JOURNAL OF CHEMICAL INFORMATION AND MODELING

ARTICLE

pubs.acs.org/jcim

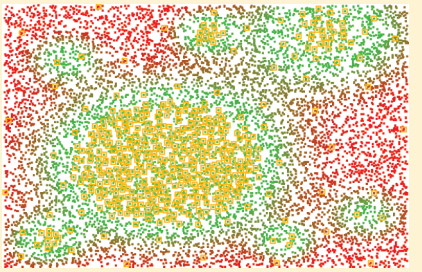
Maximum-Score Diversity Selection for Early Drug Discovery

Thorsten Meinl,*,† Claude Ostermann,‡ and Michael R. Berthold†

†Nycomed Chair for Bioinformatics and Information Mining, University of Konstanz, Konstanz, Germany

‡Max Planck Institute of Molecular Physiology, Dortmund, Germany

ABSTRACT: Diversity selection is a common task in early drug discovery. One drawback of current approaches is that usually only the structural diversity is taken into account, therefore, activity information is ignored. In this article, we present a modified version of diversity selection, which we term *Maximum-Score Diversity Selection*, that additionally takes the estimated or predicted activities of the molecules into account. We show that finding an optimal solution to this problem is computationally very expensive (it is NP-hard), and therefore, heuristic approaches are needed. After a discussion of existing approaches, we present our new method, which is computationally far more efficient but at the same time produces comparable results. We conclude by validating these theoretical differences on several data sets.



INTRODUCTION

The task of diversity selection means choosing a predefined number p of items from a set of items of size n so that the elements in the selected set are as diverse as possible. A prominent example of diversity selection can be found in early drug discovery. Usually, when a new drug is designed, so-called HTS (high-throughput screening) is performed, where several hundreds of thousands of compounds are automatically tested for their activities. This process is quite time-consuming and expensive; therefore, it is desirable to filter out redundant compounds and focus on a diverse subset of molecules, thus avoiding the retesting of duplicates.[1]

In HTS, an additional constraint is added: maximization of the selected molecules' activity. In this context, it is particularly apparent that both objectives are in conflict with each other because of the one fundamental principle underlying most of today's research in chemoinformatics, the so-called structure−activity relationship (SAR).[2] This means that molecules with a similar structure (in 2-D or 3-D) also show similar activities, and this makes it hard to select structurally diverse molecules that are also highly active. Therefore, taking both objectives together—maximizing diversity while maximizing (potential) activity—leads to Maximum-Score Diversity Selection (MSDS), which is a classical multiobjective optimization (MO) problem.

Before presenting several approaches to find good solutions for MSDS, this article first discusses the problem of pure diversity selection and how diversity can be measured. Unfortunately, most (if not all) sensible diversity definitions lead to NP-hard problems when it comes to finding optimal subsets. We provide a short formalization of MSDS and show that by adding a second objective for activity the hardness of the problem remains. Therefore, we present several heuristic approaches, among them a novel algorithm specially developed for MSDS. We compare them both in terms of solution quality and speed on several molecular and artificial data sets.

DIVERSITY SELECTION

Considering both of the involved objectives, the maximization of the subset's diversity is by far the more complicated option. Not only is the problem of finding an optimal subset computationally infeasible, as we show in the next section, but a proper definition of diversity is not straightforward. Although users, especially in chemoinformatics, tend to be able to provide a good estimation of what a diverse subset should look like, this is of course not suitable for implementation in a computer program.

In most cases, diversity is defined based on the distances $d(u,v)$ between the objects under consideration. The further two points are apart, the more dissimilar they are. The challenge is to employ the pairwise distance relation between two objects for a whole set of objects.

We will not discuss the problem of a suitable distance or dissimilarity measure d here. One can think of quite a few possible choices,[3] such as the Tanimoto or Soergel distance on fingerprints[4] or measures based on the structural overlap of two molecules (we will show an example in the Experiments section). The choice of d may certainly affect the final results but is, in many cases, dependent on the specific circumstances.

The Hypercube Coverage Measure. As we already mentioned in the Introduction, selecting diverse subsets of molecules has been performed for quite some time in the chemoinformatics community. A very intuitive and sensible definition of molecular diversity has been formally defined by Agrafiotis[5] (although it was already used more informally in an earlier publication[6]). Each molecule from the complete set I, $|I| = n$ is described by a numeric vector of length d that contains several attributes (e.g., molecular weight, charge, volume, etc.). All these vectors span a d-dimensional hypervolume, and each molecule corresponds to one point in this space. A diverse subset should then cover the

Received: October 26, 2010
Published: February 10, 2011

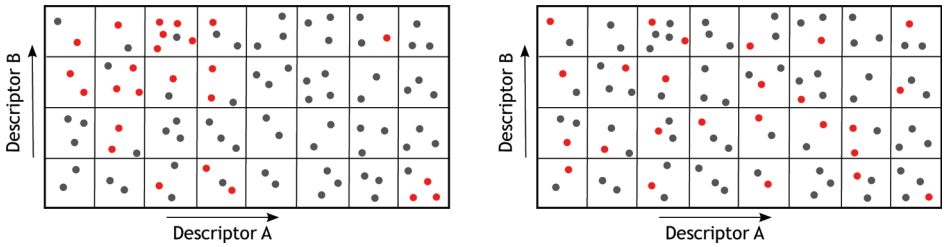


Figure 1. Both plots show the same set of molecules, characterized by two numerical attributes. The selected subset in the left plot is clearly less diverse than the one in the right plot because its selected points cover substantially less space.

space optimally. For this purpose, the hypervolume is partitioned into a set H of k hypercubes $h \in H$ of equal size

$$h := \left\{ x \in \mathbb{R}^d \middle| \forall i = 0, \ldots, d-1 : |x_i - z_i| \leq \frac{1}{2} r \right\} \quad (1)$$

where z is the center of each hypercube, and r is the length of the hypercube's edges (which is assumed to be the same in all dimensions). A subset's $S \subseteq I$ diversity is then defined as the fraction of hypercubes that contain at least one object from S

$$\delta_{\text{hc}}(S) = \frac{|\{h \in H : S \cap h \neq \emptyset\}|}{k} \quad (2)$$

Intuitively this makes sense. The more hypercubes are covered by the same amount of molecules, the better they are distributed over the whole space thus forming a diverse subset. Figure 1 shows two numerical attributes for a set of molecules. In terms of the above definition, the selected molecules in the right 2-D plot constitute a more diverse subset than those in the left plot. This hypercube-based definition of diversity also allows for a very easy selection of a diverse subset (compared to the other definitions below). As δ_{hc} is directly influenced by the numbers of occupied hypercubes, a simple approach to find an optimal subset is to select a molecule from each hypercube, preferably molecules near their centers, and repeat this process (if necessary) until p molecules have been selected.

However, one drawback of this definition is that it only works in vector spaces where the molecules can be arranged in such a way that their positions are in accordance with their original distances. Unfortunately, there are various distance definitions that do not have this property, especially with regard to molecules. One example is a substructure-based distance, where the size of the common substructure of a pair of molecules is a measure of their distance (the larger the common substructure, the smaller the distance and vice versa; more details in the Experiments section). Because only the distances between two structures are known, there is no obvious and easy way to arrange them in a (low-dimensional) vector space so that the distances inside this space are the same as the original substructure-based distances. Thus, for the general case of diversity selection, a definition is required that does not need a vector space, but works solely with the pairwise distances between objects.

