Virtual screening – an overview

W. Patrick Walters, Matthew T. Stahl and Mark A. Murcko

Recent advances in combinatorial chemistry and high-throughput screening have made it possible for chemists to synthesize large numbers of compounds. However, this is still a small percentage of the total number that could be synthesized. Virtual screening encompasses a variety of computational techniques that allow chemists to reduce a huge virtual library to a more manageable size. This review presents the current state of the art in virtual screening and discusses approaches that will allow the evaluation of larger numbers of compounds.

Medicinal chemists have always struggled with the difficult problem of deciding which compounds to synthesize. At every stage in the design process, from the discovery of the initial lead compound to the selection of a development candidate, a chemist has to choose from among thousands or millions of possible molecules. Historically, the number of compounds that could be synthesized by one chemist in a year has been small – perhaps a few hundred, but generally fewer. However, new technologies such as combinatorial chemistry and high-throughput screening (HTS) offer a much broader range of possibilities, and to contemplate the synthesis of millions or possibly billions of compounds is not unreasonable[1–4]. This new technology requires chemists to confront an unimaginably large and diverse 'chemical landscape'.

There are perhaps millions of chemical 'libraries' that a trained chemist could reasonably hope to synthesize. Each library can, in principle, contain a huge number of compounds – easily billions. Combinatorial chemists have already demonstrated, in several prototype systems, that libraries containing 1,000–100,000 compounds can in fact be assembled. Figures 1–3 give simple examples that have appeared in the literature recently. In Figure 1, the 1,4-benzodiazepine scaffold is shown, along with the components from which this scaffold might be assembled[5]. In the same fashion, Figure 2 shows a pyrrolidine library[6] and Figure 3 shows an acylpiperidine library[7]. The 'building blocks' are in many cases very simple and can readily be purchased or synthesized. In each of these three examples, it can easily be imagined that the library could contain 10^9 or more possible compounds. A reasonable conclusion from the preceding analysis is that a 'virtual chemistry space' exists that contains perhaps 10^{100} possible molecules (Box 1).

Combinatorial chemistry and HTS are advancing rapidly – 10^3–10^4 compounds per chemist-year is currently considered state-of-the-art (for certain classes of compounds or, in situations where large mixtures of compounds are synthesized together, this number can be higher). Of course, many chemistries are still not amenable to rapid synthesis. Furthermore, screening a million compounds (and doing all the necessary follow-up) is still a considerable effort. Both HTS and combinatorial chemistry labs are confronted with the need to miniaturize and automate as a way to control costs, save time, reduce the volume of waste materials, etc.

Therefore the critical and age-old question remains: how should a chemist decide what to synthesize? Put another way, how should a chemist filter the enormous virtual chemistry space? For the computational chemist, a laudable goal is to develop some kind of computer program capable of automatically evaluating very large libraries of compounds. This process is a natural extension of what molecular modeling scientists currently do, albeit on fewer compounds and in a less automated way. This process is sometimes called 'virtual screening' (VS). In this review the current state of the art in VS is discussed together with approaches that will allow us to evaluate much larger numbers of compounds.

W. Patrick Walters, **Matthew T. Stahl** and **Mark A. Murcko***, Vertex Pharmaceuticals, 130 Waverly Street, Cambridge, MA 02139-4242, USA. *tel: +1 617 577 6000, fax: +1 617 577 6680, e-mail: murcko@vpharm.com

  1359-6446/98/$19.00. PII: S1359-6446(97)01163-X

Virtual vs traditional screening

It is worth taking a moment to mention the relationship between VS carried out by computational methods, and the 'traditional' process of drug discovery. There is no dichotomy between HTS and VS. Information drives drug discovery – the more of it, the sooner, the better. It follows that HTS and medicinal chemistry should start as quickly as possible once a new target is identified. There is no reason to wait for structural information or computational analysis – indeed, in most cases there is every reason not to wait. For example, assay development and screening should be undertaken immediately by the HTS group, and chemists should immediately follow up on any screening leads or other sources of initial information (e.g. SAR on peptide substrates).

