View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by White Rose Research Online

promoting access to White Rose research papers

Universities of Leeds, Sheffield and York
http://eprints.whiterose.ac.uk/

This is an author produced version of a paper published in **Chemoinformatics: Concepts, Methods and Tools for Drug Discovery.**

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/78612

Published paper

Gillet, V.J. (2004) *Designing combinatorial libraries optimized on multiple objectives.* In: Chemoinformatics: Concepts, Methods and Tools for Drug Discovery. Humana Press , 335 - 354.
http://dx.doi.org/10.1385/1-59259-802-1:335

White Rose Research Online
eprints@whiterose.ac.uk

Designing Combinatorial Libraries Optimised on Multiple Objectives

Valerie J. Gillet

Department of Information Studies, University of Sheffield, Western Bank, Sheffield, S10 2TN; phone: (114) 2222 652, fax: (114) 2780 300, e-mail: v.gillet@sheffield.ac.uk

Abstract

The recent emphasis in combinatorial library design has shifted from the design of very large diverse libraries to the design of smaller more focussed libraries. Typically the aim is to incorporate as much knowledge into the design as possible. This knowledge may relate to the target protein itself or may be derived from known active and inactive compounds. Other factors should also be taken into account, such as the cost of the library and the physicochemical properties of the compounds that are contained within the library. Thus library design is a multiobjective optimisation problem. Most approaches to optimising multiple objectives are based on aggregation methods whereby the objectives are assigned relative weights and are combined into a single fitness function. A more recent approach involves the use of a Multiobjective Genetic Algorithm in which the individual objectives are handled independently without the need to assign weights. The result is a family of solutions each of which represents a different compromise in the objectives. Thus, the library designer is able to make an informed choice on an appropriate compromise solution.

Key Words

Combinatorial libraries, combinatorial synthesis, computational filtering, drug-like, library enumeration, genetic algorithms, MOGA, molecular descriptors, Multiobjective Genetic Algorithm, multiobjective optimisation, simulated annealing.

1. Introduction

The last decade has seen a shift from the traditional approach to chemical synthesis, based on one compound at a time, to the use of robotics allowing the synthesis of large numbers of compounds in parallel, in what are known as combinatorial libraries. The related technique of high-throughput screening allows tens to hundreds of thousands of compounds to be tested for biological activity in a single day ***(1)***. Thus, the throughput of the synthesis and test cycle has increased enormously. However, despite the increase in the number of compounds that can be handled they still represent a very small fraction of the number of drug-like compounds that could potentially be made, for example, it has been estimated that as many as 10^{40} such compounds could exist ***(2)***. Thus, it is clear that there is a need to be selective about the compounds that are made in combinatorial libraries ***(3)***.

In the early days of combinatorial synthesis the emphasis was on synthesising as many diverse compounds as possible on the assumption that maximising diversity would maximise the coverage of different types of biological activity. However, these early libraries gave disappointing results: they had lower hits rates than expected and the hits that were found tended to have too unfavourable physicochemical properties to provide good starting points for lead discovery ***(4)***.

It is now clear that if the new technologies are to be effective for drug discovery the libraries need to be designed very carefully. Consequently, the emphasis

has shifted away from large diverse libraries to the design of smaller libraries that incorporate as much knowledge about the target as is available. At one extreme, the three-dimensional (3D) structure of the biological target may be known, in which case, structure-based methods such as docking or de novo design can be used in an attempt to design compounds that will fit into the binding site ***(5,6)***. It is still the case, however, that in most drug discovery programmes the 3D structure of the target is unknown. When several actives and inactives are known it may be possible to generate a model of activity in the form of a quantitative-structure activity relationship (QSAR), the model could then be used to design libraries consisting of compounds with high predicted activities ***(7)***. Other approaches are based on similarity methods ***(8)*** where compounds are selected based on their 2D or 3D similarity to one or more known active compounds. Diverse libraries are appropriate when a library is to be screened against a range of biological targets or when little is known about the target of interest. As a general rule, the amount of diversity required is inversely related to the amount of information that is available about the target.

Whether the primary aim is to design diverse or focussed libraries, or indeed to provide a balance between the two, many other criteria should also be taken into account. For example, the compounds should possess appropriate physicochemical properties to enable them to be progressed through the drug discovery pipeline ***(9)***. In addition, the reactants should be readily available, for example, already present in in-house collections or cheap to purchase with acceptable delivery times. Thus, library design is increasingly being treated as a multiobjective optimisation problem which requires the simultaneous optimisation of several criteria. In common with most real world optimisation problems, the criteria are often in conflict, for example, achieving diversity simultaneously with drug-like properties, and thus a compromise in the

objectives is usually sought. This chapter discusses approaches for the optimisation of combinatorial libraries based on multiple objectives.

2. Methods

2.1. Reactant versus Product Based Designs

A simple two component combinatorial synthesis is shown in **Fig. 1**. The reaction involves the coupling of a-bromoketones and thioureas. Multiple products (2-aminothiazoles) can be synthesised in parallel by selecting different examples of each of the components, or reactants. The positions of variability in the reactants are indicated by the R groups.

In general, there are many more examples of the reactants available than can be handled in practice and thus selection methods must be used. For example, when designing peptides: there are 20 amino acids and hence 20´ 20 or 400 dipeptides; 8000 tripeptides; 32K tetrapeptides and so on. When designing libraries of small drug-like compounds, in general there could be tens or even hundreds of possible reactants available for each position of variability. Thus, even when libraries are limited to a single reaction scheme the numbers of compounds that could potentially be made can be very large.

Library design methods can be divided into reactant-based or product-based design. In reactant-based design, reactants are chosen without consideration of the products that will result. For example, diverse subsets of reactants are selected in the hope they will give rise to a diverse library of products. In product-based design, the selection of reactants is determined by analysing the products that will be produced.

