ARTICLES

LoFT: Similarity-Driven Multiobjective Focused Library Design

J. Robert Fischer,[†] Uta Lessel,[‡] and Matthias Rarey*,[†]

Center for Bioinformatics Hamburg, University of Hamburg, Bundesstrasse 43, D-20146 Hamburg, and Department of Lead Discovery, Boehringer Ingelheim Pharma GmbH & Co. KG, D-88397 Biberach an der Riss

Received August 3, 2009

We present LoFT, a tool for focused combinatorial library design. LoFT provides a set of algorithms, constructing a focused library from a chemical fragment space under optimization of multiple design criteria. A weighted multiobjective scoring function based on physicochemical descriptors is employed for traversing the chemical search space. The new aspect of LoFT is that a similarity-driven product-based library design approach is provided on fragment level. For this reason the feature tree descriptor is incorporated for similarity comparison of library compounds to given bioactive molecules as well as for diversifying the resulting libraries. The feature tree descriptor abstracts the molecular graph to a tree structure where the nodes are labeled with physicochemical properties. For comparison, the nodes of two trees are mapped onto each other. This strictly hierarchical mechanism is suitable for the efficient comparison of chemical fragments, allowing the evaluation of the resulting products on fragment level without explicitly enumerating them. LoFT was validated, applying three different data sets. Starting with a random reagent selection, we optimized the libraries using maximum similarity to known bioactive molecules and iteratively adding further criteria. Moreover, we compared these results with data we obtained with FTrees-FS.

INTRODUCTION

In the past, new chemical compounds were built by serial and systematic modifications guided by the similar property principle[1] and the lock and key concept.[2] The introduction of combinatorial chemistry, a technology for parallel synthesis of a large range of analogue molecules using the same reaction scheme, changed the way the pharmaceutical industry searches for novel bioactive compounds. The increased number of molecules that is synthesized can now be tested rapidly for the desired properties by high throughput screening (HTS).[3,4] However, in a cost-effective experimental screening not all possible products can be synthesized and evaluated for their bioactivity because of the size of the chemical space, estimations ranging from 10^{13} to 10^{180} virtual compounds.[5,6] For that reason, computational chemistry methods are used in a preprocessing step to select potential compounds to be synthesized and tested. While we are far from generating, storing, and virtually screening such a large number of compounds, a fragment space is a possible way of virtually dealing with combinatorial chemistry.

Fragment spaces consist of chemical fragments and a corresponding rule set, which specifies how the fragments can be connected. In some approaches, fragments are derived by cutting compounds using retrosynthetic rules, for example the RECAP rules by Lewell et al.[7] and the application-driven rule sets from Mauser et al.[8] or Degen at al. (BRICS).[9] Fragment spaces are already used for enumerating molecules using physicochemical constraints,[10] scaffold replacement,[11] ligand-based,[12,13] structure-based search,[14] and library design.[15] For compounds derived from fragment-based methods, the synthetic accessibility can theoretically be given, but chemical feasibility often suffers from different drawbacks such as reagent availability or the combination of several fragments where the reaction schemes exclude each other.[6] Moreover, in the case that nonadditivity in the structure−activity relationship to the target is determined, combinatorial synthesis should be applied.[16]

Because a fragment space does not denote an explicit scaffold, it can represent a collection of combinatorial libraries.[6,17] These virtual combinatorial libraries consist of reactants and a uniform reaction scheme. They are often represented using a Markush structure, where a common scaffold (the core) is provided with explicit links to which R-groups (the reagents) can be attached.[18] Having a common core, the resulting molecules differ in the R-groups. In recent works, fragment spaces consisting of combinatorial library collections were used for feature tree similarity searching to circumvent poor feasibility of the products.[6,17] The feature tree descriptor[13,19,20] is a topological reduced graph descriptor[21,22] representing the molecule by a tree structure. A comparison of two molecules is performed by matching their feature tree nodes on each other. On one hand, this matching procedure is more complex than using a distance metric or similarity coefficient on vector representations,[23] such as structural keys or fingerprints.[24] For that reason, substantially longer computing times have to be accepted. On the other hand, the hierarchical comparison strategy for feature trees

* Corresponding author phone: +49 40 428387351; fax: +49 40 428387352; e-mail: rarey@zbh.uni-hamburg.de.
[†] University of Hamburg.
[‡] Boehringer Ingelheim Pharma GmbH & Co. KG.

10.1021/ci900287p
Published on Web 12/18/2009

allows precalculations on fragment level,[13] implying that the products do not have to be explicitly built from the corresponding fragments. This leads to a performance speedup which has a much higher impact on the computation time than the chosen method for a single query-to-product similarity comparison.

When screening combinatorial libraries by similarity to given bioactive molecules as single objective to obtain the best products in the fragment space, two desired properties of the resulting compounds are mostly neglected: First, the compounds may exceed desired physicochemical property ranges although they are most similar to the query. Second, the set of resulting compounds are cherry-picked rather than forming a proper sublibrary. For the task of creating focused sublibraries on the basis of fragment spaces, a novel multiobjective library design method is required to obtain focused libraries with acceptable performance in all objectives. Therefore, scoring functions especially play an essential role in library design.[25]

Several approaches for combinatorial library design exist already, and several overviews and books have been published, e.g., by Agrafiotis,[26] Ghose and Viswanadhan,[27] and Weber.[28] Because of the size of the chemical space, heuristic optimization techniques such as a genetic algorithm, e.g., Brown et al.[29] or Gillet et al.,[25,30–33] or simulated annealing, e.g., Zheng et al.[34] or Agrafiotis,[35] are used to generate the libraries. Further approaches are the ultrafast algorithm by Agrafiotis,[36] a fast exchange algorithm by de Tilleghem et al.,[37] a deterministic greedy procedure by Truchon et al.,[38] or particle swarm optimization by Hartenfeller et al.[15] In most cases, an initial selection will be chosen, which is modified in iterative steps, until the procedure reaches convergence.

The two main strategies for library design are reactant-based and product-based design.[39] In reactant-based design, R-groups are chosen without taking into account the molecules that will be produced. In product-based approaches, the score of the product molecules is used for optimization. Here, usually a full enumeration of the virtual product library must be performed. This is computationally more demanding but more promising in terms of optimizing the properties of a library as a whole.[33]

Moreover, one must distinguish between compound descriptors and properties, which are derived from the structure of the single compound and descriptors that are based on the relationship of a compound with other ligands, proteins, or its environment.[28] Descriptors based on these relationships lead to more complex and expensive calculations. Because the properties in combined scoring functions of the form $\sum_i^n w_i * s_i$, where n is the number of properties, w_i the weight, and s_i the score of a single property, have to be weighted by the user. This is in contrast to other methods[31,35] which search for a set of Pareto optimal solutions where the objectives of the scoring (or fitness) function are not weighted, but the solutions are ranked by their dominance in single objectives over other solutions. In the best case, a solution should dominate all others. Otherwise, if libraries are focused on given bioactive molecules, similarity is the key property and a weighted scoring function is a good choice.

In this paper, we present a novel approach for focused library design named LoFT (Library optimizer using Feature Trees). Like the tool COLIBREE by Hartenfeller et al.,[15]

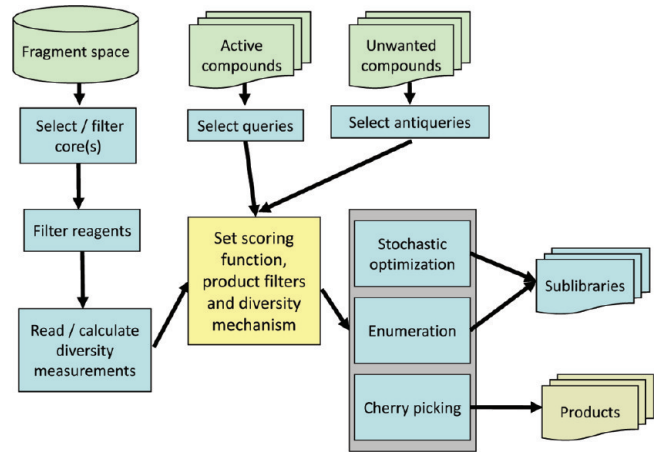


Figure 1. A possible workflow. After a fragment space is read, cores and reagents are selected and filtered by their properties. Furthermore different diversity measurements can be used. But the main aspect of LoFT is the optimization. LoFT suggests complete sublibraries with products according to the different properties and descriptors incorporated into the scoring function, product filters, and the diversity mechanism. For similarity/dissimilarity to given (anti)queries, these compounds are compared to the products using the feature tree descriptor during the optimization.

LoFT uses fragment spaces as input for focused library design to apply the strength of similarity searching in fragment spaces in library design. Moreover, applying the feature tree descriptor in combination with classical physicochemical descriptors, LoFT is able to design focused libraries in a product-based approach, but on fragment level, keeping the products within desired property ranges, with similarity to given bioactive molecules and dissimilarity to unwanted compounds. Unlike reagent-based similarity methods, the core fragments do not have to be explicitly mapped on the query structure. Using the FTrees technology for efficient product-based similarity comparisons, LoFT allows for an automatic screening approach identifying the most promising cores in a first step. In the following, we present the design of LoFT validated by some experiments using several typical drug design scenarios.

METHODS

Scope. LoFT was designed for combinatorial libraries using a core with several links where the reagents can be attached. Given an underlying fragment space, the desired library format, a set of known bioactive compounds, a set of unwanted compounds, a physicochemical property profile, and some settings for library diversity, LoFT suggests focused combinatorial libraries simultaneously optimized to the properties specified (see Figure 1 for a possible workflow). The method is limited to reagents with exactly one link (the link connecting the reagent to the core). In this case, the similarity comparison can be speeded up as described below. For reagent/reagent libraries, LoFT provides a "dummy core", allowing the user to specify two connectable link types for library generation. The resulting sublibraries can be visually inspected in different ways. FragView, a special-purpose viewer for fragment spaces, can be employed for examining the sublibrary in fragment space format, 2Ddraw[40] for browsing the product set, and Spotfire DecisionSite[41] for analyzing the physicochemical properties of the products.

Table 1. Properties Provided by LoFT[a]

property	calculation type	range	core filter	reagent filter	product filter	product scoring
number of links	correctable	[0, ∞], integer	yes	no	no	no
molecular weight	correctable	[0, ∞], double	yes	yes	yes	yes
number of non-hydrogen atoms	correctable	[0, ∞], integer	yes	yes	yes	yes
number of non-hydrogen bonds	correctable	[0, ∞], integer	yes	yes	yes	yes
smallest set of smallest rings	correctable	[0, ∞], integer	yes	yes	yes	yes
number of ring systems	correctable	[0, ∞], integer	yes	yes	yes	yes
number of H-bond acceptors	correctable	[0, ∞], integer	yes	yes	yes	yes
number of H-bond donors	correctable	[0, ∞], integer	yes	yes	yes	yes
number of rotatable bonds	correctable	[0, ∞], integer	yes	yes	yes	yes
maximal path of contiguous rotatable bonds*	correctable	[0, ∞], integer	yes	yes	yes	yes
topological polar surface area*[42]	approximating	[0, ∞], integer	yes	yes	yes	yes
number of EZ stereo centers	approximating	[0, ∞], integer	yes	yes	yes	yes
number of RS stereo centers	approximating	[0, ∞], integer	yes	yes	yes	yes
calculated logP value[65,66]	approximating	[−∞, ∞], double	yes	yes	yes	yes
molar refractivity	approximating	[−∞, ∞], double	yes	yes	yes	yes
polar surface area	approximating	[−∞, ∞], double	yes	yes	yes	yes
link type	product molecule has no links	[0, ∞], integer	yes	yes	no	no
inclusion SMARTS	nonapproximating	[0, 1], integer	yes	yes	yes, fragments must be combined	no
exclusion SMARTS	nonapproximating	[0, 1], integer	yes	yes	yes, fragments must be combined	no
user defined properties* (strictly additive)	correctable	[−∞, ∞], integer or double	yes	yes	yes	yes

[a] The table describes for each property how the values are derived (calculation type), for which filters they can be used and whether they can contribute to the product scoring. The range of the properties is stated as theoretical possible values, the number of links would rarely exceed a value of four for example. Except for the properties marked by an asterisk (*), the properties are also used by FlexNovo,[14] FragView and FragEnum[10] for filtering fragment spaces.

Descriptors and Properties. LoFT uses several descriptors for scoring and filtering. The descriptors are listed in Table 1. From the computational point of view, we distinguish between three types of descriptors depending on their ability to derive product descriptor values from fragment descriptor values:

• *Correctable additive descriptors* have the property that the descriptor value of the product can be calculated from the values of its fragments. The first group contains product descriptors achieved from adding the fragment descriptors (see also Table 1). Simple examples of such descriptors are molecular weight or the number of heavy atoms. For the maximal-path-of-contiguous-rotatable-bonds descriptor, we store the maximal path within a fragment and the maximal path starting from each link as well as the path length between two link types. In contrast to the other descriptors, these values must be updated upon fragment linking in order to calculate the maximal value. In some cases, the descriptor values must be corrected by analyzing the environment of the connected links. If the environment of the new adjacent atoms changes after connection of two links, e.g., by a change of the protonation state, bond, or atom type, the descriptor values would become incorrect if they were simply added. LoFT resolves this problem with an optional preprocessing step. Each reagent is connected to the core, and the descriptors are calculated for the reagent-core combination. Afterward, the core descriptors are subtracted from the reagent-core descriptors and stored as the new reagent descriptors (see Figure 2). Generally, a descriptor cannot be corrected, if the feature from which the descriptor is derived extends over more than two fragments. For example, this is the case when a descriptor includes the presence or absence of large molecular substructures and a small core is used.

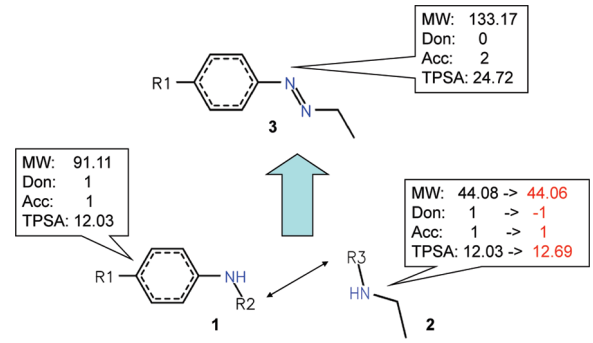


Figure 2. An example for the descriptor correction, showing a core **1** and a reagent **2**. For both fragments, exemplarily some physicochemical properties of the fragment descriptor (fd) are shown. In this case, combining both fragments results in a double bond. Subtracting the core fd from the fd of the newly combined fragment **3**, we get the reagent fd as if the reagent was connected to the core (red values). The product fd finally is the sum of the core fd and the fd of the reagents connected to it.

• For *approximating descriptors*, the descriptor values can be derived from fragment descriptors, but the values might not be correct. The correction as described above (Figure 2) might fail but will still be a better approximation in most cases than a simple addition. In general, the feature tree descriptor belongs to this group, but usually there are slight differences between molecule and fragment comparison (see Figure 3). Also further adjustments were made to cope with the problem of fragment-based similarity comparison which will be described later on. Another example for an approximating descriptor is the topological surface area (TPSA),[42] where the direct environment of each atom is examined. Using the correction mechanism only for dummy cores (see above) may still result in deviations.

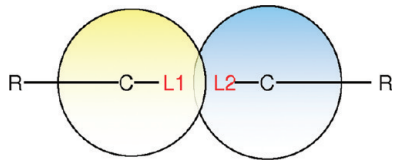


Figure 3. The algorithm for generating feature tree descriptor for fragments combines the shape descriptors of the nodes adjacent to the link nodes, as depicted. The contribution of the small region in which the van der Waals spheres of the atoms adjacent to the link overlap is included twice. This leads to slightly different results in contrast to a molecule to molecule comparison.