To clarify the relationship between VS, HTS, structural information and other components of the drug discovery process, a typical project timeline is shown in Figure 4. VS works best in an information-rich environment. However, early in a project there may not be much information available. The ability of computational chemists to contribute to a project increases as more information becomes available. The onus is on the computational group to provide suggestions and insights in a timely fashion, using whatever information and tools are at hand. For example, 2D QSAR and 2D similarity methods can be applied almost immediately, while derivation of a 3D pharmacophore will generally take longer. Ideally, X-ray crystallography and/or NMR will be used to determine the 3D structure of the macromolecular target. Once structural information becomes available, it can be used to derive new lead classes and 'fine tune' the leads that the chemists have already been pursuing. Obviously, in order to successfully apply protein structural information, it must be obtained rapidly, and preferably within a year of selecting a target.

Also, because of the many uncertainties in drug discovery and the remarkable abilities that medicinal chemists have to 'feel their way' towards solving drug design problems, it is essential for chemists to be fully involved in designing and

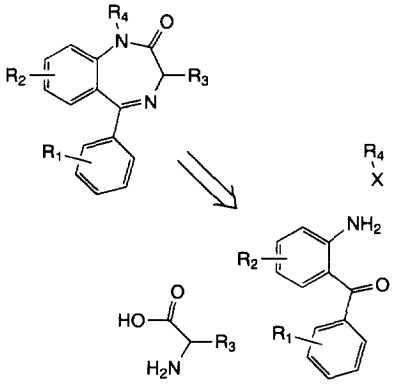

Figure 1. *Generic 1,4-benzodiazepine showing locations of four variable R groups and the components from which this combinatorial library might be constructed.*

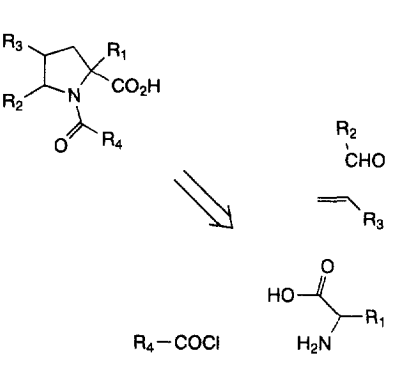

Figure 2. *Generic pyrrolidine showing locations of four variable R groups and the components from which this combinatorial library might be constructed.*

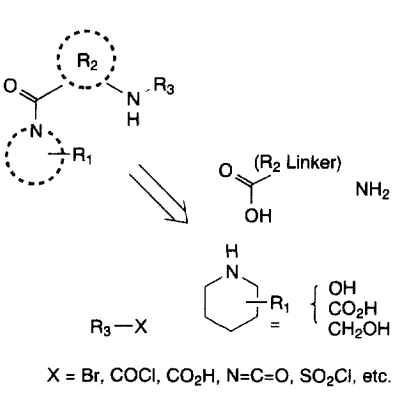

Figure 3. *Generic acylpiperidine showing locations of three variable R groups and the components from which this combinatorial library might be constructed.*

Box 1. A googol of molecules

Chemists have long been comfortable with very large numbers, beginning of course with Avogadro's Number, 6.023×10^{23} mol^{-1}, but the term 'googol' has not generally been found in chemistry textbooks. Coined in 1938 by Milton Sirotta, the nine-year-old nephew of the American mathematician Edward Kasner (1878–1955), 'googol' stands for 10 raised to the 100th power. 'Googolplex', or 10 raised to the googol, is the number 1 followed by a googol of zeros, which is still a little large even for virtual chemistry. For more insight into the googol, see Kasner, E. and Newman, J. (1940) *Mathematics and the Imagination*, Penguin Books.

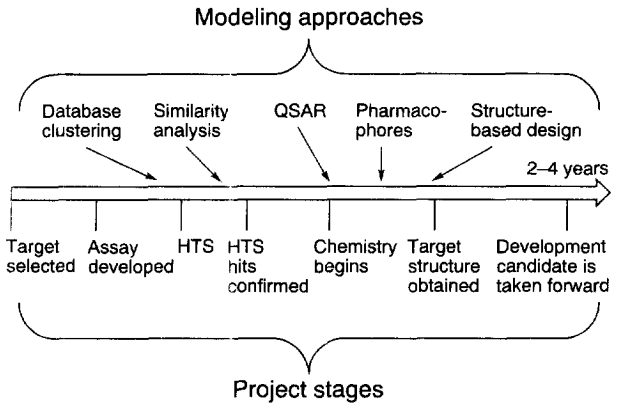

Figure 4. *Pharmaceutical project timeline. The stages of the project are shown below the arrow. Contributions which may be made by the computational chemistry group are shown above. High-throughput screening (HTS) should be carried out as soon as a target is selected. As additional information becomes available (e.g. X-ray structure of the target) it can be applied to the compound selection process.*

evaluating any virtual library. The entire process of VS does not replace the need for judgment and instinct by trained chemists. Rather, the goal of VS is to streamline the process that chemists and modelers already go through – that is, to use structural and computational information to help the chemists make their decisions.