Reactant-based design is computationally less demanding than product-based design since there are fewer molecules to consider. Consider a two component

reaction where there are 100 examples of each type of reactant. Now assume that the aim is to design a library of 100 products with configuration 10´ 10, i.e., 10 examples of each reactant. There are approximately 10^{13} different possible subsets of size 10 contained within 100 compounds, as determined by the equation below:

$$\frac{N!}{n!(N - n)!}$$

Examining this number of subsets is clearly unfeasible. Hence, a number of computationally efficient, albeit approximate, methods have been devised for performing reactant-based selection ***(10)***. Product-based design, however, is much more computationally demanding and would require the analysis of 100´ 100 potential products (i.e. 10^4 molecules). Despite the increased computational cost of product-based design it has been shown that it can result in better optimised libraries especially when the aim is to optimise library-based properties such as diversity ***(11,12)***. Product-based design is even more appropriate for targeted or focussed designs where it is the properties of the product molecules themselves that are to be optimised, e.g. similarity to a known active compound.

Product-based approaches can be divided into those that take the combinatorial constraint into account such that each reactant in one pool appears in a product with every reactant from every other reactant pool, and those that merely pick product molecules with consideration of the synthetic constraint. The latter approach is often referred to as cherry-picking and is synthetically inefficient as far as combinatorial synthesis is concerned. In this chapter the emphasis is on product-based library design methods that take the combinatorial constraint into account.

2.2. Filtering

The first step in library design is to identify potential lists of reactants. This can be done by searching databases of available compounds, for example, in-house databases or databases of compounds that are available for purchase such as the Available Chemicals Directory ***(13)***. The next step is to filter the reactant lists. This is a very important step since it can vastly reduce the computational complexity of the subsequent library design step. The aim is to remove reactants that could not possibly lead to 'good' products. A variety of filtering steps can be used. For example, removal of compounds that contain functionality that will interfere with the synthesis or that contain functional groups known to be toxic. In addition, thresholds on various physicochemical properties could also be applied, for example, removal of compounds with more than 8 rotatable bonds or molecular weights greater than 300 since compounds with these properties are not generally considered as drug-like.

2.3. Library Enumeration

Enumeration is the computational equivalent of carrying out a combinatorial synthesis. The result is a virtual library of product molecules that can then be analysed using a library design program to select compounds of interest. Two different approaches to library enumeration have been developed: fragment marking and the reaction transform approach ***(14)***.

Fragment marking involves representing a library by a central core (for example a benzodiazepine ring) which is common to all compounds in the virtual library with one or more R groups to indicate the positions of variability. The library is enumerated by creating bonds between the core template and the reactants. The reactant lists must first be 'clipped', for example, the hydoxyl group must be removed

from a carboxylic acid selected to be combined with an amine group in the formation of an amide bond. Fragment marking approaches usually require that there is a central core template that can be defined and that fragment clipping can be automated, however, this may not always possible, for example, for a Diels-Alder reaction.

The reaction transform approach is based on a computer-readable representation of the reaction mechanism which describes the transformation of the atoms in the reactants to the product. The transform is applied to the input reactants themselves to generate the products. The reaction transform approach thus more closely mimics the actual synthetic process, however, it can be difficult to construct efficient transforms. This is the approach used in the ADEPT software ***(14)***.

2.4. Design Criteria

As discussed in the Introduction, the primary design criterion is often based on either similarity or diversity. Quantifying these measures requires that the compounds are represented by numerical descriptors that enable pairwise molecular similarities or distances to be calculated or that allow the definition of a multidimensional property space in which the molecules can be placed.

A variety of different descriptors have been used in library design ***(15,16)***. They can be divided into descriptors that represent whole molecule properties; descriptors that can be calculated from the 2D graph representations of molecules including topological indices and 2D fingerprints; and descriptors calculated from 3D representations of molecules. Whole molecule properties include physicochemical properties such as molecular weight, molar refractivity and calculated logP. Topological indices are single-valued descriptors that characterise structures according to their size, degree of branching and overall shape. Many different

topological indices have been devised and they are often used together with a molecule being represented by a vector of real numbers. 2D fingerprints are binary vectors and can be divided into fragment-based methods and path-based methods. In the fragment-based methods, each bit in the vector corresponds to a particular substructural fragment and is set to “on” or “off” to indicate the presence or absence of the substructure within a molecule. In the path-based methods, all paths up to a given length in the molecule are determined and each path is hashed to a small number of bits which are then set to “on”.

The most commonly used 3D descriptors are pharmacophore keys which are usually represented as binary vectors ***(17)***. The starting point when generating a pharmacophore key is a 3D conformation of a molecule that is represented by its pharmacophoric features, that is its atoms or groups of atoms that can form interactions with a receptor such as hydrogen bond donors, acceptors, aromatic centres, anions and cations. In 3-point pharmacophore keys, each bit in the vector represents three pharmacophoric features together with a set of distance ranges that define how the features are positioned in 3D space. As with 2D fragment-based fingerprints, a bit is set to “on” to indicate the presence of a pharmacophore triplet within a molecule, otherwise it is set to “off”.

When molecules are represented by high-dimensional descriptors such as 2D fingerprints or several hundred topological indices then the diversity of a library of compounds is usually calculated using a function based on the pairwise (dis)similarities of the molecules. Pairwise similarity can be quantified using a similarity or distance coefficient. The Tanimoto coefficient is most often used with binary fingerprints and is given by the formula below:

$$S_{AB} = \frac{c}{a + b - c}$$

where there are *a* bits set to "on" in molecule *A*, *b* bits set to "on" in molecule *B*, and *c* "on" bits common to both *A* and *B*. When molecules are represented by real-numbered vectors then the comparison is usually based on Euclidean distance. Various diversity functions have been suggested for library design including the average nearest neighbours distance and the sum of pairwise dissimilarities ***(18)***.

