueller, C.; Willett, P. Clustering files of chemical structures using the Szekely-Rizzo generalization of Ward's method. *J. Mol. Graphics Modell.* **2009**, *28*, 187−195.
(39) *BCI Software*; Digital Chemistry: Sheffield, UK, 2010.
(40) *Daylight Software*; Daylight Chemical Information Systems: Aliso Viejo, CA, 2010.
(41) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.
(42) *MDL software*; Symyx Technologies Inc.
(43) *Unity 4.4*; Tripos L.P.: St. Louis, MO, 2003.
(44) Varin, T.; Saettel, N.; Villain, J.; Lesnard, A.; Dauphin, F.; Bureau, R.; Rault, S. 3D Pharmacophore, hierarchical methods, and 5-HT(4) receptor binding data. *J. Enzyme Inhib. Med. Chem.* **2008**, *23*, 593−603.
(45) Kelley, L.; Gardner, S.; Sutcliffe, M. An automated approach for clustering an ensemble of NMR-derived protein structures into conformationally related subfamilies. *Protein Eng. Des. Sel.* **1996**, *9*, 1063−1065.
(46) Paige, C. C. Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem. *Linear Algebra Appl.* **1980**, *34*, 235−258.
(47) Shi, J.; Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888−905.
(48) *MDL Drug Data Report*; Symyx Technologies Inc.: 2006.
(49) Leach, A.; Gillet, V. *An introduction to chemoinformatics*; Springer Verlag, 2007.
(50) Holliday, J. D.; Salim, N.; Whittle, M.; Willett, P. Analysis and display of the size dependence of chemical similarity coefficients. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 819−828.
(51) Horvath, D.; Marcou, G.; Varnek, A. Do not hesitate to use Tversky-and other hints for successful active analogue searches with feature count descriptors. *J. Chem. Inf. Model.* **2013**, *53*, 1543−1562.
(52) Congreve, M.; Chessari, G.; Tisi, D.; Woodhead, A. J. Recent developments in fragment-based drug discovery. *J. Med. Chem.* **2008**, *51*, 3661−3680.
(53) Ester, M.; Kriegel, H.-P.; Sander, J. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD-96*, Portland, Oregon, USA; Simoudis, E., Han, J., Fayyad, U. M., Eds.; AAAI Press: Portland, Oregon, USA, 1996; pp 226−231.
(54) Lahabar, S.; Narayanan, P. J. Ieee Singular Value Decomposition on GPU using CUDA. In *Proceedings IEEE International Symposium on Parallel & Distributed Processing*, Rome, Italy, May 23−29; IEEE: New York, 2009; Vols. *1*−*5*, pp 840−849.
(55) Rousseeuw, P. J. Silhouettes - a graphical aid to the interpretation and validation of cluster-analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53−65.