A Fast Exchange Algorithm for Designing Focused Libraries in Lead Optimization

Céline Le Bailly de Tilleghem,*,† Benoît Beck,‡ Bruno Boulanger,‡ and Bernadette Govaerts†

Institute of Statistics from the Université catholique de Louvain - 20, voie du roman pays,
1348 Louvain-la-Neuve, Belgium, and Lilly Services S.A. - 11, rue Granbonpré,
1348 Mont-Saint-Guibert, Belgium

Received July 6, 2004

Combinatorial chemistry is widely used in drug discovery. Once a lead compound has been identified, a series of R-groups and reagents can be selected and combined to generate new potential drugs. The combinatorial nature of this problem leads to chemical libraries containing usually a very large number of virtual compounds, far too large to permit their chemical synthesis. Therefore, one often wants to select a subset of "good" reagents for each R-group of reagents and synthesize all their possible combinations. In this research, one encounters some difficulties. First, the selection of reagents has to be done such that the compounds of the resulting sublibrary simultaneously optimize a series of chemical properties. For each compound, a desirability index, a concept proposed by Harrington,[20] is used to summarize those properties in one fitness value. Then a loss function is used as objective criteria to globally quantify the quality of a sublibrary. Second, there are a huge number of possible sublibraries, and the solutions space has to be explored as fast as possible. The *WEALD* algorithm proposed in this paper starts with a random solution and iterates by applying exchanges, a simple method proposed by Fedorov[13] and often used in the generation of optimal designs. Those exchanges are guided by a weighting of the reagents adapted recursively as the solutions space is explored. The algorithm is applied on a real database and reveals to converge rapidly. It is compared to results given by two other algorithms presented in the combinatorial chemistry literature: the *Ultrafast* algorithm of D. Agrafiotis and V. Lobanov and the *Piccolo* algorithm of W. Zheng et al.

1. INTRODUCTION

The process of drug discovery generally implies the following successive stages. Once a biological target has been defined, the screening of large collections of compounds is performed to identify *hits* i.e., compounds that interact with the biological target. Then *leads* are generated by chemically modifying the hits to improve chemical properties. Further chemical alterations of leads are performed to optimize criteria and convert leads into drug development candidates. Finally, preclinical trials confirm the characteristics of the new potential drug. Those different stages are summarized in Figure 1. See refs 9, 10, 12, 26, 27, and 33 for a review on drug discovery and development.

Combinatorial chemistry is widely used in this process.[2,14,15,19,23,24,35] This technique systematically combines a variety of molecular building blocks into a huge collection of compounds called a *combinatorial library*. There is an increasing need of methods to design combinatorial libraries.[28,31,32] A first possible objective is to span a wide range of chemical properties. This is known under the name *diverse library* or *exploratory library*.[1,3,37] It can be used in the hit identification stage, when chemists want to select active compounds as diverse as possible. The design of libraries using some diversity measurements is often discussed in the literature.[4–6,8,16,17,21,25] A second possible objective is to obtain a library biased toward a specific target or optimizing some chemical properties. This is known under the name *focused library* or *targeted library*.[1,3,37] It can be used in the lead optimization stage.

There is an increasing need of methods to construct combinatorial libraries. Part of those techniques are borrowed from the field of operational research, based on simulated annealing algorithms or genetic algorithms.[7,18,22,29,30,34,36,37] This paper presents a new method based on exchanges to design a focused library in lead optimization. This iterative algorithm finds rapidly in the combinatorial library a sublibrary of reasonable size that is composed of promising molecules that could be easily synthesized in laboratories thanks to their combinatorial nature.

Some notations are first fixed, and the general aim of the focused library design is defined using a real combinatorial library provided by Eli Lilly S.A. as an example. Then the principles of the *WEALD* algorithm are explained, and the results of its application on the combinatorial library show how well it performs. Finally, the *WEALD* algorithm is compared with two other existing methods, the *Ultrafast* algorithm of Agrafiotis and Lobanov[1] and the *Piccolo* algorithm of Zheng et al.[37]

2. DESIGNING FOCUSED LIBRARY IN LEAD OPTIMIZATION

Once a lead has been generated, chemists in pharmaceutical industries try to chemically modify the lead to obtain compounds having still better properties. They virtually divide the lead into *N* basic parts and set up a list of possible modifications on each part. Each modification is called a

* Corresponding author phone: (0032)-10−478817; fax: (0032)-10−473032; e-mail: lebailly@stat.ucl.ac.be.

† Institute of Statistics from the Université Catholique de Louvain.

‡ Lilly Services S.A.

10.1021/ci049787t CCC: $30.25 © 2005 American Chemical Society
Published on Web 03/30/2005

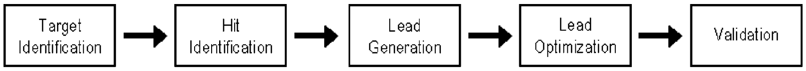

Figure 1. General scheme of drug discovery process.

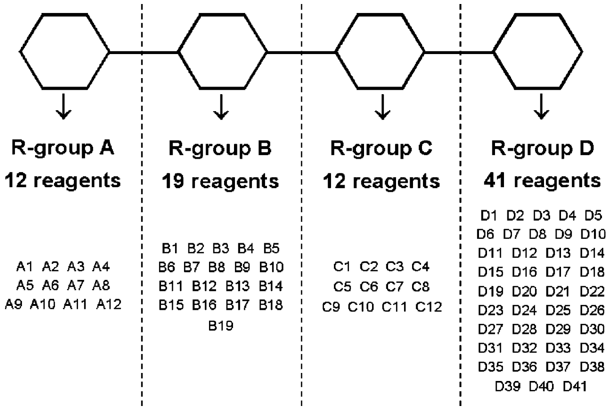

Figure 2. *Antidepressant library*: real combinatorial library composed of 4 R-groups and 12 × 19 × 12 × 41 = 112176 compounds.

reagent (or *reactant*), and each set of modifications on a part of the lead is called an *R-group* (or a *pool*).[1,3,37]

Let N be the number of R-groups and N_i the number of reagents in the i^{th} R-group ($i = 1, 2, ..., N$), $R_i = \{R_{ik}; k = 1, 2, ..., N_i\}$. The number of possible compounds obtained by combining a reagent from each R-group is $M = \prod_{i=1}^{N} N_i$. They form the combinatorial library. In most cases, its size is so huge that it is impossible to synthesize all the compounds in the library. Chemists have thus to select in each R-group n_i reagents ($n_i \leq N_i$, $i = 1, 2, ..., N$) and combine them in a full fashion to obtain a sublibrary of reasonable size. The chemists fix the desired number of selected reagents in each R-group according to its knowledge (not necessarly the same number in each R-group) and according to the number of compounds in the resulting sublibrary, $m = \prod_{i=1}^{N} n_i$. Indeed, the m selected compounds would then have to be tested in laboratories. The combinatorial nature of the compounds in the sublibrary allows an easy synthesis as far as their number is not too large compared to used plates.

In the lead optimization stage, reagents have to be selected in order to get the *best* sublibrary, i.e., a sublibrary containing compounds that optimize some chemical criteria. As there are $\binom{N_i}{n_i} = N_i!/(N_i - n_i)!n_i!$ different ways to select n_i reagents in the i^{th} R-group, the number of all possible sublibraries is $\prod_{i=1}^{N}\binom{n_i}{N_i}$, which is often huge in practice.