The *p*-Dispersion Measure. A more general diversity definition is motivated by the *p-dispersion* or *max-min* problem.[7−9] The goal is to disperse a set of facilities so that the minimum distance between a pair of facilities is maximized

$$\delta_d(S) = \min_{1 \leq i < j \leq p} \{ d(u_i, u_j) : u_i, u_j \in S \} \quad (3)$$

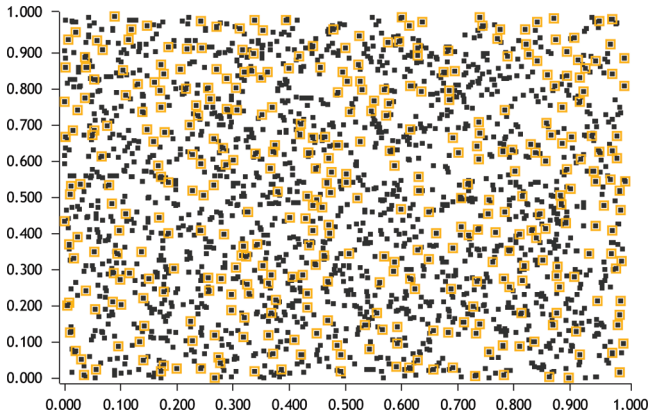


Figure 2. Subset of 200 points taken from a set of 2000 randomly distributed points, which represent a near-optimal solution for the *p*-dispersion problem.

It is easy to see that only the distances between all pairs of objects are required, regardless of whether they form a vector space or not. Figure 2 shows 2000 randomly distributed points in a 2-D space, where the points' distance is their Euclidean distance. The marked points form a subset of 200 objects, which represents a near optimal solution for eq 3. This is not necessarily *the* optimal solution because this cannot be computed efficiently—as we show below—but one that is presumably near the optimum.

Whereas this definition may be perfect for application scenarios where a large minimum distance is crucial, for our case of MSDS in molecules, a single pair of (highly active) molecules that are very close to each other will result in a very low diversity, even if the remaining molecules cover the molecule space quite evenly.

The *p*-Dispersion-Sum Measure. A similar definition, which is used even more often, is the *p-dispersion-sum* or *maximum edge weight clique problem*.[7,10−12] Instead of the minimum distance, the sum of all pairwise distances is maximized (which is equivalent to maximizing the average distance)

$$\delta_{\text{ds}}(S) = \sum_{i=1}^{p} \sum_{j=1}^{i-1} d(u_i, u_j), \; u_i, u_j \in S \quad (4)$$

Intuitively, when optimizing this objective, the selected objects are forced away from each other. If a pair of selected objects happens to be quite close, this only slightly affects overall diversity, in contrast to the *p*-dispersion case. However, it seems that in many cases this definition leads to undesirable distributions of points in the space. Figure 3 shows the same 2000 points as above, but now 200 points are selected to optimize eq 4. It is obvious that the selected points are concentrated on the corners

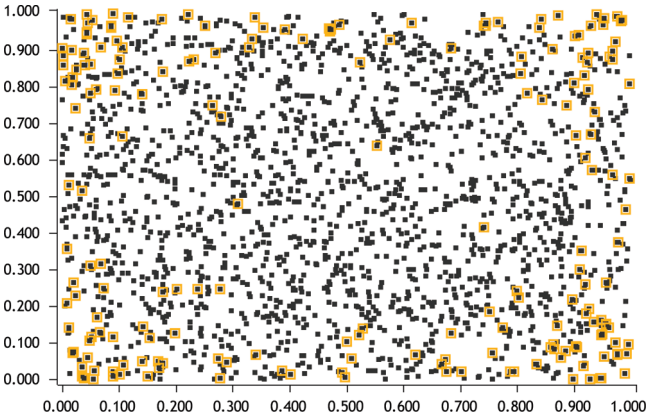


Figure 3. Subset of 200 points taken from a set of 2000 randomly distributed points, representing a near-optimal solution for the *p*-dispersion-sum problem.

of the space, and the interior is almost void of any selected point. This is obviously not a diverse distribution. Although the average distance is quite large (about 0.693 in the example), variance is also quite high: inside the corners, the distances are very small, whereas the intercorner distances are very large. Even though this is an example in 2-D space, which may not be directly transferable into molecule space, it is not so unreasonable for molecules close to the borders of the space (supposedly outliers) to be selected. Therefore, the *p*-dispersion-sum measure may not be very suitable either.

The *p*-Center Measure. A different diversity measure can be derived from the *p-center problem*.[13,14] The function to optimize is the following

$$\delta_c(S) = 1 - \max_{1\le i\le n} \min_{1\le j\le p, i\neq j} d(u_i, u_j),\ u_i \in I, u_j \in S \qquad (5)$$

In contrast to the other definitions so far, the *p*-center function cannot be solely computed with the selected objects, rather the whole set of objects I needs to be available. First, all available objects are divided into two sets of selected and unselected objects. Next, the minimum distance from each object (selected or not) to any selected object is computed. Because the goal of the *p*-center problem is to minimize the largest of these minimal distances, we define the diversity as 1 minus the maximum. Optimizing the *p*-center problem means choosing the selected objects in such a way that each object (from the set of all objects!) is as close as possible to at least one selected object. Using the 2-D example from above, this leads to a very even distribution of selected points over the whole space (Figure 4). One important difference to the other general diversity measures is the complexity of computing the diversity. Whereas the former require $O(p^2)$ computation (because they are a function of the selected objects only), *p*-center requires $O(pn)$ computations, which become significantly larger, if $p \ll n$ (the standard case in at least the application in vHTS).

The *p*-Dispersion-Min-Sum Measure. A number of problems can be encountered when applying *p*-dispersion: a single small distance can ruin diversity; in the *p*-dispersion-sum measure, many small distances can occur; and the *p*-center measure is more complex to compute. However, there is a fourth definition, which involves maximizing the sum of minimal distances to circumvent the above-mentioned deficiencies

$$\delta_{dms}(S) = \sum_{i=1}^{p} \min_{1\le j\le p, i\neq j} d(u_i, u_j),\ u_i, u_j \in S \qquad (6)$$

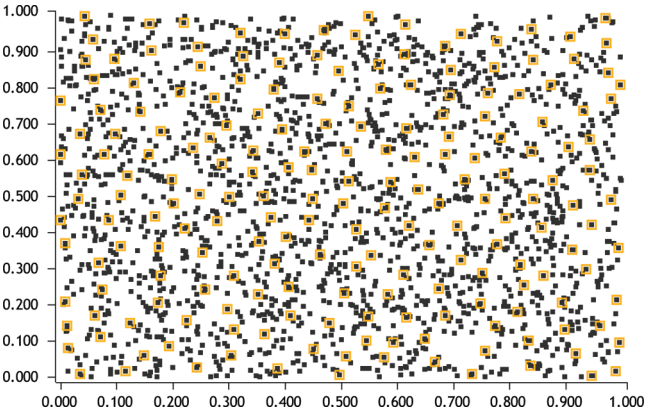


Figure 4. Subset of 200 points taken from a set of 2000 randomly distributed points, representing a near-optimal solution for the *p*-center problem.

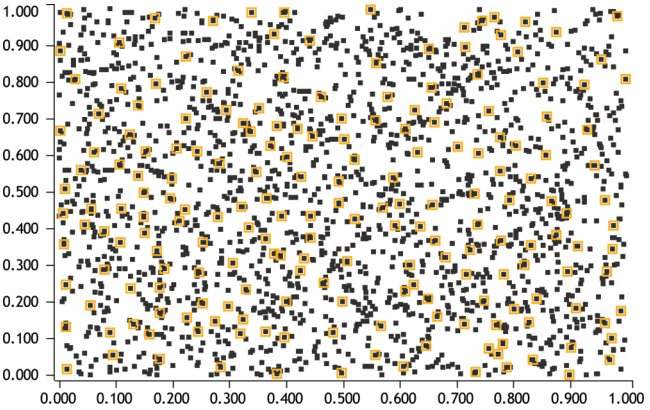


Figure 5. Subset of 200 points taken from a set of 2000 randomly distributed points, which represent a near-optimal solution for the *p*-dispersion-min-sum problem.

This means that the distances from each object to its nearest neighbor are summed. We refer to this as the *p-dispersion-min-sum* problem.

Not only is the influence of few small distances reduced, but also many small distances will heavily influence diversity, as in the corners of Figure 3, and indeed, optimizing the same distribution as before but for eq 6 results in much better coverage of the space (Figure 5). This definition of diversity has already been used in the well-known OptiSim program[4] and was mentioned in the article about the hypercube approach[5] as a replacement for non-vector spaces.