When molecules are represented by low-dimensional descriptors then the descriptors can be used to define the axes of a chemistry space. Typical descriptors are a small number of physicochemical properties or the principal components generated by the application of principal components analysis to high dimensional descriptors. Each descriptor then defines one axis and is divided into a series of bins. The combination of all bins over all descriptors defines a set of cells over a chemistry space. Molecules can be mapped onto the cells according to their physicochemical properties. A diverse library is one that occupies a large number of cells in the space, whereas, a focussed library is one where the molecules occupy a small localised region of the space.

The optimisation of physicochemical properties can be dealt with by applying simple thresholds such as Lipinski's rule-of-five ***(19)***. The rule states that if a compound violates any two of the following rules it is predicted to have poor oral absorption:

- molecular weight > 500
- log*P* > 5
- > 5 hydrogen bond donors (defined as the sum of OH and NH groups)
- > 10 hydrogen bond acceptors (defined as the number of N and O atoms).

Alternatively, they can be optimised by matching the profile of properties in the library to some collection of known drug-like molecules. The latter method will

typically allow some compounds to be present in the library that violate the more stringent rules. Several groups have developed more sophisticated methods for predicting drug-likeness ***(20)*** and, more recently, lead-likeness (since it has been recognised that lead compounds tend to be less complex than drugs) ***(21,22)***.

2.5. Optimisation Methods

The computational complexity of product-based library design has led to the development of programs that are based on optimisation techniques such as genetic algorithms and simulated annealing. The methods require the definition of a function that is able to measure the degree to which a potential solution meets the library design criteria. The optimisation technique then attempts to maximise (or minimise) the given function. Typically, many potential solutions are explored during the operation of the algorithm and thus the function must be relatively rapid to calculate.

Several groups have approached multiobjective library design by combining individual objectives into a single combined fitness function. This is a widely used approach to multiojective optimisation and effectively reduces a multiobjective optimisation problem to one of optimising a single objective.

This approach has been adopted in the SELECT library design program ***(23)***. SELECT is based on a GA and aims to identify a combinatorial subset of pre-determined size and configuration, from within a virtual, fully enumerated library. The chromosome representation in SELECT encodes potential subsets as the lists of reactants from which the library will be synthesised. Thus, the chromosome is an integer string which is partitioned according to the number of positions of variability in the library. The size of a partition is determined by the number of reactants to be selected. Thus, when configured to select an $n_A \times n_B$ subset from a virtual library of

size $N_A \times N_B$, the chromosome consist of n_A+n_B integers. Each integer corresponds to one of the possible reactants available. The standard genetic operators of crossover and mutation are used with the special condition that the same reactant must not appear more than once in a partition.

SELECT has been designed to allow optimisation of a variety of different objectives. Diversity (and similarity) is optimised using functions either based on pairwise dissimilarities and fingerprints or using cell-based measures. The physicochemical properties of libraries are optimised by minimising the difference in the distribution of the library being designed and some reference distribution, such as that seen in the World Drugs Index (WDI) ***(24)***. Cost is optimised simply by minimising the sum of the cost of the reactants. Each objective is usually standardised to be in the range 0 to 1 and user-defined weights are applied prior to summing the contributions into a weighted-sum fitness function as show below:

$$f(n) = w_1.diversity + w_2.cost + w_3.property1 + w_4.property2 + ...$$

The HARPick program also tackles multiobjective library design by combining individual objectives, via weights, into a single function. HARPick uses Monte Carlo simulated annealing as the optimisation technique ***(25)*** with library design being based on pharmacophore keys. A library is represented by an ensemble pharmacophore key which is the union of the individual molecule keys. In HARPick the pharmacophore keys are integer vectors which indicate the frequency of occurrence of each 3-point pharmacophore. The fitness function is composed of several individual functions: diversity is based on the number of unique pharmacophore triplets covered by the library and is tuned to force molecules to occupy relative voids (under-represented 3-point pharmacophores) as well as absolute voids; libraries can be optimised to fill voids under-represented in an existing library;

a function based on the number of conformations per molecule is used to control molecular flexibility; various properties are calculated that are crude measures of molecular shape with the aim being to produce an even distribution of shapes in the library; and finally a count of the total number of pharmacophores present is used to limit the inclusion of promiscuous molecules (that is, molecules that contain a large number of pharmacophore triplets). As in the SELECT program, the individual functions are combined into a single fitness function via user-defined weights.

The method has subsequently been extended to include 4-point pharmacophores and to allow pharmacophoric measures to be combined with 3D BCUT descriptors ***(26)***. BCUT descriptors were designed to encode atomic properties relevant to intermolecular interactions. They are calculated from a matrix representation of a molecule's connection table where the diagonals of the matrix represent various atomic properties such as atomic charge, atomic polarisability, and atomic hydrogen bonding ability and the off-diagonals are assigned the interatomic distances. The eigenvalues of the matrix are then extracted for use as descriptors. Five such descriptors were calculated: two based on charge; two on atomic polarisability and one based on hydrogen bond acceptors. These descriptors then define a 3D BCUT chemistry space, as for the cell-based methods described previously, with BCUT diversity being measured as the ratio of occupied cells to the total possible occupied cells. Pharmacophore diversity is based on the number of unique pharmacophores and the total number of pharmacophores in the product subset. An overall score for a library is then calculated by summing the two diversity measures. The method has been tested on a virtual library of 86140 amide products in which pharmacophores were calculated on-the-fly, i.e., during the optimisation process itself, with pharmacophore keys being stored for reuse as they are calculated.

Other similar aggregation approaches to multiobjective library design include the methods described by Agrafiotis ***(27)***, Zheng et al. ***(28)*** and Brown et al. ***(29)***.

2.6. Multiobjective Optimisation Using a MOGA

The aggregation approach to multiobjective optimisation in which multiple objectives are combined into a single fitness function is limited for a number of reasons, some of which are identified here. First, the selection of weights for the individual components is non-intuitive especially when comparing different properties for example, diversity and calculated logP. Second, the use of weights limits the search space that is explored. Third, in general the methods are restricted to finding a single solution which represents one particular compromise in the objectives; assigning a different set of weights will typically result in a different solution, one that may be equally valid but that represents a different compromise in the objectives. Thus, in practice it is usual to perform a number of trial-and-error runs using different weights in order to identify a 'good' solution.