Along the next sections, a real combinatorial library is used to show the performance of the *WEALD* algorithm for optimizing sublibraries and to compare it to other methods. This library has been developed in the context of a new serotoninergic antidepressant and is referred as the *antidepressant library*.

The library is composed of $N = 4$ R-groups. The different R-groups are represented in Figure 2. There are $N_1 = 12$, $N_2 = 19$, $N_3 = 12$ and $N_4 = 41$ reagents in each R-group. The size of the whole combinatorial library is thus $M = \prod_{i=1}^{N} N_i = 12 \times 19 \times 12 \times 41 = 112176$. 112176 are too many compounds to synthesize as well as test.

Chemists at Eli Lilly want to select $n_i = 3$ ($i = 1, 2, ..., N$) reagents in each R-group to obtain a sublibrary of $m = \prod_{i=1}^{N} n_i = 3^4 = 81$ compounds that optimizes 8 properties of interest, denoted $P1$, $P2$, ..., $P8$. As the chemists were interested in two targeted receptors, properties $P1$ and $P2$ measure respectively the binding to the first targeted receptor and the functional assay on the second targeted receptor. Both properties have to be maximized. On the other side, $P3$, $P4$, $P5$ and $P6$ measure the binding to four other undesirable receptors, and those 4 criteria have to be minimized as the drug candidate has to be selective. Finally properties $P7$ and $P8$ are respectively the probability of nonmutagenocity and the opposite of the metabolization rate, both to be maximized. Those 8 criteria are predicted using statistical models build on historical data using common chemical descriptors (e.g. molecular weight, number of bounds, number of hydrogen atoms...) as input variables.

As there are $\prod_{i=1}^{N}\binom{N_i}{n_i} = \binom{12}{3} \times \binom{19}{3} \times \binom{12}{3} \times \binom{41}{3} \simeq 5 \times 10^{11}$ different ways to select three reagents in each R-group and combine them, there is a clear need for a method to find the one that optimizes simultaneously the 8 criteria of interest.

3. A WEIGHTING EXCHANGE ALGORITHM FOR LIBRARY DESIGN (*WEALD*)

3.1. Principles. This paper proposes a new method to select reagents in R-groups to obtain compounds that optimize some properties. The algorithm is iterative as most of the algorithms in this domain.

First, n_i reagents are selected at random in the i^{th} R-group and combined to form an initial sublibrary with $m = \prod_{i=1}^{N} n_i$ compounds. Then exchanges are performed on the current sublibrary to improve the quality of its compounds. The principle of exchanges was introduced in the experimental D-optimal design by Fedorov[13] in 1972. This principle is applied to select a reagent in the current sublibrary and replace it by a new one. But exchanges are not done in a naive way at random. First a criteria has been developed to quantify in one index the quality of a sublibrary. This is explained in section 3.2. Second a weighting method summarizes all the information accumulated till each iteration to guide the reagents exchanges. This is explained with more details further in section 3.3.

3.2. Quality of a Sublibrary. The algorithm objective is to select in a combinatorial fashion compounds that optimize 8 properties. In front of such a multiobjective problem, the literature often proposes to aggregate those variables into a unique value between 0 and 1 that is a compromise of all the properties values. This is the *desirability* principle introduced by Harrington in 1965.[20] The quality of a compound is quantified by a *fitness* value lying between 0 and 1 that summarizes the properties of interest. The higher the fitness is the better the compound fits the criteria.

Let P be the number of properties to optimize. In the *antidepressant library*, P equals 8, and the fitness of a molecule is computed using the Derringer principle[11] with the next formula

$$fitness = \prod_{i=1}^{P}[f_i(P_i)]^{w_i}$$

where P_i is the i^{th} property, f_i is a function that maps the i^{th} property in the interval [0, 1], and w_i is a weight chosen to emphasize (or not) the i^{th} property with $\sum_{i=1}^{P} w_i = 1$.

Figure 3 displays three general forms of functions f_i proposed by Derringer. The first and the second graphs show increasing and decreasing functions that can be used when the property P_i has to be maximized or minimized, respectively. The last graph shows a function f_i that can be used when the property P_i must reach an optimal value OPT_{Pi}.

The graphs of Figure 4 display the values of the 8 properties to optimize computed for all the compounds in the combinatorial library. The best molecule is represented with a dark circle on each graph and has a fitness value of 0.6908.

The overall quality of a sublibrary is quantified by a *loss* value lying between 0 and 1 that summarizes the differences between the fitness of its *m* compounds and their ideal value 1:

$$loss = \frac{1}{m}\sum_{j=1}^{m}[1 - fitness_j]^2$$

Smaller is the loss better is the sublibrary.

To summarize, a *fitness* value is associated with each compound (the highest, the best), and a *loss* value is associated with each sublibrary (the smallest, the best). In the *WEALD* algorithm, exchanges between reagents are repeatedly performed to decrease the loss of the current sublibrary.

3.3. Exchanges Methodology. The *WEALD* algorithm starts by selecting at random n_i reagents in the i^{th} R-group and combining them to form an initial sublibrary with $m = \prod_{i=1}^{N} n_i$ compounds. Then, repeatedly, exchanges are performed to obtain a better sublibrary. In an exchange, one reagent is ejected from the current sublibrary and replaced by another one of the same R-group to decrease the loss of the current sublibrary.

First the reagent to be ejected out of the current sublibrary is selected. For each current selected reagent, the mean of the fitness associated with the compounds in the sublibrary containing this reagent is computed. The worst one is ejected, i.e., the one with the smallest average fitness.

In a second step, a new reagent is selected to replace the ejected one in the set of reagents of the same R-group that are not yet in the current sublibrary. This choice is drawn randomly according to probabilities of selection associated with each reagent. At the beginning of the algorithm, in each R-group, all reagents have the same probability to enter the sublibrary. Then, at each iteration, probabilities of selection are adapted using all the fitness already computed according to the following rules:

Rule 1: If a reagent has never been tested for an exchange, it is associated with a selection probability of 1.

Rule 2: If a reagent has already been tested for an exchange, it is associated with a probability of selection which is the ratio between the average fitness of all the compounds already explored that contains this reagent and the maximum fitness computed till that step.

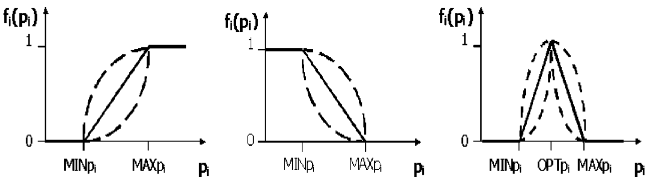

Figure 3. Different kinds of functions f_i proposed by Derringer[11] to map chemical properties in the interval [0, 1].

Rule 3: If the average fitness of the compounds obtained using a reagent is lower than the 10th percentile of all the fitness computed till that step and if more than 1% of the compounds in the whole combinatorial library have been explored, the probability of selection for this reagent will be set at 0.

All these selection probabilities are of course standardized to sum to 1 in each R-group. Rule 1 ensures that a reagent not explored yet will have the highest selection probability for next exchanges. According to rule 2, exchanges with a reagent that provides good compounds will be preferentially tested. And with rule 3 a bad reagent will definitely be excluded from the sublibrary and never more tried for an exchange.