One can think of quite a lot of other diversity measures, but we will continue with the ones presented above because they are either commonly used and/or are easily motivated.

SCORE ESTIMATION

Not only is a diversity measure needed for MSDS, but also a score for each molecule. This can either be a real measured value or, more likely, a prediction of the molecule's activity. We will not go into detail here because the estimation of activity is a huge research area of its own (detailed overviews can be found in refs 15−17). For all later experiments, we use the measured activities as scores, although this cannot be performed in pratice of course. However, this does not (or only slightly) influence the validation

dx.doi.org/10.1021/ci100426r |*J. Chem. Inf. Model.* 2011, 51, 237–247

of the presented methods, given that the chosen score estimation is sufficiently reliable.

MULTIOBJECTIVE DIVERSITY SELECTION

As already mentioned, Maximum-Score Diversity Selection is not only the selection of a maximally diverse subset but is also extended by adding a second objective of maximizing the score of the selected molecules. In contrast to the diversity objective, which is based on a pairwise relation, this objective is easier to address because the score is a property of each single molecule.

MSDS can be formulated as follows: On the one hand, there is a score for each molecule, and ideally the highest-scoring molecules are selected. On the other hand, there are pairwise distances between all molecules, and the selected molecules should maximize the chosen diversity measure. The number of selected molecules is fixed beforehand. Formally, this is written as

$$\text{Maximize } f_1(S) = \sum_{u \in S} \sigma(u) \tag{7}$$

$$\text{Maximize } f_2(S) = \delta(S) \tag{8}$$

where $|S| = p$, σ returns the score of an object, and δ is any reasonable diversity function. This is the formulation of a classical multiobjective optimization (MO) problem. Because there is usually no single "optimal" solution in MO, the terms *non-dominated* or *Pareto optimal* solution are commonly used. Both describe a solution that is optimal in the sense that all other solutions are worse in at least one objective. All non-dominated solutions form the so-called *Pareto front*. The user can then select the preferred solution from the front, emphasizing one (or more) of the objectives. He may, for example, prefer more active subsets over more diverse subsets or vice versa.

Various approaches can be used to solve MO problems,[18] the most simple probably being evolutionary approaches such as genetic algorithms (GA). Usual GA implementations can only deal with a single objective function, but there are also special algorithms for MO problems that evolve a set of non-dominated solutions in one run. Examples are SPEA2,[19] NPGA[20], and NSGA-II.[21] In the Algorithms for MSDS section, we briefly describe our implementation with NGSA-II for solving MSDS.

Another way of solving MO problems is to determine a weighting for the different objectives beforehand and build a single objective function. On the one hand, this yields only a single solution for the chosen weighting, and therefore, many solutions with different weightings usually have to be calculated in order to find a set of nondominated solutions. On the other hand, it is possible to use a wide range of single-objective algorithms without having to modify them. The weights on each objective let the user indicate preferences for the corresponding objectives. For MSDS, the combined function is the following

$$\text{maximize } f_c(S) = (1-\alpha)\cdot\delta(S) + \alpha\cdot\sum_{u \in S}\sigma(u) \tag{9}$$

where α is a user-defined parameter in the range of $[0,1]$. Large α lead to subsets with more highly scored molecules, whereas small α emphasize more diverse subsets. Using this single objective function makes it possible to apply special algorithms.

Maximum-Score Diversity Selection Is NP-Hard. One may wonder why we mentioned genetic algorithms above. The reason for this is that the optimization of all general diversity measures presented above, i.e., selecting a subset that maximizes the function, is an NP-hard problem. The well-known maximum clique problem[22] easily reduces to all three dispersion-problems[8], and the dominating set problem reduces to *p*-center[14] (we omit details for the proofs). This essentially means that exact solutions can only be computed for very small problems. Current state-of-the art exact algorithms for, for example, the *p*-dispersion-sum problem currently find solutions for less than 100 objects[23] only. However, in the real world, scenarios such as chemoinformatics selections range from a few hundred molecules out of a few thousand up to 500000 out of 5000000 molecules.

The above results only hold for the task of pure diversity selection, i.e., without taking the activity objective into account. However, it is easy to reduce any of the "pure" problems to the multiobjective MSDS. For example, we take the NP-hard *p*-dispersion-sum problem, which is also known as the maximum edge-weight clique problem. As the name indicates, the diversity selection problem is modelled as a graph problem: Each molecule is a node in the graph, and there are edges between all pairs of nodes labeled with the corresponding molecules' distances. Thus, we have a complete graph for which a clique of size p is searched. Finding a clique of size p in a complete graph is easy, but finding the clique with the largest sum of edge labels is an NP-hard optimization problem. By also assigning a label to each node (the score of the molecule) and selecting a clique that has the largest sum of both edge *and* node labels, we end up with the MSDS problem. If all node labels are equal, the original *p*-dispersion-sum problem is a special case of MSDS, proving that MSDS is no easier than *p*-dispersion-sum. Thus MSDS is also an NP-hard optimization problem. Please note, that this does *not* imply, that *all* instances of MSDS are hard to solve (e.g., instances where node labels are much larger than edge labels are easy because only the node labels have an impact on the target function), but MSDS in general is NP-hard.

Therefore, heuristics are needed to find good or near-optimal solutions. In both refs 8 and 14, special heuristics are presented that find good solutions for *p*-dispersion and *p*-center, respectively. The algorithm for *p*-dispersion can easily be modified to solve *p*-dispersion-sum as well. The heuristic for *p*-center even gives an approximation bound, in that its solutions are at most a factor of 2 from the optimal solution. No special heuristics are currently known for *p*-dispersion-min-sum. In any case, meta-heuristics, such as the already mentioned genetic algorithms or simulated annealing,[5] can be applied to find suitable subsets.

ALGORITHMS FOR MSDS

Now that we have discussed several measures for diversity and motivated the use of heuristic approaches, we present several algorithms that can be applied to MSDS. The first two heuristics were designed to optimize the respective diversity measures only. In order to be applicable to MSDS, the input data, which is essentially the complete distance matrix of all molecules, needs to be preprocessed. Because the distance matrix is the same as a complete edge-labeled graph, we decided to incorporate the node labels, i.e., the molecules' score values, into the adjacent edge labels. By using a weighting parameter α, the focus between activity and diversity can be adapted. By automatically running

Table 1. Algorithm 1: *p*-Dispersion Optimization Heuristics

Input: A complete edge-labeled graph G, the number of elements to select k
Output: A subset of objects optimizing the p-dispersion(-sum) measure
$E' \leftarrow E$ sorted in ascending order by labels;
Sel $\leftarrow V_G$;
while |Sel| $>k$ **do**
  $\{u,v\} \leftarrow pop(E')$;
  if $u \in$ Sel **then** Sel $\leftarrow$ Sel $- \{u\}$;
  else if $v \in$ Sel **then** Sel $\leftarrow$ Sel $- \{v\}$;
end
for each $u \in$ Sel **do**
  for each $v \in V_G$ — Sel **do**
    Sel$' \leftarrow$ Sel $- \{u\} + \{v\}$;
    if δ(Sel$'$) $> \delta$(Sel) **then** Sel $\leftarrow$ Sel$'$;
  end
end
return Sel

the algorithm multiple times with varying values of α, a set of solutions, similar to an approximated Pareto front, is created. The modified distances are determined by

$$d_{\text{new}}(u_i, u_j) = (1 - \alpha)[\sigma(u_i) + \sigma(u_j)] + \alpha d(u_i, u_j) \qquad (10)$$

Erkut's Heuristic for *p*-Dispersion and *p*-Dispersion-Sum. In his paper on the p-dispersion problem,[8] Erkut presented a heuristic approach. Starting with the complete set of all n nodes in the graph, nodes are iteratively deleted until only p nodes remain. The decision of which nodes to remove is based on a list of all edges in the graph, sorted in ascending order by their labels (i.e., distance). The shortest edge is removed from the list, and one of its two nodes (the choice is arbitrary) is deleted from the set. This process is repeated until only p nodes remain. This set is the starting point for the second phase, in which a local search is performed. Each node currently included in the solution set is exchanged with all currently unselected nodes, one at a time. If any such swap improves the objective function δ (either p-dispersion or p-dispersion-sum), it is accepted, otherwise it is rejected. The algorithm in Table 1 shows this procedure in pseudocode.