• For the calculation of *nonapproximating descriptors*, the fragments must be explicitly combined. Important examples for this type are lists of inclusion or exclusion substructure patterns used for filtering. The usage of such descriptors increases the run time of LoFT by a high factor. To avoid this, substructure matching can be evaluated and stored as binary values for each fragment. During optimization, these values are added and typical patterns such as "carboxyl groups only allowed for one reagent" can be easily simulated.

Besides these internal descriptors, LoFT allows the usage of user-defined additive descriptors as integer or double values, e.g., the reactant price or availability.

Similarity Comparison Using the Feature Tree Descriptor. To achieve products similar to the query and diverse reagents, the feature tree descriptor[13,19,20] is used. A feature tree abstracts the exact molecular topology by an undirected and unrooted tree structure instead of employing a bit string or vector for molecule representation. By cutting the acyclic nonterminal bonds, condensing simple cycles to single nodes, and applying special cutting rules to complex cycles, the molecular graph is mapped on a feature tree. Subsequently, the tree nodes are labeled with the sterical and physicochemical properties of the corresponding "building blocks" of the molecular graph, the "features". Because the nodes of the tree are connected in the same way as these building blocks, the overall arrangement of functional groups is retained. Figure 4 depicts Imatinib (Gleevec), a kinase inhibitor, and its corresponding feature tree.

An algorithm for the comparison of two feature trees is the match search algorithm,[13,19,20] aligning the trees by mapping the nodes onto each other. Therefore, the match search algorithm benefits from the fact that by removing a single edge, a tree disaggregates into two independent components (rooted subtrees). Searching an optimal alignment, the matching of two subtrees depends only on the matching of the subsequent smaller subtrees (see Figure 5). This allows the application of a dynamic programming scheme.[43] The algorithm is recursively called for all subtree combinations without the current root nodes and the returned similarity values are stored in a dynamic programming matrix avoiding multiple calculations of the same subtree combinations. A maximum weighted bipartite matching algorithm finally assigns subtrees to each other, resulting in the best alignment and therefore the highest similarity value. Figure 6 illustrates an alignment of imatinib and the best hit from screening the Bionet data set.[44]

Moreover, because of the modular structure of the feature tree descriptor, it can be applied directly to molecular fragments.[13] The open valence of a molecular fragment is mimicked by a linker node. Employing the dynamic programming matrix, we preset the corresponding cells of the fragment link edges by the comparison values of the fragments to be connected. For that reason, the fragments do not have to be explicitly combined. In the case of combinatorial library design, query-to-product similarity comparisons will be evaluated. The query-to-product comparison is calculated in two steps: First, the similarity values of query and reagents (query-to-reagent comparison) are evaluated, and second, query and core are compared (query-to-core comparison). In the latter case, the cells of the dynamic programming matrix which correspond to the core link edges are preset by the similarity values of the corresponding link edges of the reagent (see Figure 7).

To focus a combinatorial library toward known binding motifs, LoFT allows the user to specify multiple molecules as queries, guiding the optimization process by highest possible similarity to them. Furthermore, it also allows for so-called antiqueries, to which highest dissimilarity is preferred. For a single (anti)-query-to-product comparison, the matrix of a query-to-core comparison is preset by the similarity values of a query-to-reagent comparison. If only reagents providing a single link are allowed, the comparison can be accelerated by reusing precomputed values of query-to-reagent comparisons. LoFT employs a global matrix (see Figure 8) as already applied for the SwiFT extension of the match search algorithm,[45] to store the already computed values of each query edge to reagent link edge combination.

Another modification allows the reuse of computed values within the query/core matrix. Because of the focused library enumeration process, a query-to-product comparison only differs by the exchange of a single reagent. The cells, which depend only on the remaining reagents, do not have to be computed again. In the case of a core providing two links, 50% of the cells can be reused (see Figure 9).

Moreover, a size filter was incorporated, starting the similarity comparison only if the number of heavy atoms of the product is in a certain range compared to the number of heavy atoms of the query. This reduces the run time of the optimization as shown later (see Figure 18), because not all similarity comparisons have to be computed.

Filtering. Filter rules can be applied to reagents and cores and therefore shrink the combinatorial space by removing unwanted fragments. The rules can also be disposed to the products generated during the optimization process. If a product fails the filter criterion, it is scored with zero. To allow a simple use of complex filter rules, logical expressions can be stated using AND, OR, NOT and parentheses. For each allowed property (see Table 1), a minimum and a maximum value is declared.

The logical language was further extended by the use of the term "VIOLATE[x]{expr_1,..., expr_n}" to allow that x of the given n expressions ($x \leq n$) fail. By this additional expression, for example the frequently used rule of five by Lipinski et al.[46] problems of oral absorption can be incorporated easily as shown in the following example:

VIOLATE[1]{ACC[0,5], DON[0, 10], MW[0, 500], CLOGP[$-\infty$,5]}

Scoring. For scoring a single property, LoFT uses desirability functions as proposed by Derringer et al.[47] to fit properties into the interval [0, 1]. This mapping is also used by Le Bailly de Tilleghem et al.[37] In LoFT, the mapping is

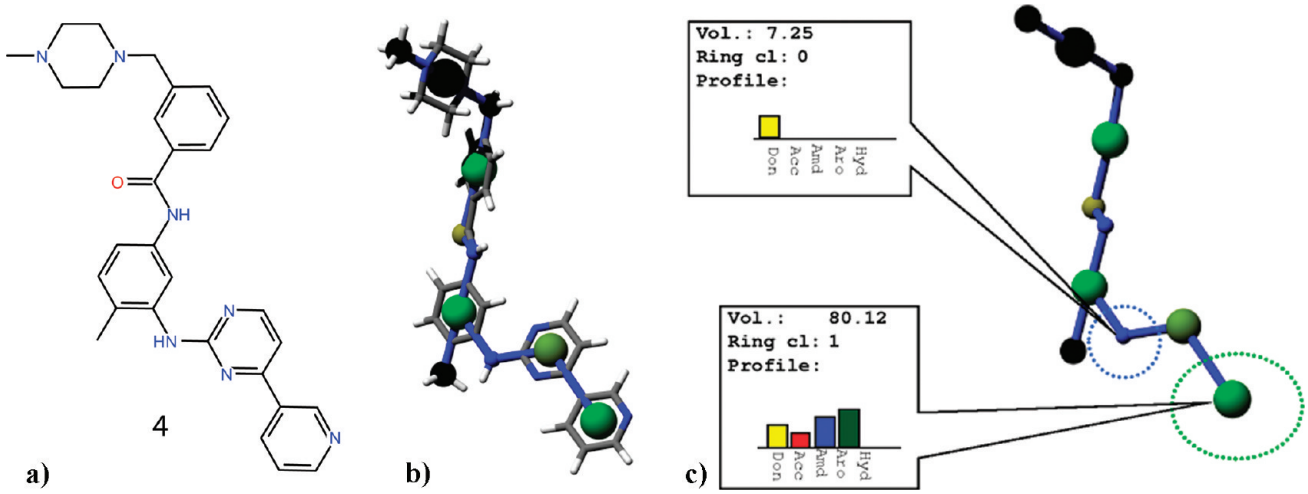


Figure 4. The molecular graph of imatinib (Gleevec) (**4**), a kinase inhibitor depicts (a) the molecular graph and (b) molecule and feature tree (for better understanding, the feature tree is shown with the same 3D coordinates) and (c) shows the feature tree and exemplarily the profiles of two nodes. (a) The structure was generated with 2Ddraw,[40] and (b, c) with FlexV.[67]

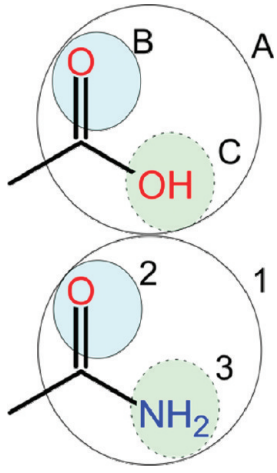


Figure 5. The optimal alignment of the two exemplary subtrees A and 1 depends only on the matching of the root nodes (carbons) and on the matching of all combinations of smaller subtrees starting from the root nodes (B-2/C-3 and B-3/C-2).

done by setting four points of a trapezoid and defining the behavior of the trapezoid shoulder (see Figure 10). This way a differentiation between highly appreciated, acceptable, and unwanted properties can be achieved.

The score assessing the feature tree similarity is limited to the range [0, 1]. To avoid result sets where the products are too similar to the given queries, a minimum and maximum similarity can be defined. Values beyond these thresholds are set to zero.

In order to allow multiple query molecules, consensus scoring is realized, employing the maximum similarity or the sum of the similarity values divided by the number of queries (see Table 2a and 2b). Additionally, molecules can be defined as antiqueries to generate libraries which differ from them. Here, the dissimilarity value is used for scoring (see Table 2c and 2d). Again, thresholds for a minimum and maximum dissimilarity to the antiqueries can be defined. A product is scored by the weighted sum over the property scores (Table 2e). To score a single reagent, the sum of the product scores is divided by the number of products which contain it (Table 2g). The overall score for the library is then calculated either by the arithmetic, the geometric, or the quadratic mean (see Table 2i,2j, and 2k, respectively) over all reagent scores. Alternatively, the maximum or minimum reagent score can be applied.

The selected scoring mode can drastically change the composition of the generated sublibrary. Employing the maximum score might result in sublibraries, where few products have a very good score, while the other products fail certain criteria resulting in a lower score. On the other hand, using the minimum reagent score might result in a sublibrary where the products have similar scores but lack products with a very high score.

The scoring scheme can be used to sort the reagents before starting the optimization process. Afterward the maximum number of reagents for each core link can be set, shrinking huge fragment spaces, and makes them applicable to an enumerator or cherry picking algorithm. Additionally, the best scoring reagents can be chosen to generate a promising start sublibrary for optimization as an alternative to a random reagent selection.

Diversity Considerations within a Sublibrary. As the optimization process is guided by similarity to given molecules, the resulting focused libraries (sublibraries) will consist of similar reagents and products. To achieve a more diverse sublibrary, LoFT offers several diversity mechanisms. First, the same clustering module as used by FTrees Release 2[19,48] was incorporated. It provides a single and a complete linkage clustering algorithm (see, for example, the review by Downs et al.[49]). To calculate the distance matrix for clustering, LoFT provides the feature tree descriptor as well as the maximum common subgraph (MCS, see Raymond et al.[50] for a review). The resulting cluster IDs can be assigned to the reagents and, for each core link, the maximum number of reagents from one cluster can be specified for the optimization algorithms. If this value is exceeded, the corresponding reagents are penalized, depending on the number of reagents from the same cluster. Therefore, the user specifies the value where a penalty is applied first (point A in Figure 10) and a value (point B in Figure 10) where the maximum penalty (a value between 0 and 1) is deployed. The penalty increases uniformly for numbers of reagents from the same cluster lying between A and B.

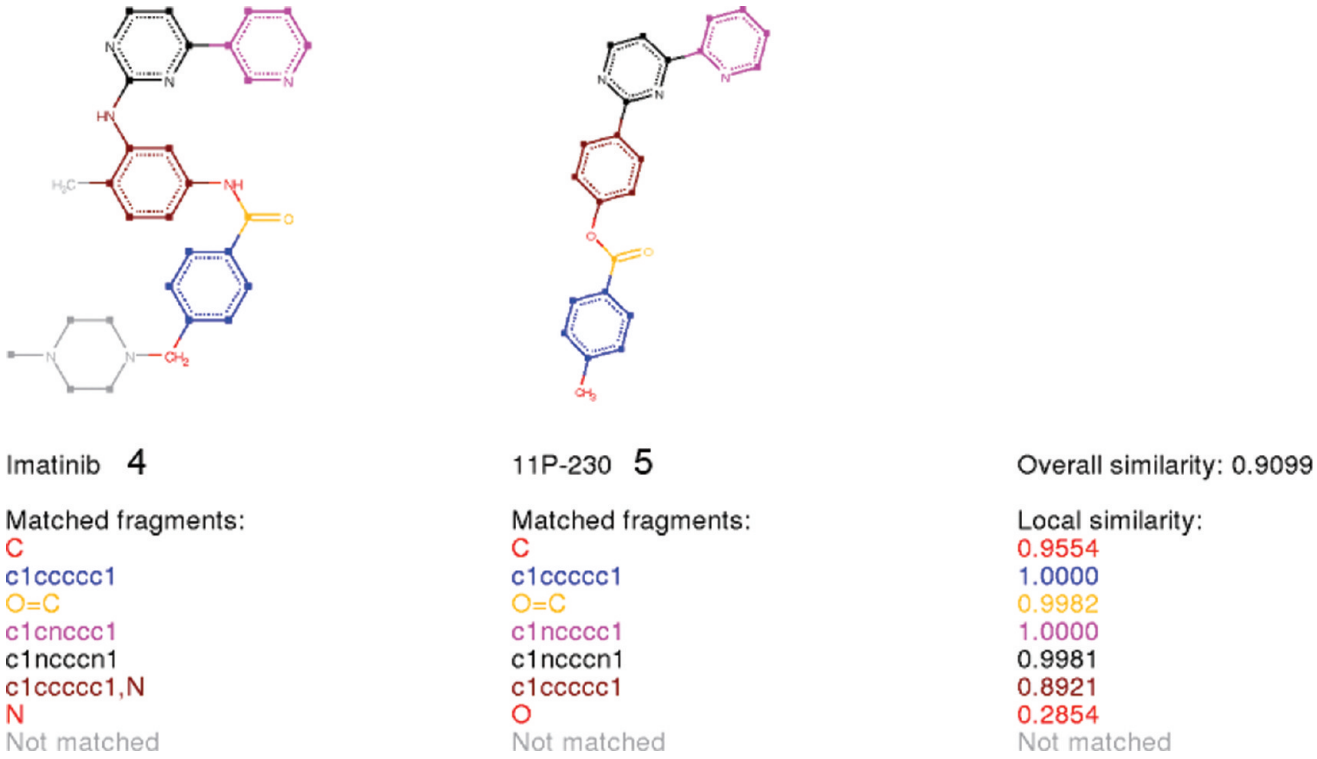


Figure 6. Matching of imatinib (**4**) and the most similar hit **5** from screening the Bionet set[44] with the free accessible FTrees Web Interface.[68] The figure shows the alignment of the molecular building blocks and was generated with FTreesXL.[69]

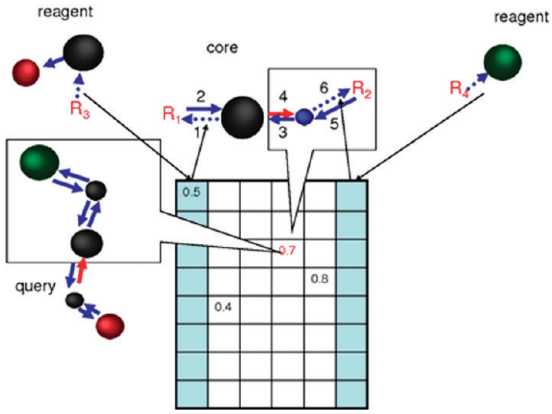


Figure 7. A query-to-core comparison. Each nondirected edge is represented by two directed edges, and a split of the nondirected edge results in two directed subtrees. In the example above, the split was set including the directed edges colored in red. The directed edges are arbitrarily numbered, and R1 connects to R3, and R2 to R4, respectively. The similarity of the corresponding subtree combination is written to the marked cell. For that reason, the cells (light blue) corresponding to the link edges (dotted arrows) of the core feature tree are preset with the similarity values of the query-to-reagent comparisons in the dynamic programming matrix for query and core.

