Bioorganic & Medicinal Chemistry 20 (2012) 5396–5409

Contents lists available at SciVerse ScienceDirect

Bioorganic & Medicinal Chemistry

journal homepage: www.elsevier.com/locate/bmc

Using self-organizing maps to accelerate similarity search

Fanny Bonachera [a,b], Gilles Marcou [a], Natalia Kireeva [a], Alexandre Varnek [a], Dragos Horvath [a,*]

[a] Laboratoire d'Infochimie UMR 7177, Université de Strasbourg, 1, rue B. Pascal, 67000 Strasbourg, France
[b] Unité de glycobiologie structurale et fonctionnelle UMR 8576, Université Lille I, Bâtiment C9, 59655 Villeneuve d'Ascq Cedex, France

ARTICLE INFO

Article history:
Available online 21 April 2012

Keywords:
Chemical space navigation
Virtual screening
Similarity searching
Fuzzy pharmacophores
Kohonen maps

ABSTRACT

While self-organizing maps (SOM) have often been used to map and describe chemical space, this paper focuses on their use to accelerate similarity searches based on vectors of high-dimensional real-value descriptors for which classical, binary fingerprint-based similarity speed-up procedures do not apply. Fuzzy tricentric pharmacophore (FPT) and ISIDA substructure counts are herein explored examples. Similarity search speed-up was achieved by positioning compounds on a SOM, then searching for analogues only in the neurons neighbouring the ones in which the query compounds reside. Smaller neighbourhood means shorter virtual screening (VS) time, but lower analogue retrieval rates. An enhancement criterion, conciliating the opposite trends is defined. It depends on map definition and build-up protocol (training set choice, map size, convergence criteria,...). The main goal is to discover and validate SOMs of optimal quality with respect to this criterion. Increasing the size of the training set beyond a certain limit is shown to be unnecessary and even detrimental, suggesting that one SOM built on a relatively small but diverse training set may be an effective VS enhancer of a much larger database. Also, using an excessively large number of training iterations may lead to over-fitting. Gradual training with en-route checking of VS enhancement propensity is the best strategy to follow. Maps were successfully challenged to accelerate the large-scale VS of 12,000 queries against 160,000 compounds, and shown to provide a meaningful mapping of activity-annotated compounds in chemical space.

1. Introduction

Similarity-based virtual screening is an integral part of modern in silico drug discovery.[1]

This approach is based on the paradigm that structurally close compounds may also have similar activity against a given target.[2] A candidate molecule from some large structural database will be considered as potentially active if a similarity search shows that it is related (in terms of its molecular descriptors, and using an appropriate similarity measure[3]) with one or more known actives. The results of these searches are ranked lists of all screened compounds, along with similarity scores. The highest ranked compounds of these lists are assumed to be the closest to the query in terms of activity.[4]

However, with the ever-growing size of databases, similarity searches become more and more time-consuming. While this is not a practical concern when binary fingerprints are used for similarity searches—for which very fast search methods have been designed—the situation is different if one wishes to employ information-rich, high-dimensional real number vectors, such as fuzzy pharmacophore descriptors. While chemically relevant, and intrinsically very powerful in similarity searching (particularly well suited for 'lead hopping',[5]) they require floating-point operations instead of fast bit-wise matching and, furthermore, are not eligible for search acceleration procedures developed for binary fingerprints.[6–8]

Similarity search speed-up techniques rely on some kind of 'divide and conquer' approach aimed to split the original search space into sub-domains, and then decide beforehand, in function of the given query, which sub-domain(s) may be ignored during the search. Real-value descriptors do not naturally provide a 'granular' search space, by contrast to binary fingerprints. However, self-organizing maps (SOMs) may be used with such descriptors in order to obtain the needed 'tessellation' of the chemical space. Furthermore, SOM-driven tessellation is non-linear with respect to the chemical space (SOMs being a particular type of neural networks),[9,10] and easy to visualize (output being a 2D grid of 'nodes' or 'neurons'). Each node is represented by a weight vector (the code vector) of the same dimension as the molecular descriptors. The fact that the self-organizing maps are able to preserve the topological properties of the input space have made them popular in Chemoinformatics.[11] The two-dimensional generated maps are:

Abbreviations: SOM, Self-organizing maps; VS, Virtual screening; VLH, Virtual hit list; B, Brute; R, Refinement; HR, HyperRefinement; FPT, Fuzzy pharmacophore triplets; DB, Data base; QS, Query set.

* Corresponding author.
E-mail addresses: dhorvath@unistra.fr, d.horvath@wanadoo.fr (D. Horvath).

0968-0896/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.bmc.2012.04.024

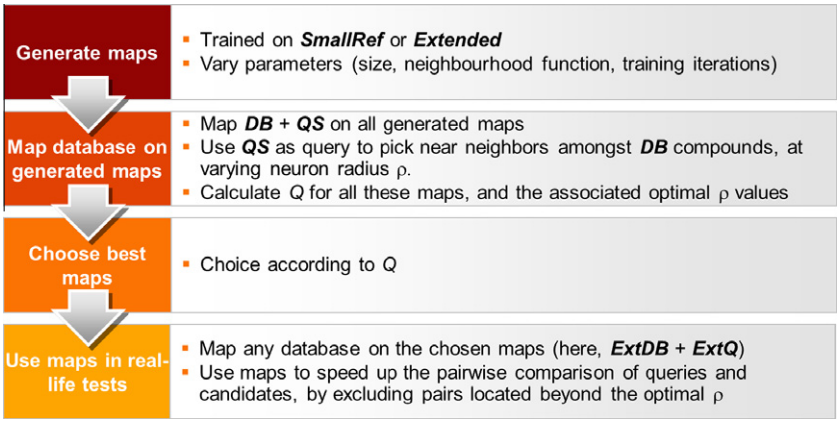

Fig. 1. Workflow of building, assessment, selection and testing of SOMs with good similarity-based virtual screening enhancement propensities.

- Easy to understand two-dimensional representations of the high-dimensional input.
- Clustered representations of the input: objects that are mapped in the same node or in adjacent nodes can be considered as similar.

The applications of Kohonen self-organizing maps in Chemoinformatics research are varied. It has been demonstrated that Kohonen maps can effectively be used to produce topology-preserving maps of small molecules, providing ways to compare compounds and assess similarity.[12,13] They have also proven to be very effective in classifying and clustering compounds as well as describing the chemical space of a database,[14] detecting novelties,[15,16] studying structure–activity relationships,[17] mapping pharmacophores,[18] selecting virtual screening candidates,[19,20] predicting or mapping properties,[21] or comparing chemical libraries.[22]

The use of Kohonen self-organizing maps to accelerate search has already been published in the image retrieval domain.[23,24] The association of classification of images on a SOM with a k-nearest neighbors similarity scoring function proved to be a quick and effective way of retrieving images.

Here, SOMs will be used to enhance similarity-based virtual screening (VS). While typically tree-based acceleration methods[6,7] are used to speed up search in large databases, these have several drawbacks compared to the use of self-organizing maps:

- SOMs as virtual screening enhancers are compatible with various similarity metrics, whereas trees are constructed with respect to one particular metric. Although SOMs operate on the basis of Euclidean metric, they are nevertheless able to successfully accelerate Tanimoto similarity-based searches, as will be shown in this work. Actually, any similarity metric can be used 'on the fly', in as far as its neighbourhood behaviour[25] is correct.
- Furthermore, there is no need to rebuild the SOMs if the database is extended. This is systematically required with search trees. New compounds need just be mapped into corresponding nodes of the existing map. Building a tree requires the entire database, whereas building a relevant and robust SOM only needs a representative fraction of the database. Actually, as will be shown in this work, oversized SOM training sets may have detrimental influence on SOM performance.
- A tree model can sometimes miss hits (because some branches are ignored during a search), whereas in the herein presented approach the hit retrieval rate is tunable, and may be balanced off against the time gain by specifying the size of the searched map neighbourhood around the residence node of the query compound.

The VS benchmark consists in retrieving, out of a compound database (DB, ~55,000 molecules), the virtual hits (nearest neighbours below a specified dissimilarity threshold) for each of the members of a 'query set' (QS) of 2000 compounds. Similarity (both Euclidean and Tanimoto) is calculated in terms of vectors of fuzzy tricentric pharmacophore (FPT) descriptors, and ISIDA sequence fragment counts. The CPU times required to match each QS member against every DB compound are determined and used as reference values. To speed up this virtual screening protocol, a set of SOMs have been trained, in the considered (pharmacophore or fragment count-based) chemical spaces, on two training datasets of different sizes (of about 11,000 and 53,000 molecules, respectively). After the training phase, both QS and DB were then mapped thereon, and QS members were only matched against DB compounds residing on the same or on neighbouring neurons of the SOM (at increasing size of the squared neighbouring neuron area around the query neuron). The smaller this area, the less DB compounds must be matched against each query: virtual screening time is therefore decreasing. However, since DB compounds outside this area are by default considered 'dissimilar', the retrieval rate of the initially established sets of virtual hits will also decrease. A best compromise criterion, conciliating thee opposite trends, is defined in order to characterize the SOM speed-up versus retrieval rate efficiency, and thus compare the relative performances of the various SOMs. The workflow is shown in Figure 1.

The final goal of this work is to understand how to train maps with optimal virtual screening enhancement proficiency, by studying the impact of map build-up choices (training set size, map size and geometry, imposed convergence criteria, choice of neighbourhood functions) on their proficiency. The following text is structured as follows: in Methods, after presentation of the employed data sets and molecular descriptors, the SOM technology is briefly introduced. The systematic scan, in pharmacophore descriptors space, of combinations of considered SOM parameters is described. The definition of the VS enhancement criterion recommended as the objective score of SOM quality is given. Two of the map topologies covered by this systematic scan are also investigated in conjunction to ISIDA fragment counts, in order to assess the impact of the nature of descriptors on the VS enhancement proficiency. Eventually, a real-life VS experiment in pharmacophore descriptors space, validating the usefulness of SOMs as VS accelerators is presented. The Results section devotes a lengthy discussion to the problem of proper map training: reaching convergence while avoiding over-fitting. The impact of training set choice on map quality is the next important chapter, followed by a presentation of the to-date best ranked map in terms of the herein proposed

VS enhancement criterion. Next, an overview of the issues encountered when using SOM-driven VS acceleration in fragment count spaces is provided, in order to (a) illustrate the overall robustness of the method, irrespectively of the employed descriptor spaces and (b) highlight specific issues pertaining to the use of these topology-specific descriptors, conceptually different from pharmacophore pattern-rendering FPT. Eventually, the behaviour of top-ranking maps in the real-life FPT-based VS experiment is illustrated. The Conclusion section points to the key observations made in these investigations, and ends with a pragmatic discussion on the usefulness of the approach.