Both construction and local improvement take at most $O(n^2 \log n)$ time. In practice about half the time is usually spent in fully sorting the list of edges, which offers slight potential for improvement because only about half the edges (for our data sets) are actually needed. The experiments show that this simple heuristic performs surprisingly well for both p-dispersion and the p-dispersion-sum measures.

Hochbaum and Shmoys Heuristic for *p*-Center. In addition to the NP-hardness proof of the p-center problem, Hochbaum and Shmoys developed an approximate algorithm, which guarantees that its solutions are twice the optimal solution at most.[14]

The algorithm itself is as simple as the one for p-dispersion, see Algorithm 2 in Table 2. However, the reason why it works is much harder to understand as it is motivated by the complexity proof, which includes transformations into the dominating set and the maximum independent set problem. Therefore, here we only present the algorithm and refer to the original publication for details.

The runtime of this algorithm is $O(n^2 \log n)$, the same as for Erkut's p-dispersion heuristics.

Table 2. Algorithm 2: *p*-Center Optimization Heuristics

Input: A complete edge-labeled graph G, the number of elements to select k
Output: A subset of objects optimizing the p-center measure
low $\leftarrow$ 1;
high $\leftarrow |V_G| \times (|V_G| - 1)/2$;
while high $>$low $+1$ **do**
  mid $\leftarrow$ (high + low)/2;
  max $\leftarrow$ length of midshortest edge in G;
  $G' \leftarrow G$ without edges longer than max;
  Sel $\leftarrow \emptyset$;
  Avail $\leftarrow V_G$;
  for each $u \in$ Avail **do**
    Sel$\leftarrow$Sel $\cup \{u\}$;
    for each $\{u, v\} \in E_{G'}$ **do**
      Avail $\leftarrow$ Avail — $\{v\}$;
      for each $\{v, w\} \in E_{G'}$ **do**
        Avail $\leftarrow$ Avail — $\{w\}$;
      end
    end
  end
  if | Sel| $\leq k$ **then**
    high $\leftarrow$ mid;
    Sel$' \leftarrow$ Sel;
  else low $\leftarrow$ mid;
end
return Sel$'$

Genetic Algorithms. Genetic algorithms in general are an easy and simple way of solving optimization problems, as only a suitable genetic representation of solutions is needed, plus a function, which is subsequently optimized. Fortunately, multiobjective genetic algorithms only differ from the single-objective variants in the way individuals/solutions are selected for the next generation. The encoding of potential solutions as chromosomes, genetic operators such as crossover and mutation and fitness functions (for each single objective) can be used without any modifications.

For the experiments NSGA-II (Nondominated Sorting Genetic Algorithm II[21]) was used. With the exception of problems with many objectives, where it is usually outperformed by SPEA2,[19] it is still one of the best (and simplest) multiobjective genetic algorithms. The main challenge of multiobjective GAs is the selection of individuals for reproduction because in contrast to single objective problems there is no global ranking of all individuals on the basis of which selection could occur. Instead, NSGA-II partitions the whole population into several fronts. The first front is formed by all nondominated individuals, which are removed. The second front consists of all remaining individuals that are now nondominated, and so on. The front, in which an individual has been placed, is then used as a rank during selection. In unbiased tournament selection,[24] which is used in our implementation, if two individuals compete in a tournament, the one with the lower rank is chosen. If both have the same rank, the so-called *crowding distance*, which measures the density of individuals, is used to break ties. In order to maintain a good spread of solutions, individuals in less dense regions of the search space are preferred over ones with many close neighbors. However, in the Experiments section, it becomes clear that even this technique cannot prevent the

solutions from concentrating on only a limited region of the search space.

An individual in the current case is a combination (in mathematical terms) of p distinct items from the complete set of n. They can either be represented as a bit string of length n, where exactly p bits are set corresponding to the selected items or as an integer array of length p containing the indices of the selected items. Genetic operators exist for both representations.[25] The fitness function for the score objective simply sums the scores of all selected items, whereas for the diversity objective, $\delta(S)$ is evaluated. It is easy to see that usually $\delta(S)$ has a different value range than the score sum (e.g., p-dispersion is much smaller, p-dispersion-sum is much greater). Therefore, it is even more desirable to present a set of nondominated solutions to the user becausee the parameter α in a combined objective function does not only need to control the balance between the two objectives but would also have to level out the different value ranges of the two objectives. When using multiobjective evolutionary algorithms, the parameter α is not needed at all because they compare each objective independently and only use the nondominated relation between possible solutions for ranking and selection.

Score Erosion. Because the complexity of Erkut's and Hochbaum and Shmoys' heuristic is at least quadratic in the number of items, it is still quite time-consuming to process data sets with several thousands of objects. Therefore, we propose a much faster algorithm for MSDS.

The motivation for the algorithm comes from its application in HTS but is equally applicable to any other MSDS problem. In virtual HTS, once the molecules' scores have been calculated by various virtual screening programs, the molecules are sorted according to their scores. The usual approach would be to select the "top p" molecules; however, this completely ignores the diversity objective. Therefore, reasoning suggests reducing the score of all molecules i that are similar to the just selected molecule s, by a certain amount after each molecule. The force of this erosion depends on the distance between i and s and a parameter β

$$\sigma_{t+1}(u_i) = (1 - e^{-\delta(u_i,s)/\beta}) \cdot \sigma_t(u_i) \qquad (11)$$

Algorithm 3 in Table 3 shows the pseudocode for this approach.

First, the highest ranked molecule is selected. Next the distance to all remaining molecules is computed, and the scores of all molecules are decreased in proportion to the distance. The user-defined factor β controls how much the score is eroded. Using a large value for β decreases the scores of similar molecules quite extensively; therefore, the diversity objective is favored, whereas a small β focuses more on the activity objective. After the erosion step, the molecules are reranked on the basis of the changed score values. These operations are iteratively performed until p elements have been selected. The experiments demonstrate that this algorithm performs as well as the other heuristics for some diversity definitions and has a better complexity of only $O(pn)$ compared to $O(n^2 \log n)$.

In another article,[26] an algorithm similar to Score Erosion is presented. The greedy algorithm the authors developed uses a similar update formula as eq 11 but instead of multiplying the old value with the redundancy value ($\equiv$ diversity) to the selected pattern, they subtract it from the significance value ($\equiv$ score)

$$\sigma_{t+1}(u_i) = \sigma_t(u_i) - \beta \times d(u_i, u_s) \qquad (12)$$

Table 3. Algorithm 3: Score Erosion

Input: A list with activities A, the complete distance matrix D, the number of items to select k, and the weighting parameter β
Output: A subset of objects that, depending on β, is a good trade-off between activity and diversity

```
Sel = ∅;
best ← max_k A [k];
for i ← 1 to k do
    Sel ← Sel ∪ {best};
    A [best] = −∞;
    best′ = best;
    for j ← 1 to n do
        A[j] ← A[j] · (1 − e^(−(D[best,j]/β)));
        if A [j] > A [best′] then best′ ← j;
    end
    best ← best′;
end
return Sel
```

At first glance, this seems a little odd because activity and distance are two different concepts, and subtracting one from the other does not have have an intuitive meaning. However, we also tried this update rule in our experiments, and as we describe in the next section, it appears to work quite well in some cases.