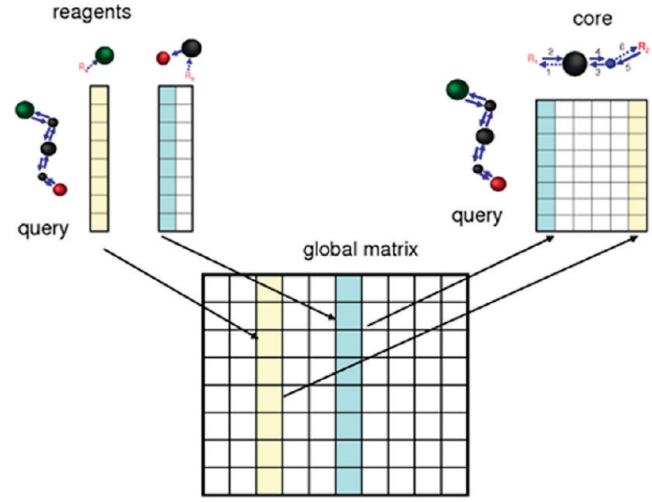


Figure 8. Use of a global matrix for query-to-reagent comparisons. The similarity values for each query edge to reagent link edge combination are stored in the global matrix. The matrix can be accessed via query edge id and the unique reagent id to preset the corresponding cells of the query-to-core comparison. In this example, the yellow with respect to blue cells of the query-to-reagent comparisons are first computed and then stored in the global matrix and retrieved by a query-to-product comparison.

Furthermore, the calculated dissimilarity values (1 − similarity value) can be used directly for diversity measurement. In this case, either the minimum distance of two reagents assigned to the same core link or the average distance of all reagents assigned to the same core link can be exploited. As in the above-mentioned case of using cluster IDs, a user-defined value can be applied for penalizing critical solutions where the maximum distance threshold is exceeded. Note that the score will not become negative: $\text{pscore}_i = \max(0, \text{pscore}_i - \text{pen}_{\text{cluster}} - \text{pen}_{\text{min}} - \text{pen}_{\text{avg}})$. Therefore, the maximum number of reagents from the same cluster, the minimum distance of two reagents, and the average distance can be applied in parallel. For huge fragment spaces, the distance values between reagents are computed on demand; otherwise the triangular diversity matrix can be applied for storing already computed values.

Diversity Considerations between Sublibraries. During stochastic optimization, subsequent sublibraries might differ only by one reagent. In case that the *n* best sublibraries are stored, the minimum number of different reagents between two sublibraries can be specified. A newly generated sublibrary will be compared to the list of already stored

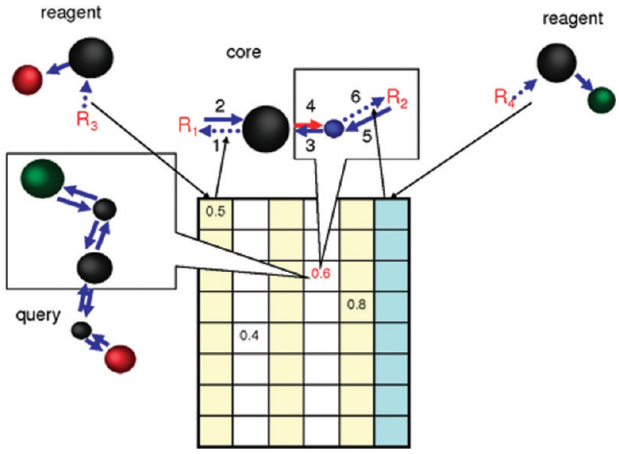


Figure 9. Dynamic programming matrix for another query-to-core comparison following the one from Figure 7. Enumerating a sublibrary, only a single reagent is exchanged in each step. Consequently, only the dynamic programming matrix cells of core edges depending on the new reagent have to be recalculated. Here, the only change is the substitution of the reagent on the right. Therefore, the values of the yellow cells can be reused; the blue cells have to be preset and the white cells must be a newly computed for this link combination. Then the match-search is called for all query-edge-to-core-edge combinations.

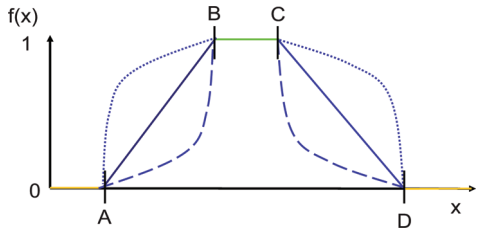


Figure 10. Mapping property values into the range [0, 1]. For each property, the points A, B, C, and D can be set. If, for example, A and B as well as C and D are set to the same values, a 0/1 decision will be computed. The function can also slowly (for the left shoulder: $(x - A)^{1/2}/(B - A)^{1/2}$ with respect to $(D - x)^{1/2}/(D - C)^{1/2}$ for the right shoulder), uniformly $((x - A)/(B - A)$ with respect to $(D - x)/(D - C))$, or quickly $((x - A)^2/(B - A)^2$ with respect to $(D - x)^2/(D - C)^2)$ decrease to undesirable response values.

solutions. If there is a sublibrary with a better score and less different reagents as the value specified, the new solution is discarded. Otherwise the new sublibrary is inserted in the list and all conflicting solutions with too many identical reagents and a lower score are removed. Note that this is a heuristic approach in which the resulting set of sublibraries might depend on the order in which the sublibraries were generated.

For cherry picking, the maximum similarity can be specified for each pair of molecules in the list containing the *n* best products. The same approach as applied for sublibraries is used: Only a product, for which no other conflicting (too similar) product has a higher score, will be inserted. All conflicting products with a lower score are removed from the list. A similar approach has already been integrated in FTrees-FS[13] and FlexNovo.[14]

Optimization Algorithms. LoFT provides several algorithms for traversing the chemical search space, trying to maximize the sublibrary score applying the scoring function described above. In the following, the different approaches are summarized briefly:

• *Simulated annealing*[51] belongs to the most widely used stochastic optimization techniques. In analogy to a cooling process, a temperature value starting with a high value is gradually decreasing during the process. If the temperature is high, the algorithm allows the change toward solutions with (far) lower score with high probability. While the temperature is decreasing, the acceptance probability for lower scoring solutions is reduced until only solutions with higher score will be accepted.

• *Threshold acceptance*[52] accepts lower scores, if ΔE, the energy of the current library minus the energy of the new library, is within a certain threshold.

• *Great deluge*[53] accepts lower scores, as long as they are about a predefined value, the so-called "water level". This value is increased during the optimization process.

• *Hill climbing* allows only solutions with better scores without stepping back. Therefore, this algorithm allows a fast evaluation but is likely to end in a local maximum.

In general, the library design algorithms exchange a single reagent and evaluate the score. Only those products containing the new reagent have to be rescored. The reagent to be exchanged is selected either randomly or the reagent with the worst score is chosen. To avoid the repeated substitution of the same reagent in the latter case, the worst reagent is chosen only every *i*th time, doing a random exchange otherwise. In all cases, a new step is accepted, if the resulting sublibrary gets a better score. The approaches differ in the tolerance criterion which defines how a degradation of the solution score is handled. All library design algorithms store the *n* best sublibraries seen.

• An *enumerator* was foremost implemented for validation. It enumerates all possible solutions and saves the *n* best sublibraries. Because of combinatorial explosion, it is only applicable to very small data sets.

• These library design algorithms are accomplished by a *cherry picking* algorithm, which selects the *n* best products according to the defined scoring function. Therefore, all possible products are generated and scored. In this case for performance reasons shrinking huge fragment spaces to the best scoring fragments is highly recommended. The cherry picking algorithm tests all products independently without generating a sublibrary.

Multiple Core Evaluation. Instead of choosing a single core for sublibrary optimization, a set of fragments providing the same links can be selected as cores. Equally to the consideration of multiple queries, the cores can be weighted individually in order to control their influence on the sublibrary score.

RESULTS AND DISCUSSION

Recently, feature tree similarity searching in fragment spaces built on synthesis protocols for combinatorial libraries was successfully applied.[6,17] One advantage of virtual screening in fragment spaces built from combinatorial libraries is that the synthesis protocols of the hits detected are given. Furthermore, these hits can be easily explored making use of combinatorial synthesis. But to benefit from this option, small sublibraries have to be designed that contain 10 × 10 reagents, for example, which lead to 100 products that should be all more or less similar to the query. This task can be taken over by LoFT using feature tree

Table 2. Different Scoring Functions Incorporated in LoFT

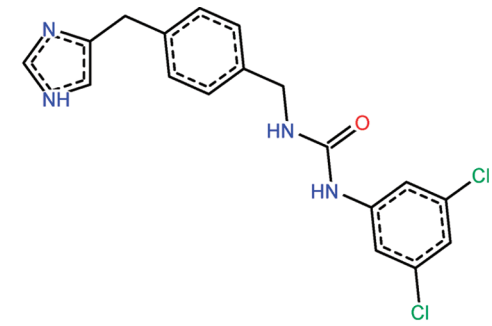
formula		description
$\mathrm{qsim}(p) = \max(\mathrm{sim}(i,p) \mid i \in q)$	(a)	Maximum similarity of product p to any query molecule i from query set q.
$\mathrm{qsim}(p) = \frac{1}{\lvert q\rvert}\sum_{i=1}^{\lvert q\rvert}\mathrm{sim}(i,p)$	(b)	Sum of the similarity values of product p compared to all queries q divided by the number of queries.
$\mathrm{asim}(p) = 1 - \max(\mathrm{sim}(i,p) \mid i \in a)$	(c)	Maximum dissimilarity (1-similarity) of product p to any antiquery molecule i from the antiquery set a.
$\mathrm{asim} = \frac{1}{\lvert a\rvert}\sum_{i=1}^{\lvert a\rvert}(1 - sim_i)$	(d)	Sum of the dissimilarity values of product p compared to all antiqueries a divided by the number of antiqueries.
$\mathrm{pscore}(p) = \sum_{i=1}^{n} w_i s_i$	(e)	Weighted sum over all property scores to score a product p where n is the number of properties, w_i the weight and s_i the score of a single property.
$P(r) = \{\text{products } p \mid p \text{ contains } r\}$	(f)	Set of all products which contain reagent r.
$\mathrm{rscore}(r) = \frac{1}{\lvert P(r)\rvert}\sum_{p\in P}\mathrm{pscore}(p)$	(g)	The score of a reagent r is the sum of the product scores divided by the number of products. Only products which contain r are taken into account.
$R(l) = \{\text{reagents } r \mid l \text{ contains } r\}$	(h)	All reagents r which are part of sublibrary l.
$\mathrm{libscore}(l) = \frac{1}{\lvert R(l)\rvert}\sum_{r\in R}\mathrm{rscore}(r)$	(i)	Arithmetic mean score over the reagent scores of sublibrary l.
$\mathrm{libscore}(l) = (\prod_{r\in R}\mathrm{rscore}(r))^{1/\lvert R\rvert}$	(j)	Geometric mean score over the reagent scores of sublibrary l.
$\mathrm{libscore}(l) = \sqrt{\frac{1}{\lvert R(l)\rvert}\sum_{r\in R}\mathrm{rscore}(r)^2}$	(k)	Quadratic mean score over the reagent scores of sublibrary l.

similarity. Concentrating on similarity alone may result in sublibraries with unfavorable physicochemical properties (depending on the project's needs, the queries, and the library scaffolds). For example, it might be good to design a library with products similar to a query with low TPSA, because the compounds searched for have to penetrate the blood−brain barrier.[42,54] Finally, if a broader exploration of the initial fragment search hit is desired, it can be helpful to limit the similarity among the products within the designed library.

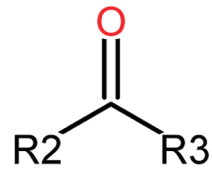
Having this application scenario in mind, we planned the following validation study. Starting with a random reagent selection, we want to show that LoFT is able to design sublibraries, selecting similar reagents as that for FTrees-FS in a cherry picking approach. Moreover, we want to show that the sublibraries pass user-defined criteria with respect to physicochemical properties while high similarity to given query compounds is obtained. Finally, certain dissimilarity between the products in a library can be achieved, if desired.

Experimental Process. We use three different combinatorial libraries from the freely available Knowledge Space[55] to target the histamine H3 receptor, serotonin 5-HT_{2A} receptor, and cyclin-dependent protein kinase 2 (CDK2) and use known bioactive molecules from the literature to generate sublibraries with products similar to these query compounds. To increase the virtual combinatorial libraries, we added reagents from vendor catalogues, for example, standard amines and carboxylic acids.

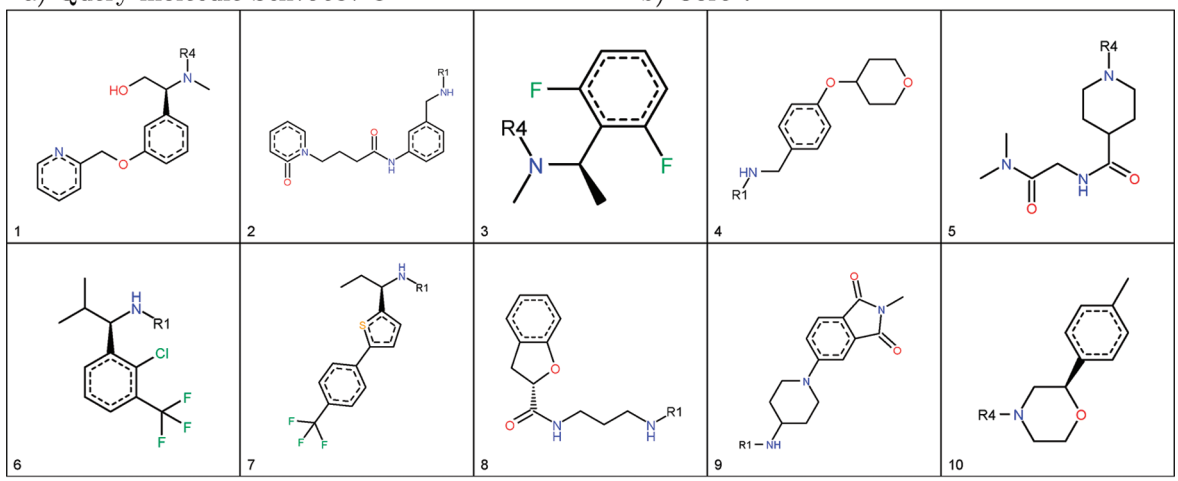
To evaluate LoFT, we generate sublibraries of size 5×5 and $4 \times 4 \times 4$, respectively, with different requirements, which are suitable for visual inspection. First, we generate a random sublibrary which is also the starting sublibrary of the optimization process. Second, to compare LoFT with FTrees-FS, an FTrees-FS search is performed and the 10 and 12, respectively, highest scoring reagents are selected. The resulting sublibrary is scored by LoFT. Afterward, a LoFT-simulated annealing run with 200 000 iterations is