2. Methods

2.1. Data sets

There are six, partly overlapping data sets to which this work refers. These are as diverse as possible compound collections, randomly extracted from a large panel of sources, in order to minimize set-related artefacts. Care has been taken to include both chemically 'dense' series of analogues, 'sparse' sets of drugs and reference compounds (in which only the diverse marketed structures stand alone, not surrounded by analogues—unless several related structures have been marketed as drugs) and finally, commercial organic compound databases of more or less drug-like compounds, part of chemical series or singletons. Also, since SOMs are unsupervised learning methods, no particular attention was paid to ensure that SOM training sets are distinct from the VS molecules.

- The DataBase DB represents a pool of 55,613 molecules including random subsets of 11 different analogue series used to model structure–property relationships in literature,[26] marketed drugs and biological reference compounds, 1870 ligands from the Pubchem database tested on the hERG channel,[27] and a majority of randomly picked ZINC compounds.
- The query set **QS**, is composed of 2000 molecules, regroups the remainders of the 11 above-cited series, further marketed drugs and biological reference compounds and commercially available molecules (picked randomly from the ZINC database,[28]). There is no overlap between DB and QS. These molecules serve as starting points (queries) for similarity-based virtual screening against DB, in order to assess the ability of our SOM-driven screening tool to find, for all of the 2000 queries, a maximum of their nearest neighbours amongst DB compounds, with a minimum of effort. Thus, QS and DB are the data sets used to assess SOM quality, not to train the SOMs. The SOM training sets, which only partially overlap with QS, are the following:
- The large SOM training set Extended of 53,206 molecules is basically a subset of previously available molecules (DB + QS), excluding the analogue series members, the Pubchem compounds and some 900 ZINC molecules. More precisely, out of the 2000 QS molecules, 149 are members of both SmallRef and Extended, while the others are never used to train maps.
- The small SOM training set SmallRef of 11,168 molecules features all drugs and biological reference compounds seen in Extended, but significantly less ZINC molecules.
- Eventually, the external database ExtDB of roughly 160,000 molecules from the corporate collection of one of our industrial partners: this was used, in the final stage, to verify the performance of the map-enhanced virtual screening tool, deployed on multiple processors, under real-life conditions. This set has been screened against 12,491 query compounds—taken basically from SmallRef, and completed with randomly picked commercial compounds: let us refer to this query set as ExtQ.
- A subset of ligands from the database of useful decoys (DUD),[29] regrouping the binders to the ten most ligand-rich targets displayed in Table 2, and used to assess the proficiency of maps in telling various ligand classes apart.

2.2. Descriptors and metric: the chemical space of the similarity-based virtual screening approach

Fuzzy pharmacophore triplets (FPT)[30,26] represent fuzzy counts of monitored triplets of potential pharmacophore points (PPP), at given topological inter-feature distances, that is, 'edge lengths' of the considered triangles. As discussed in the original publication ,[30] six different PPP types (hydrophobic, aromatic, hydrogen bond donor and acceptor, positive and negative charge) were assigned to the atoms within every microspecies present (at significant population levels) in the proteolytic equilibrium of the molecule at a given pH of 7.4. Molecular fingerprints are averages of microspecies fingerprints, accounting for their relative population levels. In this work, only FPT1 descriptors, corresponding to the default set-up of fuzzy triplets in the original work[30] were considered. In the *SmallRef* set, 4418 different pharmacophore triplets were found to be populated (out of the 4494 defined for the FPT1 set-up). Therefore, this subset of 4418 triplets has been systematically used to construct the 4418-dimensional descriptor vectors positioning the molecules in the Chemical Space (CS). The averages $\langle D_i \rangle_{SmallRef}$ and standard deviations $\sigma(D_i)_{SmallRef}$ of the 4418 vector elements were taken over the *SmallRef* compounds and employed to normalize the descriptors prior to both virtual screening and SOM build-up/mapping. Therefore, the finally employed descriptor vectors are $d_i = [D_i - \langle D_i \rangle_{SmallRef}]/\sigma(D_i)_{SmallRef}$, which, for *SmallRef* molecules, corresponds to classical 'Z-transformed' [31] vectors. Please note that, with other sets, *SmallRef* averages and deviations are employed rather than the actual values over the respective sets.

For each query compound of QS, both its Euclidean and its Tanimoto distance to every DB molecule (the latter being defined as 1-Tanimoto index) was computed, on hand of the d_i values. Generically speaking, let the inter-molecular dissimilarity score (whether Euclidean or Tanimoto) between two molecules m and M be denoted $\Sigma(M,m)$. An empirical distance threshold of 0.25 was picked to delimit 'virtual hits' in terms of Tanimoto distances (meaning a maximum of 25% of tolerated dissimilarity). In Euclidean space, the equivalent cutoff value (roughly corresponding to the same number of 'virtual hits') was found to be of 9.0. Therefore, for each query compound, two virtual hit lists (*VHL*)—based on Tanimoto VLH_T and Euclidean VLH_E distances, respectively—were established. They include the first 300 (or fewer) nearest neighbours from DB, found to be within the respective cutoff radii with respect to the query. Virtual screening has been carried out by a FORTRAN executable on a x86_64 Intel CPU, and the CPU times t_E^{ref} and t_T^{ref} for the complete virtual screening according to Euclidean, respectively Tanimoto distance matrices were measured using the unix *time* command. These times include, next to the actual time span of the computation of the $QS \times DB$ distance matrices, the overheads for data input and virtual hit list output.

Alternatively, ISIDA[32–35] atoms and bond sequences of length between 3 and 7 atoms, and atom-centred fragments based on sequences of atoms, of radii between 2 and 4 were also investigated here, in the role of alternative chemical spaces. Unlike fuzzy pharmacophore triplets, these descriptors are molecular topology-focused, showing how many times each of the observed substructures of atoms and bonds are being encountered in every molecule. The total number of considered fragments is open-ended. However, rare fragments occurring in less than 5 molecules of Extended were discarded, leaving a total of 2836 atom-centred fragments (out of ≈ 13,000), and 5792 sequences (of ≈19,000) in

the eventually considered integer-value descriptor vectors. By contrast to the FPT-based chemical space, ISIDA fragment counts were employed as such, without normalization, in Tanimoto dissimilarity scoring. The above-mentioned distance threshold of 0.25 was used to define fragment-similarity-based hits. Note that, in the present work, pairwise dissimilarity was also calculated on the basis of the truncated fingerprints serving to build the maps, excluding rare fragments, and the VHL were built on this basis as well.

2.3. Build-up of the self-organizing maps (SOM)

2.3.1. The SOM_PAK software

SOM_PAK, first published in 1992[36] is a program package written in ANSI C that contains all necessary tools to build and exploit self-organizing maps.

The package provides tools to initialize, train maps, evaluate the quantization error (see below 'evaluating maps quality') and visualize maps. To initialize a map, the user needs to specify the following input parameters:

- The name of the file containing the molecular descriptors of the training compounds.
- The map topology: with SOM_PAK, rectangular or hexagonal arrangements of the neurons in a 2D lattice are supported. Only the 'rectangular' option has been used here. By contrast to the multidimensional chemical space (4418-dimensional, according to the number of monitored fuzzy triplets), also referred to as 'input space' in the following, the rectangular 2D lattice of neurons is the 'output space' of the map. In output space, molecules are located onto their 'winning neuron', the one of minimal Euclidean distance to that molecule, in input space. Distances in output space are defined between two neurons, as the Euclidean distance between the associated lattice nodes. Implicitly, in output space the dissimilarity between two molecules is given by the distance between their winning neurons.
- The wanted dimensions of the resulting map, in terms of two integers defining $length \times width$ of the rectangular neuron lattice. For each of the $length \times width$ neurons, a characteristic 'code vector', positioning them in the chemical space of input molecules, need to be fitted. The number of fittable parameters in a SOM therefore equals to $length \times width \times dimension\ of\ the\ chemical\ space$.
- The neighbourhood function, which can be 'Gaussian' or 'bubble'. When updating the code vectors of each neuron, the shift of the code vector of a neuron induced by a given molecule must be obviously a decreasing function of the distance between this neuron and the 'winning neuron' associated to that molecule. This function may be either a Gaussian or a bubble function (that is, return one if the distance is below a given threshold or 'learning' radius, and zero otherwise). Likewise, the width of the Gaussian bell is also tunable (the 'learning radius' being here associated to the half-width of the Gaussian bell). SOM_PAK will decrease the initial, user-input learning radius linearly with the number of iterations, in order to enhance the convergence. To the same purpose, the absolute value of molecule-induced shifts is also being dampened as fitting proceeds, by multiplying them to a learning parameter taking an user-defined value α at the beginning of the fit procedure, and linearly decreasing with respect to the iteration number.

2.3.1.1. Map initialization. The reference vectors (code vectors) were initialized using the RANDINIT program. All reference vectors components are first set to random values evenly distributed in the area of corresponding training data input vectors components.

2.3.1.2. Map training. After initialization, we have used the VSOM program to train the reference vectors. This program uses rough random map parameterized at the initialization step. During this training phase, VSOM finds the best matching node (or neuron) for each training input data vector and uses the neighbourhood function to update the nodes neighbours.

The important parameters are *rlen* (the number of training steps), *alpha* (initial learning rate parameter, which decreases linearly to zero during training) and the *radius* (initial radius of the training area, which decreases linearly to one during training) parameter.

Further training processes, taking as input already trained maps instead of randomly initialized ones, may be used to continue the refinement.

2.3.2. Building the SOMs

Various maps have been built using the two training sets (*SmallRef* and *Extended*). We chose to vary X and Y dimensions from 8×6 to 30×30, which covers a range of 48– 900 neurons. Practically, the 36 explored map geometries were: 6×12, 6×14, 8×10, 8×6, 10×10, 10×14, 10×20, 12×8, 14×6, 18×18, 18×20, 20×20, 22×24, 22×26, 22×28, 22×30, 24×22, 24×24, 24×26, 24×28, 24×30, 26×22, 26×24, 26×26, 26×28, 26×30, 28×22, 28×24, 28×26, 28×28, 28×30, 30×22, 30×24, 30×26, 30×28, 30×30. Each such map has been built, based on every training set, in both 'Gaussian' and 'bubble' version, leading to a total of '$2 \times 2 \times 36 = 144$ processed SOMs. With ISIDA descriptors, only the 22×28 and respectively 10×10 'bubble' maps were considered.

Map fitting is driven by the objective to minimize the quantization error *QE*, the average Euclidean distance between each molecular descriptor vector and the closest code vector (the one of the neuron to which it was assigned). In terms of fitting strategies, the calibration of the SOMs based on *Extended* was expected to be more difficult to achieve, therefore different fitting strategies were used:

Three successive runs were employed for map calibration.

- Brute (B) training at *rlen* of 1000, *alpha* of 0.05 and a *radius* equalling the *lenght* of the neuron lattice.
- Refinement (R) at *rlen* of 10,000, *alpha* of 0.02 and a *radius* of *max* (6,*length*/2).
- HyperRefinement (HR) at variable *rlen* (typically 50,000, which will also be referred as 'Standard HR'—other values being indicated in the text), *alpha* of 0.01 and a *radius* of *max* (3,*length*/4).