Multiobjective Evolutionary Algorithms (MOEAs) are a class of algorithms that are based on optimising each objective independently and thus avoid the need to assign weights to individual objectives ***(30)***. They exploit the population nature of evolutionary algorithms in order to explore multiple solutions in parallel. The MOGA is one example of a MOEA and is based on a GA ***(31)***. In MOGA, the fitness ranking in a traditional GA is replaced by Pareto ranking. Pareto ranking is based on the concept of dominance, where, in a given population, one solution dominates another if it is better in all objectives and a non-dominated solution is one for which no other solution is better in all the objectives. In MOGA, an individual is assigned fitness according to the number of individuals by which it is dominated. Parent selection is

then biased towards the least dominated individuals so that all non-dominated solutions have equal chance of being selected and they have a higher chance of being selected than solutions that are dominated. The non-dominated individuals form what is known as the Pareto surface. In the absence of further information, all solutions on the Pareto surface are equally valid with each one representing a different compromise in the objectives.

The MOGA algorithm has been adopted in the MoSELECT library design program ***(32-34)***. MoSELECT derives from the earlier SELECT program with the original GA being replaced by a MOGA. Thus, in MoSELECT different objectives such as diversity, similarity, physicochemical property profiles and cost are treated independently to generate a family of different compromise solutions as will be shown in the Results section.

2.7. Varying Library Size and Configuration

Many library design methods require that the size (number of products) and configuration (numbers of reactants selected for each component) of the library are specified upfront. However, it is often difficult to determine optimum values a-priori and usually there is a trade-off between these criteria and the other criteria to be optimised. Consider the design of a library where the aim is to maximise coverage of some cell-based chemistry space. It is clear that as more products are included in the library the chance of occupying more cells increases. Thus, an optimal library is likely to be one that represents a compromise in size and diversity.

MoSELECT has been adapted so that size and configuration can be optimised simultaneously with other library design criteria. Size is allowed to vary by using a binary chromosome representation. The chromosome is partitioned, as before, with

one partition for each position of variability. However, now each partition is of length equal to the number of reactants available with each reactant represented by a binary value. The value “1” indicates that a reactant has been selected and the value of “0” indicates that it has not been selected for the final library. Thus the chromosome is now of length N_A+N_B (as opposed to n_A+n_B as described earlier). The application of the genetic operators results in different reactants being selected and deselected and library size (and configuration) are varied by altering the number of bits set to “1”.

As described previously, diversity and library size are usually in conflict with larger libraries resulting in greater cell coverage. Thus, when optimising on diversity alone there will be a tendency to select very large libraries. Thus, in MoSELECT size is included as an objective alongside diversity with each objective being handled independently. This allows the trade-off between size and diversity to be explored in a single run.

2.8 Multiobjective Design Under Constraints

The MOGA approach allows the mapping of the entire Pareto surface with solutions at the extremes being identified as well as a range of solutions in between the extremes. When optimising size and diversity this means that a wide range of solutions are possible, from libraries consisting of a single product up to the library size that achieves maximum cell coverage. While having the ability to map the entire Pareto surface can provide useful insights into the shape of the search space of a particular library design problem, in practice there are often external constraints that must be taken into account. For example, constraints on library size may arise from the equipment available or simply on the basis of cost.

Library configuration can be a factor in cost as well as library size itself. Typically it is desirable to minimise the total number of reactants required. Thus, if the aim is to synthesise a library of 400 products from two positions of variability then the most efficient use of reactants is achieved for the configuration 20´ 20. Other configurations (40´ 10; 25´ 16 etc.) would require access to a greater number of unique reactants.

Constraints can be implemented within the MOGA to direct the search towards restricted regions of the search space. Constraints are handled by penalising solutions that violate the constraints. Such infeasible solutions are allowed to exist within the population (rather than being removed entirely) since their presence may lead to feasible solutions later in the search through the use of crossover. They are penalised so that they have a lower chance of being selected for reproduction and so that they do not appear in the final solution set. In the example described in the next section, constraints are applied on library size and configuration, however, they could equally be applied to any of the objectives incorporated within the library design.

3. Results: Designing 2-aminothiazole Libraries

The two-component 2-aminothiazole library shown in **Fig. 1** is used to illustrate different library design scenarios using the SELECT and MoSELECT programs.

As discussed, the starting point for library design is to identify available reactants, for example, by searching in-house databases and/or by identifying reactants that can be purchased. In this case, substructure searches were performed on the ACD. When constructing a query it is often necessary to place constraints on the compounds to be returned as hits. Thus, the a-bromoketone substructure was

constrained so that it should not be embedded within a ring and explicit hydrogens were attached to one of the nitrogen atoms in the thiourea query with the additional constraint that substitution on the sulphur atom was prohibited.

Once initial sets of reactants were found computational filters were applied to remove reactants that are known to be undesirable. This was done using the ADEPT software ***(14)*** with the following compounds being removed: reactants having molecular weight greater than 300; reactants having more than 8 rotatable bonds; and a series of substructure searches were performed to remove reactants containing undesirable substructural fragments. After filtering there were 74 a-bromoketones and 170 thioureas remaining, which represents a virtual library of 12850 product molecules. The next step in the design process was to enumerate the full virtual library which was done using the transform method in ADEPT.

The virtual library was then characterised using the Cerius^2 default topological descriptors and physicochemical properties ***(35)***. The 50 default descriptors were reduced to three principal components using principal components analysis and this defined a three dimensional chemistry space into which the virtual library could be plotted. The chemistry space consisted of 1134 cells and when the virtual library was mapped into the space it was found to occupy 364 of the cells, this thus represents the maximum cell coverage that is achievable.