Limiting the search space: basic assumptions

Restricting the search space is reasonable given our obvious inability to conceive of, let alone properly treat, 10^{100} molecules. There are two basic assumptions that can be used to help limit or restrict the search space.

> Assumption #1: The concept of 'maximally diverse' libraries is not sensible. Focused, synthesizable, drug-like libraries are needed.

If there are 10^{100} molecules in our virtual chemistry space but we can only synthesize 10^6 compounds, we have essentially no chance to 'cover diversity space' uniformly. It is foolish even to try. Instead, attention should be limited to libraries that are more practical:

- Focused on specific problems or classes of problems – for example, a hydroxamate library of matrix metalloprotease (MMP) inhibitors.
- Synthesizable, with reactions that will work in high yield with reagents that can be bought or prepared cheaply.
- Enriched with molecules that have the properties normally associated with drugs – for example, not too large and not too lipophilic.

Attempts must therefore be made to limit the size of the virtual library, but how large is a 'practical' library? There are perhaps 10,000 common and synthetically reasonable scaffolds in the chemical literature (there are many more known scaffolds, but not all are so easy to make). For comparison, there are 2,500 different scaffolds found in known drugs[8]. When all known scaffolds are examined, whether present in drugs or not, it is found that side-chains are attached at an average of three positions. Finally, there are around 1,000 different side-chains in known drugs. By multiplying these numbers we get:

$$10{,}000 \text{ scaffolds} \times (1{,}000 \text{ side-chain groups})^3 = 10^{13} \text{ compounds}$$

This number, while still huge, is comprehensible. Although the assumptions made here are reasonable and have greatly limited the search space, if the parameters are changed the effect can be dramatic. For example, if 100,000 scaffolds are allowed with 10,000 different side-chains used at four different positions, the total number of compounds is 10^{21}.

Many of the molecules in the virtual library are impractical or undesirable for a variety of reasons. First, certain combinations of functional groups are not synthetically compatible. Second, some molecules cannot be readily synthesized with current technology (but this can be a dangerous argument). Third, it is very rare to find certain combinations of functional groups in drugs or drug-like molecules, and it is reasonable to eliminate compounds that contain such combinations. Figure 5 shows, in schematic form, some of the different reasons for eliminating compounds.

Hence, a practical virtual library might comprise, say, 10^{15} compounds. This is far smaller than 10^{100}, but it still represents a huge problem that is not yet tractable with current technology. Nevertheless, it is at least tractable in principle.

One way to further simplify the problem can be gleaned from watching chemists design experiments. A good medicinal chemist does not need to make every compound in a series – methyl, ethyl, propyl, isopropyl, *n*-butyl and so on. In other words, the chemist does not fully enumerate a chemical series. Rather, the chemist's goal is to obtain the maximum amount of information from each compound,

regardless of the size of the library being synthesized, and this should be the goal of the computational chemist as well.

> Assumption #2: Because the search space is so huge, the trap of fully enumerating the virtual library must be avoided.

As chemists get closer to development candidates, they may begin to fine tune their structures and make smaller changes. But even in such cases, the chemist does not make every conceivable compound within a series. By analogy, a computer program might be expected to make subtle changes to interesting lead compounds that are discovered in a first pass treatment of a virtual library.

There are several ways to avoid full enumeration. Examples include evolutionary methods and similarity clustering, which will be discussed below.

Other ways to limit the search space

The search strategy, in practice, will depend on the task at hand. This will vary from case to case and can be influenced by several situational variables.

Situational variables

Lead hunting or lead optimization? Looking for leads is a very different excercise than optimizing a compound class. This changes the kinds of molecules that must be considered in the virtual library.