A study of the convergence of the maps was conducted in order to make sure that the above-mentioned conditions are sufficient, and that the map quality criteria are not biased due to their failure to converge. To this purpose, based on *Extended*, using Gaussian neighbourhood and for two different sizes (a modest 10×10 and a larger 22×28), series of successive maps were saved at every step of a succession of training runs. Training always included a B and a Refinement phase, followed by HR stages of different lengths. Map quality criteria were then monitored throughout these series, in order to check how many iterations were required before they reached a stable value. In parallel, a similar study was conducted for the bubble 22×28 configuration, based on *SmallRef*, in order to verify the set size impact on the convergence rate.

2.4. Maps visualization with SomView

The software used for map visualization and to make all SOM figures of this paper is an in-house developed software, SomView. It allows us to open .fvs files (lists of coordinates corresponding to the best-matching nodes in the map for each training data sample,

created with the *visual* program), together with the associated .sdf files of mapped molecules, and to easily visualize rectangular self-organizing maps.

The maps are depicted as rectangular grids in which each node is a pie colored by labels or by label majority, depending on the chosen option. If an .sdf file is provided, the contents of each node can be viewed in 2-dimensional structures. Other nodes visualization options are available : node color and size by topographic index (see below—'Evaluation of self-organizing maps quality'), node size by quantization error.

2.5. Map-enhanced similarity searching

2.5.1. Assigning compounds onto map neurons

Before performing similarity search tests, both QS and DB sets are mapped on the current Kohonen map κ, that is, each of the compounds is being assigned to the closest map neuron. This is performed with the *visualize* tool. Formally, each compound is thus being assigned a 2D position on the map grid, coded by two integers: the neuron position (N_x,N_y). Unlike in complete similarity-based screening, a given QS molecule will now be compared to a DB compound only if the host neurons of the two structures are acceptably close to each other. The highest acceptable distance between two neurons for which direct similarity scoring needs to be performed is called the *neuron radius* ρ. Let $(N_x,N_y)|_M$ and $(N_x,N_y)|_m$ be the residence neurons of molecules M and m respectively. These two molecules are subjected to an actual similarity calculation of their descriptor vectors only if $|N_{x,M} - N_{x,m}| \leqslant R$ and $|N_{y,M} - N_{y,m}| \leqslant R$. For a query compound M located on the query neuron in Figure 2, only *DB* molecules located on neurons within the box of size $2R+1$ around the query neuron would qualify for explicit $\Sigma(m,M)$ calculations. Otherwise, the inter-molecular dissimilarity score $\Sigma(M,m)$ will be by default assumed infinitely large. In other words, unless they reside on neurons that are no more than ρ units apart, on either dimension of the map grid, M and m do no longer count as neighbours. This working hypothesis allows a quick, and -if the map is meaningful- effective discarding of (m,M) pairs for which the actual costly calculation of $\Sigma(M,m)$ in the 4000-plus dimensional descriptor space may be spared. Figure 3 illustrates how the inter-molecular dissimilarity matrix calculation is impacted by the choice of ρ. When rebuilding the virtual hit lists *VHL* using map κ, at a given ρ value, the lower ρ—and the lower the quality of κ, the higher the odds that neighbour pairs originally seen in the *VHL* issued from the exhaustive pairwise comparison (see subSection 2.2) will now be missed. Reversely, the lower ρ and the better the quality of κ, the lesser the effort to calculate the $QS \times DB$ dissimilarity matrix.

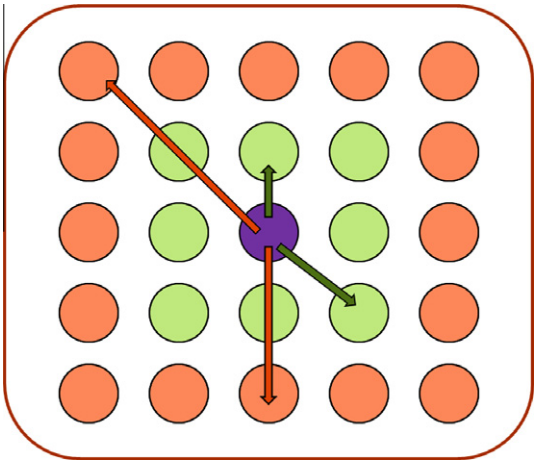

Fig. 2. Radius around the query node (in purple): green neurons correspond to $\rho = 1$, orange neurons correspond to $\rho = 2$.

2.5.2. Virtual screening enhancement factor of a map

Let the *retrieval rate* RR_Σ represent the average, over all query compounds, of the fraction of retrieved nearest neighbours (using κ, at ρ), with respect to the total number of neighbours in the exhaustive VHL_Σ at metric $\Sigma = \{E(uclidean), T(animoto)\}$. Note: for the roughly 25% of singletons having no nearest neighbours at all, *RR* will always count as 1.0: map-driven virtual screening at any ρ values cannot be held responsible for failing to detect near neighbours when there are none. Let f_Σ represent the time it took to perform the $QS \times DB$ dissimilarity matrix calculation, using κ at ρ, compared to the reference time t_Σ^{ref}. The '$RR - f$' plot of $RR_\Sigma^{\kappa@\rho}$ versus $f_\Sigma^{\kappa@\rho}$ at increasing ρ values describes a curve originating at $\rho = 0$ ($\Sigma(M,m)$ calculations restricted only to molecules m residing on the same neuron as the query M—the point at lowest time fraction and at lowest retrieval rate). Increasing ρ will eventually lead to picking all the possible (m,M) pairs for explicit dissimilarity calculation, thus $RR = 1$ and $f = 1$ if the box of size ρ covers the entire map. In between theses extremes, the virtual screening enhancement factor with respect to dissimilarity metric Σ, for the map κ needs to be optimized by scanning for the best time enhancement $1 - f_\Sigma^{\kappa@\rho}$ versus retrieval rate compromise, over increasing radii:

$$Q_\Sigma^\kappa = \max_\rho \left[RR_\Sigma^{\kappa@\rho} \times \left(1 - f_\Sigma^{\kappa@\rho}\right)^2 \right] \quad (1)$$

In Eq. 1, time enhancement is squared, in order to emphasize that time effectiveness is, in this case, a more stringent demand than a perfect retrieval of all the possible near neighbours. Eventually, since a map is expected to be an efficient VS enhancer irrespective of the actually employed dissimilarity metric, the average criterion Q^κ is taken as the arithmetic mean of Q_T^κ and Q_E^κ as defined in Eq. 1. When comparing FPT-trained to fragment count-trained maps, only Q_T^κ will be considered, as the latter were not employed in conjunction with the Euclidean metric.

m	M	Σ(m,M)	NeurXY	$\Sigma^{\rho=1}$(m,M)	$\Sigma^{\rho=2}$(m,M)
1	2	0.123	(3,8)-(4,8)	0.123	0.123
1	3	0.751	(3,8)-(7,1)	∞	∞
…	…	…	(…)-(…)	…	…
i	j	0.057	(4,3)-(6,1)	∞	0.057
i	j+1	0.438	(4,3)-(4,5)	∞	0.438
…	…	…	(…)-(…)	…	…
N-1	N	0.632	(1,3)-(1,3)	0.632	0.632

Fig. 3. Illustration showing how the choice of the neuron radius ρ impacts on the calculation of the inter-molecular dissimilarity score matrix $\Sigma(M,m)$. Given the actual values in column 3, and the neuron co-ordinates of compounds in the pairs (column 4), it can be seen that at small ρ values, most of the pairwise comparisons are no longer carried out: at $\rho = 1$, dissimilarity is supposed to be infinite unless compounds are not residing within the same or within adjacent neurons. This is justified in most cases—coloured in green, where the effort of the calculation of high Σ values is spared. Sometimes, this may cause pairs of similar compounds to be not recognized as such: situation depicted in red. Further increasing of the radius may eventually bring such pairs back into the subset of potentially close neighbours. Orange boxes represent situations of dissimilar molecules, nevertheless located on neurons within the range of ρ.

2.5.3. Real-life testing: large library comparison with the map-enhanced virtual screening tool

In order to allow rapid matching of large sets of commercial compounds against a typical corporate collection a SOM-driven virtual screening tool has been designed to automatically break up the problem into tractable sub-problems, and then deploy each sub-problem on an independent CPU. Such a comparison tool may serve either with the goal to discover similar, potential hits close to already existing corporate compounds, or to seek for original pharmacophore patterns, not yet represented in the corporate collection. First, the guiding map (some optimal configuration discovered during benchmarking) must be installed, step at which the compounds of the corporate collection *ExtDB* are assigned to their winning neurons of the new map, and the FPT1 descriptors of compounds residing on a same neuron are grouped together in a common, neuron-specific file. This operation must be done only once, for a given map: then, accordingly reshuffled *ExtDB* fingerprints are ready to be confronted to an arbitrary number of external queries *ExtQ*. External queries are first encoded under the form of FPT1 descriptors as well, then they are assigned to their winning neurons too. Eventually, given the neuron radius ρ recommended for use with the installed map, it is straightforward to list, for each query, located in x, y, the subset of reference compounds which require to be confronted therewith. For a given (x,y) and the corresponding sets (X,Y) with $|x-X| \leqslant \rho$ and $|y-Y| \leqslant \rho$, the query file and its associated reference FPT1 files are dispatched for exact Tanimoto-based compound dissimilarity scoring on some free node on a cluster or—in the present test—on an available CPU out of the four of the used x86_64 RedHat workstation. The master script then pauses until a new computational resource is available to dispatch new query/reference file bundles, until all the query compounds have been confronted—each with the relevant subset of *ExtDB*, the one residing on neurons close to its own. Hits outside the employed neuron radius ρ will never be retrieved and are not known to this date (the baseline, complete computation of the similarity matrix of $12{,}491 \times 1.6\,10^5 = 2$ billion floating-point Tanimoto score calculations in 4418 dimensions has not been attempted). The number of reported *ExtQ–ExtDB* hits (at Tanimoto $\geqslant 0.75$), and the physical time taken to complete the calculations (deployment waiting times included) have been monitored with respect to several maps, found to be optimal or near-optimal during the benchmarking work. Some of these maps were also used at variable ρ values, in order to check in how far the choice of ρ in order to optimize Q as outlined in Eq. 1 actually selects a radius values which performs well in this real-life test.

3. Results and discussion

3.1. Monitoring map convergence

Map training is an iterative process through time. While training, the software learns the code vectors of the neurons from sample vectors of the input data. The only stopping criterion is the number of steps, decided by the user at the beginning of the training process. The question addressed here is how the tuning process of the code vectors affects the VS enhancement propensities of maps. These are different form the actual training objective (quantization error)—yet, it is clear that a poorly trained map (i.e., with almost random code vectors) may not significantly enhance the VS process. With a random map, the retrieval rate of similar compound pairs should linearly scale with the number of compound pairs subjected to explicit Σ value calculations, since there is no meaningful grouping of related compounds on a same or on neighbouring neurons.