The SELECT program was then used to design a 15´ 30 library that was simultaneously optimised on diversity (measured by the number of occupied cells) and to have a drug-like molecular weight profile (measure by the RMSD between the profile of the library and the profile of molecular weights found in the WDI. The resulting library was found to occupy a total of 234 cells and its molecular weight profile is shown in **Fig. 2** together with the profile of molecular weight found in WDI.

When optimising a library on diversity alone the best library found occupies 282 cells and when optimising on molecular weight profile alone, the best library was found to occupy 169 cells. Thus, when optimising both objectives simultaneously using the weighted-sum approach in SELECT, the resulting library represents a compromise in the two objectives.

Performing a single run of SELECT with one set of weights does not allow the library designer to explore the relationship between the two objectives and a single somewhat arbitrary solution was produced.

The relationship between molecular weight profile and diversity was then explored using the MOGA approach implemented in MoSELECT. The result was a total of 11 different libraries with each library representing a different trade-off between the objectives, as shown by the crosses in **Fig. 3**. The most drug-like library (the library with the best molecular weight profile) is the least diverse (169 cells occupied) whereas the most diverse library (282 cells occupied) has the least drug-like profile. The SELECT solution found previously is shown by the solid diamond.

Thus far, the size and configuration of the libraries was fixed. The relationship between library size and diversity was investigated by performing multiple runs of the SELECT program with each run configured to find a library of increasing size. The results of performing this exercise are shown in the **Fig. 4** where it can be seen that diversity (cell coverage) increases as library size increases.

MoSELECT allows the trade-off in library size and diversity to be investigated in a single run. The libraries found are shown by the solid squares (superimposed on the SELECT results) in **Fig. 5**. Thus, the full range of library sizes is explored, from very small libraries with low diversity up to a library size of 1392

which has the maximum diversity that is possible: it occupies all 364 cells that are occupied by the full virtual library.

The remaining library designs are based on applying the MOGA under various constraints. In **Fig. 6**, the libraries are constrained to contain between 250 and 500 products. Finally, the libraries are constrained to contain between 15 and 20 reactants in each component. The libraries found when no constraint is placed on configuration are shown by the crosses in **Fig. 7A** and the libraries found when the constraints are applied are shown by the solid squares. **Fig. 7B** illustrates that the constrained (more efficient) libraries were found without any loss in diversity.

4. Discussion and Notes

Combinatorial library design is a complex procedure that can be divided into several steps as indicated above. A wide variety of different computational tools are available that can be applied to the different steps, however, effective use of the tools can require considerable user interaction in order to maximise the chances of finding useful compounds. Thus, the tools should not be considered as black boxes.

For a given reaction scheme the first step is usually to identify available reactants. Care should be taken when constructing substructural queries to ensure that the compounds retrieved are indeed capable of undergoing the reaction, for example, when searching for primary amines it may be desirable that hits are restricted to those that contain a single amine group. Visual inspection of the results can be used to ensure that the substructural query was correctly specified and it can also be useful in determining which computational filters to apply. For example, the presence of highly flexible molecules in the answer set may suggest the use of a filter to remove reactants where the number of rotatable bonds is above some threshold value. Filters

are extremely important since the early removal of undesirable compounds can simplify the later stages of library design.

Once the reactant pools have been filtered, the next step in product-based designs is usually to enumerate the full virtual library. This can be a very time consuming step and hence a useful precursor can be to enumerate carefully chosen subsets that will give an indication of the success or otherwise of the full virtual experiment. Thus, in a two component reaction it can be useful to take the first reactant in the first pool and combine it with all the reactants in the second pool (to generate $1' n_B$ products). This should then be followed by the enumeration of one reactant in the second pool with all reactants in the first pool to give $n_A' 1$ products. If either of these two partial enumeration steps fail then the full enumeration will also fail. Thus, troublesome reactants can be identified early.

The next step is to determine the descriptors to use for the library optimisation. It is important that descriptors are chosen that are relevant to the type of compounds that the library is being designed for. The descriptors should result in a high degree of similarity between compounds that are known to have the desired properties. Thus, if some active compounds are known then, ideally, these should cluster together within the descriptor space. Another criterion to take into account when choosing descriptors is the number of compounds in the virtual library. Some descriptors can be costly to compute, especially 3D descriptors when the conformational flexibility of the compounds is taken into account. Thus, it is important to be aware of the computational resources that will be required for a given library design strategy.

Finally the optimisation step itself usually involves human intervention. With the traditional aggregation approaches to library design the user must decide on appropriate weights for the various objectives being optimised. This can involve

several trial-and-error experiments where different combinations of weights are applied. In the novel library design method based on a MOGA, the user no longer needs to determine relative weights however a family of different compromise solutions is found and hence the user must apply his or her own knowledge to decide which library represents the best compromise in the objectives.

5. References

1. Wolcke, J. and Ullmann, D. (2001) Miniaturized HTS technologies- uHTS. *Drug Discov Today*, **6**: 637--646.
2. Valler, M. J. and Green, D. (2000) Diversity screening versus focussed screening in drug discovery, *Drug Discov. Today*, **5,** 286--293.
3. Rose, S. and Stevens, A. (2003) Computational design strategies for combinatorial libraries, *Current Opin. Chem. Biol*. **7**, 331--339.
4. Martin, E. J. and Critchlow, R. E. (1999) Beyond mere diversity : Tailoring combinatorial libraries for drug discovery. *J. Comb. Chem.* **1**, 32--45.
5. Beavers, M. B. and Chen, X. (2002) Structure-based combinatorial library design: methodologies and applications. *J. Mol. Graph. Model*. 20, 463--468.
6. Leach, A. R., Bryce, R. A. and Robinson, A. J. (2000) Synergy between combinatorial chemistry and *de novo* design. *J. Mol. Graphics Model.* **18**, 358--367.
7. Kubinyi, H. (2002) From narcosis to hyperspace: The history of QSAR. *Quantit. Struct.-Act. Relat.* **21**, 348--356.
8. Barnard, J. M., Downs, G. M. and Willett, P. (1998) Chemical similarity searching. *J. Chem. Inf. Comput. Sci.* **38**, 983--996.