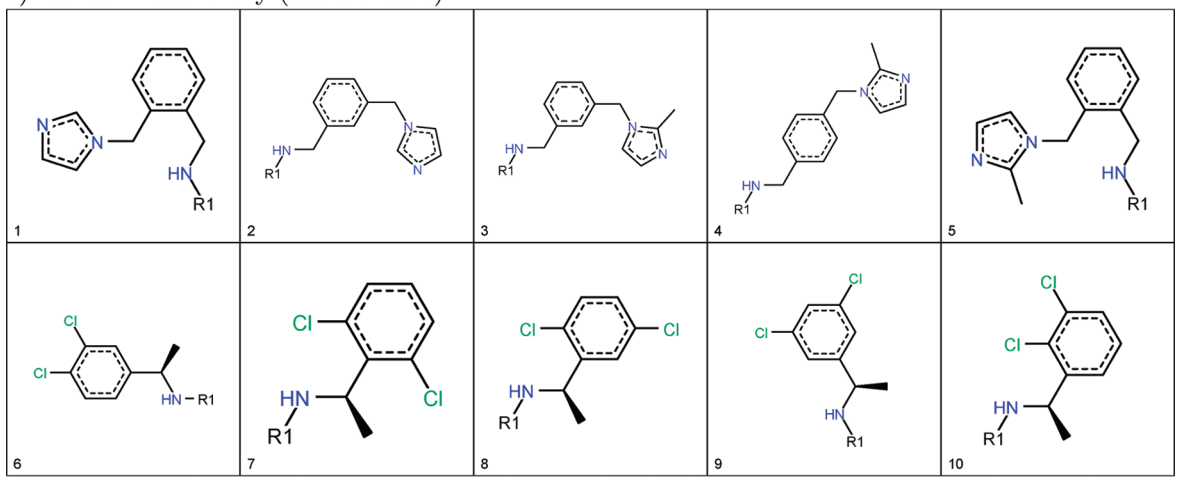
a) Query molecule Sch79687 **6**



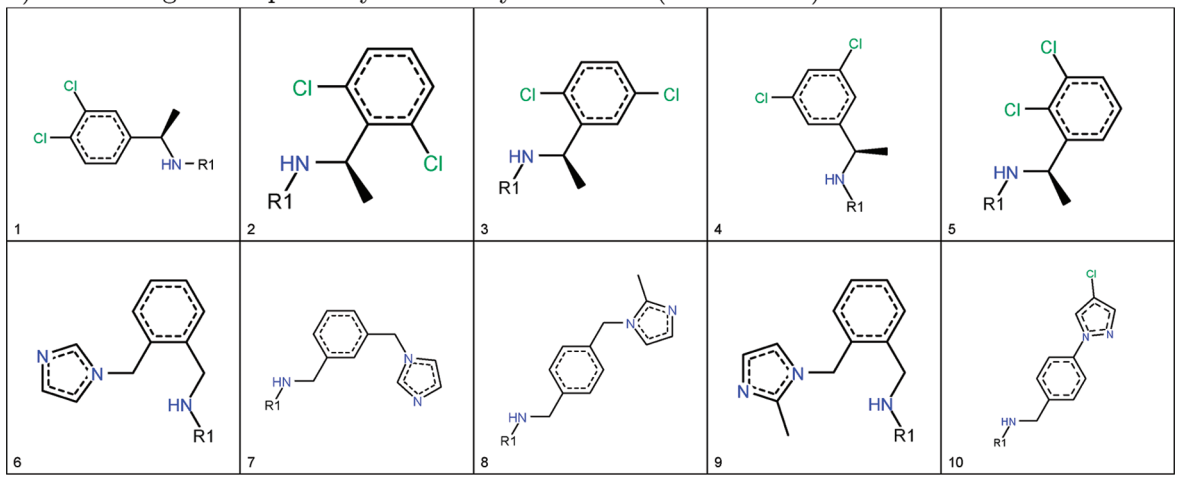
b) Core **7**



c) Random sublibrary (score 0.6933)

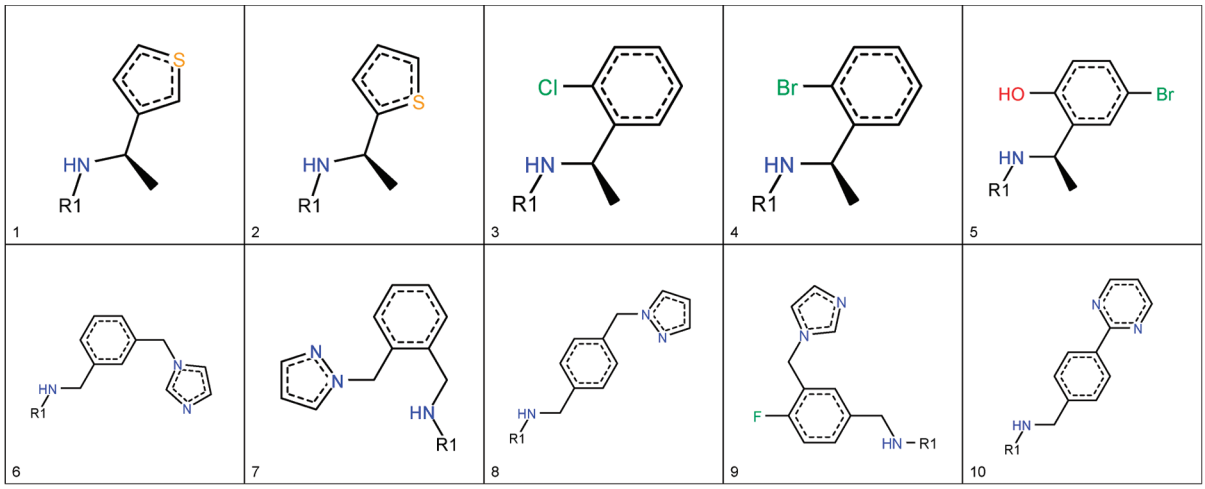


d) Best 5 reagents respectively selected by FTrees-FS (score 0.9311)

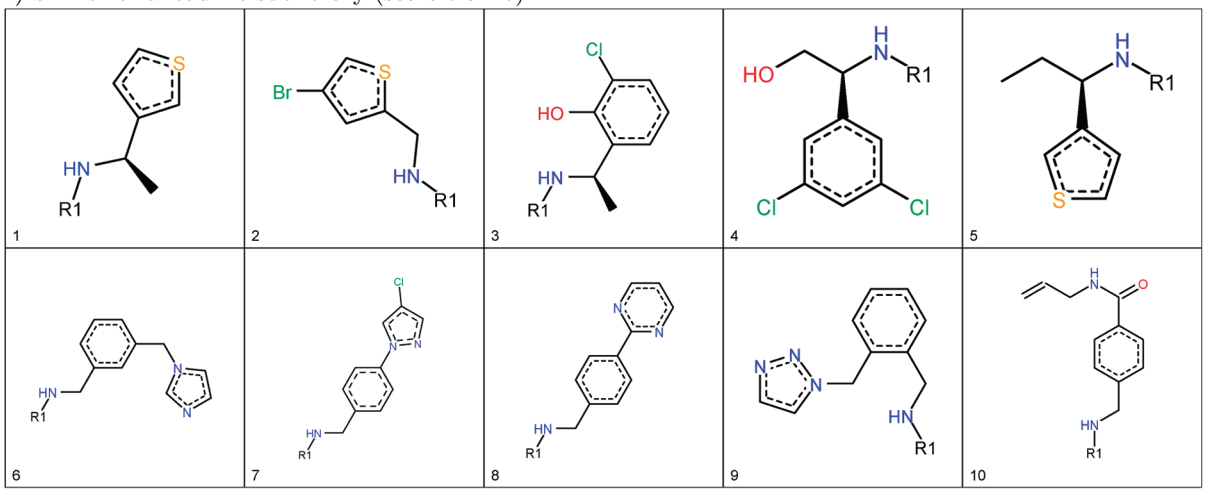


e) Similar sublibrary (score 0.9282)

Figure 11. Continued.



f) Similar and leadlike sublibrary (score 0.9120)



g) Similar, leadlike, and diverse sublibrary (score 0.9012)

Figure 11. Different 5 × 5 sublibraries for the query molecule **6** and core **7** depicted in Figure 11a and 11b, respectively. Except for the comparison with the randomly chosen library, the similarity decreases with each additional criterion. Reagents 1−5 link to R2 of the core and reagents 6−10 to R3.

performed using feature trees similarity to the query as a single criterion. The applied parameters are shown in the Supporting Information. Especially, the results shown are computed using an identical seed of 1. Using other seeds results in slightly different selections. Scores of optimization runs with different seeds are shown in the Supporting Information as well. To achieve products with physicochemical properties within certain ranges, we do another simulated annealing run with a leadlike filter based on similar criteria like those published by Oprea et al.[56] For the CDK2 case study, we incorporate these values (points B and C; in parentheses point A and accordingly D is shown, see Figure 10) in the scoring function weighted with 0.05 each and weight the similarity to the query with 0.7:

- Molecular weight ≤ 450 (600)
- Number of rings ≤ 4 (6)
- (−6) −3.5 ≤ clogP ≤ 4.5 (7)
- Number of donors ≤5 (8)
- Number of acceptors ≤ 8 (12)
- Number of rotatable bonds ≤ 10 (15)

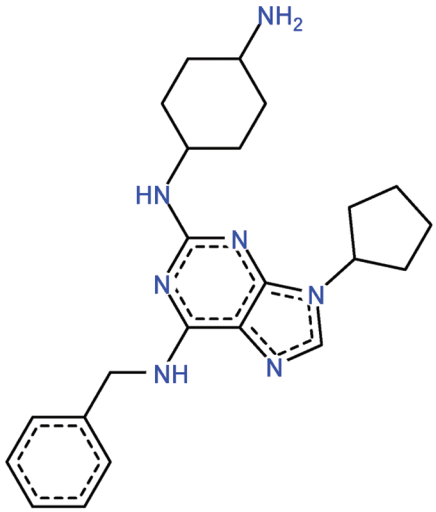
To achieve leadlike and diverse products with maximum similarity to the query compound, the reagents were clustered by a complete linkage clustering using a maximum similarity of 0.8 in a preprocessing step. During optimization, only one reagent from each cluster is allowed in the sublibrary; otherwise, the reagent scores are penalized with −0.2.

At last, we investigate how LoFT performs using different sublibrary sizes and show the similarity values of 50 × 50 sublibraries for 5-HT_{2A} in detail.

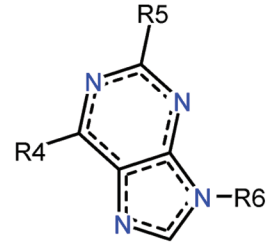
Histamine H3 Receptor. The histamine H3 receptor controls the release of neurotransmitters such as serotonin, acetylcholine, and histamine itself. H3 receptor antagonists have potential, for example, to be used as antiobesity drugs or for epilepsy treatment.[57] We use a known antagonist from the literature[57] (**6**, see Figure 11a) and optimize a urea library with 32 585 reagents and core **7** (Figure 11b).[58] All reagents can connect to both links of the symmetrical core. A randomly distributed sublibrary is illustrated in Figure 11c. Using FTrees-FS as a gold-standard to find similarity products, we depict in Figure 11d the five best reagents for each core link selected by FTrees-FS. The result of a LoFT optimization with similarity to the query as single objective is shown in Figure 11e. Both designed sublibraries show a reasonable similarity but exceed leadlike criteria (see Figure 14) and consist of very similar reagents, e.g., differing only by their substitution patterns. Using the leadlike filter, we obtain a sublibrary in which the properties are distributed within the leadlike ranges (see Figure 11f and Figure 14).

Moreover, when only one reagent from each cluster is allowed, the reagent selection becomes more diverse (see Figure 11g).

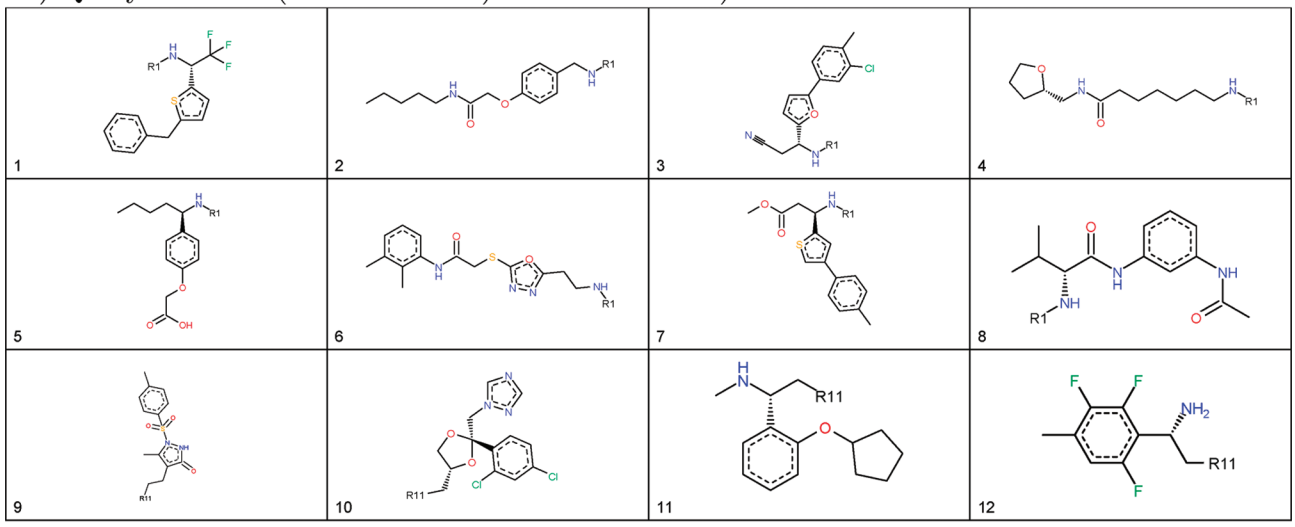
Cyclin-Dependent Kinase 2 (CDK2). CDK2 plays an important role at two stages of the cell cycle[59] and is therefore a well-known target for cancer treatment.[60] We use a known inhibitor from literature[61] (**8**, Figure 12a). Figure 12 shows the 4 × 4 × 4 sublibraries of this case study. We use a purine library with core **9** (see Figure 12b)[62] and 29 649 reagents (23 486 reagents connectable to link both R4 and



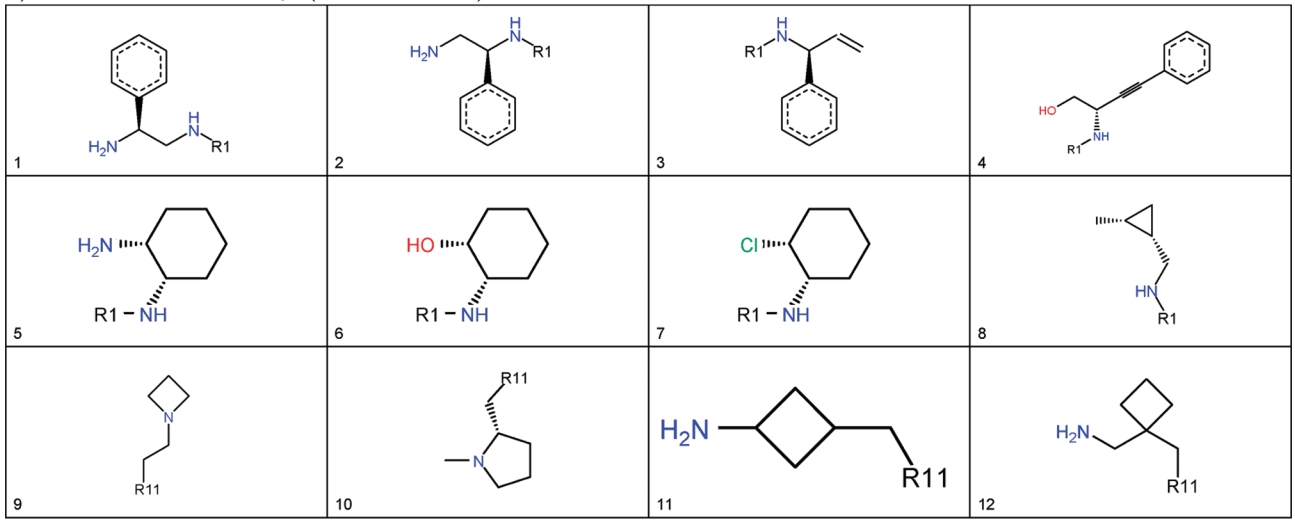
a) Query molecule (ZINC03591113) **8**



b) Core **9**

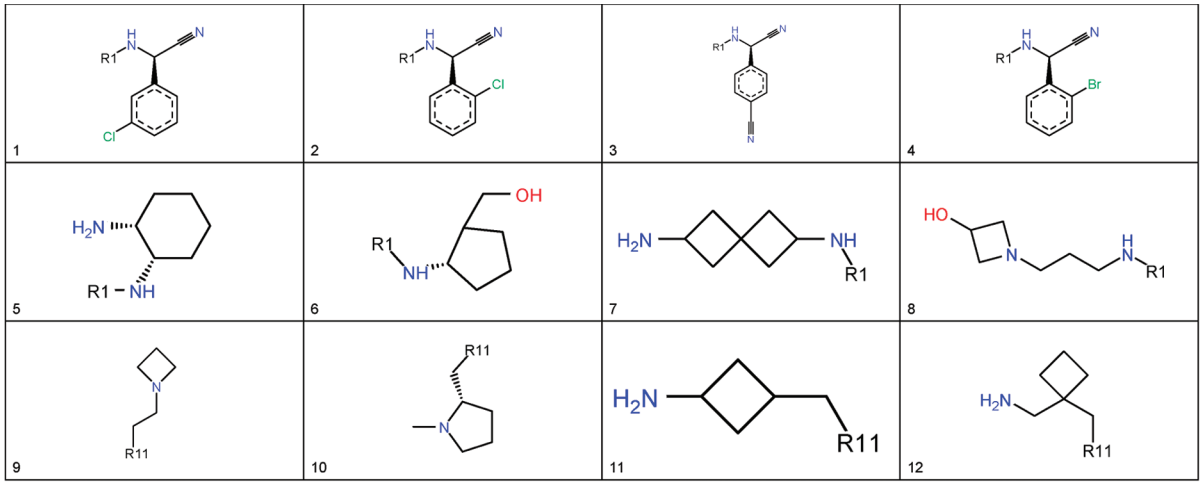


c) Random sublibrary (score 0.0000)