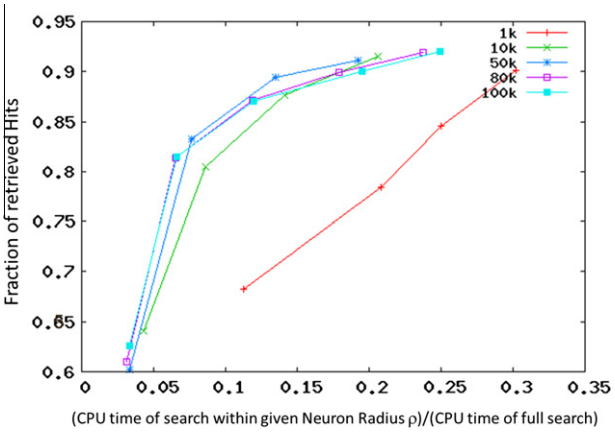

Fig. 4. Retrieval rate—time fraction ($RR-f$) curves after different training steps, of a 22×28 map, Bubble neighbourhood with the *SmallRef* dataset.

Map convergence depends both on the training set size and the map size. Figure 4 shows the $RR-f$ plots (in terms of Euclidean-driven VS, $\Sigma = E$) for a succession of maps trained on the *SmallRef* set, of dimensions 22×28, rectangular topology and bubble neighbourhood function. The first map in the series was obtained after 1000 Brute steps of training from a randomly initialized set of code vectors, whereas successive maps were each obtained by further training of their predecessor, by the indicated number of steps: the second map evolved from the first after further 10,000 Refinement iterations (thus accumulates a total of 11,000 training steps, etc.). Note that here map labels correspond to the number of training iterations in the *last* fitting round they have been submitted to. Therefore, this map built on 11,000 cumulated B + R iterations will be simply labelled '10k'. It is the starting point of further HyperRefinement runs of various length (50, 80, 100 and 200 thousand steps). The map labelled '50k' in Figure 4 accumulates thus a total of 61,000 training steps, etc. In terms of the map training strategies outlined in Section 2.3.2, this map is the product of three successive steps: B, R and HR. Other maps correspond to even longer HR runs. This refitting scheme starting from a common 'ancestor' ensures that the relative behaviour of successive maps is purely due to the fitting process, and not due to the stochastic choice of initial code vectors.

This figure shows that, with *SmallRef*, the convergence is quickly obtained. The B training step is insufficient, leading to relatively large and not very homogeneous nodes. However, the following R run significantly improves the $RR-f$ characteristic of the map, pushing it into the near-optimality area. Relevant nodes hosting query compounds shrank (at $\rho = 0$ only 64% of expected hits were found to reside on the same neuron of the query). Yet, after extending the scope of the search to $\rho = 1$ (query neuron and its neighbours), more than 80% of hits are covered, and more effectively than at $\rho = 0$ of the Brute map. A slight improvement is still witnessed after HR. However, further pushing of the fitting process at 80 or 100 thousand additional steps does no longer bring any improvement—on the contrary, high retrieval rates seem to become relatively more expensive in terms of computational effort (larger neuron radii needed). Apparently, fitting starts by assigning the few gross families of compounds to some neurons of the map, while all the others are empty (it is highly unlikely to generate, at the random initialization step, a 4418-dimensional code vector which by chance strongly correlates to an existing pharmacophore pattern in a molecule. Only a dwindling minority of possible code vectors encode chemically meaningful code vectors, anyway). Further refinement aims at splitting the gross families on the few populated neurons into more specifically defined subfamilies, to be

hosted on the so-far empty neurons in the vicinity of the original attractor.

Over-fitting artefacts are even stronger when the larger training set *Extended* is employed. Figure 5, first of all, shows that achieving convergence is globally more difficult with larger sets: B+R steps alone are not yet able to push the $RR - f$ curve into the optimality zone. HR at 50k additional steps is necessary to achieve this. More aggressive refinement, notably at 200k additional steps, clearly illustrates an unwanted 'upwards' bend of the $RR - f$ curves, signalling a loss of the ability to effectively retrieve the nearest neighbours.

The distribution of the molecule populations in the neurons of the corresponding maps supports the same explanation suggested during the discussion of *SmallRef*-trained maps: it can be seen that more aggressive map training results in a steady homogenization of population sizes allocated to every neuron. At a certain point, however, this homogenization appears to be artefactual, and no longer match the 'natural' compound family sizes found in the data set. Too much refinement is not necessarily beneficial. First, different families may be spread out over zones of different sizes, which makes the choice of the optimal ρ value a frustrating exercise: low ρ is effectively dealing with families that were not dispatched over large areas in spite of continued fitting. High ρ values are a must in order to ensure decent retrieval rates within families that have been dispatched over many neighbouring neurons, but are time-wasting choices with respect to the queries in 'localized' subfamilies. At a certain point in the fitting process, the selective spreading of specific families over much larger zones, while localized families remain confined to a restrained set of neighbouring neurons, may therefore cause an overall loss of efficacy of the map as a generic virtual screening enhancer—as observed here. Interestingly, albeit SOMs are unsupervised learning methods, and albeit the fact that the minimized criterion is the quantization error, not the quality of the $RR - f$ characteristic, the typical signature of over-fitting artefacts can nevertheless be evidenced in terms of $RR - f$ quality.

In parallel the Quantization error monotonously decreases upon more aggressive training: from 11.61 for the B map (1k), to 11.50 for B + R (10k), to 10.13 at HR (50k), 9.93 (80k), 9.62 (100k), 9.51(200k). As can be seen the most significant quantization error decrease (during the additional 50k steps of HR) is also matched by the most significant improvement of the $RR - f$ curve. However—and expectedly, since quantization error is the objective function minimized at SOM training—quantization error cannot signal over-fitting.

However, convergence of maps seems to be heavily map size dependent. Unlike the previously shown results, fitting of the much smaller 10×10 map (Gaussian neighbourhood function) does not seem to follow the same pattern. Brute and Refinement appear insufficient in this configuration (the additional R steps actually seem to slightly decrease map quality). Further training, however, fails to reveal clear signs of reaching an over-fitted configuration and aggressive fitting of the 10×10 map actually seems to favour apparition of massive nodes, while sparsely populated nodes are rare (they do exist, though, and particularly in the Brute 1k map. Node population level plots at various fitting stages can be accessed upon request). *Per se*, this behaviour is not surprising: the more degrees of freedom in a model (here: the more neurons in a map), the more likely it is prone to over-fitting artefacts. Practically, though, it is not easy to predict (without running this very time-consuming scan at successive training levels) whether a peculiar map, at peculiar geometry, is being over-fitted or not, after the HR step—or whether, on the contrary, it might still benefit from more training. However, the general trend emerging from Figure 6 clearly evidences over-fitting of 100k maps with respect to 50k ones, the effect being overall visible for the *SmallRef* set, and quite specific for large maps, in agreement with the previous discussion.

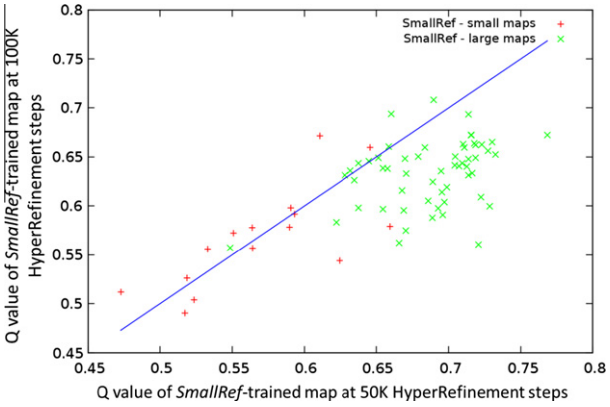

Fig. 6. On *X*, the plot monitors *Q* factor of *SmallRef*-trained maps after a 50k HR run against, on *Y*, the corresponding value for a map obtained after a 100k HR task. Green dots correspond to 'large' maps—of more of 200 neurons, red dots stand for small maps (from 48 to 200 neurons). The straight line is the diagonal: points below diagonal are over-fitting-prone maps.

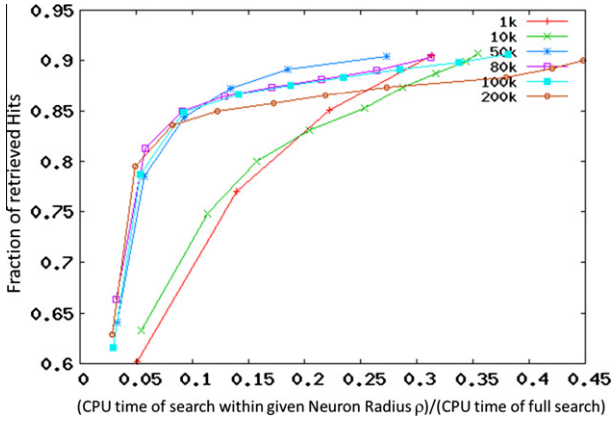

Fig. 5. $RR - f$ curves after different training steps, of a 22×28 map, Bubble neighbourhood with the *Extended* dataset.

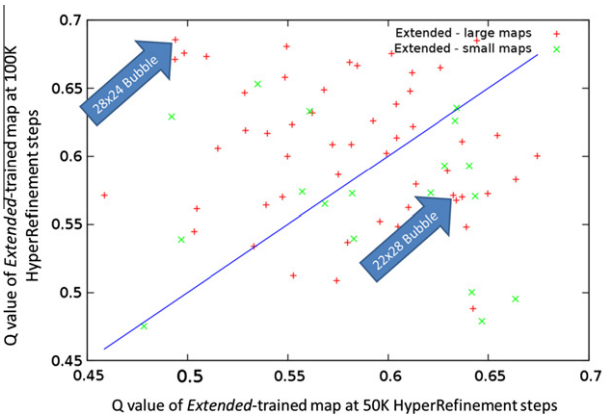

Fig. 7. On *X*, the plot monitors *Q* factor of *Extended*-trained maps after a 50k HyperRefinement run against, on *Y*, the corresponding value for a map obtained after a 100k HyperRefinement task. Green dots correspond to 'large' maps—of more of 200 neurons, red dots stand for small maps (from 48 to 200 neurons). Arrows pinpoint the behaviour of two very similar maps which nevertheless differ significantly: the upper significantly improves at 100k, while the lower suffers from over-fitting after 50k HyperRefinement steps.

When using the Extended training set, however, the optimal training effort appears to display an almost chaotic dependence (Fig. 7) on the precise map geometry: sometimes HR at 100k steps triggers significant over-fitting artefacts—as had been the case with the 22 × 28 map studied in detail previously, see Figure 5. However, the opposite scenario seems to occur equally often: 50k HR steps may as well be insufficient. Note that the most striking example thereof occurs with the 28 × 24 Bubble map—a very similar set-up to the above-discussed 22 × 28 case study.