9. van de Waterbeemd, H. and Gifford, E. (2002) ADMET in silico modelling: towards prediction paradise? *Nat. Rev. Drug Discov.* **2**, 192--204.
10. Gillet, V. J. and Willett, P. (2001) Dissimilarity-based compound selection for library design in *Combinatorial Library Design and Evaluation. Principles, Software Tools and Applications in Drug Discovery* (Ghose, A. K. and Viswanadhan, A. N. eds), Marcel Dekker, New York, pp 379--398
11. Gillet, V. J., Willett, P. and Bradshaw, J. (1997) The effectiveness of reactant pools for generating structurally diverse combinatorial libraries. *J. Chem. Inf. Comput. Sci.* **37**, 731--740.
12. Jamois, E. A., Hassan, M., and Waldman, M. (2000) Evaluation of reactant-based and product-based strategies in the design of combinatorial library subsets. *J. Chem. Inf. Comput. Sci.*, **40**. 63--70.
13. ACD. Available Chemicals Directory, MDL Information Systems, Inc. 14600 Catalina Street, San Leandro, CA 94577. http://www.mdli.com.
14. Leach, A. R., Bradshaw, J., Green, D. V. S., Hann, M. M. and Delany III J. J. (1999) Implementation of a system for reagent selection and library enumeration, profiling and design. *J. Chem. Inf. Comput. Sci.* **39**, 1161--1172.
15. Brown, R. D. (1997) Descriptors for diversity analysis. *Perspect. Drug Discov. Design*. **7/8**, 31--49.
16. Bajorath, J. (2001) Selected concepts and investigations in compound classification, molecular descriptor analysis, and virtual screening. *J. Chem. Inf. Comput. Sci.* **41,** 233--245.
17. Pickett, S. D., Mason, J. S. and McLay, I. M. (1996) Diversity profiling and design using 3D pharmacophores: Pharmacophore-Derived Queries (PDQ). *J. Chem. Inf. Comput. Sci.* **36**, 1214--1223.

18. Waldman, M., Li, H., and Hassan, M. (2000) Novel algorithms for the optimization of molecular diversity of combinatorial libraries. *J. Mol. Graphics Model.* **18**, 412--426.
19. Lipinski, C. A., Lombardo, F., Dominy, B. W. and Feeney, P. J. (1997) Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Delivery Rev.* **23**, 3--25.
20. Clark, D. E. and Pickett, S. D. (2000) Computational methods for the prediction of 'drug-likeness'. *Drug Discov. Today* **5**, 49--58.
21. Hann, M. M., Leach, A. R., and Harper, G. (2001) Molecular complexity and its impact on the probability of finding leads for drug discovery. *J. Chem. Inf. Comput. Sci.* **41**, 856--864
22. Oprea, T. I. (2000) Property distributions of drug-related chemical databases. *J. Comput-Aided Mol. Des.* **14**, 251--264.
23. Gillet, V. J., Willett, P., and Bradshaw, J. (1999) Selecting combinatorial libraries to optimise diversity and physical properties. *J. Chem. Inf. Comput. Sci.* **39**, 167--177.
24. WDI: The World Drug Index is available from Derwent Information, 14 Great Queen St., London W2 5DF, UK.
25. Lewis, R. A., Pickett, S. D., and Clark, D. E. (2000) Computer-aided molecular diversity analysis and combinatorial library design, in *Reviews in Computational Chemistry Volume 16* (Lipkowitz, K. B. and Boyd, D.B., eds) VCH Publishers, New York, pp 1--51.
26. Mason, J. S. and Beno, B. R. (2000) Library design using BCUT chemistry-space descriptors and multiple four-point pharmacophore fingerprints:

Simultaneous optimisation and structure-based diversity. *J. Mol. Graph. Model.* **18**, 438--451.

27. Agrafiotis, D. K. (2002) Multiobjective optimisation of combinatorial libraries. *J. Comput.-Aid. Mol. Design* **5/6**, 335--356.
28. Zheng, W., Hung, S. T., Saunders, J. T. and Seibel, G. L. (2000) PICCOLO: A tool for combinatorial library design via multicriterion optimization, in *Pacific Symposium on Biocomputing 2000* (Atlman, R.B., Dunkar, A.K., Hunter, L., Lauderdale, K. and Klein, T.E., eds), World Scientific, Singapore, pp. 588--599.
29. Brown, J. D., Hassan, M., and Waldman, M. (2000) Combinatorial library design for diversity, cost efficiency, and drug-like character. *J. Mol Graph. Model*. **18**, 427--437.
30. Coello, C. A., van Veldhuizen, D. A., and Lamont, G.B. (2002) *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York.
31. Fonseca, C. M. and Fleming, P. J. (1995) An overview of evolutionary algorithms in multiobjective optimization, in *Evolutionary Computation* Vol. 3, No. 1, (De Jong, K. ed) The Massachusetts Institute of Technology, pp. 1-16.
32. Gillet, V. J., Khatib, W., Willett, P., Fleming, P., and Green, D. V. S. (2002) Combinatorial library design using a Multiobjective Genetic Algorithm. *J. Chem. Inf. Comput. Sci.* **42**, 375--385.
33. Gillet, V. J., Willett, P., Fleming P., and Green D. V. S. (2002) Designing focused libraries using MoSELECT. *J. Mol Graph. Model*. **20**, 491--498.

34. Wright, T., Gillet, V. J., Green, D. V. S., and Pickett, S. D. (2003) Optimising the size and configuration of combinatorial libraries. *J. Chem. Inf. Comput. Sci.* **43**, 381--390.