EXPERIMENTS

Several experiments were carried out in order to show how good the solutions are that are produced by the different approaches described in the previous sections. Two tests were performed with publicly available data sets: a small one from BindingDB.org and a larger one from PubChem. For the third test, an in-house data set from Nycomed was used. All experiments were performed on an eight-core Xeon workstation inside the KNIME data analysis framework.[27]

Before a detailed discussion on the experiments, we want to mention an important fact about the genetic algorithm. In the very first experiments, the generated Pareto fronts had very poor coverage compared to that of the other heuristics. Only a very small part in the middle was discovered, and solutions to both borders (high activity/high diversity) were lacking completely. We partially solved this problem by adding the solution with the highest activity to the initial population. Because this is a Pareto-optimal solution, it will survive in all generations and help in better covering the activity-accentuated part of the front. However, diverse solutions are still lacking in most cases, as we show below.

CDK2 Data Set. The first data set that was used is publicly available from BindingDB.org,[28,29] consisting of 1376 molecules, which have been tested for their activity against the CDK-2 protein. The data set contains the molecules' activities as IC_{50} values and their 2-D structure. In our experiments, we set the subset size to 137 (10% of the database), resulting in a search space of about $1.8 \cdot 10^{193}$ possible solutions.

One crucial parameter on molecular data sets is the choice of the distance function. For the experiments, the size of the maximum common substructure MCSS was used as a measure of distance/similarity. The distance between two molecules u and v is 1 minus the squared size (sum of nodes and edges) of their maximum common substructure *mcss* divided by the product

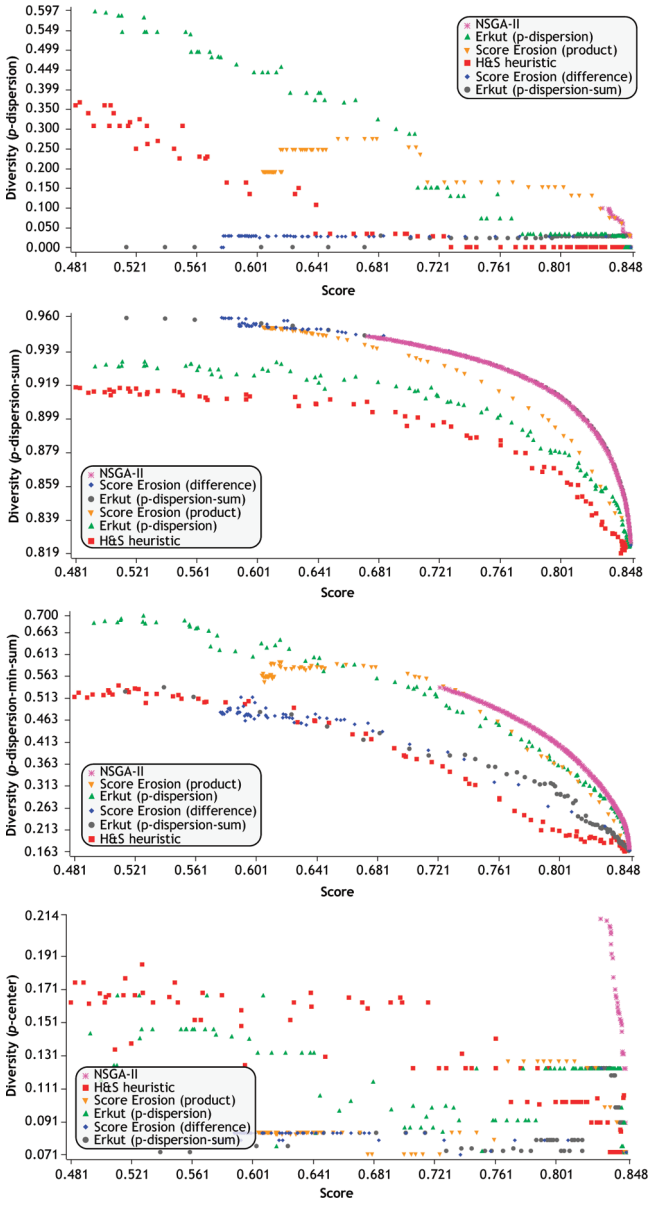


Figure 6. Approximations of the Pareto front on the CDK2 data set for the different algorithms and diversity measures.

of the sizes of both molecules

$$\delta(u,v) = 1 - \frac{\text{size}(\text{MCSS}(u,v))^2}{\text{size}(u) \times \text{size}(v)} \qquad (13)$$

The MCSS was computed with the substructure miner MoSS.[30,31] Searching for frequent substructures in two molecules is equivalent to computing (one of) their MCSS.

The remaining settings for the different algorithms are as follows:

Multiobjective GA: Population size = 300, 2000 generations, mutation rate = 10%, uniform crossover

***p*-center**: 100 samples with α uniformly sampled from 0 to 1 (inclusive)

Score Erosion: 100 samples, with β uniformly sampled from 0 to 1 (inclusive)

Erkut: 100 samples with α uniformly sampled from 0 to 1 (inclusive)

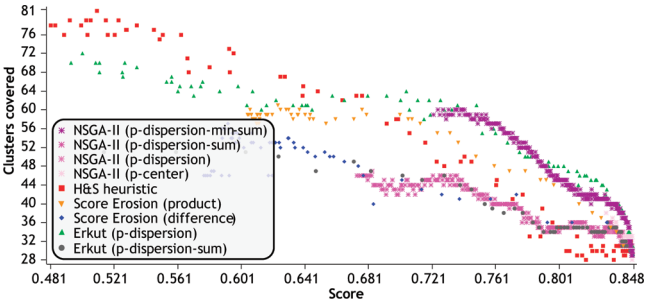


Figure 7. Number of covered clusters on the CDK2 data set plotted against the activity value of all generated subsets.

Figure 6 shows the results on the CDK2 data set. Several important conclusions can be drawn from the diagrams:

- The approximated Pareto fronts for *p*-dispersion and *p*-center are very scarcely populated. This is not a big surprise because the minimum or maximum functions inside their definitions evaluate to the same value for many different subsets, as long as the one minimal/maximal pair of molecules is the same.
- In *p*-dispersion, Erkut's heuristic and Score Erosion (with the product update rule) are able to find good solutions, whereas all other algorithms fail to do so. Except for the genetic algorithm, this is not remarkable, as they are not designed to do so. The poor quality of the GA's solutions is a bit disappointing, though.
- In *p*-dispersion-sum, both Erkut's heuristic and Score Erosion (with the difference update rule) find a good approximation of the front, with the GA following closely behind.
- *p*-dispersion-min-sum is best solved by Erkut's heuristic for the *p*-dispersion problem, with Score Erosion and in parts the genetic algorithm taking second place.
- Except for the genetic algorithm, neither heuristic is able to find any good solutions to the *p*-center problem, not even its special heuristic.

The main motivation for MSDS was that an optimized subset is more diverse than pure "top-*p*" selection and is more active than a random or purely diverse subset. In HTS, one is usually interested in the number of molecule clusters that are covered. The more the better because each cluster is a potential independent starting point for further optimization. Therefore, all molecules in the CDK2 data set were assigned to clusters (or singletons) in the same way as for all real projects at Nycomed (aided by the ClassPharmer program[32]), yielding 73 clusters and 31 singletons. Then, for each generated subset, the number of clusters that are covered by its molecules was determined. Figure 7 shows the subsets's normalized activity plotted against the clusters count.