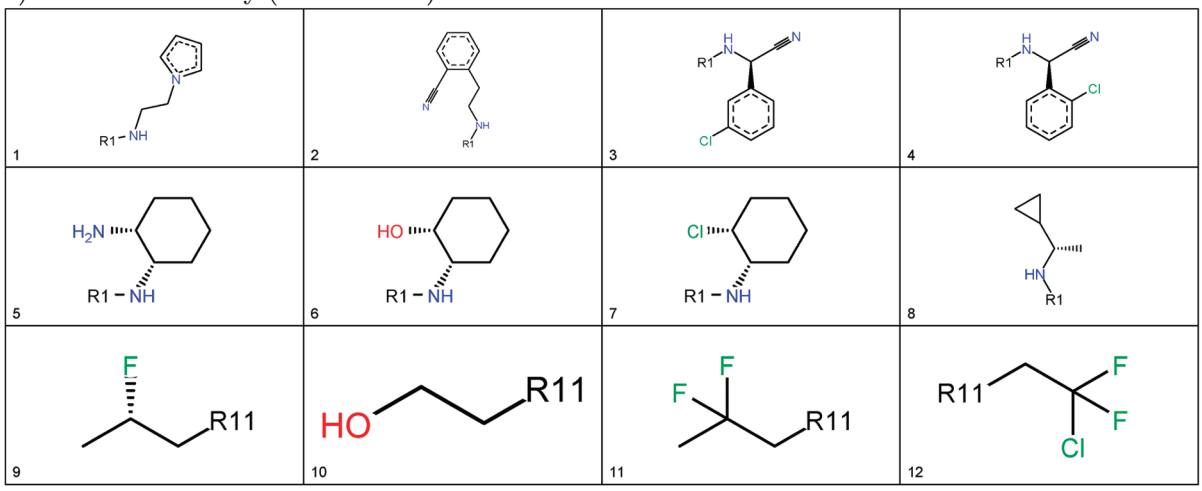


d) Best 4 reagents respectively selected by FTrees-FS (score 0.9507)

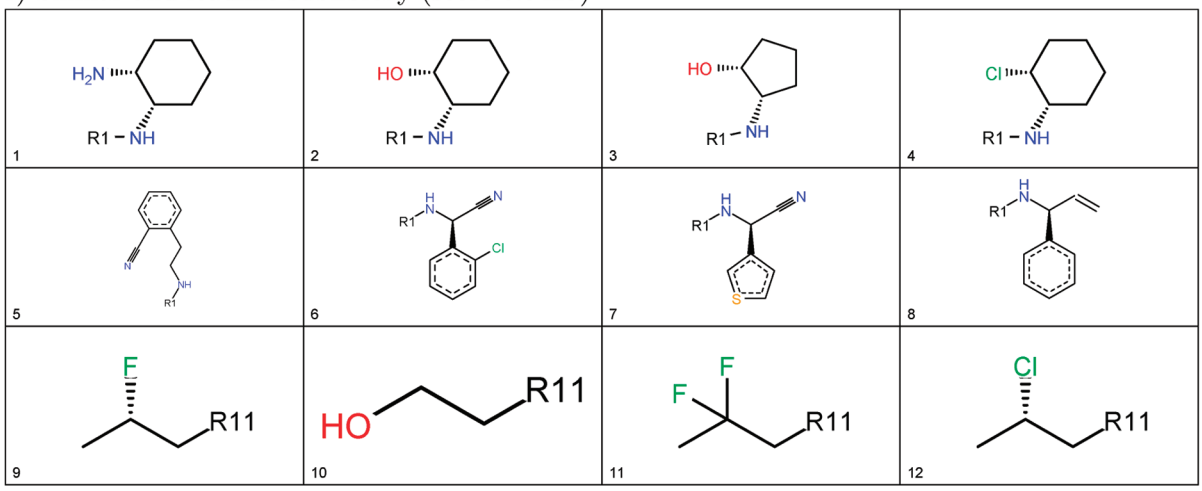
Figure 12. Continued.



e) Similar sublibrary (score 0.9634)



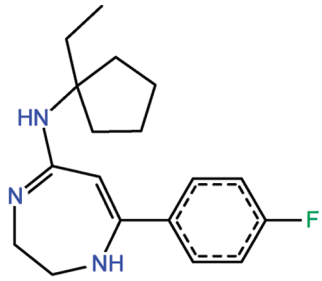
f) Similar and leadlike sublibrary (score 0.9429)



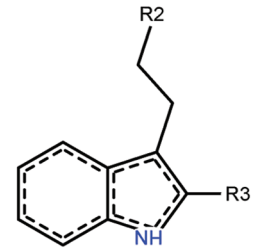
g) Similar, leadlike, and diverse sublibrary (score 0.9429)

Figure 12. Different 4 × 4 × 4 sublibraries for the query molecule **8** and core **9** shown in Figure 12a and 12b, respectively. Reagents 1−4 link to R4 of the core, reagents 5−8 to R5, and reagents 9−12 to R6. Because of the properties and the size of the query molecule, the leadlike criteria are incorporated into the scoring function instead of using a filter. Consequently, for core link R4, reagents which do not consist of rings are more likely to be selected. For the diverse sublibrary, the reagents at R4 and R5 are switched in contrast to the other sublibraries, because feature trees cannot distinguish between the link positions. In doing a randomized reagent substitution the optimization reaches convergence, mapping either of the reagents of R4 or R5 to the aromatic ring of query **8**.

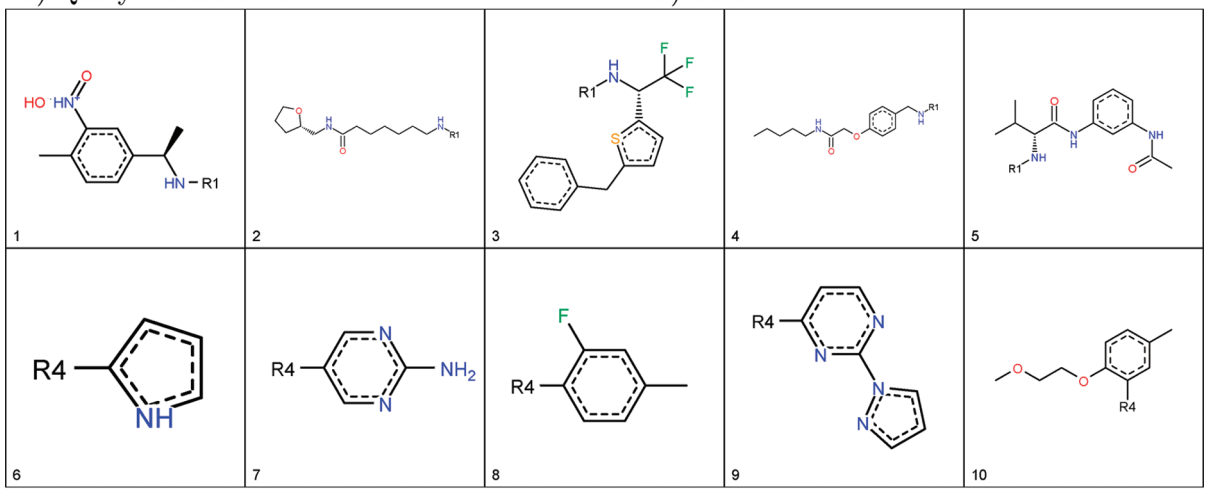
R5; 6163 connectable to R6). Again, too similar reagents are frequently selected. Furthermore, using similarity as single criterion leads to products exceeding leadlike properties. This is not surprising using query **8** and core **9** with three open valences. Therefore, in this case it is more reasonable to incorporate the leadlike criteria into the scoring function instead of applying a filter on the products. This allows single products to exceed property thresholds (see



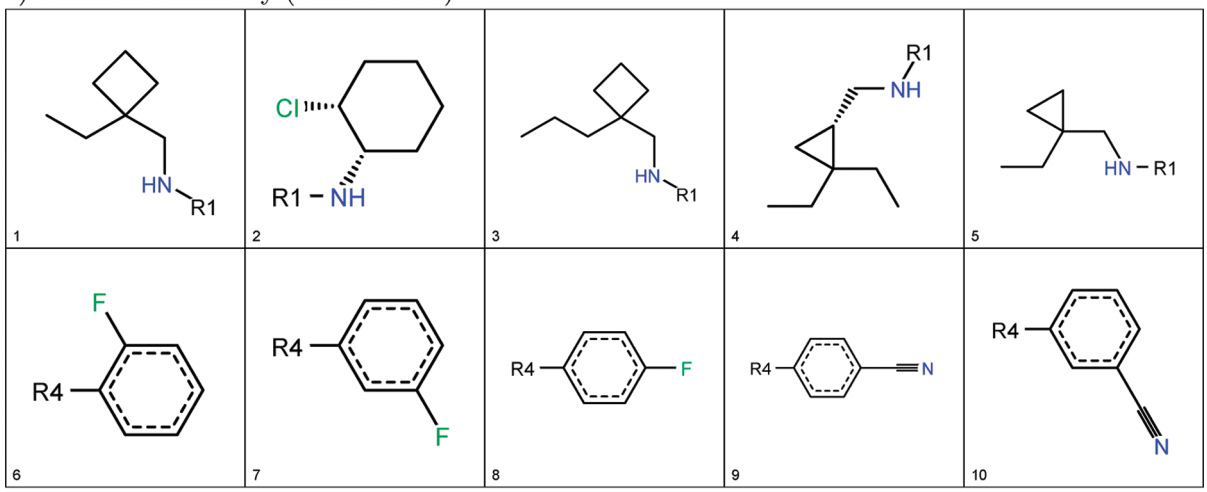
a) Query molecule **10**



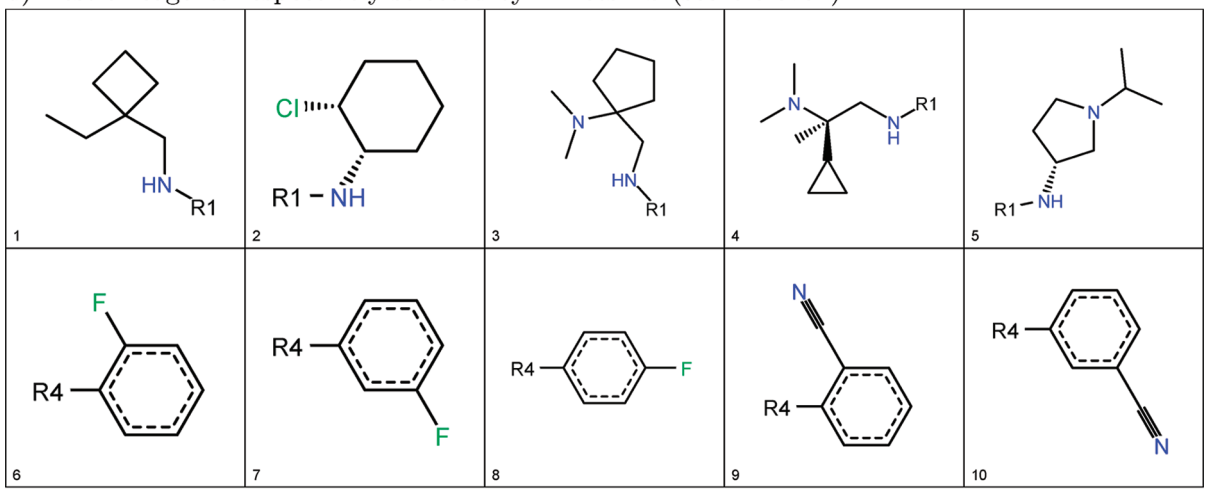
b) Core **11**



c) Random sublibrary (score 0.5653)

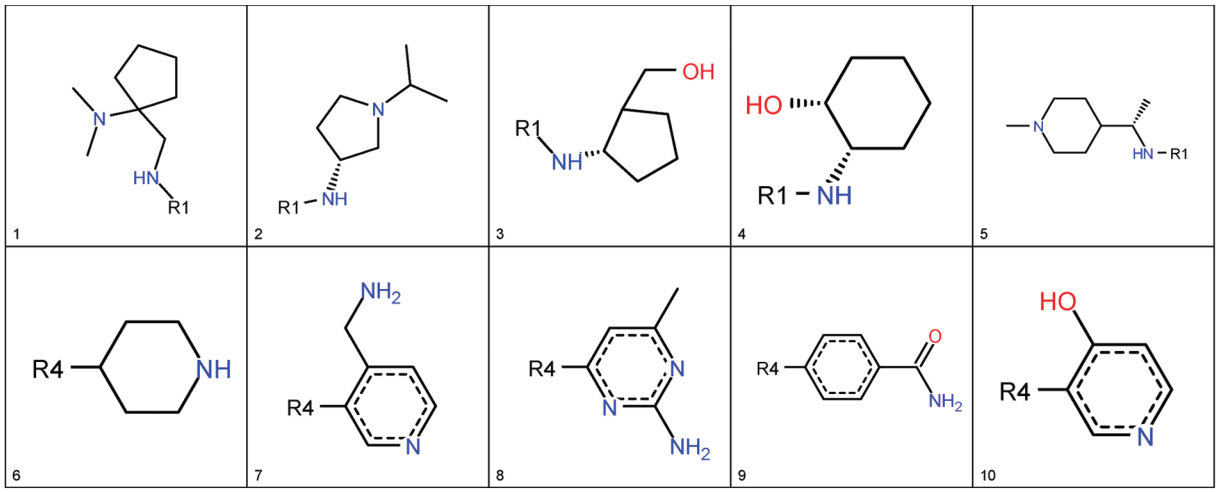


d) Best 5 reagents respectively selected by FTrees-FS (score 0.8751)

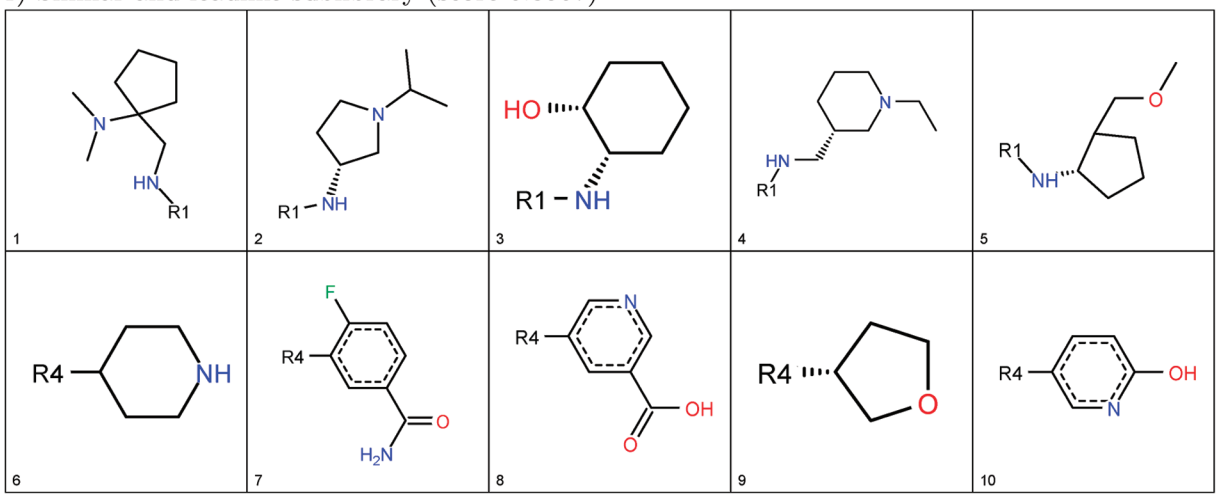


e) Similar sublibrary (score 0.8754)