Finally, the exchange between the entering reagent drawn randomly according to selection probabilities and the ejected worst one is applied if the loss of the new sublibrary is lower than the loss of the current sublibrary. If it is not the case, another reagent of the same R-group is tried to replace the ejected one, and if no exchange with the ejected reagent decreases the loss it is kept till the convergence of the algorithm.

The algorithm ends when there is no more possible exchange with a reagent having strictly positive selection probability.

The different steps of the *WEALD* algorithm are summarized next:

Step 0: - Initialize a sublibrary at random $S^0 = S^0{}_1 \times S^0{}_2 \times ... \times S^0{}_N$ where $S^0{}_i = \{R_{ik}; k = 1, 2, ..., n_i\}$ is a set of selected reagents of the i^{th} R-group, R_i, and record its loss L^0;

- Set the best sublibrary $S^b = S^0$ and its loss $L^b = L^0$;

- Initialize the selection probabilities for each reagent, R_{ij}, by $w_{ij} = 1/N_i$ ($j = 1, ..., N_i$ and $i = 1, ..., N$);

- Go to step 1.

Step 1: If $w_{ij} = 0\ \forall i$ and $\forall j$ stop and output S^b else execute step 2.

Step 2: - Select the worst reagent of S^b, denoted R_{IK};

- Select the set of possible reagents for an exchange, denoted $R^* = R_I \setminus S^b{}_I = \{R_{It}, t = 1, ..., N_i - n_i\}$;

- Go to step 3.

Step 3: - If R^* is empty or if $w_{It} = 0\ \forall t$, keep R_{IK} in S^b till convergence and never try it again for an exchange; update the selection probabilities; return to step 1.

- Else generate a trial sublibrary S^t by replacing R_{IK} in S^b by a reagent of R^*, denoted R_{IT}, chosen according to the selection probabilities; compute its loss L^t; go to step 4;

Step 4: - If $L^t < L^b$ set $S^b = S^t$ and $L^b = L^t$; update the selection probabilities; return to step 1;

- If $L^t \geq L^b$ set $R^* = R^* \setminus R_{\mathbf{IT}}$ and return to step 3;

3.4. Application on the *Antidepressant Library*. We applied the *WEALD* algorithm (implemented in the S-Plus software) on the *antidepressant library* (12 × 19 × 12 ×

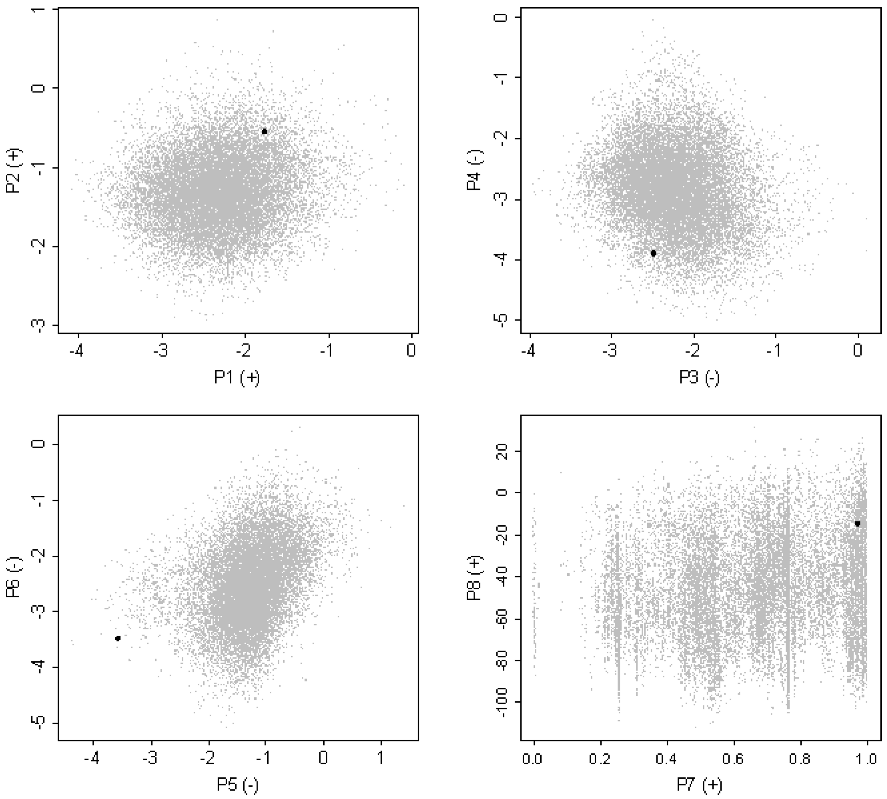

Figure 4. Values of the 8 properties *P*1, *P*2, ..., *P*8 computed for the 112176 compounds of the *antidepressant library*. *P*1 = binding to a targeted receptor; *P*2 = functional assay on another targeted receptor; *P*3, *P*4, *P*5 and *P*6 = binding to other receptors; *P*7 = probability of nonmutogenicity and *P*8 = − metabolization rate. Besides properties names, (+) stands for maximized property and (−) stands for minimized property. The dark circle represents the best molecule (fitness = 0.6908).

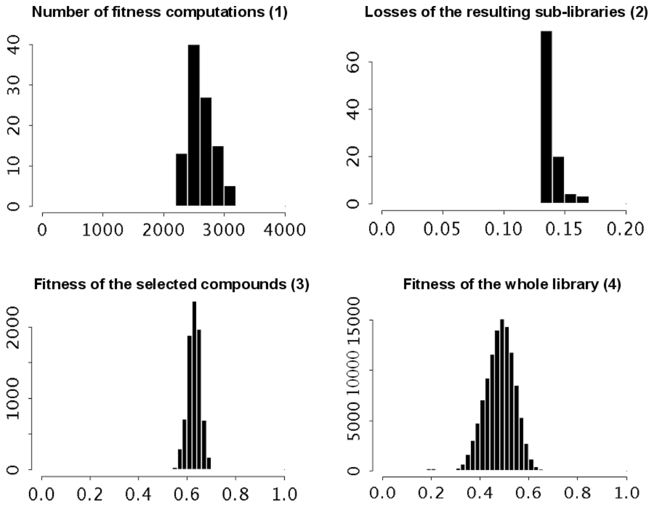

Figure 5. Results of the application of the *WEALD* algorithm on 100 different initial sublibraries. (1) Number of fitness computations, (2) Losses of the final sublibraries, (3) Fitness of the compounds in the final sublibraries and (4) Fitness of the 112176 compounds in the whole library.

41) to select $n_i = 3$ reagents in each R-group. By combining them, $m = 81$ compounds are obtained, and they have to optimize 8 criteria of interest. As described in section 3.2, fitness values are used to quantify the quality of compounds and a loss value to measure the quality of a sublibrary. As the *WEALD* algorithm is not deterministic and as it may depend on the starting sublibrary, it was applied 100 times, each time with a different random initial sublibrary. This section analyzes the rate of convergence of the *WEALD* algorithm and the quality of the sublibraries obtained.

The first important result is that the *WEALD* algorithm converges rapidly. On average, 2613 fitness are computed before the algorithm stops. This represents only 2.33% of the whole combinatorial library. In addition, this fast rate of convergence is quite stable, whatever is the initial sublibrary. Indeed, each time the *WEALD* algorithm had been applied, the convergence was reached by computing between 2220 and 3114 fitness (between 2% and 2.8% of the whole library). This can be seen on the histogram (1) of Figure 5 representing the distribution of the numbers of fitness computations for the 100 runs of the algorithm.