These results are quite interesting, since the Hochbaum and Shmoys heuristics discovers the most clusters by far, although with very low scores. However, this does not imply that the *p*-center measure is best suited for molecules: First, the *p*-center values computed by the heuristics were rather low compared to, for example, the genetic algorithm. Second, the subsets generated by the GA with the *p*-center measure cover far fewer clusters, although its *p*-center values are highest. However, one has to keep in mind that this behavior is dependent on the chosen clustering approach, and there may exist other sensible clusterings for which the coverage is different. The remaining observations are similar to what was already discussed above: Score

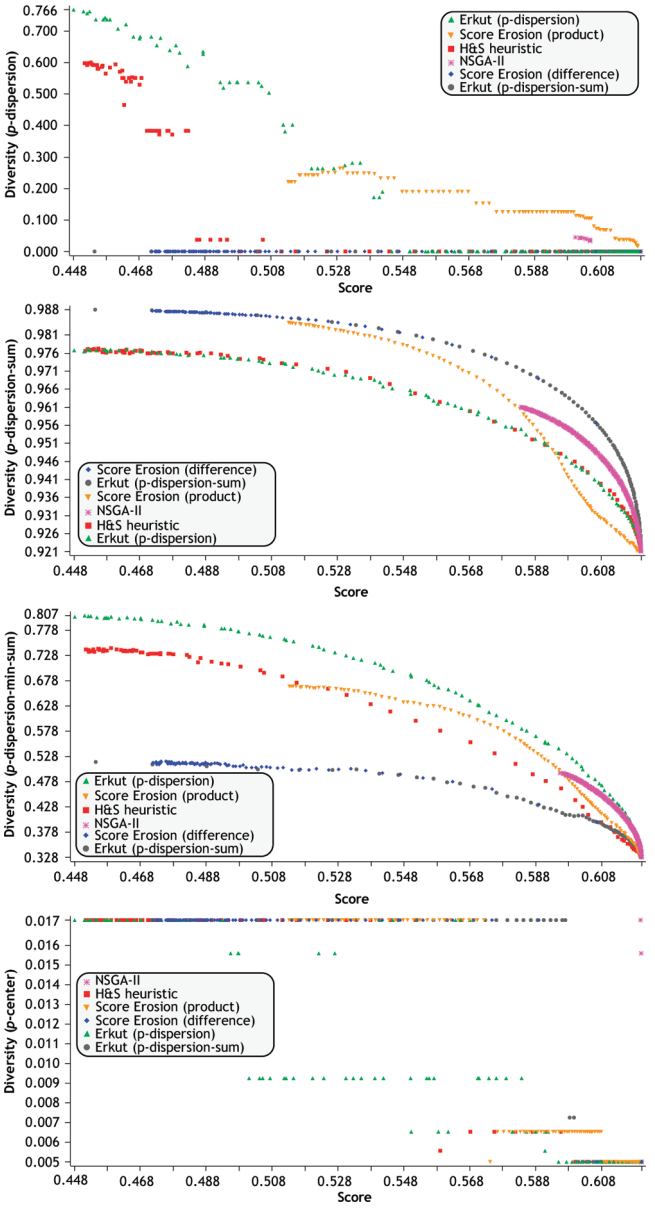


Figure 8. Approximations of the Pareto front on the AID 884 data set for the different algorithms and diversity measures.

Erosion with the product update rule works fairly well as does Erkut's *p*-dispersion heuristic.

AID 884 Data Set. The second publicly available data set is PubChem's AID 884 bioassay. It consists of 13082 tested molecules. We filtered out all molecules that did not have a value for the activity at 0.457 μM, which we took as score values for the molecules (because the most molecules had a value there). For the remaining 12156 molecules, we computed their pairwise distances in the same way as for the CDK2 data set and performed MSDS with mostly the same settings. Only the subset size p was set to 1000, and the number of generations for the genetic algorithm was reduced to 33 because otherwise the runtimes would have been too long (several days per run).

Figure 8 shows the results on the AID 884 data set. The results are qualitatively comparable to the CDK2 data set; however, some facts are noteworthy. First, the genetic algorithm's approximation of the Pareto front is significantly worse. This can be due

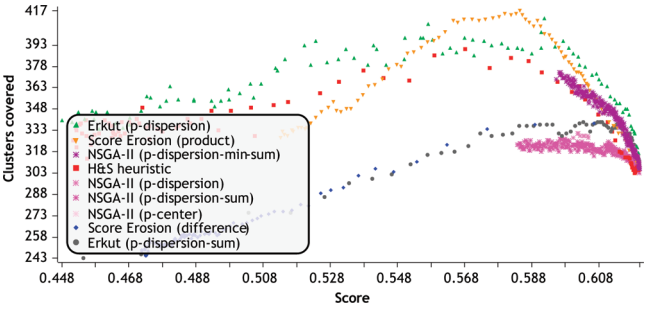


Figure 9. Number of covered clusters on the AID 884 data set plotted against the activity value of all generated subsets.

to the much larger number of possible subsets ($6.77 \cdot 10^{1498}$) because the AID 884 data set is about 10 times the size of the CDK2 data set. Also, the results from the *p*-center measure appear to be a little unusual, as only about five classes of subsets with different *p*-center values are found. Also, the curves representing the approximated Pareto fronts have degenerated. Still, the genetic algorithm finds the best solutions in this case, too.

We also investigated cluster coverage on the AID 884 data set following the same procedure as above. As shown in Figure 9, the results are different in comparison to the CDK2 data set. As opposed to the Hochbaum and Shmoys heuristic, this time Score Erosion and Erkut's heuristics cover significantly more clusters than any other approach. The fact that for Score Erosion the number of clusters decreases below an activity value of about 0.58 is due to an inappropriate choice of the β parameter's range. The same holds for almost all other heuristics. Concerning the genetic algorithm, optimizing the *p*-dispersion-min-sum measure gives the best results, which is again a clear indication that this measure is the preferred one for molecular data sets.

In-House Data Set. The third data set is an in-house data set from Nycomed consisting of 1572 molecules with IC_{50} values. The protocol was identical to the AID 884 data sets, except for a subset size of 150; however, the difference lies in the way the clusters have been defined. Whereas for the former two, the results from ClassPharmer were used directly, for this data set they were refined by the project group leading to 86 clusters (including 13 singletons). The approximated Pareto fronts are shown in Figure 10.

It can be seen that qualitatively the results are comparable to the other data sets. More interesting is the analysis of the cluster coverage, which is shown in Figure 11. It is apparent that in this case as well, the *p*-dispersion-min-sum measure correlates the best with the number of covered clusters as can be seen from the solutions found by the genetic algorithm with the corresponding diversity measure. Even better coverage is obtained by Score Erosion. The reason that this algorithm also produces many inferior solutions is once again attributed to the choice of the β parameter. It was steadily increased from 0 to 1, and higher values of β seem to lead to worse solutions. In practice, this is not a real problem because this effect is easily noticeable and can be compensated by an appropriate choice of sampling β. Because Score Erosion is by far the fastest heuristic, as shown in the next section, it is easy to experiment with different parameter settings.

Performance Analysis. Looking at the absolutes runtimes of the above experiments, one can already see that Score Erosion is the fast algorithm: It takes about 47 s per iteration on the AID 884 data set, whereas Hochbaum and Shmoys heuristic takes 114 s per iteration and Erkut's heuristic even 470 s (all experiments were performed on an eight-core Intel system running at