3.2. Impact of the training set size. Top performance maps.

The study of map convergence has already shown that a training set size increase does not only render convergence more difficult to achieve, but also erratically affects the position of the borderline between optimal training and over-fitting. However, how does the set size affect absolute map quality? Let Q_s represent the Q score for a map trained on the *SmallRef* set—plotted on Y in Figure 8, and Q_e (on X) its counterpart of the *Extended*-trained map. The strong prevalence of above-diagonal points in Figure 8 clearly shows that, all other things being equal, maps trained on *SmallRef* tend to be more potent VS enhancers than those built on the basis of *Extended*. All in all, the dominant maps at $Q_s > 0.7$ are all based on *SmallRef* and built with 50k HR steps. By contrast, no set-up whatsoever, all sizes, neighbouring functions and training strategies confounded, led to an *Extended*-trained map of $Q_e > 0.7$. The dominant maps are all large (with more than 300 neurons). By contrast, neither one of employed neighbouring function (Gaussian vs Bubble) displays any dominance in terms of associated high quality maps.

It may seem puzzling that increasing the training set size should result in a global decrease of map performances, albeit the *Extended* training set actually makes up for a large majority of the DB used in the similarity screening simulation producing the Q criteria. Yet, Kohonen nets are unsupervised learning algorithms: more input information does not necessarily lead to maps of improved Q scores. Furthermore, the non-linear training procedure is prone, like any complex objective function minimizations, to risks of being trapped in local minima, etc. In light of the insights provided by the detailed study of convergence, it can thus be said that any potential benefits stemming from the supplementary information provided by *Extended* are being cancelled by the increased difficulty of map training in presence of more input molecules. This notwithstanding, please note that the so-called *SmallRef* set is already a consistent collection of eleven thousand diverse and relevant molecules, thus some two order of magnitude larger than a typical structure-activity learning set. The flattening out and eventual decrease of the map performance with training set size allegedly happens somewhere within the range between 10 and 50 thousands compounds. Albeit this work does not furnish a formal proof, the further reducing of the training set below *SmallRef* is not likely to witness a further grow of map quality. Our argument in this respect is the coherent behaviour of *SmallRef*-trained maps in the convergence study: at 50k HR steps, these seem indeed to reach an optimum—thus fully exploit all the chemical information in the training set. *SmallRef* does hence not appear to be as large as to jeopardize map convergence, by contrast to *Extended*, for which the study showed a chaotic behaviour in the convergence study, and failed to highlight an optimal number of training iterations.

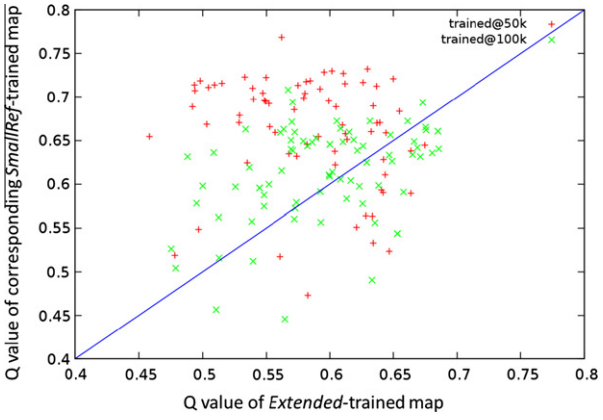

Fig. 8. On X, the plot monitors Q factor of *Extended*-trained maps against, on Y, the corresponding value for the equivalent map built on hand of *SmallRef*, all other parameters being equal. Green dots correspond to aggressively trained maps at 100k HR steps, by contrast to 50k step-maps rendered as red points.

3.3. SOM-driven virtual screening enhancement in fragment descriptor spaces

The first general conclusion that can be drawn from Table 1 is that VS enhancement using Kohonen maps is generally applicable to various descriptor spaces and metrics. In particular, the novelty introduced in terms of the Tanimoto score (the absence of average/variance rescaling of fragment counts, by contrast to the FPT terms) does not conflict with the perception of similarity 'prone' by the Kohonen nets (which is Euclidean). Hit retrieval rates and time enhancement factors are quite similar to the ones reported for the FPT descriptor space (90 % of hits retrieved in ≈10% of time), meaning that molecule pairs perceived as similar by either rescaled Euclidean, rescaled Tanimoto or non-rescaled Tanimoto metrics are well localized in a same or in closely neighbouring neurons.

Both maps show a largely consensual behaviour in both FPT and sequence count spaces, with the 22 × 28 configuration furthermore being a top performer in all the three spaces. Their initially established proficiency rank (22 × 28 > 10 × 10) is upheld in fragment count spaces. The only marked difference is the much greater performance gap with respect to the *SmallRef* set in atom-centred fragment space.

Over-training is as much an issue in fragment space descriptors at it was in FPT space: note that, in all but three cases, the optimal Q_T scores were reached within the 50k HR stage. The two cases witnessing top Q scores being reached after the 200k HR stage correspond to poor-quality maps. While, in the ISIDA sequence count space, *SmallRef* is clearly a sufficient training set, atom-centred fragment-based SOMs require training on the *Extended* set. This is surprising, since this 2836-dimensional descriptor was the shortest of all the three herein considered vectors. Expectedly,

Table 1
Optimal training stage (labels as explained in Section 3.1: 10k = B + R, >10k = additional HR) and (Tanimoto-based) Q_T factors at optimal training stages, for the two map configurations evaluated in fragment count-based descriptor spaces

Descriptor type	Training set	Map	Optimal training stage	Q_T
Atom-centred	*SmallRef*	10 × 10	200k	0.40
Atom-centred	*Extended*	10 × 10	100k	0.69
Atom-centred	*SmallRef*	22 × 28	200k	0.49
Atom-centred	*Extended*	22 × 28	50k	0.69
Sequences	*SmallRef*	10 × 10	10k	0.65
Sequences	*Extended*	10 × 10	50k	0.70
Sequences	*SmallRef*	22 × 28	80k	0.73
Sequences	*Extended*	22 × 28	50k	0.73
FPT	*SmallRef*	10 × 10	50k	0.63
FPT	*Extended*	10 × 10	200k	0.69
FPT	*SmallRef*	22 × 28	50k	0.72
FPT	*Extended*	22 × 28	80k	0.66

the more detailed the descriptor, the more input (coverage) of the chemical space should be required in order to allow for meaningful mapping. Note that this effect is not due to rare fragments, seen in less than 10 of the molecules—these were, currently, ignored both at map training and at similarity-based VS steps.

Both fragment descriptors were treated as fixed size vectors, although in reality they are open-ended. Practically, a SOM can be built and exploited only on hand of a fixed size descriptor—some pre-selection of fragments for SOM building is compulsory, and these may be indeed training set dependent. However, a same subset of selected fragments was used with both *SmallRef* and *Extended*—this potential problem does not explain the noted differences. Molecules containing fragments ignored at SOM build-up would, if these 'exotic' fragments were taken into account at the similarity scoring step, appear as dissimilars that were 'abusively' located within the same neuron as the query compound. This is however a minor issue, merely causing some (potentially avoidable) time loss due to the undue computation of a dissimilarity score. The insight that SOM nodes must not be structurally homogeneous has been an evidence all since the development of this technology: compound similarity implies a positioning within a same, or neighbouring neurons, but residence within a same neuron does *not* imply the existence of an underlying structural similarity. This problem is thus general, given the finitude of SOM maps versus the quasi-infinitude of drug-like compounds—but however minor, for it engenders only a suboptimal—nevertheless very significant—VS acceleration, and no hit misses.

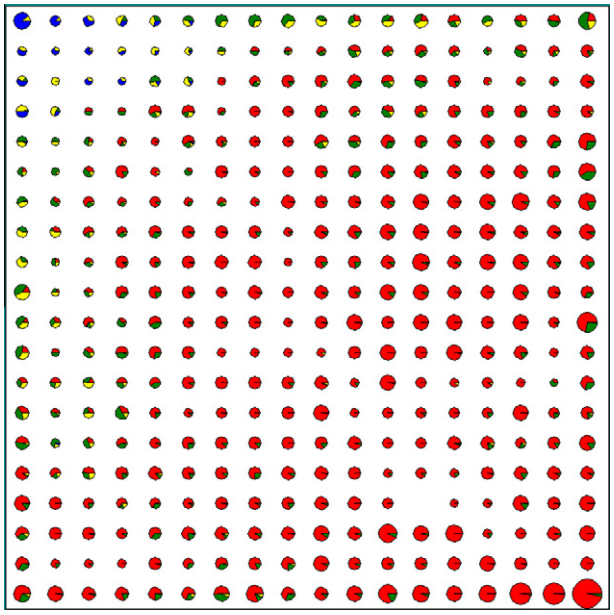

Fig. 9. Repartition of compounds on the 18 × 20 rectangle bubble map. Neuron circle sizes represent the number of resident compounds. Neurons are coloured proportionally to the fraction of compounds at given number of Lipinksi rule violations. Red = 0 violations, green = 1, yellow = 2, blue = 3.

3.4. An overview of the best pharmacophore space (FPT) map

Amongst studied maps, the best quality criterion $Q = 0.77$ corresponds to a SOM trained on the *SmallRef* dataset, which contains 18 × 20 = 360 neurons and has been trained in three steps (B + R + HR of 50k iterations), with a rectangular topology and a bubble neighbourhood function.

Figure 9 visualizes the mapping of *SmallRef* on this SOM, where molecules have been colour-coded by the number of Lipinski[37] rule violations, in order to convey a general idea of the chemical space mapping quality. The map coherently accounts for drug-likeness,[38] with a clear drug-likeness gradient on the north-west (non-drug-like) to south-east (drug-like), albeit it was not trained by any means to account for drug-likeness (SOMs are unsuperwised learning methods, anyway). Furthermore, the map is relatively homogeneously populated by the *SmallRef* compounds, with only one empty neuron and a largest neuron (on the far bottom right) with 300 compounds (the smallest molecules in the set, of 100 < MW < 200), where 291 thereof break no Lipinski rules.