35. Cerius^2 is available from Accelrys Inc., 9685 Scranton Road, San Diego, CA 92121.

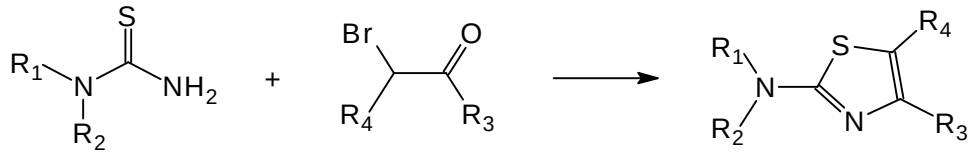

Fig. 1. A 2-aminothiazole library synthesised from a-bromoketones and thioureas.

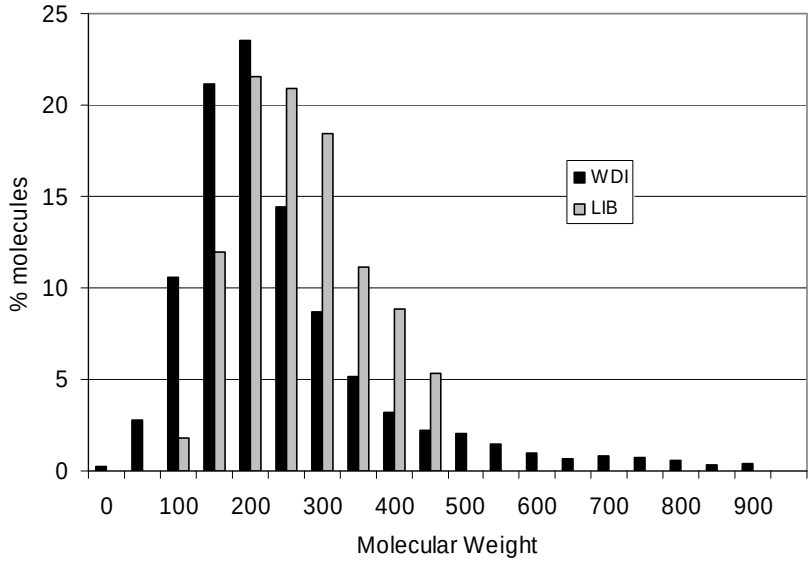

Fig. 2. The molecular weight profile of the library designed using SELECT (LIB) is shown together with the profile of molecular weights in WDI.

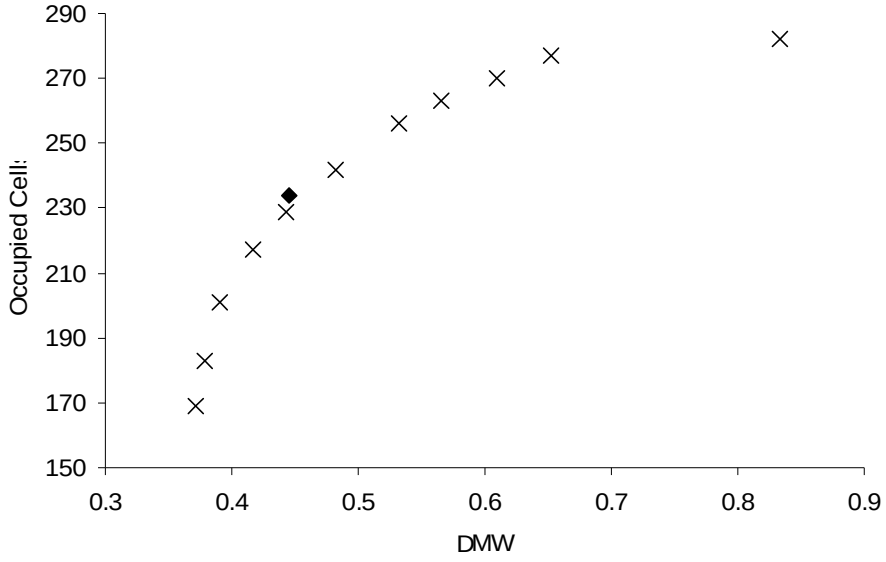

Fig. 3. A family of libraries (shown by the crosses) is found when optimising molecular weight profile simultaneously with cell based diversity when using the MoSELECT program. The single SELECT solution is shown by the solid square.

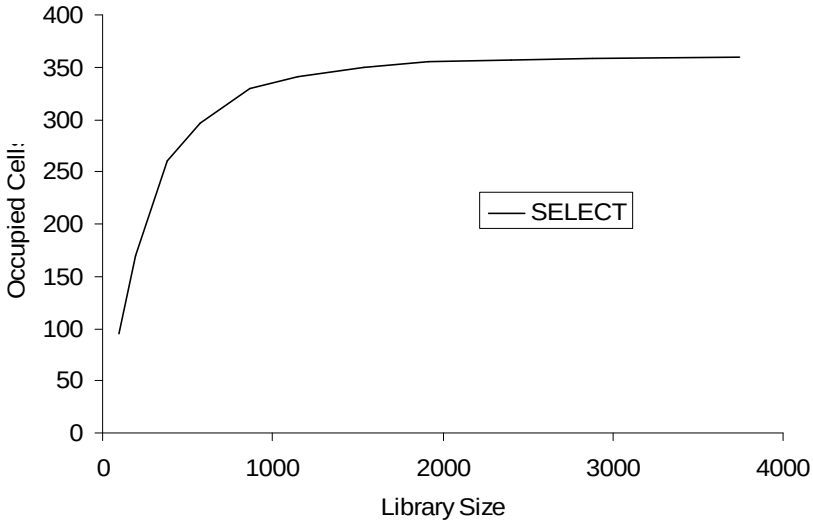

Fig. 4. Exploring library size and diversity with the SELECT program requires multiple runs with different input values.