dx.doi.org/10.1021/ci100426r |*J. Chem. Inf. Model.* 2011, 51, 237–247

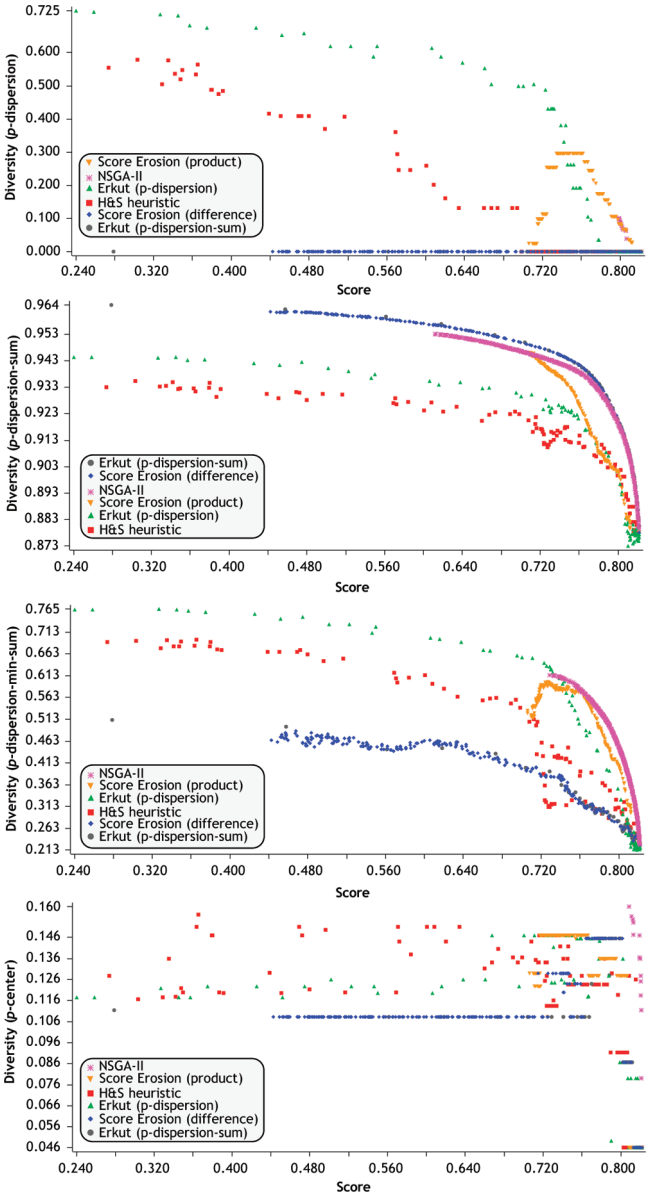


Figure 10. Approximations of the Pareto front on the in-house data set for the different algorithms and diversity measures.

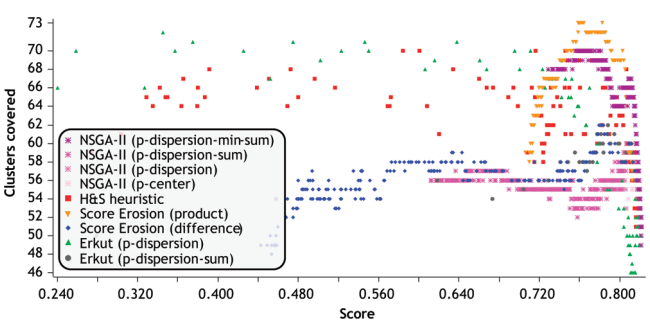


Figure 11. Number of covered clusters in the in-house data set plotted against the activity value of all generated subsets.

1.83 GHz with 16GB of main memory). Also on the much smaller CDK2 data set the numbers are likewise: 0.15 s for Score Erosion, 0.53 s for Hochbaum and Shmoys heuristics, and 0.94 s

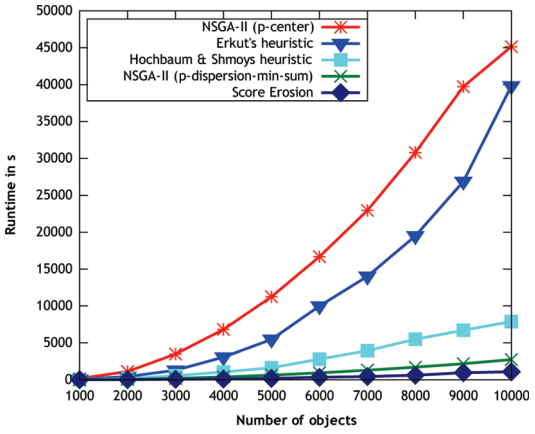


Figure 12. Runtime behavior of the heuristics with increasing data set sizes and a fixed percentage of selected objects.

for Erkut's heuristic per iteration. The runtimes for the genetic algorithm were considerably higher and also depend heavily on the selected number of generations.

In order to systematically verify the claimed runtime complexities of the three problem-specific heuristics, we performed additional experiments on artificial data sets of sizes 1000−10000. We did not use real world data sets here because it is relatively difficult to select a subset from a real world molecular data set that does not change the search space structure. This is quite a simple process with artificial data sets because three activity spots are created with the same parameters (location and width) each time. This ensures that the structure remains almost the same for all experiments, and only the number of objects varies. The results are depicted in Figure 12. The reported runtimes for the genetic algorithm result from 8 parallel threads; therefore, they are not directly comparable to the other heuristics. It is however quite clear that the p-center measure is much more time-consuming to evaluate than the other measures (which are identical in complexity). Erkut's heuristic nicely shows the claimed quadratic increase [in fact $O(n^2 log\ n)$] in runtime. In addition the runtimes for Hochbaum and Shmoys' heuristic and for Score Erosion increase quadratically in the number of objects but at a much lower rate. The fact that Score Erosion also shows a quadratic increase lies in the fact that its complexity is $O(pn)$, and because p was chosen to be $0.1n$, this also yields $O(n^2)$.

The second part of the runtime analysis investigates the effect of p, the number of selected items, on a constant-sized data set of 2000 objects. For Score Erosion, linear increase in runtime over the whole range of the experiment is expected, whereas for the p-center heuristic, the runtimes should stay almost constant as it examines all edges in any case, regardless of how many objects are to be selected. Erkut's algorithm has to be examined more thoroughly. For small p, substantial time is spent creating the sorted list of edges, which requires $O(e \log e) = O(n^2 \log n)$ time. During the optimization step, each currently selected node is interchanged with each nonselected node, and the win (or loss) of this exchange has to be computed by looking at the weights of all adjacent edges. This leads to a complexity of $O(p(n - p)p) = O(p^2 n - p^3)$. This means that by increasing p the increase of runtime should slow down or even reverse. Figure 13 clearly demonstrates that our conjectures are correct. Still, Score Erosion and the Hochbaum and Shmoys heuristic are much faster

dx.doi.org/10.1021/ci100426r |J. Chem. Inf. Model. 2011, 51, 237–247

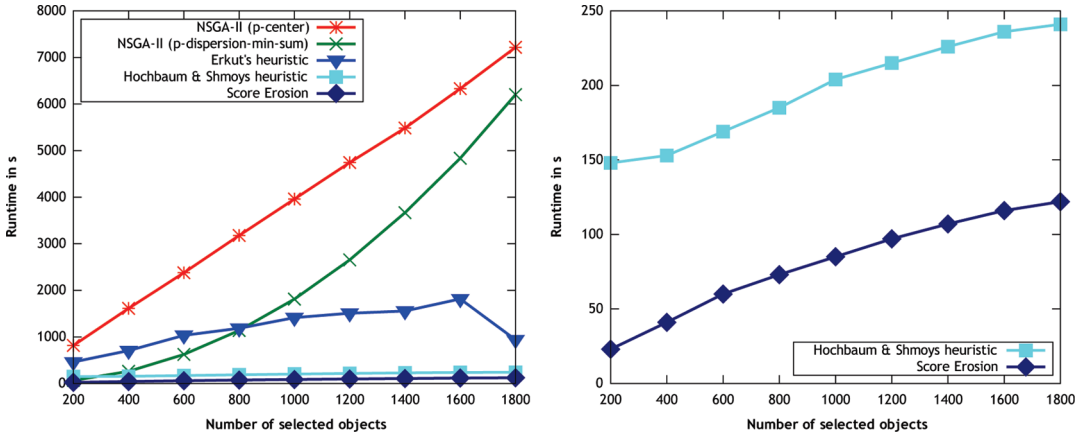


Figure 13. Runtime behavior of all heuristics with increasing number of selected items and a fixed data set size. The right diagram shows a section from the left diagram.

than Erkut's, and in practice, the selection of almost all objects from the complete set is rather uninteresting.