For a better view of the quality of this map, a set of 2170 active compounds from the DUD database ([29]) have been mapped on it and coloured according to their associated targets (see Table 2). Figure 10 shows their repartition on the map. This DUD subset has been selected to regroup targets with a maximum of associated ligands. Unfortunately, nothing is known about potentially promiscuous ligands, which may bind to other targets out of the ten selected, in addition to the 'officially' assigned one. Since six targets out of them are kinases, for which the discovery of selective ligands is notoriously difficult (ATP-mimicking ligands will hit many kinases, for they all have a more or less well conserved ATP binding pocket), pharmacophorically similar compounds will reside on a same neuron, even if they are formally assigned to distinct categories with respect to the targets they bind. Ache, Fxa and Src ligands are well separated from the others. Src is a tyrosine kinase, represented by a set of specific inhibitors. Intriguingly, Cox2 ligands share a node ((18,6), the biggest with 263 compounds—more than 10% of the total) with p38 kinase ligands. Albeit cyclooxygenase 2 and the p38 kinase are functionally different, Cox2 also happens to recognize ligands with large aromatic moieties and hydrogen bond acceptors or anions. Indeed, the two ligand series are structurally very close, especially as far as pharmacophore patterns can tell. Discovery of the right-hand p38 ligand by similarity-driven virtual screening, with the left-hand Cox2 compound of Figure 11 as a query, is an example of potentially promising 'lead hopping'—changing of scaffold while preserving the actual pharmacophore pattern. Whether the respective Cox2 and p38 ligands

Table 2
Label, targets and number of DUD compounds found for each target

Nb	Target	Nb of actives	Colour
1	ache (Acetylcholinesterase)	106	Red
2	cox2 (Cytochrome c oxidase subunit II)	409	Yellow
3	dhfr (Dihydrofolate reductase)	408	Light blue
4	egfr (Epidermal growth factor receptor kinase)	427	White
5	fgfr1 (Fibroblast growth factor receptor 1 kinase)	97	Grey
6	fxa (Factor X)	146	Pink
7	pdgfrb (Beta-type platelet-derived growth factor receptor kinase)	110	Gray blue
8	p38 (p38 mitogen-activated protein kinase)	342	Dark blue
9	src (Proto-oncogene tyrosine-protein kinase)	49	Dark red
10	vegfr2 (Vascular endothelial growth factor receptor kinase 2)	76	Green

The colors represent classes on the map.

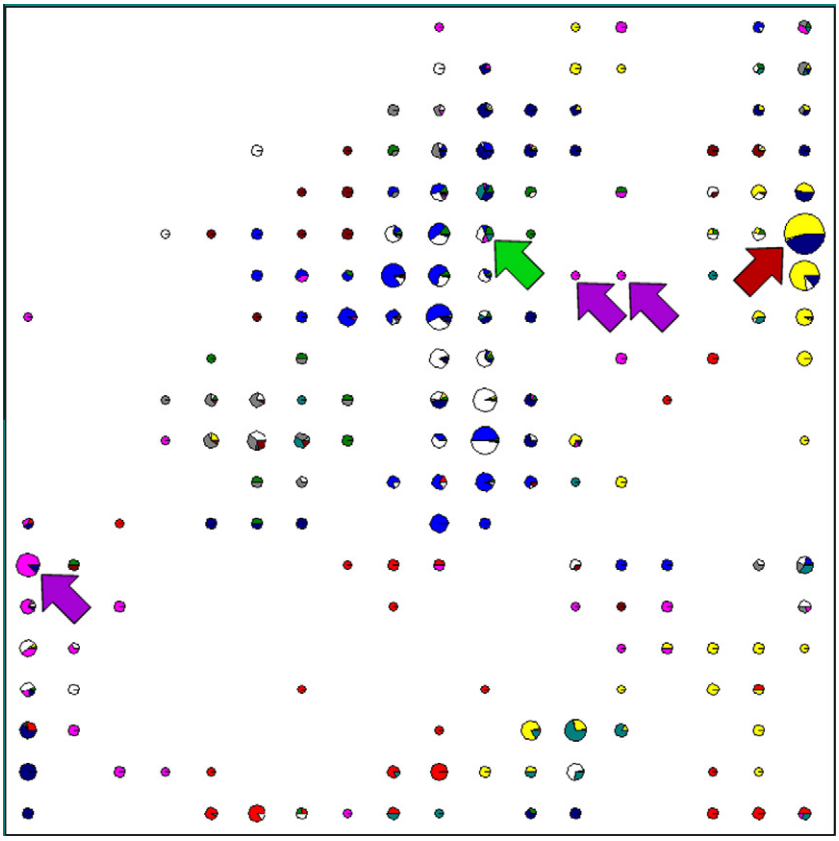

Fig. 10. Mapping of the 2170 DUD compounds on the 18 × 20 rectangle bubble map. Neurons are coloured according to the class they display. See Table 2 for colors. The red arrow points to neuron (18,6), the green arrow points to neuron (11,6) and the violet arrows point to neurons (1,14), (13,7) and (14,7).

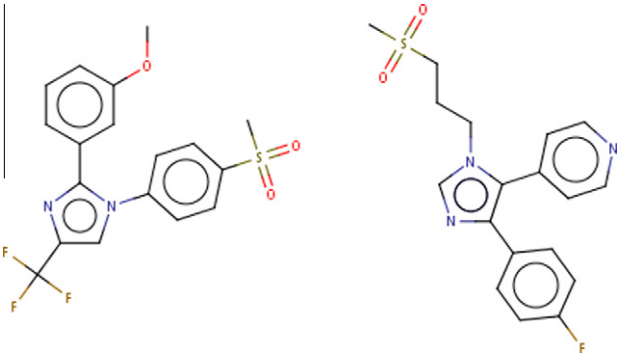

Fig. 11. Left: a Cox2 inhibitor, and right a P38 inhibitor of the DUD data set. Albeit binding to different targets, they clearly display a highly similar overall pharmacophore pattern and reside both in neuron (18,6)—red arrow in Figure 10.

are actually binders to both targets is unknown to us at this point. All ligands are indeed in the 'Lipinski-compliant' area of the map, with the bigger ones closer to the top left areas of the map.

A closer look at one heterogeneous neuron (11,6) shows that some of the depicted targets share structurally close ligands. This node regroups ligands binding respectively to dhfr, egfr, fxa, pdgfrb, vegfr2 (see Fig. 12). Despite their different primary targets, it is visible that they share similar pharmacophoric patterns (according to our descriptors). Knowing that Dihydrofolate reductase readily binds heterocyclic bases and favours negatively charged compounds, like the kinases, this is actually not surprising at all—the most atypical resident of the node is the fxa inhibitor, which nevertheless features extended aromatic systems 'ornated' with several hydrogen bond acceptors (here, carbonyl groups instead of pyridine nitrogens) and donors (amide NH groups, instead of phenylamines). Again, the affinity of these ligands for the alternative targets is not known.

Factor X inhibitors are spread all over the map (but usually in nodes with a good purity). However, this is not a weak point of the mapping per se, but a general limitation of global similarity-based virtual screening: overall pharmacophore pattern similarity is by no means a *necessary* condition for two compounds to bind a same target (see detailed discussion of the Neighbourhood Behaviour problem[25]). Indeed, the necessary conditions for two compounds to be recognized by a same target is they possess key anchoring groups actually interacting with the receptor—that is, a *binding pharmacophore*. But the overall pharmacophore pattern encoded by the fuzzy triplet fingerprint is a function of all putatively interacting groups—thus, imposing a strict similarity of the entire pattern is far too stringent. First, molecular moieties that never interact with the site may arbitrarily vary. Second, a protein site possess many putative interaction centres, and not all of these are systematically used to bind all the ligands. Ligands of comparable activities may exploit some different anchoring points—only few anchoring points are recurrently used by all the known actives, and these form the consensus binding pharmacophore. But two molecules may share a same consensus pharmacophore, yet be widely dissimilar—with respect to the other, much more numerous, irrelevant or occasionally relevant groups—which are accounted for in the overall pharmacophore pattern fingerprint, by definition. This is the case here (Fig. 13): on one hand, the benzamidine group in the left-most fxa ligand is a hallmark of fxa activity, and is positively involved in the interaction with the active site. It also has a prominent role in shaping the overall pharmacophore

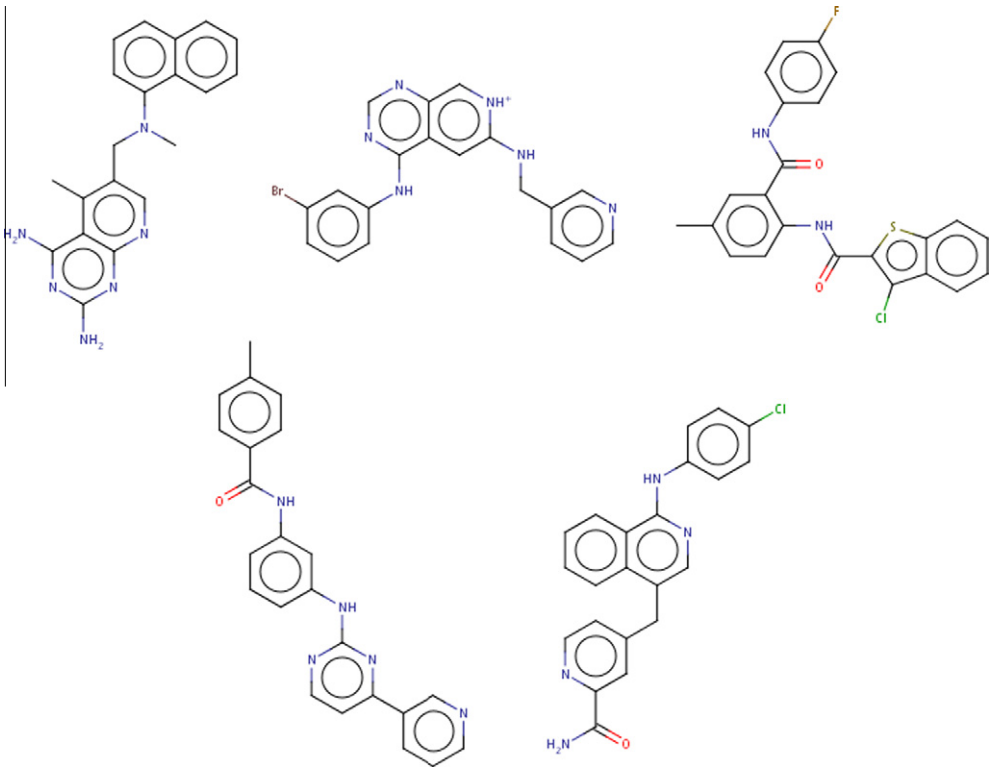

Fig. 12. Example of five compounds found in the heterogeneous node (11,6)—green arrow in Figure 10. From left to right: dhfr inhibitor, egfr inhibitor, fxa inhibitor, pdgfrb inhibitor, vegfr2 inhibitor.