Figure 13. Continued.



f) Similar and leadlike sublibrary (score 0.8567)



g) Similar, leadlike, and diverse sublibrary (score 0.8537)

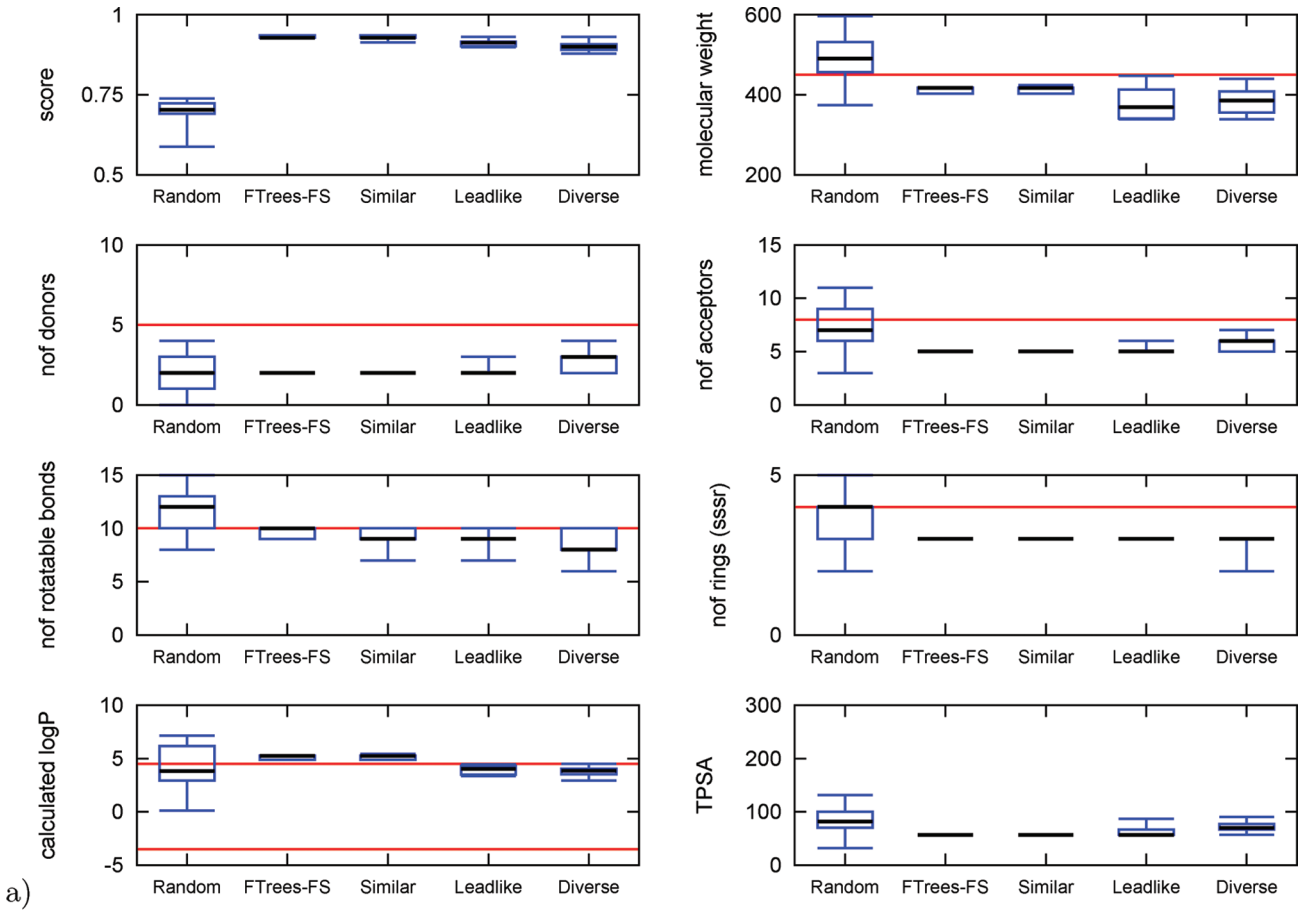
Figure 13. Different 5 × 5 sublibraries for the query molecule **10** and core **11** illustrated in Figure 13a and 13b, respectively. Reagents 1−5 link to R2 of the core and reagents 6−10 to R3.

Figure 14 and Figure 15). Even if single products cannot satisfy all objectives, the overall property profiles of these sublibraries become more leadlike compared to the one using similarity as a single criterion. Especially, the number of rings is decreased to four. Furthermore, Figure 12g shows the disadvantage of the feature tree descriptor: Although the reagents 1−4 would fit better to R5, and 5−8 to R4, respectively, the feature tree descriptor cannot distinguish between those links and therefore the reagents are interchanged.

Serotonin 5-HT$_{2A}$ Receptor. Serotonin (5-hydroxytryptamine) is the biogenic amine of tryptophan and plays an important role as a neurotransmitter, controlling numerous (patho)physiological processes. The serotonin receptor 5-HT$_{2A}$ is a G protein-coupled receptor (GPCR), and 5-HT$_{2A}$ antagonists have potential utility in treating depression, schizophrenia, and sleep disorders.[63] A known antagonist[63] (**10**, see Figure 13a) was used to optimize an indole library with core **11** (see Figure 13b)[64] and 25 567 reagents (23 486 reagents connectable to link R2; 2081 connectable to R3). In comparison to the case studies of query **6**, we get diverse products which are similar to the query and obey the leadlike criteria (see Figure 14).

Furthermore, we evaluated the workflow also for a larger sublibrary size of 50 × 50 for query **10** and core **11** (Figure 13a and 13b). Figure 16a shows the similarity values of the 2500 products using LoFT with different optimization criteria. For the randomly chosen reagents, the similarity values of the first 1700 products fluctuate around a value of 0.75, which is slightly above a random similarity value, because the core matches to the scaffold of the query structure. The other products do not pass the size filter, and the similarity is scored with 0.0. The sublibrary of the simulated annealing run shows high similarity values while they decrease with the more stringent conditions to the sublibrary. Figure 16b shows that the product properties of a similar, diverse, and leadlike sublibrary are within the given ranges.

Comparison with FTrees-FS. The results presented above show very similar sublibraries applying FTrees-FS as well as LoFT using similarity as the only criterion. If different possible core placements result in compounds similar to the query (see, for example, Figure 17), a fully enumerated sublibrary generated from FTrees-FS results will contain products with low similarity. This is because reagents from different core placements might result in products of different size. In contrast, LoFT optimizes the sublibraries focusing on a single core placement. To cope better with the demands of library design, LoFT allows starting all recursive com-



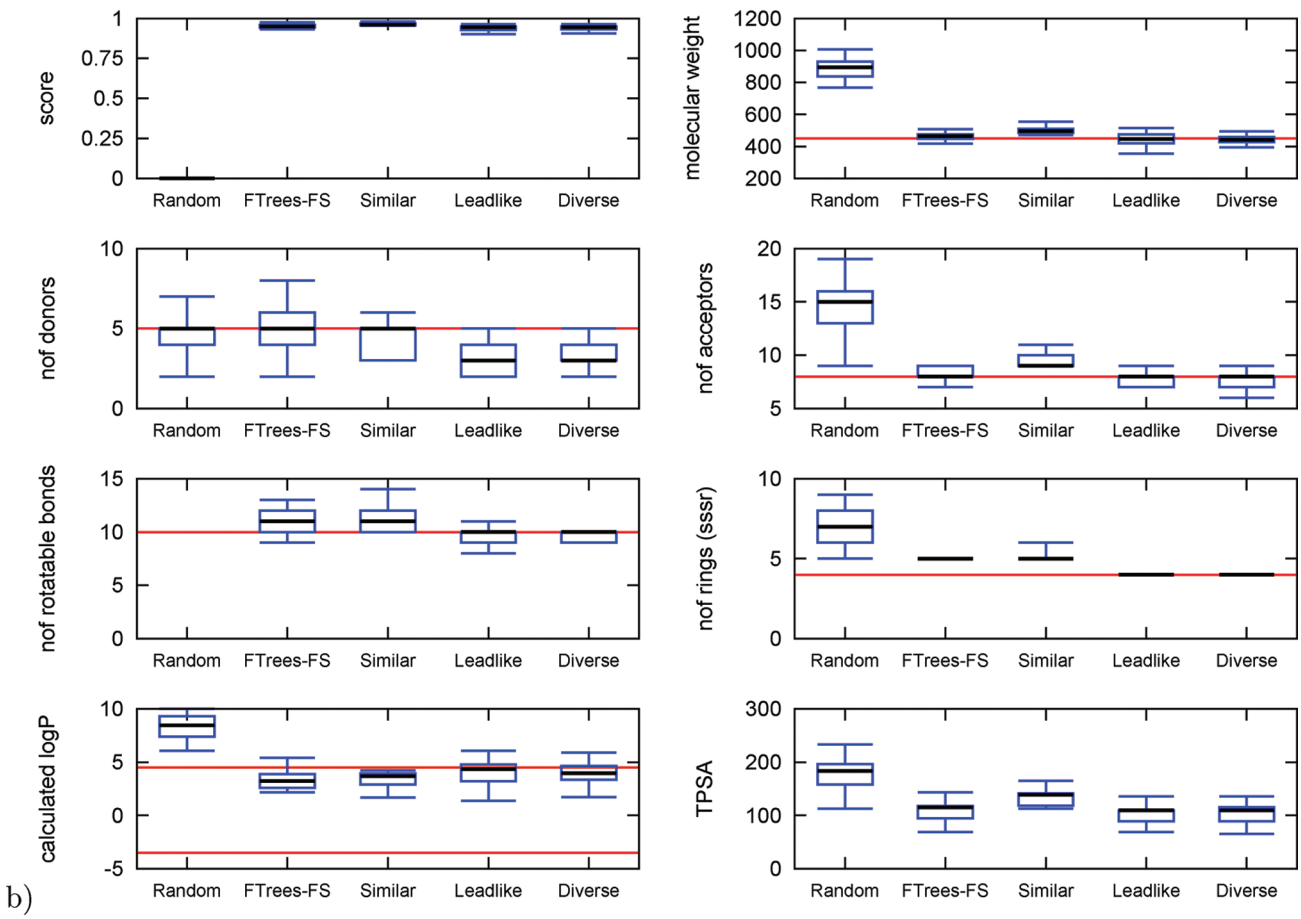


Figure 14. Continued.

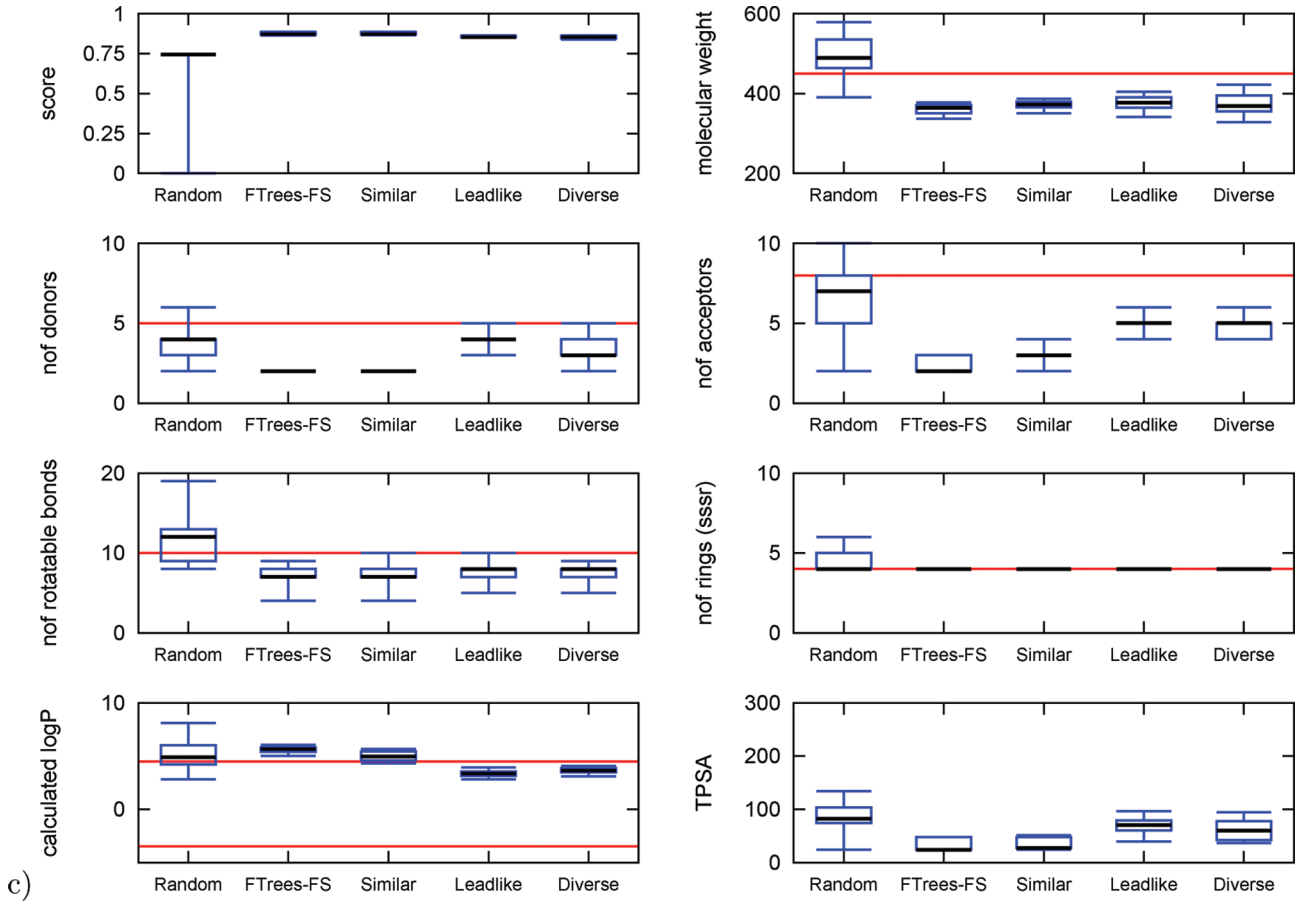


Figure 14. Box-whisker-plots for selected product properties for the random (1), FTreesFS (2), similar (3), similar and leadlike (leadlike, 4), and the similar, leadlike, and diverse sublibraries (diverse, 5) for H3 (a), for CDK2 (b), and for 5-HT_{2A} (c), respectively. The black lines indicate the median, and the boxes the interquartile range. Furthermore, the minimum and maximum property value is displayed. The red lines indicate the leadlike criteria thresholds. When the leadlike filter criteria are used, the properties are within given ranges. For H3 and $5HT_{2A}$, the feature tree similarity to the query is the only scoring objective. When the leadlike criteria are incorporated into the scoring function, a few product properties exceed the criteria thresholds (CDK2, see Figure 15).