The second important result is that the quality of the sublibraries obtained with the *WEALD* algorithm is quite high. Indeed, losses of the 100 sublibraries are quite small, on average 0.1382. They are also stable if we analyze the 100 losses represented on histogram (2) of Figure 5.

The fitness of the compounds in the 100 sublibraries are high, on average 0.6291, which is the 99, 71st percentile of the fitness of the whole combinatorial library. One can conclude by analyzing those numbers or comparing histograms (3) and (4) of Figure 5 that the *WEALD* algorithm provides a sublibrary containing most of the best compounds (highest fitness) of the initial combinatorial library.

Those 100 runs of the *WEALD* algorithm allow for the conclusion that it converges rapidly and provides a sublibrary with the most promising compounds whatever is the initial sublibrary. In real cases applications, the *WEALD* algorithm can be applied only one time as it is stable. Nevertheless, to prevent a local minimum it can be recommended to repeat the application a few times (2 or 3) with other initial sublibraries.

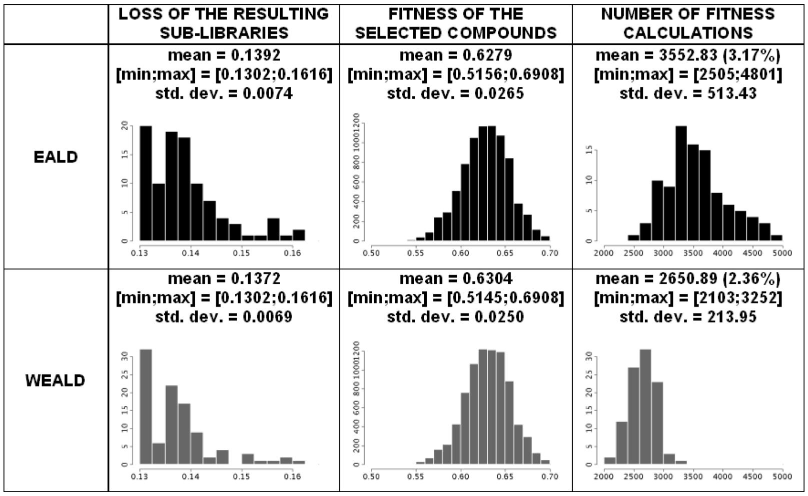

Figure 6. Results of the application of the *EALD* algorithm and the *WEALD* algorithm on 100 initial sublibraries.

3.5. What Happens When Applying the *WEALD* Algorithm without Using Selection Probabilities? We applied the principles of the *WEALD* algorithm on the *antidepressant library* but without using selection probabilities (algorithm called *Exchange Algorithm for Library Design − EALD*) to confirm that those proposed weights help finding more rapidly a sublibrary 3^4 containing compounds that optimize the 8 properties of interest.

Both algorithms, *WEALD* and *EALD*, are applied on 100 different initial sublibraries, and the rates of convergence and the quality of the obtained sublibraries are compared. The results of those simulations are summarized in Figure 6. We conclude that using probabilities of selection accelerates the convergence because the *WEALD* algorithm only computes on average 2651 fitness which is only 2, 36% of the whole combinatorial although *EALD* computes on average 3553 fitness which corresponds to 3, 17% of the whole combinatorial library. Moreover, the *WEALD* algorithm provides slightly better sublibraries than *EALD* according to the average losses that are smaller with *WEALD* (0.1372) than with *EALD* (0.1392) or according to the fitness of the selected compounds that are a little higher with *WEALD* (0.6304) than with *EALD* (0.6279).

In conclusion, using the probabilities of selection to guide exchanges allows a faster convergence through better sublibraries.

4. COMPARISON WITH THE *ULTRAFAST* ALGORITHM

Agrafiotis and Lobanov[1] proposed an iterative algorithm to design a focused library. As most of the algorithms in this domain, it starts with a randomly chosen sublibrary, and its quality is evaluated by a single value that summarizes the properties to optimize. Then it optimizes each R-group in sequence until no further improvement is possible.

More precisely the different steps are summarized as follows:

Step 0: - Initialize a sublibrary at random: $S^0 = S^0{}_1 \times S^0{}_2 \times ... \times S^0{}_N$ where $S^0{}_i = \{R_{ik};\ k = 1, 2, ..., n_i\}$ is a set of selected reagents of R-group R_i and record its loss L^0;

- Set the current sublibrary $S^c = S^0$ and its loss $L^c = L^0$;

- Set the best sublibrary $S^b = S^0$ and its loss $L^b = L^0$;

- Go to step 1.

Step 1: Perform steps 2 to 4 for $i = 1, ..., N$ (i.e. for each R-group R_i) then go to step 5.

Step 2: Perform step 3 for $j = 1, ..., N_i$ (i.e. for each reagents R_{ij} of the i^{th} R-group R_i).

Step 3: Evaluate the sublibrary $S = S^c{}_1 ... \times S^c{}_{i-1} ... \times R_{ij} \times S^c{}_{i+1} ... \times S^c{}_N$ and record its loss L_{ij}.

Step 4: Sort the reagents in the i^{th} R-group R_i in ascending order of loss L_{ij} and select the n_i reagents with the smallest losses: $S^t{}_i = \{R_{ik};\ k = 1, 2, ..., n_i\}$.

Step 5: Evaluate the trial sublibrary $S^t = S^t{}_1 \times S^t{}_2 \times ... \times S^t{}_N$ and record its loss L^t. If $L^t < L^b$ set $S^c = S^b = S^t$ and $L^c = L^b = L^t$ then return to step 1 else stop and output S^b.

This is a systematic search through the solutions space: for each R-group R_i, all possible reagents are combined with the current selected ones of the other R-groups and the n_i best reagents are selected.

The *WEALD* algorithm and the *Ultrafast* algorithm were compared on the *antidepressant library* to select three reagents in each R-group. Both algorithms were applied on 50 different initial sublibraries chosen at random. By analyzing the sublibraries obtained with each algorithm, one can conclude that the *WEALD* algorithm converges 3 times faster than the *Ultrafast* algorithm, but the quality of the sublibraries is nevertheless slightly higher (losses of the obtained sublibraries smaller and fitness of the selected compounds in these sublibraries higher). In addition, the *WEALD* algorithm is less sensitive to the choice of the initial sublibrary (standard deviations smaller). Those results are summarized in Figure 7.