pattern, as it is a carrier of a positive charge and of several H bond donors. However, it is not *compulsory*, as proven by the right-most fxa binder. The two extreme fxa ligands are beyond doubt *dissimilar* in terms of overall pharmacophore patterns: one is cationic and rich in H-bond donors, the second is neutral (actually, as FPT descriptors are pH-dependent, at 7.4 the latter fingerprint captures some low contributions from the anionic species obtained by deprotonation of the sulfonamide at a predicted pH of about 8.0). No surprise, thus, to see these ligands on remote neurons, beyond the neuron radius of this map. Starting the search with the left-hand ligand as a query cannot find the right-hand molecule—and rightly so, not because of the map and its neuron radius, but because the pharmacophore dissimilarity score of these compounds is high: they would have not selected each other in all-pair-based similarity scoring, either. The middle ligand, however, is reasonably similar to the right-hand molecules (and their residence neurons are adjacent)—yet, it is still far from the left-hand benzamidine, although the *o*-aminopyridine moiety (shown as charged) is a bioisostere of benzamidine. However, due to pH-sensitivity of the fingerprints, in actual FPT calculations the charged form as shown in the Figure contributes only roughly 50 %, whereas the neutral one (with the pyridine N as an acceptors) is equivalently important: the pK_a of this *o*-aminopyridine is estimated at roughly 7 by the ChemAxon plugin[39] called by the FPT generator. True, a medicinal chemist may decry the failure to pick the middle ligand starting from the benzamidine as a query, but this is not due to employing the map as VS enhancer, but again due to the intrinsically high dissimilarity of pharmacophore patterns. Rule-based pharmacophore flagging (stating that both benzamidine and *o*-aminopyridine are charged, thus equivalent) would have perhaps been more satisfactory for the end user—in this situation. In many others, apparently insignificant structural changes triggering

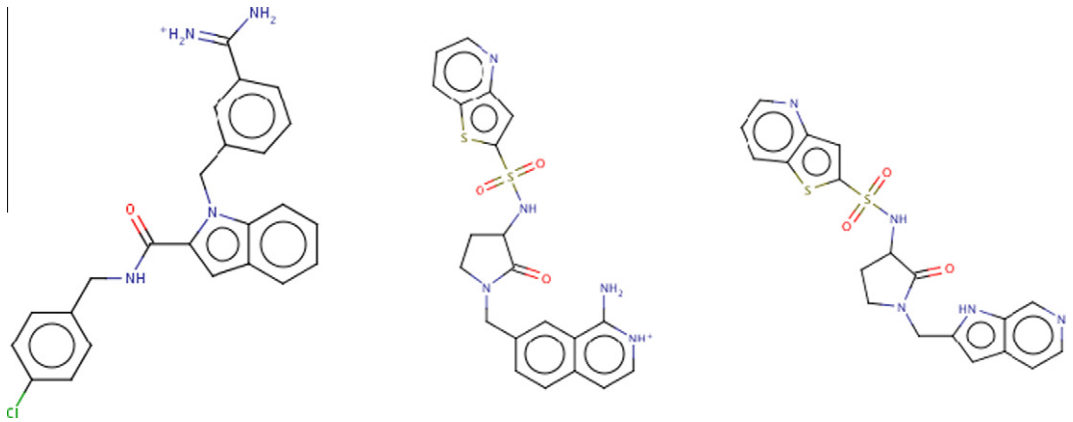

Fig. 13. Comparison of three fxa ligands residing in three different neurons (from left to right: neuron (1,14), neuron (13,7), neuron (14,7)—violet arrows in Figure 10).

important pK_a shifts will immediately result in puzzling *activity cliffs* [40] unless pH-dependence of the descriptors is not accounted for.[30]

3.5. Real-life virtual screening enhancement tests and benchmark

A preliminary real-life test of VS enhancement has been performed on a subset of 4 maps selected for their good a priori performances—at the point of this undertaking, which preceded the discovery of the best map discussed above. The goal of this simulation was to double-check in how far the actual VS enhancement propensities of these maps match their relative ranking by the *Q* criterion—that is, whether *Q*-based design of maps will lead to effective VS enhancement tools.

The four selected maps, which all displayed *Q* values between 0.5 and 0.7 were :

- Top1: 22 × 28 rectangle bubble trained on the SmallRef set
- Top2: 20 × 40 rectangle gaussian trained on the Extended set
- Top3: 28 × 30 rectangle bubble trained on the SmallRef set
- Small: 6 × 6 rectangle bubble trained on the Extended set

Each of the four maps was used to confront the *ExtQ* set to the *ExtDB* compounds. In addition, for the map top2, the virtual screening was conducted with various neuron zone sizes, below and above the optimally established $\rho = 3$. For each run, we monitored the effective physical time the virtual screening took to complete, as well as the number of detected pairs of Tanimoto dissimilarity below 0.25 (i.e. the number of retrieved hits).

3.5.1. Hit rate analysis

The total number of (*ExtQ, ExtDB*) compound pairs within 0.25 of Tanimoto dissimilarity is not known, since the cumbersome systematic comparison of $\sim 2 \times 10^9$ fingerprint pairs has not been attempted. Note—the retrieved hits are unequally distributed among the external compounds: if the 'average' external compound turns out to have slightly above 2 neighbors among the ExtDB molecules, this is because some 7000 of the 12k do not have any hit at all, while 1500 feature 10 hits or more (only the 10 top hits are returned and counted, anyway). This is not surprising, given the ubiquity of series in corporate databases: compounds that are related to an in-house motif are more likely to return many hits, for there are multiple 'incarnations' of that motif. The tool therefore does detect diversity holes. (see Fig. 14).

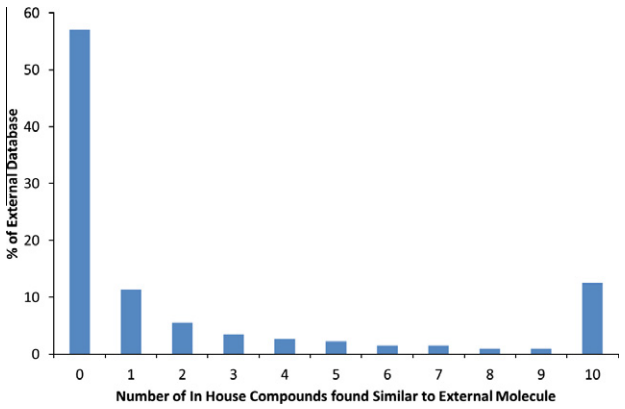

Fig. 14. Percentage of ExtQ (on *Y*) found to have *N* (on *X*) near neighbours among in-House compounds (ExtDB). The almost 60% at '0' are brand-new pharmacophore patterns, not present in the corporate database ExtDB. The others may still be potentially interesting, if they are original from a scaffold-centric view. Distribution obtained with map 'top2', where the average number of neighbours/external compound is of 2.17.

3.5.2. Map-dependent hit rate: how many of the pairs of neighbours escape detection when using maps for acceleration, and what time gain does one get in compensation?

The optimality criterion *Q* used to choose these maps, as a 'best' compromise between speed-up and hit loss (while taking care that hit loss does not exceed 15%) may not be a direct quality indicator for a real-scale experiment because (a) the sizes and the nature of the involved sets are not the same—the peculiarities of compound distribution in a corporate database were not accounted for, and (b) the real-scale experiment has been parallelized on multiple (4) CPUs, thus biasing time gain measures over the original map optimality.

The Table 3 displays, for each map, at nominal or variable neuron radius ρ, the number of retrieved hits and the time it took to complete the screening.

It is clear from above that the initial 'small' 6 × 6 bubble map was the best in detecting similar pairs (the baseline number of which is not known, albeit the scanning of ρ with the map top2 suggests that convergence at full map coverage should stabilize this number somewhere not far beyond 30000). This is achieved in a relative good time of less than 3 h, whereas bigger—thus finer—maps such as top2 witness an aggressive increase in computer effort at increased ρ. Conclusion: similar compounds which, for whatever reasons, are not located on close-by neurons, are at risk to be dispatched anywhere in the map—retrieving them by increasing *R* may be very costly. In terms of the speed-up, solutions top1 and, in a lesser extent, top3 are very satisfying. All in all, the good behaviour of the maps as evidenced at their primary benchmarking stage (2000 external compounds (QS) × 53,000 playing the role of 'in-house molecules' (DB)) was confirmed in this real-scale experiment.

As hinted by the primary benchmarking, the optimal neuron zone size ρ for top2 is indeed 3, and the top1 and top3 maps have clear speed-up advantages over the 'small' solution. The second-best ranked map top2 was slightly deceiving—albeit its best performances happened at the nominal neuron radius value as assigned by the *Q*-driven set-up process, it was outperformed by both top3 and small maps in the real-life VS simulation. Obviously, small *Q* variances as the ones between the 'top' maps are not relevant. It is not reasonable to expect that *Q* may represent some universal quality criterion. While high *Q* may not guarantee excellent map performances under any arbitrary VS conditions, low *Q* proves a map to be a bad performer.

3.6. General remarks about map-enhanced virtual screening

Training the Kohonen maps represented, per se, a negligible amount of time compared to the extensive testing of all the map

Table 3
Results of real-scale inter-set similarity assessment

Map	Neuron radius ρ	# Detected similar pairs	Time (min)	*Q*
Small	1	29718	158	0.51
Top1	1	27107	65	0.71
Top2	1*	24588	74	—
Top2	3	27096	139	0.70
Top2	5*	27707	260	—
Top2	10*	28571	597	—
Top3	2	27837	96	0.69

Neuron radii values labelled by * were checked out for testing purposes, and do not represent the nominal values associated to that map. The last column reports the *Q* factor of the map, based on the benchmarking study.

versions—different topologies, neighbourhood functions, training sets, at various refinement levels—with respect to their virtual screening propensities, in terms of the herein advocated *Q* factor formalism. Full Euclidean and Tanimoto score calculation of each of the 2000 queries against the 55 thousand DB molecules took ≈6400 s in the FPT chemical space, 3400 with atom-centred fragments and 8500 with sequence counts. Whilst virtual screening with efficacious maps at proper choices of neuron radii were much faster than that, most effort went into testing all combinations of inefficacious maps at inappropriate choices of ρ. The entire scan of reported FPT-based alternative scenarios roughly took two CPU weeks on an x86_64 Unix workstation. The herein gained insight allows to drastically reduce the number of scenarios to be revisited, if the approach were to be extended to further new chemical spaces. So far, the *SmallRef* set has proven to provide for satisfactory training in two of three visited descriptor spaces, and the FPT-based tests showed that subtle choices like the one of the neighbouring function may be of no importance. Also, 100,000 iterations are an absolute maximum in terms of map training effort—which does not preclude the user to check whether better results would not be available at earlier map fitting stages. This notwithstanding, the scan of a dozen of various map, differing in both global size and length/width ratio is recommended. The best 22×28 map geometry emerging as the best set-up in FPT descriptor space has shown excellent performances with fragment counts, but may yet not be optimal one. Following these guidelines, finding a well-suited Kohonen map to enhance VS in a new chemical space may be achieved, at most, in a matter of few CPU days—and used as such for a very long time, because its training set, if sufficiently diverse, does not have to be synchronized in any way to the database of screened compounds (otherwise, larger training sets should systematically yield better results). External, objective or coincidental, factors may strongly impact the relative time gain induced by the use of the Kohonen maps. In case of large-scale deployment of such similarity scoring, using a cluster or computer grid, when on each node the number of actual similarity calculations becomes relatively small as the total job has been split amongst many CPUs, deployment/job queuing time loss may eventually overshadow the actual gain from the intrinsically faster similarity scoring. Like any standardized quality index, the *Q* factor may not accurately render the effective benefits of the method in all possible real life circumstances.