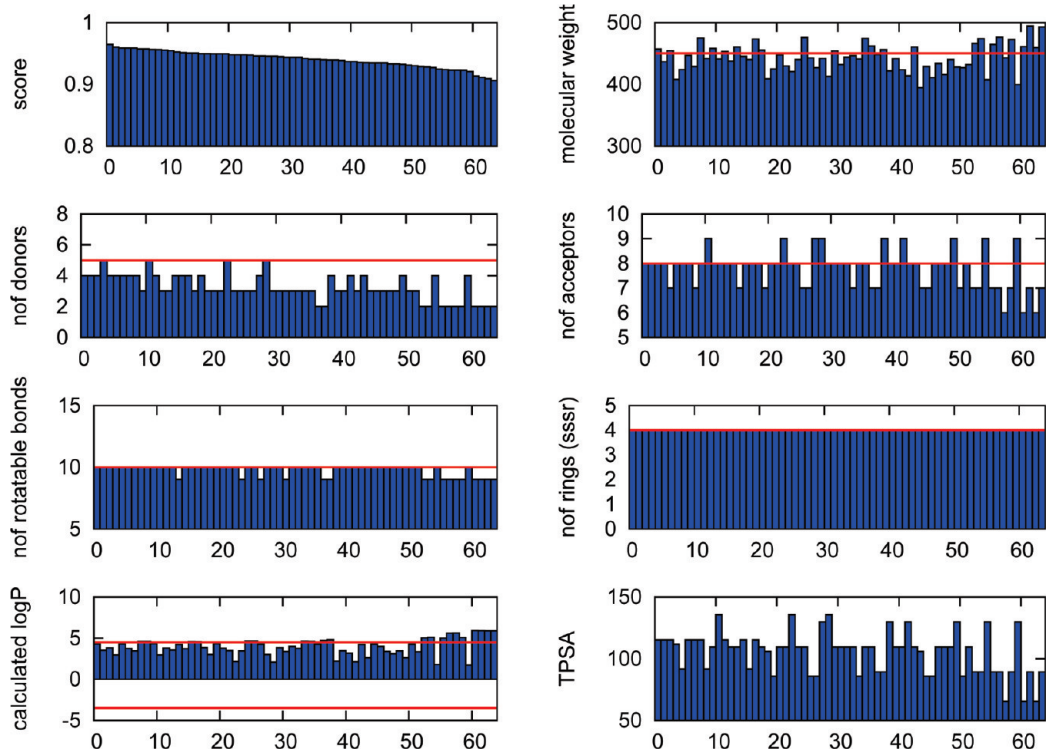


Figure 15. Property distributions for the 64 products of the similar, leadlike, and diverse sublibrary for CDK2. When the leadlike criteria are incorporated in the scoring function, there are a few products that exceed single criteria.

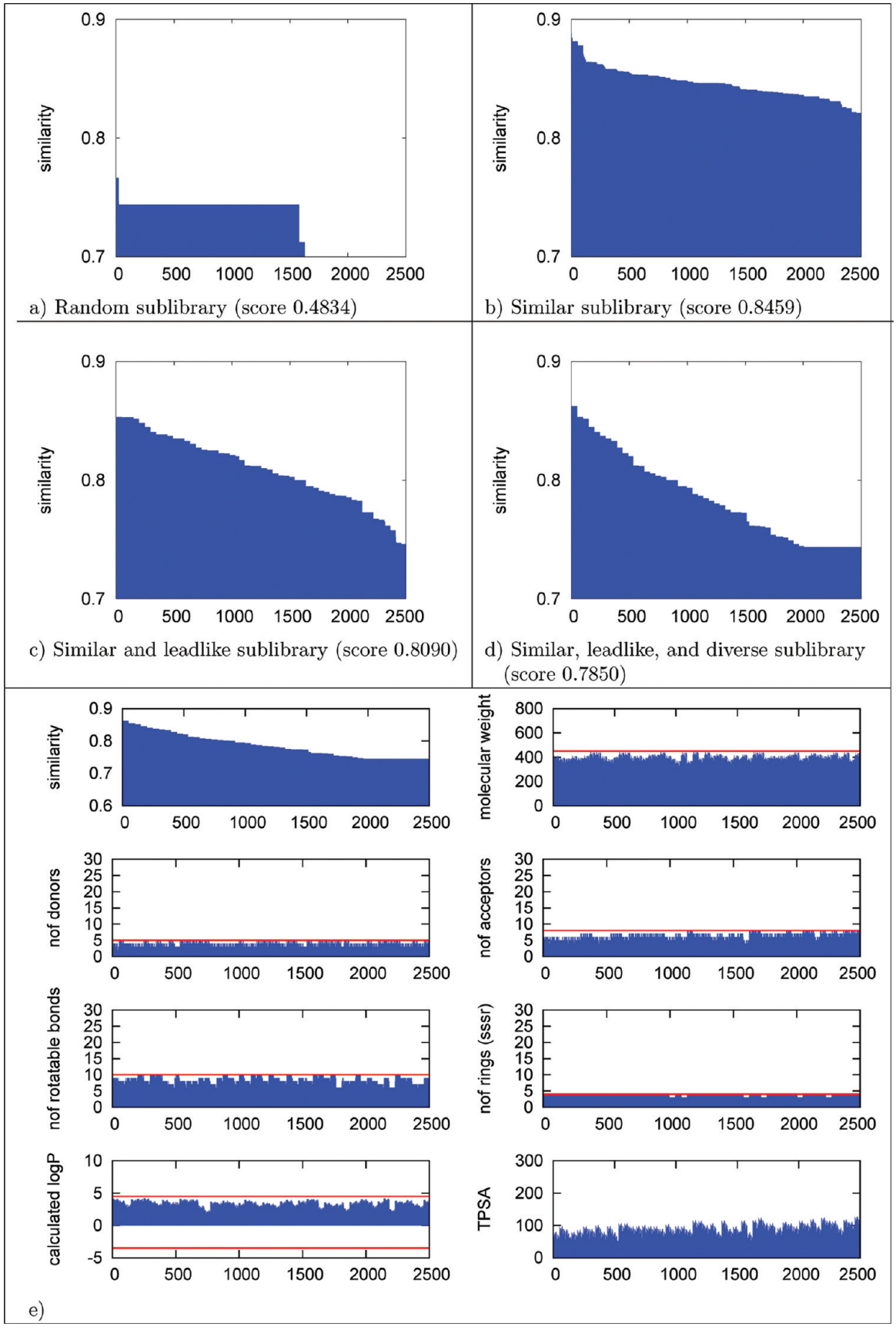


Figure 16. (a−d) Similarity values of the 2500 products of a 50 × 50 sublibrary for the 5HT2a core (Figure 13b). The product values were sorted starting with the highest one. (a) The first similarity values of the random sublibrary (score 0.4834) are relatively high, because the core (Figure 13b) matches to the scaffold of the query (see Figure 13a). The last products do not pass the size filter for similarity comparisons and are scored with 0.0. (b) The sublibrary with similarity as a single criterion has a score of 0.8459, while the scores are decreasing with the additional requirements (c, d). (e) depicts the product values of some key properties of the similar, leadlike, and diverse sublibrary. The filter thresholds are visualized by the red lines, showing that the property values are within the given ranges.

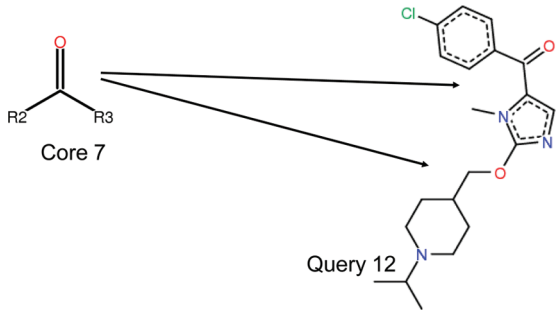


Figure 17. Two possible placements of core 7 on query 12.[57] The automatic placement depends on the available reagents and the project-specific constraints. The sublibraries corresponding to this query and core combination can be found in the Supporting Information.

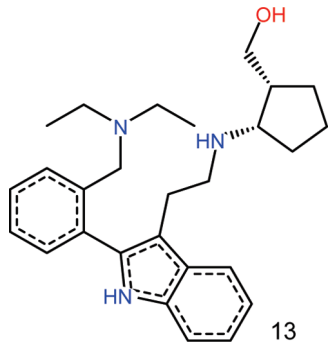


Figure 18. Molecule 13 is part of the 50 × 50 5HT2a library which was optimized using similarity as the only criterion. Here, the indole core is placed on the large ring of the query structure whereas FTrees and FTrees-FS are not allowed to map in a similar way with the standard parameter settings.

parisons from the core regardless of the subtree proportion. Figure 18 depicts molecule **13**, which is part of the 50 × 50 5HT2a sublibrary, where this more relaxed comparison strategy is advantageous. The indole core can be placed on the large ring of the query structure, although the long carbon chains result in single nodes which lead to subtrees of unequal size. The physicochemical properties of compounds selected with FTrees-FS are in most cases similar to those of the query molecules. Consequently, performing FTrees-FS searches based on a query with acceptable physicochemical properties in most cases results in acceptable ranges of those properties for the selected compounds. But in practice sometimes the query molecules consist of some unfavorable properties, and for those cases it is important to flexibly integrate these optimization parameters in the scoring function of the design tool. Figure 14 shows that it is possible to influence the physicochemical profile of the sublibraries according to the requirements defined.

Run Time and Memory Usage. The sublibraries are computed on an Intel Core 2 Duo 3 GHz with 4 GB RAM and OpenSuse 10.2. Figure 19 shows the run times of the different case studies. There is a significant decrease in run times, if the filter criteria are additionally applied. This is because that for a huge number of products, the filter criteria are exceeded and no similarity comparisons have to be computed. When the leadlike criteria are incorporated in the scoring function (CDK2 study), there is no decrease in run time, because all similarity comparisons have to be computed. Using cluster IDs as diversity criterion does not affect the run time notably.

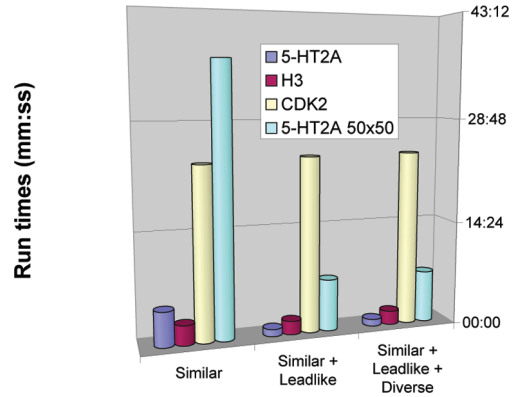


Figure 19. Run times of the different case studies. The run times decrease significantly if filter criteria are used, because no similarity comparison is calculated if a product does not pass the filter. When the leadlike criteria are incorporated in the scoring function (CDK2 study), there is no decrease in run time, because all similarity comparisons have to be computed. Furthermore, the additional diversity criterion does not affect the run times considerably.

Modulating sublibrary sizes and the number of optimization steps, we can see that LoFT scales linearly (see Figure 20). Moreover for the H3 study, the dynamic programming matrix of core **7** is very small (four columns) and therefore the run time is dictated by the query-to-reagent comparisons. If the global matrix is fully computed once, there is no large difference in terms of run time between an optimization using 10 000, 20 000, 50 000, 100 000, or 200 000 steps.

The memory usage of LoFT depends on the size of the fragment space, because the molecular structures and feature trees of the fragments are held in memory for the whole program run. Also the size of the query feature trees determines the size of the global matrix together with the number of reagents used. For the H3 case study, the observed maximal memory size was 2.6 GB, and for CDK2 and 5-HT_{2A} it was 2.2 GB. In most scenarios, the reagent feature trees are only needed for filling the global matrix and the molecular graph of a reagent is only needed for calculating the fragment descriptors. In a future version, we will dynamically load the data from file if requested, reducing the memory needs substantially. Furthermore, for speeding up the computation and avoiding redundant calculations, LoFT stores several results, e.g., the score of each product or the similarity of a reagent to each query molecule link, increasing the memory needs as well. Because the feature tree descriptor abstracts the molecular structure and isomeric structures cannot be distinguished, the SwiFT approach[45] could be applied to decrease the number of computations as well as the size of the diversity matrix and the size of the global matrix (see Table 3).

CONCLUSION

We have presented a novel tool for focused library design which is especially designed to generate project-specific combinatorial libraries. Besides an optimization of diverse physicochemical properties, the products should be similar to given active compounds. For those similarity searches, the feature tree descriptor was incorporated. In contrast to existing methods for library design, the incorporated match search algorithm for similarity comparison allows a comparison on product level, avoiding the explicit enumeration of all products of the sublibrary.

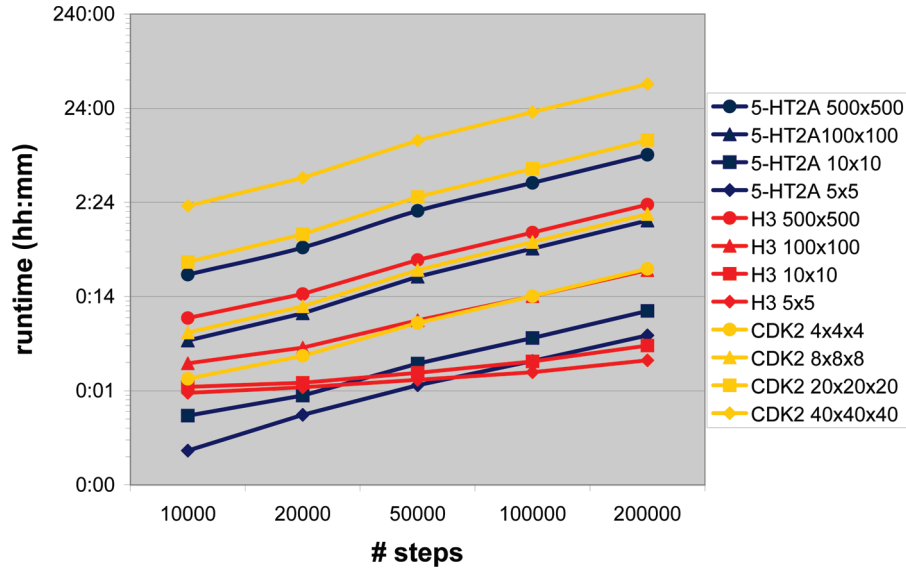


Figure 20. Correlation of sublibrary size, number of steps, and run time for the case studies. For the run time, a logarithmical scale is used. In the H3 case study, the dynamic programming matrix is very small (four columns). Therefore for the 5 × 5 sublibrary, the run time is dictated by the query-to-reagent comparisons. If the global matrix is fully computed once, there is no further noteworthy computation overhead. For the CDK2 40 × 40 × 40 sublibrary optimization taking 200 000 steps, 3.2 × 108 query-to-product similarity comparisons are computed within 43 h and 24 min (∼2000 comparisons/s). The run times are decreased substantially by incorporating the size filter for the similarity comparisons. For example, allowing the number of heavy atoms of the product only in the range 0.6 to 1.5 with respect to the number of heavy atoms of the query molecule, the run time is decreased to 16 h and 9 min, whereas the score decreases from 0.9115 to 0.9102 as well. For the 20 × 20 × 20 sublibrary, the run time is reduced from 11 h and 4 min to 4 h and 42 minutes and the score increased from 0.9263 to 0.9284. For the 5-HT2a 500 × 500 sublibrary optimization taking 200 000 steps, the run time is reduced from 7 h and 46 min to 2 h and 28 min whereas the score drops from 0.7915 to 0.7914.

Table 3. Redundancy on Feature Tree Level for the Three Fragment Spaces[a]

fragment space	no. reagents	nonredundant no. reagents on ftree level	no. dissimilarity comparisons for diversity matrix	no. dissimilariy comparisons without redundancy	percent similarity comparisons, %
H3	32585	21401	5.3×10^9	2.29×10^9	∼43
CDK2	29649	19194	4.39×10^9	1.84×10^9	∼42
5-HT_{2A}	25567	15672	3.27×10^9	1.23×10^9	∼37

[a] For calculating the diversity matrix, using a SwiFT preprocessing, we can save more than 57% of calculations and memory size. For all data sets, ∼250 MB and less than a minute is required to identify redundant feature trees.