Figure 8 shows the evolution of the loss according to the number of fitness computations for the two algorithms applied on the same initial sublibrary. The losses at the end of the two methods are quite the same, but the rate of convergence of the *WEALD* algorithm is clearly higher than the *Ultrafast* algorithm. With the *Ultrafast* method, only 6 iterations are performed but for each one, $(12-3) \times (3^3) + (19-3) \times (3^3) + (12-3) \times (3^3) + (41-3) \times (3^3) = 1944$ fitness are first computed to select the three best reagents in each R-group, and then $3^4 = 81$ fitness are computed to evaluate the quality of the sublibrary obtained by combining the three best reagents in each R-group. At each iteration around 2000 fitness have to be computed with the *Ultrafast* algorithm although with the *WEALD* algorithm, this number could be only $1 \times 3^3 = 27$ if only one exchange is tried or

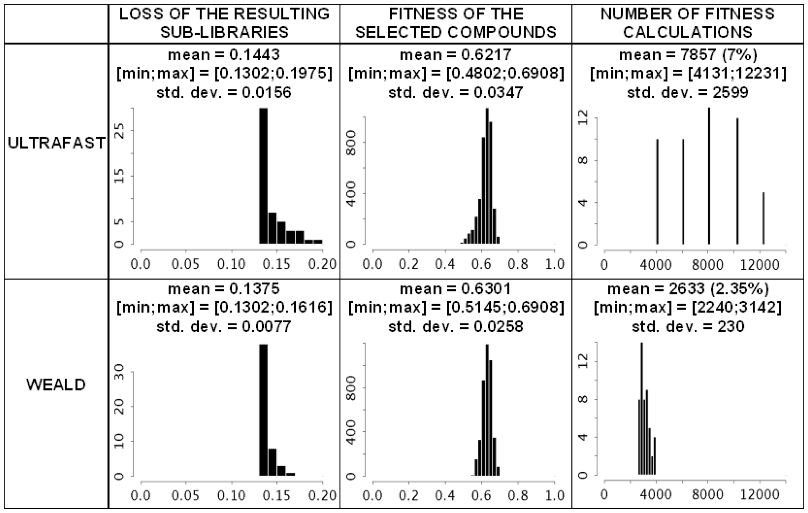

Figure 7. Results of the application of the *Ultrafast* algorithm and the *WEALD* algorithm on 50 different initial sublibraries chosen at random.

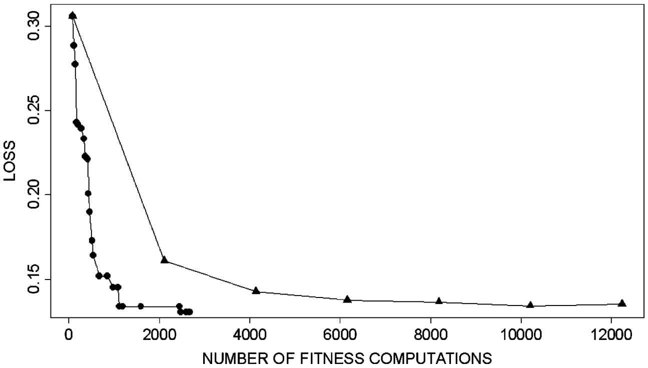

Figure 8. Loss as a function of the number of fitness computations when applying the *Ultrafast* algorithm (triangles) and the *WEALD* algorithm (circles) on the same initial sublibrary.

$N_e \times 3^3$ if N_e exchanges are tried before having an exchange that decreases the loss.

A drawback of the *Ultrafast* algorithm is that the fitness computed at each iteration are entirely neglected when performing next iterations. This is not the case with the *WEALD* algorithm that uses computed fitness to guide exchanges by using the selection probabilities as explained in section 3.3. In addition, with those selection probabilities, when the *WEALD* algorithm has converged, one does not need to perform all possible exchanges to see that, indeed, the algorithm has converged because part of the reagent is associated with a selection probability of 0.

5. COMPARISON WITH THE *PICCOLO* ALGORITHM

Simulated annealing is a well-known iterative method to solve discrete and combinatorial optimization. Zheng et al.[37] suggest applying its principles to solve the problem of focused library design by also using a single criteria to summarize the properties of interest.

Let t_0, r_0 and μ be respectively the initial temperature, the number of iterations in an annealing series and the temperature reducing factor; the general framework of the simulated annealing is the following:

Step 0: - Generate an initial sublibrary S^0 randomly and record its loss L^0;

- Set $t = t_0$, $r = r_0$ and flag = False;

- Set the current sublibrary $S^c = S^0$ and its loss $L^c = L^0$;

- Set the best sublibrary $S^b = S^0$ and its loss $L^b = L^0$;

- Go to step 1.

Step 1: Generate a trial sublibrary S^t by perturbing the current one S^c (*cfr* below). Calculate $\Delta L = L^t - L^c$. If $\Delta L < 0$ then execute step 3; otherwise execute step 2.

Step 2: Compute $P = e^{-\Delta L/t}$ and compare it with a random value y from a uniform distribution in [0, 1]. If $y < P$ then execute step 4; otherwise execute step 3.

Step 3: Set $S^c = S^t$. If $L^b \geq L^t$ then set $S^b = S^t$. If $\Delta L < 0$ then set flag = True.

Step 4: Set $r = r - 1$. If $r > 0$ then return to step 1.

Step 5: If flag = True, then set flag = False, $r = r_0$, $t = t\mu$ and repeat step 1; otherwise stop and return S^b.

The perturbation scheme used when applying those principles to design a focused library is also based on exchanges. Zheng et al. propose to first choose the R-group with a probability proportional to the average of $N_i/\sum_{i=1}^{N} N_i$ and $n_i/\sum_{i=1}^{N} n_i$. In this chosen R-group, they use a uniform random sampling approach to select the reagent that enters the sublibrary and the one that is ejected.

Three parameters must be fixed at the beginning of the *Piccolo* algorithm: t_0, r_0 and μ. The initial temperature t_0 is generally determined in order that the initial probability for accepting bad moves be approximately equal to a prescribed value P_0. We chose t_0 to obtain $P_0 = 0.5$. For the number of iterations in an annealing series, r_0, and the temperature reducing factor, μ, 6 different combinations of those parameters were tested: $\mu = 0.6$ or 0.9 and $r_0 = 12$, 20 or 41 (the minimum, the mean or the maximum number of reagents per R-group).

The *WEALD* algorithm and the *Piccolo* algorithms (*Piccolo* with the 6 different combinations of parameters) were applied on 30 different initial sublibraries. The extreme results are displayed in Figure 9.

On the quality point of view, the smallest losses and the highest fitness were obtained with $r_0 = 41$ and $\mu = 0.9$. But the sublibraries obtained with the *WEALD* algorithm are slightly better moreover with less fitness computations. This may be due to the fact that, thanks to the selection probabilities used in the *WEALD* algorithm, sublibraries are faster guided through sublibraries with compounds having

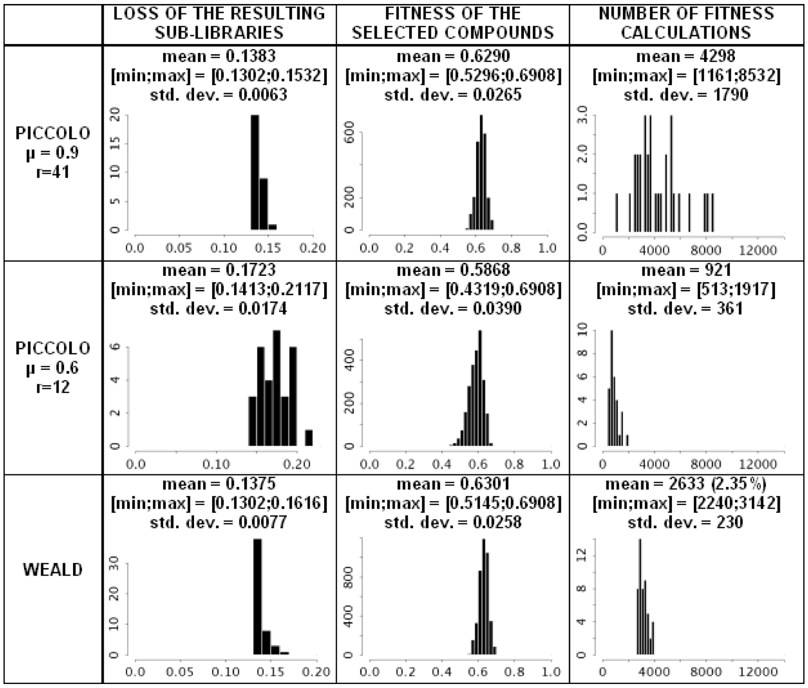

Figure 9. Results of the application of the *Piccolo* algorithm with the two extremes parameters r_0 and μ combinations ($r_0 = 41$ and $\mu = 0.9$; $r_0 = 12$ and $\mu = 0.6$) and the *WEALD* algorithm on 30 initial sublibraries.