RELATED WORK

As we already mentioned, diversity selection has a long tradition in chemoinformatics. Early approaches applied clustering techniques on descriptors for the molecules.[33] The cluster centers were then used as a set of divers molecules. Already then it was observed that genetic algorithms in general do not find the best solutions compared to specialized algorithms. Nevertheless, they were successively used in several other articles about denovo design of diverse libraries.[34,35] However, the evolved individuals are not molecules by themselves but rather several building blocks that are later on synthesized to molecules. In the latter publication, diversity selection has even been combined with other objectives rendering it a multiobjective optimization problem. Another approach to diversity selection are greedy algorithms that iteratively choose new molecules that have the greatest distance to all already selected molecules.[4,36] Many diversity selection approaches bear the problem that it is unclear which specific diversity funtions they are optimizing, i.e., they are not relying on a formula that can be used to score the diversity of any subset of molecules. Exceptions are the hypercube measure[5,6] and the *p*-dispersion-min-sum measure.[4]

CONCLUSIONS

In this article, we have introduced the concept of Maximum-Score Diversity Selection and motivated its usefulness in early drug discovery. The experiments on several molecular data sets have shown that when a subset of molecules is carefully selected (and does not simply consist of the top-*p* molecules) many more clusters are able to be covered. With respect to the selection of more diverse subsets, we discussed several measures for diversity, with *p*-dispersion-min-sum being the most intuitive and presumably also the most appropriate measure for molecular data sets. We have presented several approaches for finding high-score but diverse subsets, among which our novel Score Erosion algorithm is clearly the fastest heuristic and finds solutions of comparable quality. We also showed that using genetic algorithms, which are quite popular especially for multiobjective problems, can lead to insufficient coverage of the solution space, even if several tweaks are applied.

AUTHOR INFORMATION

Corresponding Author

*E-mail: Thorsten.Meinl@uni-konstanz.de.

ACKNOWLEDGMENT

We thank Nycomed for partial funding and support of this project, especially Olaf Nimz and Andrea Zaliani, as well as Ulrik Brandes for fruitful discussions.

REFERENCES

(1) Matter, H. Selecting optimally diverse compounds from structure databases: A validation study of two-dimensional and three-dimensional molecular descriptors. *J. Med. Chem.* **1997**, *40*, 1219–1229.

(2) Selassie, C. R. D. History of Quantitative Structure−Activity Relationships. In *Burger's Medicinal Chemistry and Drug Discovery*, 6th ed.; Abraham, D. J., Ed.; JohnWiley & Sons, Inc.: New York, 2003; Vol. 1: Drug Discovery; Chapter 1, pp 1−48.

(3) Pickett, S. D.; Mason, J. S.; McLay, I. M. Diversity profiling and design using 3D pharmacophores: Pharmacophore-derived queries (PDQ). *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 1214–1223.

(4) Clark, R. D. OptiSim: An extended dissimilarity selection method for finding diverse representative subsets. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 1181–1188.

(5) Agrafiotis, D. K. Stochastic algorithms for maximizing molecular diversity. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 841–851.

(6) Cummins, D. J.; Andrews, C. W.; Bentley, J. A.; Cory, M. Molecular Diversity in chemical databases: Comparison of medicinal chemistry knowledge bases and databases of comercially available compounds. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 750–763.

(7) Moon, D. I.; Chaudhry, S. S. An analysis of network location problems with distance constraints. *Manage. Sci.* **1984**, *30*, 290–307.

(8) Erkut, E. The discrete *p*-dispersion problem. *Eur. J. Oper. Res.* **1990**, *46*, 48–60.

(9) Dellacroce, F.; Grosso, A.; Locatelli, M. A heuristic approach for the max−min diversity problem based on max-clique. *Comput. Oper. Res.* **2009**, *36*, 2429–2433.

(10) Pisinger, D. Upper bounds and exact algorithms for *p*-dispersion problems. *Comput. Oper. Res.* **2006**, *33*, 1380–1398.

(11) Park, K.; Lee, K.; Park, S. An extended formulation approach to the edge-weighted maximal clique problem. *Eur. J. Oper. Res.* **1996**, *95*, 671–682.

(12) Aringhieri, R.; Cordone, R.; Melzani, Y. Tabu Search versus GRASP for the maximum diversity problem. 4OR. *Q. J. Oper. Res.* **2008**, *6*, 45–60.

(13) Hakimi, S. L. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Oper. Res.* **1965**, *13*, 462–475.
(14) Hochbaum, D. S.; Shmoys, D. B. A Best possible heuristic for the k-center problem. *Math. Oper. Res.* **1985**, *10*, 180–184.
(15) Katritzky, A. R.; Kuanar, M.; Slavov, S.; Hall, C. D.; Karelson, M.; Kahn, I.; Dobchev, D. A. Quantitative correlation of physical and chemical properties with chemical structure: Utility for prediction. *Chem. Rev.* **2010**, *110*, 5714–5789.
(16) Todeschini, R.; Consonni, V. Molecular Descriptors for Chemoinformatics. In *Methods and Principles in Medicinal Chemistry*; Mannhold, R., Kubinyi, H., Folkers, G., Eds.; Wiley-VCH: Weinheim, Germany, 2009; Vol. I: Alphabetical Listing.
(17) Todeschini, R.; Consonni, V. Methods and Principles in Medicinal Chemistry. In *Molecular Descriptors for Chemoinformatics*; Mannhold, R., Kubinyi, H., Folkers, G., Eds.; Wiley-VCH: Weinheim, Germany, 2009; Vol. II: Appendices, References.
(18) Branke, J., Deb, K., Miettinen, K., Słowiński, R., Eds.; *Multiobjective Optimization: Interactive and Evolutionary Approaches (Lecture Notes in Computer Science)*; Springer: Berlin, 2008; Vol. 5252.
(19) Zitzler, E.; Laumanns, M.; Thiele, L. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*; Swiss Federal Institute of Technology: Zurich, Switzerland, 2001.
(20) Horn, J.; Nafpliotis, N.; Goldberg, D. E. A. Niched pareto genetic algorithm for multiobjective optimization. *IEEE Conf. Evol. Comput.* **1994**, *1*, 82–87.
(21) Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. Evol. Computation* **2002**, *6*, 182–197.
(22) Garey, M. R.; Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*; W.H. Freeman and Company: New York, 1979.
(23) Sørensen, M. M. New facets and a branch-and-cut algorithm for the weighted clique problem. *Eur. J. Oper. Res.* **2004**, *154*, 57–70.
(24) Sokolov, A.; Whitley, D. *Unbiased Tournament Selection*; Genetic and Evolutionary Computation Conference, 2005.
(25) Meinl, T.; Berthold, M. R. Crossover Operators for Multiobjective k-Subset Selection, 2009.
(26) Xin, D.; Cheng, H.; Yan, X.; Han, J. *Extracting Redundancy-Aware Top-k Patterns*. International Conference on Knowledge Discovery and Data Mining, 2006.
(27) Meinl, T.; Cebron, N.; Gabriel, T. R.; Dill, F.; Kötter, T.; Ohl, P.; Thiel, K.;Wiswedel, B.; Berthold, M. R. *The Konstanz Information Miner 2.0*, 2009.
(28) Liu, T.; Wen, Y. L. X.; Jorissen, R. N.; Gilson, M. K. BindingDB: A web-accessible database of experimentally determined proteinligand binding affinities. *Nucleic Acids Res.* **2007**, *35*, D198–D201(1).
(29) The Binding Database. http://www.bindingdb.org/ (accessed May 24, 2007).
(30) Borgelt, C. *On Canonical Forms for Frequent Graph Mining*; University of Magdeburg: Magdeburg, Germany, 2005.
(31) Borgelt, C.; Berthold, M. R. Mining Molecular Fragments: Finding Relevant Substructures of Molecules, 2002.
(32) Simulations Plus, Inc., ClassPharmer. http://www.simulationsplus.com/Products.aspx?grpID=1&cID=13&pID=12 (accessed January 26, 2010).
(33) Holliday, J. D.; Ranade, S. S.; Willett, P. A Fast algorithm for selecting sets of dissimilar molecules from large chemical databases. *Quant. Struct.—Act. Relat.* **1995**, *14*, 501–506.
(34) Brown, R. D.; Clark, D. E. Genetic diversity: Applications of evolutionary algorithms to combinatorial library design. *Expert Opin. Ther. Pat.* **1998**, *8*, 1447–1459.
(35) 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*, 375–385.
(36) Clark, R. D.; Langton, W. J. Balancing representativeness against diversity using optimizable k-dissimilarity and hierarchical clustering. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 1079–1086.