Applying the feature tree descriptor together with physicochemical product properties, such as molecular weight or TPSA, which can be directly retrieved from the fragments, we demonstrated that LoFT is able to design promising sublibraries within reasonable computing times. The optimization process can be guided using several molecules as queries or antiqueries at the same time. We achieve sublibraries in which the products are similar to the query and all sublibraries pass predefined property criteria as opposed to fragment space search methods which pick the most similar products from combinatorial space. In addition, the combination of fragment spaces and library design allows the consideration of multiple scaffolds with multiple, individual alignments to query compounds. The computing times make an interactive optimization cycle applicable.

ACKNOWLEDGMENT

The authors thank Jörg Degen (ZBH) for the implementation of the fragment space library, Tobias Lippert (ZBH) for the joint implementation of a generic simulation module, and Patrick Maass (ZBH) for helpful discussions and for the supply of basic code functionalities. We are indebted to Herbert Köppen, Bernd Wellenzohn, and Alexander Weber at Boehringer Ingelheim for pushing LoFT in the right direction by their valuable input. Moreover, we thank Sally Hindle, Holger Claussen, and Marcus Gastreich at BioSolveIT for their support.

Supporting Information Available: An automatically generated document containing the parameters used for computation and the 5 × 5 or 4 × 4 × 4 sublibraries and histograms for additional query compounds. This material is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Johnson, E. G.; Maggiora, G. M. *Concepts and Applications of Molecular Similarity*; Wiley: New York, 1990.
(2) Fischer, E. Einfluss der Konfiguration auf die Wirkung der Enzyme. *Ber. Dtsch. Chem. Ges.* **1894**, *27*, 2985–2993.
(3) Böhm, H. J.; Schneider, G. *Virtual Screening for Bioactive Molecules*; John Wiley & Sons, Inc.: New York, 2000.
(4) Terrett, N. K. *Combinatorial Chemistry*; Oxford University Press: Cary, NC, 1998.
(5) Dobson, C. M. Chemical space and biology. *Nature.* **2004**, *432* (7019), 824–828.
(6) Lessel, U.; Wellenzohn, B.; Lilienthal, M.; Claussen, H. Searching Fragment Spaces with Feature Trees. *J. Chem. Inf. Model.* **2009**, pp 270−279.

(7) Lewell, X. Q.; Judd, D. B.; Watson, S. P.; Hann, M. M. RECAP-retrosynthetic combinatorial analysis procedure: a powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. *J. Chem. Inf. Comput. Sci.* **1998**, *38* (3), 511–522.
(8) Mauser, H.; Stahl, M. Chemical Fragment Spaces for de novo Design. *J. Chem. Inf. Model.* **2007**, *47* (2), 318–324.
(9) Degen, J.; Wegscheid-Gerlach, C.; Zaliani, A.; Rarey, M. On the Art of Compiling and Using 'Drug-Like' Chemical Fragment Spaces. *ChemMedChem* **2008**, *3* (10), 1503–1507.
(10) Pärn, J.; Degen, J.; Rarey, M. Exploring fragment spaces under multiple physicochemical constraints. *J. Comput.-Aided. Mol. Des.* **2007**, *21* (6), 327–340.
(11) Maass, P.; Schulz-Gasch, T.; Stahl, M.; Rarey, M. Recore: a fast and versatile method for scaffold hopping based on small molecule crystal structure conformations. *J. Chem. Inf. Model.* **2007**, *47* (2), 390–399.
(12) Schneider, G.; Lee, M. L.; Stahl, M.; Schneider, P. De novo design of molecular architectures by evolutionary assembly of drug-derived building blocks. *J. Comput.-Aided. Mol. Des.* **2000**, *14* (5), 487–494.
(13) Rarey, M.; Stahl, M. Similarity searching in large combinatorial chemistry spaces. *J. Comput.-Aided. Mol. Des.* **2001**, *15* (6), 497–520.
(14) Degen, J.; Rarey, M. FlexNovo: Structure-Based Searching in Large Fragment Spaces. *ChemMedChem.* **2006**, *1* (8), 854–868.
(15) Hartenfeller, M.; Proschak, E.; Schüller, A.; Schneider, G. Concept of combinatorial de novo design of drug-like molecules by particle swarm optimization. *Chem. Biol. Drug Des.* **2008**, *72* (1), 16–26.
(16) Patel, Y.; Gillet, V. J.; Howe, T.; Pastor, J.; Oyarzabal, J.; Willett, P. Assessment of additive/nonadditive effects in structure−activity relationships: implications for iterative drug design. *J. Med. Chem.* **2008**, *51* (23), 7552–7562.
(17) Boehm, M.; Wu, T.-Y.; Claussen, H.; Lemmen, C. Similarity Searching and Scaffold Hopping in Synthetically Accessible Combinatorial Chemistry Spaces. *J. Med. Chem.* **2008**, *51* (8), 2468–2480.
(18) Leland, B. A.; Christie, B. D.; Nourse, J. G.; Grier, D. L.; Carhart, R. E.; Maffett, T.; Welford, S. M.; Smith, D. H. Managing the Combinatorial Explosion. *J. Chem. Inf. Comput. Sci.* **1997**, *37* (1), 62–70.
(19) Rarey, M.; Dixon, J. S. Feature trees: a new molecular similarity measure based on tree matching. *J. Comput.-Aided. Mol. Des.* **1998**, *12* (5), 471–490.
(20) Rarey, M.; Hindle, S.; Maass, P.; Metz, G.; Rummey, C.; Zimmermann, M. Feature Trees: Theory and Applications from Large-Scale Virtual Screening to Data Analysis. In *Pharmacophores and Pharmacophore Search*; Langer, T.; Hoffmann, R. D., Eds.; Wiley-VCH: Weinheim, 2005; Vol. 32, pp 81−116.
(21) Gillet, V. J.; Downs, G. M.; Holliday, J. D.; Lynch, M. F.; Dethlefsen, W. Computer storage and retrieval of generic chemical structures in patents. 13. Reduced graph generation. *J. Chem. Inf. Model.* **1991**, *31* (2), 260–270.
(22) Gillet, V. J.; Willett, P.; Bradshaw, J. Similarity searching using reduced graphs. *J. Chem. Inf. Comput. Sci.* **2003**, *43* (2), 338–345.
(23) Willett, P.; Barnard, J. M.; Downs, G. M. Chemical Similarity Searching. *J. Chem. Inf. Model.* **1998**, *38* (6), 983–996.
(24) Gillet, V. J.; Nicolotti, O. Evaluation of reactant-based and product-based approaches to the design of combinatorial libraries. *Perspect. Drug Discovery Des.* **2000**, *20*, 265–287.
(25) Gillet, V. J.; Khatib, W.; Willett, P.; Fleming, P. J.; Green, D. V. S. Combinatorial library design using a multiobjective genetic algorithm. *J. Chem. Inf. Comput. Sci.* **2002**, *42* (2), 375–385.
(26) Agrafiotis, D. K.; Martin, E. J. Advances in Combinatorial Library Design. *J. Mol. Graph. Model.* **2000**, *18*, 317–319.
(27) Ghose, A. K.; Viswanadhan, V. N. *Combinatorial Library Design and Evaluation*; Marcel Dekker: New York, 2001.
(28) Weber, L. Current Status of Virtual Combinatorial Library Design. *QSAR Comb. Sci.* **2005**, *24* (7), 809–823.
(29) Brown, R. D.; Martin, Y. C. Designing combinatorial library mixtures using a genetic algorithm. *J. Med. Chem.* **1997**, *40* (15), 2304–2313.
(30) Gillet, V. J.; Willett, P.; Bradshaw, J.; Green, D. V. S. Selecting Combinatorial Libraries to Optimize Diversity and Physical Properties. *J. Chem. Inf. Model.* **1999**, *39* (1), 169–177.
(31) Gillet, V. J.; Willett, P.; Fleming, P. J.; Green, D. V. S. Designing focused libraries using MoSELECT. *J. Mol. Graph. Model.* **2002**, *20* (6), 491–498.
(32) Gillet, V. J. *Computational Medicinal Chemistry for Drug Discovery*; Bulinck, P.; de Winter, H.; Langenaeker, W.; Tollenaere, J. P., Eds.; Marcel Dekker: New York, 2004.
(33) Gillet, V. J. Applications of Evolutionary Computation in Drug Design. *Struct. Bonding (Berlin)* **2004**, *110*, 133–152.
(34) Zheng, W.; Hung, S. T.; Saunders, J. T.; Seibel, G. L. PICCOLO: a tool for combinatorial library design via multicriterion optimization. *Pac. Symp. Biocomput.* **2000**, 588–599.
(35) Agrafiotis, D. K. Multiobjective optimization of combinatorial libraries. *J. Comput.-Aided. Mol. Des.* **2002**, *16* (5−6), 335–356.
(36) Agrafiotis, D. K.; Lobanov, V. S. Ultrafast algorithm for designing focused combinational arrays. *J. Chem. Inf. Comput. Sci.* **2000**, *40* (4), 1030–1038.
(37) Le Bailly de Tilleghem, C. L.; Beck, B. T.; Boulanger, B.; Govaerts, B. A fast exchange algorithm for designing focused libraries in lead optimization. *J. Chem. Inf. Model.* **2005**, *45* (3), 758–767.
(38) Truchon, J.-F.; Bayly, C. I. GLARE: A New Approach for Filtering Large Reagent Lists in Combinatorial Library Design Using Product Properties. *J. Chem. Inf. Model.* **2006**, *46* (4), 1536–1548.
(39) Gillet, V. J.; Willett, P.; Bradshaw, J. The Effectiveness of Reactant Pools for Generating Structurally-Diverse Combinatorial Libraries. *J. Chem. Inf. Comput. Sci.* **1997**, *37* (4), 731–740.
(40) Fricker, P. C.; Gastreich, M.; Rarey, M. Automated Drawing of Structural Molecular Formulas under Constraints. *J. Chem. Inf. Model.* **2004**, *44* (3), 1065–1078.
(41) Spotfire DecisionSite, 9.1.1; Tibco Software Inc., 212 Elm St., Somerville, MA.
(42) Ertl, P.; Rohde, B.; Selzer, P. Fast Calculation of Molecular Polar Surface Area as a Sum of Fragment-Based Contributions and Its Application to the Prediction of Drug Transport Properties. *J. Med. Chem.* **2000**, *43* (20), 3714–3717.
(43) Bellman, R. On the Theory of Dynamic Programming. *Proc. Natl. Acad. Sci. U.S.A.* **1952**, *38*, 716–719.
(44) Key Organics Limited U. K. Bionet Screening Compounds Database. http://www.keyorganics.ltd.uk/screenin.htm, accessed January 30, 2009.
(45) Fischer, J. R.; Rarey, M. SwiFT: An Index Structure for Reduced Graph Descriptors in Virtual Screening and Clustering. *J. Chem. Inf. Model.* **2007**, *47*, 1341–1353.
(46) Lipinski, C. A.; Lombardo, F.; Domini, B. W.; Feeney, P. J. Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery. *Adv. Drug Delivery Rev.* **1996**, *23*, 3–25.
(47) Derringer, G.; Suich, R. Simultaneous-Optimization of Several Response Variables. *J. Qual. Tech.* **1980**, *12* (4), 214–219.
(48) FTrees, 2.02; Biosolve IT GmbH: An der Ziegelei 75, 53757 St. Augustin, Germany.
(49) Downs, G. M.; Barnard, J. M. Clustering Methods and Their Uses in Computational Chemistry. *Rev. Comput. Chem.* **2003**, *18*, 1–40.
(50) Raymond, J. W.; Willett, P. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *J. Comput.-Aided. Mol. Des.* **2002**, *16* (7), 521–533.
(51) Kirkpatrick, S.; Gelatt, C. D., Jr.; Vecchi, M. P. Optimization by Simulated Annealing. *Science* **1983**, *220* (4598), 671–680.
(52) Dueck, G.; Scheuer, T. Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. *J. Comp. Phys.* **1990**, *90* (1), 161–175.
(53) Dueck, G. New Optimization Heuristics: The Great Deluge Algorithm and the Record-to-Record Travel. *J. Comp. Phys.* **1993**, *104* (1), 86–92.
(54) Kelder, J.; Grootenhuis, P. D.; Bayada, D. M.; Delbressine, L. P.; Ploemen, J. P. Polar molecular surface as a dominating determinant for oral absorption and brain penetration of drugs. *Pharm. Res.* **1999**, *16* (10), 1514–9.
(55) BiosolveIT GmbH, KnowledgeSpace. http://www.biosolveit.de/KnowledgeSpace. Accessed January 2, 2009.
(56) Oprea, T. I.; Davis, A. M.; Teague, S. J.; Leeson, P. D. Is there a difference between leads and drugs? A historical perspective. *J. Chem. Inf. Comput. Sci.* **2001**, *41* (5), 1308–1315.
(57) Celanire, S.; Wijtmans, M.; Talaga, P.; Leurs, R.; de Esch, I. J. P. Keynote review: histamine H3 receptor antagonists reach out for the clinic. *Drug Discovery Today* **2005**, *10* (23−24), 1613–1627.
(58) Lau, J. F.; Jeppesen, C. B.; Rimvall, K.; Hohlweg, R. Ureas with histamine H3-antagonist receptor activity--a new scaffold discovered by lead-hopping from cinnamic acid amides. *Bioorg. Med. Chem. Lett.* **2006**, *16* (20), 5303–5308.
(59) Nigg, E. A. Cyclin-dependent protein kinases: key regulators of the eukaryotic cell cycle. *Bioessays* **1995**, *17* (6), 471–480.
(60) Wadler, S. Perspectives for cancer therapies with cdk2 inhibitors. *Drug Resist. Update* **2001**, *4* (6), 347–367.
(61) Huang, N.; Shoichet, B. K.; Irwin, J. J. Benchmarking sets for molecular docking. *J. Med. Chem.* **2006**, *49* (23), 6789–6801.
(62) Gray, N. S.; Wodicka, L.; Thunnissen, A. M.; Norman, T. C.; Kwon, S.; Espinoza, F. H.; Morgan, D. O.; Barnes, G.; LeClerc, S.; Meijer, L.; Kim, S. H.; Lockhart, D. J.; Schultz, P. G. Exploiting chemical libraries, structure, and genomics in the search for kinase inhibitors. *Science* **1998**, *281* (5376), 533–538.
(63) Swain, C. J.; Teran, A.; Maroto, M.; Cabello, A. Identification and optimization of 5-amino-7-aryldihydro-1,4-diazepines as 5-HT2A ligands. *Bioorg. Med. Chem. Lett.* **2006**, *16* (23), 6058–6062.

(64) Smith, A. L.; Stevenson, G. I.; Lewis, S.; Patel, S.; Castro, J. L. Solid-phase synthesis of 2,3-disubstituted indoles: discovery of a novel, high-affinity, selective h5-HT2A antagonist. *Bioorg. Med. Chem. Lett.* **2000**, *10* (24), 2693–2696.
(65) Ghose, A. K.; Crippen, G. M. Atomic physicochemical parameters for three-dimensional-structure-directed quantitative structure−activity relationships. 2. Modeling dispersive and hydrophobic interactions. *J. Chem. Inf. Comput. Sci.* **1987**, *27* (1), 21–35.
(66) Wildman, S. A.; Crippen, G. M. Prediction of Physicochemical Parameters by Atomic Contributions. *J. Chem. Inf. Model.* **1999**, *39* (5), 868–873.
(67) FlexV, 1.9.0; BioSolveIT GmbH: An der Ziegelei 75, 53757 St. Augustin, Germany.
(68) Fischer, J. R.; Fricker, P.; Rarey, M.; Gastreich, M.; Hindle, S.; Sonnenburg, F.; Lemmen, C. FTreesWeb; A Web Interface to Feature Trees. http://www.zbh.uni-hamburg.de/FTreesWeb, accessed February 14, 2009.
(69) FTreesXL, 1.1.4; BiosolveIT GmbH: An der Ziegelei 75, 53757 St. Augustin, Germany.

CI900287P