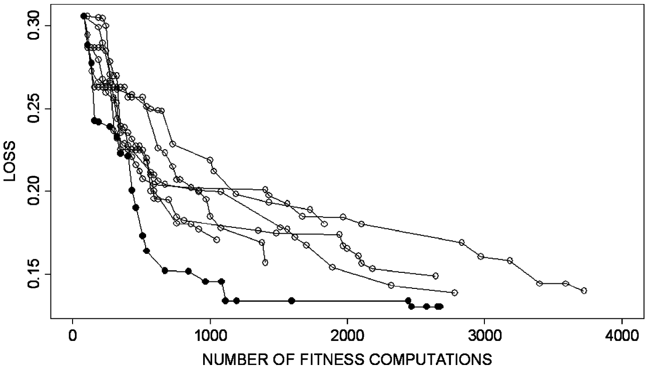

Figure 10. Loss as a function of fitness computations when applying the *Piccolo* algorithm (empty circles) with the 6 parameters combinations ($\mu = 0.6$ or 0.9 and $r_0 = 12$, 20 or 41) and the *WEALD* algorithm (dark circles) on the same initial sublibrary.

good properties than the *Piccolo* algorithm that uses a random sampling approach to select reagents that enter the sublibrary.

On the convergence rate point of view, the smallest number of fitness computations was reached with $r_0 = 12$ and $\mu = 0.6$: less than 1% of the fitness of the whole library had been calculated. It is faster than the *WEALD* algorithm but too fast to give good results: the average loss of the 30 sublibraries is 0.1723 against 0.1375 with *WEALD* and the average fitness of the compounds in the 30 sublibraries is 0.5868 against 0.6301 with *WEALD*.

We conclude that, on the *antidepressant library*, the *WEALD* algorithm has the best quality/price (loss/number of fitness computations) ratio, as it can also be seen in Figure 10 representing the evolution of the loss according to the number of fitness computations for the two algorithms applied on the same initial sublibrary.

6. ALGORITHMS COMPARISON ON A SECOND COMBINATORIAL LIBRARY

We reach the same conclusions if the three algorithms, *Ultrafast*, *Piccolo* and *WEALD*, are applied on another real combinatorial library. This second library has been built at Eli Lilly in the context of the development of a new molecule acting on the regulation of glucose in type II diabetes. It will be further referred to as the *diabetes example*. This combinatorial library is composed of three R-groups, with respectively 47, 50 and 47 reagents leading to $M = 47 \times 50 \times 47 = 110450$ possible compounds. The three algorithms were used to select a combinatorial sublibrary with $m = 5 \times 5 \times 5 = 125$ compounds optimizing 10 criteria of interest, $P1$, $P2$, ..., $P10$.

In the same context, two molecules are well-known, and chemists want to obtain a new molecule that has a similar structure as the first one but a completely different structure as the second one. Properties $P1$ and $P2$ are appropriate distance measures and have to be respectively minimized and maximized. Properties $P3$ and $P4$ measure binding and functional assay on a targeted receptor and both have to be maximized. To quantify selectivity, properties $P5$, $P6$, $P7$, $P8$ and $P9$ measure the binding to five other undesirable receptors. Finally property $P10$ is the opposite of the metabolization rate and has to be maximized. The distances $P1$ and $P2$ are based on chemical descriptors and are computed although properties $P3$, $P4$, ..., $P10$ are predicted using statistical models developed on historical data. The multicriteria optimization is converted to a single objective optimization using a fitness value to summarize the properties of each compounds as described in section 3.2.

WEALD converges nearly 4 times faster than *Ultrafast* and 3 times faster than *Piccolo* (with parameters $r_0 = 300$ and $\mu = 0.95$), with a number of fitness computations of 3549, 13830 and 9890, respectively. *WEALD* provides also better sublibraries than *Ultrafast* with an average fitness for the selected compounds of 0.5815 and 0.5309, respectively. With a higher number of fitness computations, *Piccolo* provides slightly better sublibraries with an average fitness of 0.5990 (the 99, 64th percentile of fitness distribution) against 0.5815 with *WEALD* (the 99, 53th percentile of fitness distribution).

Figure 10 shows the evolution of the loss according to the number of fitness computations for the three algorithms applied on the same initial sublibrary that has a loss of 0.689. The losses at the convergence of *Piccolo* and *WEALD* are quite the same, but the rate of convergence of *WEALD* is clearly higher than *Piccolo*. One can also see that *WEALD* provides a faster better sublibrary than *Ultrafast*.

7. SIMULATIONS WITH DIFFERENT COMBINATORIAL LIBRARY DIMENSIONS

The good performance of the *WEALD* algorithm is only demonstrated through the previous sections by means of the *antidepressant library* and the *diabetes example*. This section compares the three algorithms with other combinatorial libraries to check if the *WEALD* algorithm still has the best quality/price ratio. More precisely, this section addresses the following question: what is the impact of the combinatorial library dimensions (the number of R-groups, the number of reagents, the number of selected reagents...) on the performance of the three algorithms? We thus have performed a set of simulations to check that the *WEALD* algorithm ensures a good balance between the loss of the resulting sublibrary and the corresponding number of fitness computations to attain it whatever the dimensions of the whole combinatorial library are.

Simulations are performed on a set of variations of the *antidepressant library* used in sections 2 to 5. The next 4 factors vary to obtain different combinatorial library dimensions:

***Nreag*:** the average number of reagents per R-group. It takes three values, 10, 30 or 100, by resampling reagents of the *antidepressant library*.

***MaxMin*:** the ratio of the maximum and minimum number of reagents per R-group. It takes two values: 1 or 3.

***N*:** the number of R-groups. It takes three values: 4, 3 (by choosing at random one of the 4 R-groups of the *antidepressant library* and fixing at random a reagent in this R-group) or 2 (by choosing at random two of the 4 R-groups of the *antidepressant library* and fixing at random a reagent in each of the two R-groups).

***Nselec*:** the number of selected reagents per R-group. It takes values 2, 3, 4 or 5.

A complete factorial design was used, and for each of the $3 \times 2 \times 3 \times 4 = 72$ combinations of factors 100 initial sublibraries were initialized at random and the three algorithms were applied.

By analyzing the average loss and the average number of fitness computations for each of the 72 cases, one can conclude that the *WEALD* algorithm converges always faster than the *Ultrafast* and the *Piccolo* algorithms. The *WEALD* algorithm and the *Piccolo* algorithm reach very similar losses (same quality of resulting sublibraries), smaller than losses attained with the *Ultrafast* algorithm.