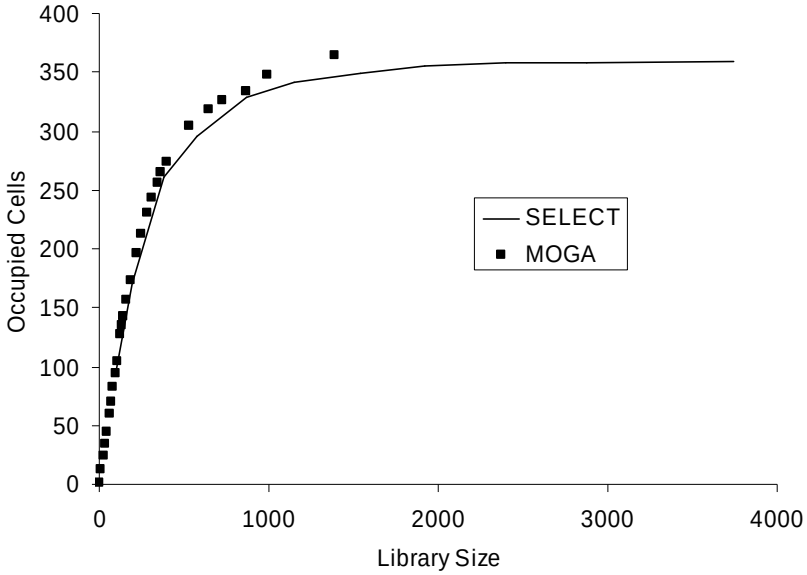

Fig. 5. Library size and diversity can be explored in a single run using the MoSELECT program. The family of solutions found is shown by the solid squares and is superimposed on the SELECT curve repeated from **Fig. 4**.

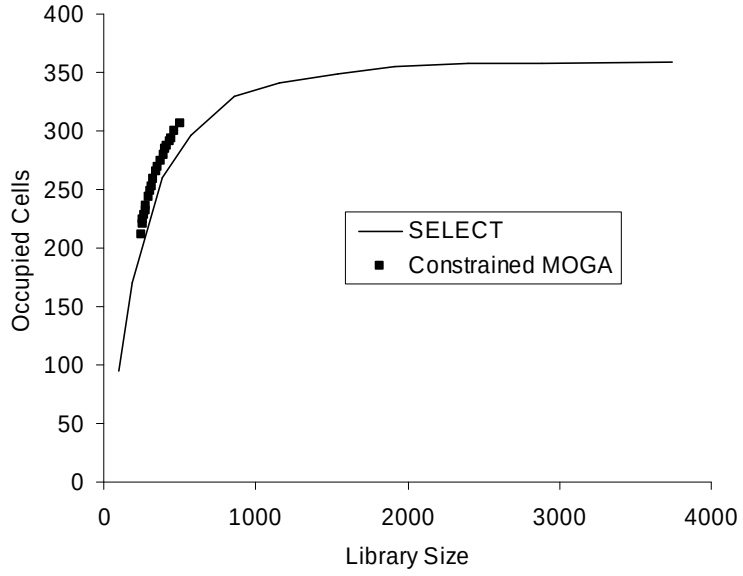

Fig. 6. Library sized is constrained to between 250 and 500 products.

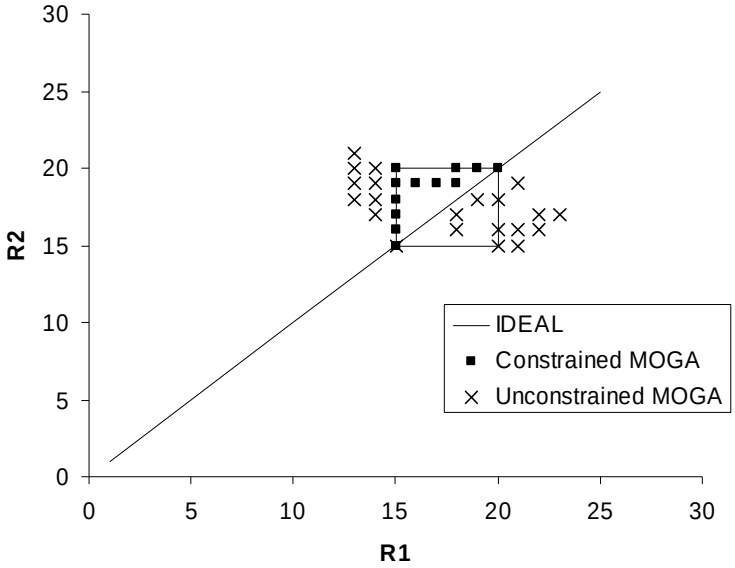

Fig. 7A. The crosses show a run where library size is constrained, but no constraints are placed on library configuration. The solid squares show the effect of also constraining configuration so that between 15 and 20 reactants are used from each pool. The solid line shows the ideal solution in terms of efficiency, that is, equal numbers of reactants are selected from each reactant pool.

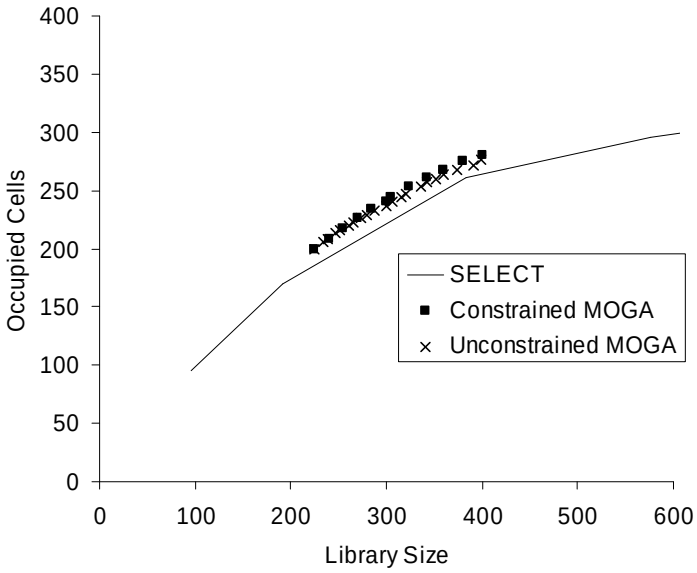

Fig. 7B. No loss of diversity is seen in the configuration-constrained library relative to the less efficient unconstrained solutions.