4. Conclusions

Using Kohonen self-organizing maps is an effective way to accelerate similarity searches in a database of small compounds. The acceleration tests performed on 57613 compounds, using the first 2000 as query, have proven that mapping the molecules on a SOM can considerably accelerate similarity searches without significant losses of virtual hits, as proven by high quality scores *Q*, specifically developed to the purpose of synthetically capturing the compromise between speed-up and hit loss. The best maps may retrieve about 90% of the relevant neighbours of the query in about 10% of the total time required to scan the entire database.

During the training phase, attention should be focused on the convergence of the maps. Failure to converge results in suboptimal performances, but—somehow surprisingly for an unsupervised learning method—excessive map training was shown to lead to over-fitting artefacts. There seem to be no unequivocal rules on how to establish the optimal number of training iterations—convergence behaviour being notably determined by both map geometry and training set size. A gradual training strategy, with intermediate checks of VS enhancement scores, is advised in order to discover an optimal map.

Furthermore, care is advised while choosing the training set (which must be representative of the targeted chemical space: here—drug-like molecules). Two training sets have been compared, the *SmallRef* (11,168 compounds) and the *Extended* (53,206) set. With FPT descriptors, the smaller training set is sufficient to create maps that depict correctly the chemical space of our database. Surprisingly, the employment of *Extended* training set was not only unhelpful to further increase map performance, but often detrimental (its specific impact varied in function of map geometry). Too many training molecules tend to render convergence more difficult to achieve, and thus cancel out any potential benefits from the additional information they provide. Note, however, that the 'small' set is already a significant and diverse compound collection. In fragment count-based chemical spaces, only two map geometries (10×10 and 22×28) have been investigated, and so far, a similar behaviour was evidenced with ISIDA atoms-and-bond sequence counts, where training on *SmallRef* and *Extended* yield maps of comparable performances. However, an opposite trend (better results upon training on the*Extended*) was noticed with atom-centred fragment counts. Was *SmallRef* large and diverse enough to cover all the relevant pharmacophore patterns, but not all the atom-centered fragment types? The question cannot be answered unless a systematic scan of many map geometries should be first run in atom-centred fragment space. Furthermore, some quantitative measure of the coverage of chemical space by a given training set should be tentatively related to its proficiency as training set—for while it is clear that increasing the training set size above a certain limit is unnecessary or even detrimental, it is not clear what this descriptor space-dependent limit should be. Presumably, the key aspect here is not the set size per se, but rather the covered chemical space volume and, perhaps, density/redundancy of included compounds. More work will be needed to shed light on these issues.

The above-mentioned observation underlines a major advantage of SOMs as VS enhancers: the applicability of a predefined map, built on hand of a relatively small but diverse set of molecules, to accelerate VS within a much larger, independent database is proof that, indeed, there is nothing to be gained by a systematic (and time-consuming) rebuild of the map each time new molecules are added to the database to screen. This interpretation in further supported by the real-life performance tests, which have shown that the maps may successfully stand the challenge of scaling down the effort of a 2×10^9-fold matching of 4000-dimensional fingerprints to hardly more than one hour on a 4-CPU workstation, without dramatic losses of virtual hits. This, even though the herein screened corporate compound collection, completely unrelated to the other training/test sets, was as realistic a challenge case as one may encounter in drug discovery.

The winner map of the *Q*-based benchmarking study has been visually rendered, with respect to its ability to monitor drug-likeness. A clear-cut separation of drug-like and non-druglike compounds can be observed along its diagonal, showing that this is a potentially meaningful chart of medicinal chemical space. Furthermore, mapping of diverse series of binders to 10 different targets (both inter-related kinases and widely differing enzymes) lead to a coherent and sensible picture, highlighting no inherent weakness of the map as such, but rather the well-known pitfalls and limitation of global similarity-based search of active analogues. Given that map training was never oriented towards discrimination of activity classes, and that the DUD reference binders were never employed at any stage of training or map selection, this is a remarkable result underlining the excellent propensity of the Q-score to recognize meaningful chemical space maps.

Pragmatically, although some insights on how to build an optimal map have been gained (neighbourhood function does not seem to matter, and more than 100,000 training iterations are virtually

certain to cause over-fitting artefacts), other aspects (choice of the optimal geometry), some other aspects (optimal geometry, optimal training set properties) cannot be settled without scanning through a reasonable pool of alternatives. Fortunately, the discovery of the absolute optimum of similarity-based virtual screening acceleration problem is *not* required: near-optimal solutions are more than welcome. On one hand, the unlikely potential loss of some virtual hits due to this approach is a less serious problem than the miss of real hits—for not all these similar analogues will obey the similarity principle and display the same desired activity as the query. Loss of real hits (false negatives) is an issue in experimental screening as well—yet, a less stringent one than false positives, which require much more effort to analyse and discard. Or, Kohonen-map driven virtual screening is by nature unable to induce any (additional) false positives the unaided similarity-based VS would have not retrieved. Therefore, a suboptimal mapping at low neuron radius will merely result in additional false negatives, and, at increased neuron radius, in a relative loss of time compared to the absolutely optimal map—still, a net gain in time compared to unaided VS. Implementing the procedure (the one-time fitting and testing of a few maps, the (quick) mapping of the reference database thereon, and the (instantaneous) positioning of the query compound thereon, at each VS run) may only benefit similarity-driven VS: the better the chose map, the larger the benefit. Eventually, the neuron radius could be left as a tunable parameter in the hands of the user, to be adjusted with respect to the stringency of the goal of complete virtual hit retrieval. The method is supposed to be implemented and publicly available on our similarity-based Virtual Screening server, http://infochim.u-strasbg.fr/webserv/VSEngine.html by mid-2012.

References and notes

1. Lyne, P. *DDT* **2002**, *7*, 1047.
2. *Concepts and Applications of Molecular Similarity*; Johnson, M., Maggiora, G., Eds.; Wiley Interscience, 1990.
3. Willett, P. *Annu. Rev. Inform. Sci.* **2009**, *43*, 1.
4. Willett, P. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 983.
5. Lipp, E. *Genet. Eng. Biotechn. N.* **2008**, *28*, 22.
6. Smellie, A. *J. Chem. Inf. Model.* **2009**, *49*, 257.
7. Bentley, J. *Commun. ACM* **1975**, *18*, 509.
8. Gionis, A.; Indyk, P.;M.R., VLDB '99 Proceedings of the 25th International Conference on Very Large Data Bases (1999).
9. Kohonen, T. *Biol. Cybern.* **1982**, *43*.
10. Kohonen, T. *Proc. IEEE* **1990**, *78*.
11. Schneider, G.; Wrede, P. *Prog. Biophys. Mol. Biol.* **1998**, *70*, 175.
12. Polanski, J.; Gasteiger, J. *Acta Pol. Pharm.* **1999**, *56*, 112.
13. Anzali, S.; Mederski, W.; Osswald, M.; Dorsch, D. *Bioorg. Med. Chem. Lett.* **1997**, *8*, 11.
14. Anzali, S.; Gasteiger, J.; Holzgrabe, U.; Polanski, S. J.; Wagener, A. *Perspect. Drug Disc.* **1998**, *9–11*, 273.
15. Hristozov, D.; Oprea, T.; Gasteiger, J. *J. Comput. Aided Mol. Des.* **2007**, *21*, 617.
16. Hristozov, D.; Oprea, T.; Gasteiger, J. *J. Chem. Inf. Model.* **2007**, *47*, 2044.
17. Aoyama, T.; Suzuki, Y.; Ichikawa, H. *J. Med. Chem.* **1990**, *33*, 905.
18. Polanski, J. *Adv. Drug Deliv. Rev.* **2003**, *55*, 1149.
19. Polanski, J.; Jarzembek, K.; Gasteiger, J. *Comb. Chem. High Throughput Screen.* **2000**, *3*, 481.
20. Selzer, P.; Ertl, P. *J. Chem. Inf. Model.* **2006**, *46*, 2319.
21. Gasteiger, J.; Li, X. *Angew. Chem., Int. Ed. Engl.* **1994**, *33*, 643.
22. Sadowski, J.; Wagener, M.; Gasteiger, J. *Angew. Chem., Int. Ed. Engl.* **1995**, *34*, 2674.
23. Im, D.-J.; Lee, M.; Lee, Y.; Kim, T.; Lee, S.; Lee, J.; Lee, K.; Cho, K. *Lect. Notes Comput. Sci.* **2005**, *3481*, 334.
24. Oh, K.; Zaher, A.; Kim, P. *Lect. Notes Comput. Sci.* **2002**, *2383*, 131.
25. Horvath, D.; Jeandenans, C. *J. Comput. Inf. Comp. Sci.* **2003**, *43*, 680.
26. Bonachera, F.; Horvath, D. *J. Chem. Inf. Model.* **2008**, *48*, 409.
27. Bolton, E.; Wang, Y.; Thiessen, P.; Bryant, S. *Annu. Rep. Comput. Chem.* **2008**, *4*.
28. Irwin, J.; Shoichet, B. *J. Chem. Inf. Model.* **2005**, *45*, 177.
29. Huang, N.; Shoichet, B.; Irwin, J. *J. Med. Chem.* **2006**, *49*, 6789.
30. Bonachera, F.; Parent, B.; Barbosa, F.; Froloff, N.; Horvath, D. *J. Chem. Inf. Model.* **2006**, *46*, 2457.
31. Kornhuber, J.; Terfloth, L.; Bleich, S.; Wiltfang, J.; Rupprecht, R. *Eur. J. Med. Chem.* **2009**, *44*, 2667.
32. Varnek, A.; Fourches, D.; Horvath, D.; Klimchuk, O.; Gaudin, C.; Vayer, P.; Solovev, V.; Hoonakker, F.; Tetko, I.; Marcou, G. *Curr. Comput. Aided Drug Des.* **2008**, *4*, 191.
33. Varnek, A.; Fourches, D.; Hoonakker, F.; Solovev, V. P. *J. Comput. Aided Mol. Des.* **2005**, *19*, 693.
34. Solovev, V. P.; Varnek, A. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1703.
35. Varnek, A.; Fourches, D.; Solovev, V. P.; Baulin, V. E.; Turanov, A. N.; Karandashev, V. K.; Fara, D.; Katritzky, A. R. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1365.
36. T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, Report A31 (1996).
37. Lipinski, C.; Lombardo, F.; Dominy, B.; Feeney, P. *Adv. Drug Deliv. Rev.* **1997**, *23*, 3.
38. Proudfoot, J. R. *Bioorg. Med. Chem. Lett.* **2002**, *12*, 1647.
39. ChemAxon, pka calculator plugin, 2007. http://www.chemaxon.com/marvin/chemaxon/marvin/help/calculator-plugins.html.
40. Maggiora, G. *J. Chem. Inf. Model.* **2006**, *46*, 1535.