Those simulations have also been used to build a prediction model of the number of computed fitness, *Nfitness*, as a function of the 4 factors, *Nreag*, *MaxMin*, *N* and *Nselec*. A full second-order regression model was fitted on the data using the logarithm of *Nfitness* as the response, and nonsignificant effects were removed. The following model has finally been obtained: $\ln(Nfitness) = 2.044 + 0.026 \times Nreag - 0.393 \times Nselec + 0.530 \times N + 0.329 \times Nselec \times N$. We conclude that, as $N > 2$, all three factors, *Nreag*, *N* and *Nselec*, have a positive effect on the number of fitness computations. We also conclude that *MaxMin*, the ratio of the maximum and minimum number of reagents per R-group, seems to have no significant effect on the number of computed fitness.

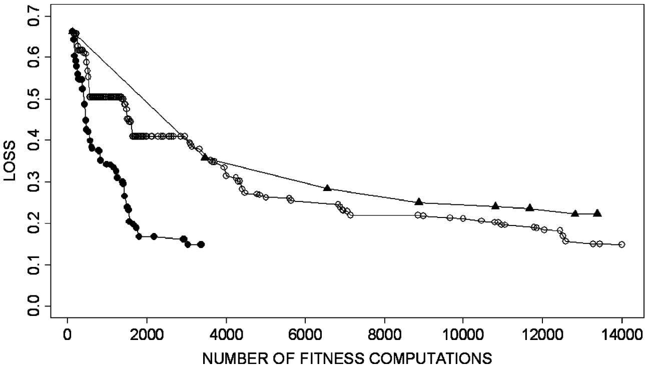

Figure 11. Loss as a function of the number of fitness computations when applying the *WEALD* algorithm (dark circles), the *Ultrafast* algorithm (triangles), and the *Piccolo* algorithm (empty circles) on the same initial sublibrary.

As R^2 is 0.950 (Adjusted R^2 is 0.947), one may hope to use this model for prediction when applying the *WEALD* algorithm to solve a new combinatorial library design optimization.

8. *WEALD* AS A TOOL TO SEARCH FOR THE BEST COMPOUNDS IN A COMBINATORIAL LIBRARY

In the previous sections, the three presented algorithms focused on the selection of combinatorial sublibraries that optimize some drugability properties. The combinatorial nature of the selected compounds makes the synthesis in laboratories directly in plates easier, but this is not always a priority. One may also want to select the *m* best compounds in a sublibrary whatever reagents are used. In such a context, the *WEALD* algorithm may also be used after some slight modifications.

Three points are adapted to this more general optimization problem:

Initialization: *m* compounds are selected at random in the set of all possible compounds.

Choice of the ejected reagent: the worst compound in the current sublibrary is first selected (the one with the smallest fitness), and one of its reagents is selected at random to be ejected.

Selection probabilities: the selection probabilies are computed as before except that probabilities are not any more set at zero when a reagent is considered as "bad" (see rule 3 of section 3.3).

This adaptation of the *WEALD* algorithm was applied on the *antidepressant library* to select a sublibrary with $m = 100$ best compounds. Hundreds of simulations were performed, each time with a different initial sublibrary. We conclude that, by computing on average 6937 fitness (6.18% of the whole library), the sublibrary obtained at the end of the algorithm contains on average 80 of the 100 best compounds. Those numbers are averaged over the 100 runs. For one sublibrary provided by this adapted *WEALD* algorithm, Figure 12 shows the rank of the selected compounds in the full library. This shows that the 100 selected

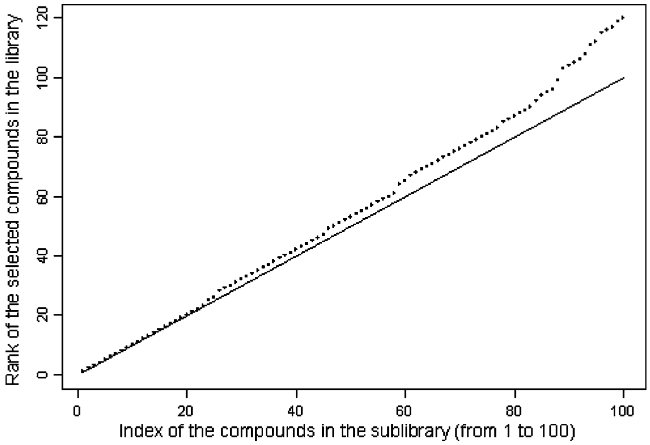

Figure 12. WEALD applied to select the 100 best compounds: the points are the ranks of the selected compounds (indexed by numbers from 1 to 100) in the whole combinatorial library and the straight line is the ideal rank (1 to 100).

compounds are in the set of the 120 best compounds, and around 80 of the 100 best compounds were found.

9. CONCLUSION

Combinatorial chemistry has revolutionized the drug discovery process. It can be used in screening for a hit or a lead, as well as in the lead optimization stage. It allows for the generation of a huge compounds library by systematically combining building blocks composed of reagents.

The *WEALD* algorithm is a new iterative method based on exchanges to explore a combinatorial library and select a sublibrary of reasonable size that optimize druglike properties. The *WEALD* algorithm can be used to obtain either a combinatorial sublibrary containing compounds generated by combinations of a selected reagents subsets or either a sublibrary composed of the *m* best compounds whatever reagents are used. With both objectives, the same methodology is applied to summarize the criteria to optimize: the quality of a molecule is quantified by a desirability index and the quality of a sublibrary is quantified by a loss value.

The *WEALD* algorithm is compared to two other algorithms developed for library design, the *Ultrafast* algorithm of Agrafiotis and Lobanov[1] and the *Piccolo* algorithm of Zheng et al.[37] The three algorithms are applied on two real combinatorial libraries, the *antidepressant library* and the *diabetes example*, to select a combinatorial sublibrary. The *WEALD* algorithm reveals to have the best quality/price ratio: it converges rapidly, exploring only a small part of the whole combinatorial library (2.33% with the *antidepressant library* and 3.21% with the *diabetes example*), and provides sublibraries with most of the best compounds, i.e. compounds with the highest fitness values. Comparisons on other simulated combinatorial libraries allow for the conclusion that the *WEALD* algorithm converges the fastest and also provides sublibraries of quality similar to the ones obtained with the *Piccolo* algorithm whatever the dimensions of the combinatorial libraries are.

The advantage of the *WEALD* algorithm is the weighting methodology developed to guide the exchanges between reagents. According to some fixed rules, a probability of selection is associated with each reagent, and they are updated at each iteration taking into account the quality of the explored compounds. Those selection probabilities are used to choose the new reagent that enters the sublibrary and replace the worst actual one. Those weights allow a faster convergence through better sublibraries.

ACKNOWLEDGMENT

The authors thank Eli Lilly for providing the two real combinatorial libraries, the *antidepressant library* and the *diabetes example*. A special thanks to Philippe Remusat of Lilly Services S.A. who spent a lot of his time implementing the *WEALD* algorithm in *SAS* and making its application possible at Eli Lilly, linking the *WEALD* algorithm to all other softwares used in the lead optimization stage.

REFERENCES AND NOTES

(1) Agrafiotis, D. K.; Lobanov, V. S. Ultrafast Algorithm for Designing Focused Combinatorial Arrays. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1030−1038.

(2) Agrafiotis, D. K.; Lobanov, V. S.; Salemme, F. R. Combinatorial Informatics in the Post-Genomics Era. *Nat. Rev. Drug Discov.* **2002**, *1*, 337−346.

(3) Agrafiotis, D. K. Multiobjective Optimization of Combinatorial Libraries. *J. Comput.-Aided Mol. Des.* **2002**, *16*, 335−356.

(4) Agrafiotis, D. K. Stochastic Algorithms for Maximizing Molecular Diversity. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 841−851.

(5) Agrafiotis, D. K.; Lobanov, V. S.; Rassokhin, D. N.; Izrailev, S. The Measurement of Molecular Diversity. In *Virtual Screening of Bioactive Molecules*; Böhm, H.-J., Schneider, G., Eds.; Wiley-VCH: Weinheim, 2000; pp 265−300.

(6) Agrafiotis, D. K.; On the Use of Information Theory for Assessing Molecular Diversity. *J. Chem. Inf. Comput. Sci.* **1997**, *37(3)*, 576−580.

(7) Clark, D. E.; Westhead, D. R. Evolutionary Algorithms in Computer-Aided Molecular Design. *J. Comput.-Aided Mol. Des.* **1996**, *10(4)*, 337−358.

(8) 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 Commercially Available Compounds. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 750−763.

(9) Cunningham, M. J. Genomics and Proteomics: the New Millenium of Drug Discovery and Development. *J. Pharmacol. Toxicol. Meth.* **2000**, *44(1)*, 291−300.

(10) Debouck, C.; Metcalf, B. The Impact of Genomics on Drug Discovery. *Annu. Rev. Pharmacol. Toxicol.* **2000**, *40*, 193−207.

(11) Derringer, G. C.; Suich, R. Simultaneous Optimization of Several Response Variables. *J. Qual. Technol.* **1980**, *12(4)*, 214−219.

(12) Drews, J. Drug Discovery Today and Tomorrow. *Drug Discov. Today* **2000**, *5*, 2−4.

(13) Fedorov, V. V. *Theory of Optimal Experiments*; Academic Press: NY, 1972.

(14) Floyd, C. D.; Leblanc, C.; Whittaker, M.; Combinatorial Chemistry as a Tool for Drug Discovery. *Prog. Med. Chem.* **1999**, *36*, 91−168.

(15) Gallop, M. A.; Barrett, R. W.; Dower, W. J.; Fodor, S. P. A.; Gordon, E. M.; Applications of Combinatorial Technologies to Drug Discovery. 1. Background and Peptide Combinatorial Libraries. *J. Med. Chem.* **1994**, *37(9)*, 1233−1251.

(16) Gillet, V. J.; Willett, P.; Bradshaw, J.; Green, D. V. S. Selecting Combinatorial Libraries to Optimize Diversity and Physical Properties. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 169−177.

(17) Gillet, V. J.; Willett, P.; Bradshaw, J. The Effectiveness of Reactant Pools for Generating Structurally Diverse Combinatorial Libraries. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 731−740.

(18) Gillet, V. J.; Willett, P.; Fleming, P. J.; Green, D. V. Designing Focused Libraries Using MoSELECT. *J. Mol. Graph. Model.* **2002**, *20(6)*, 491−498.

(19) Gordon, E. M.; Barrett, R. W.; Dower, W. J.; Fodor, S. P. A.; Gallop, M. A. Applications of Combinatorial Technologies to Drug Discovery. 2. Combinatorial Organic-Synthesis, Library Screening Strategies, and Future-Directions. *J. Med. Chem.* **1994**, *37(10)*, 1385−1401.

(20) Harrington, J. The Desirability Function. *Ind. Quality Control* **1965**, *21(10)*, 494−498.

(21) Hassan, M.; Bielawski, J. P.; Hempel, J. C.; Waldman, M. Optimization and Visualization of Molecular Diversity of Combinatorial Libraries. *Mol. Divers.* **1996**, *2*, 64−74.

(22) Judson, R. Genetic Algorithms and Their Use in Chemistry. *Rev. Comput. Chem*. **1997**, *10*, 1−73.

(23) Lam, K. S. Application of Combinatorial Library Methods in Cancer Research and Drug Discovery. *Anticancer Drug Des.* **1997**, *12(3)*, 145−167.

(24) Leach, A. R.; Hann, M. M. The *in silico* World of Virtual Libraries. *Drug Discov. Today* **2000**, *5*, 326−336.

(25) Lewis, R. A.; Mason, J. S.; McLay, I. M. Similarity Measures for Rational Set Selection and Analysis of Combinatorial Libraries: The Diverse Property-Derived (DPD) Approach. *J. Chem. Inf. Comput. Sci.* **1997**, *37(3)*, 599−614.

(26) Neamati, N.; Barchi, J. J. New Paradigms in Drug Design and Discovery. *Curr. Top. Med. Chem.* **2002**, *2*, 211−227.

(27) Ohlstein, E. H.; Ruffolo, R. R.; Elliott, J. D. Drug Discovery in the Next Millenium. *Annu. Rev. Pharmacol. Toxicol.* **2000**, *40*, 177−191.

(28) Pickett, S. D.; McLay, I. M.; Clark, D. E. Enhancing the Hit-to-Lead Properties of Lead Optimization Libraries. *J. Chem. Inf. Comput. Sci.* **2000**, *40(2)*, 263−272.

(29) Sheridan, R. P.; Kearsley, S. K. Using a Genetic Algorithm to Suggest Combinatorial Libraries. *J. Chem. Inf. Comput. Sci.* **1995**, *35(2)*, 310−320.

(30) Singh, J.; Ator, M. A.; Jaeger, E. P.; Allen, M. P.; Whipple, D. A.; Soloweij, J. E.; Chowdhary, S.; Treasurywala, A. M. Application of Genetic Algorithms to Combinatorial Synthesis: a Computational Approach for Lead Identification and Lead Optimization. *J. Am. Chem. Soc.* **1996**, *118*, 1669−1676.

(31) Stanton, R. V.; Mount, J.; Miller, J. L. Combinatorial Library Design: Maximizing Model Fitting Compounds with Matrix Synthesis Constraints. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 701−705.

(32) Terrett, N. K.; Gardner, M.; Gordon, D. W. Combinatorial Synthesis − The Design of Compound Libraries and their Application to Drug Discovery. *Tetrahedron* **1995**, *51*, 8135−8173.

(33) The Global Biodiversity Institute/International Institute of Tropical Agriculture. *Training Course on Biodiversity, Biotechnology, and Law*, Ibadan, Nigeria, 1−24 March 2000, *http://www.aaas.org/international/ssa/gbdi/*.

(34) Venkatasubramanian, V.; Chan, K.; Caruthers, J. M. Evolutionary Design of Molecules with Desired Properties Using the Genetic Algorithm. *J. Chem. Inf. Comput. Sci.* **1995**, *35(2)*, 188−195.

(35) Venton, D. L.; Woodbury, C. P. Screening Combinatorial Libraries. *Chem. Intell. Lab. Sys.* **1999**, *48(2)*, 131−150.

(36) Weber, L.; Wallbaum, S.; Broger, C.; Gubernator, K.; Optimization of the Biological Activity of Combinatorial Compound Libraries by a Genetic Algorithm. *Angew. Chem., Int. Ed. Engl.* **1995**, *34*, 2280−2282.

(37) Zheng, W.; Hung, S. T.; Saunders: J. T.; Seibel, G. L. Piccolo: A Tool for Combinatorial Library Design via Multicriterion Optimization. *Pac. Symp. Biocomput.* **2000**, *5*, 588−599.

CI049787T