What kind of library to construct? Typically, there are three kinds of libraries: focused, targeted and general. Focused libraries are aimed at a family of related targets – for example, a library designed against chymotrypsin-fold serine protease inhibitors. Targeted libraries are aimed at a single therapeutic target such as HIV-1 protease. General libraries are, as the name implies, designed to be of broad interest for HTS against any target.

How rapid are the assays? Enzyme, cellular and pharmacokinetic (PK) assays will all be essential at various points in the discovery process. The throughput of these assays helps determine how rapidly new compounds should be synthesized. For example, when nearing the selection of a development candidate, PK properties will probably be focused on. If the assays used to determine clearance half-life can only handle 10 compounds per week, perhaps only 100 compounds per week need to be designed and synthesized (typically, at least 90% of the compounds will not have suitable *in vitro* potency or cellular activity and will not need to be carried forward into PK assays). During the early stages of a project, when *in vitro* enzyme potency is the main concern, the assays may be able to generate 100,000 data points per week, so much larger numbers of compounds will want to be considered (and synthesized).

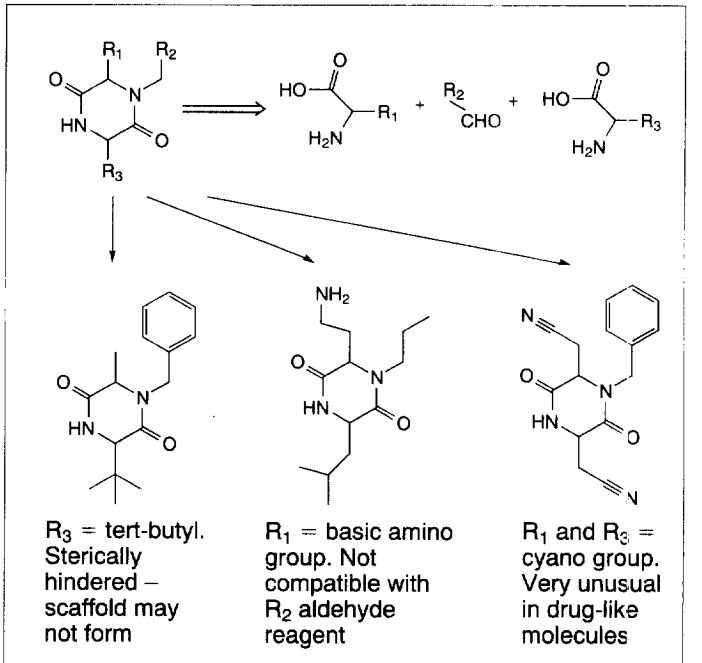

Figure 5. *Generic piperazinedione library and three specific molecules within the library that may be rejected for various reasons.*

How much time is available? For example, does management expect a result in one month or one year?

How many chemists are working on the project? With few chemists, even when the chemistry is reasonably straightforward, the number of distinct lead classes that can be effectively pursued will be quite small.

How high-throughput is the chemistry? In cases where the total number of compounds being made is high, it may be possible to synthesize compounds that are more novel, unusual or 'daring'. By contrast, when compound production is low, a more conservative strategy may be adopted.

Where is the compound headed? There may be many different goals for a particular compound, both in the short term and in the long term. For example, is a novel lead class required? Is a compound needed to prove a biological

hypothesis about a new mechanism of action (i.e. a pharmacological probe)? Or, is an attempt to select a true development candidate being made?

How close to achieving the goal? Does an existing compound need minor or major changes? Is a radically different compound class required, perhaps as a second-generation lead?

Other information

There are many different classes of information that can play a vital role in determining the search strategy. Such information can be broken down into several categories (Box 2).

- Knowledge about compounds that interact with the target (and close homologs).
- Structure–function information about the receptor of interest, including X-ray and NMR structures, homology models, dynamic motion, calorimetric data and effects of point mutations.
- General knowledge about drugs and drug-like molecules.

Applying information to direct virtual screening

Given the preceding discussion, it is clear that there is a great deal of information that can be used to focus the VS effort. The goal is to encode as much of this information as possible in a computer program to evaluate large numbers of compounds automatically. This program can then be used by the chemists to help them make the difficult choices of compound selection.

It is clear that the ability to evaluate large numbers of compounds computationally with a high degree of accuracy will be greatest in those cases where the most information is available. For example, more success is likely in situations where structural information about the protein target exists (without high-resolution structural information it is possible to apply pharmacophore or homology models, but the accuracy of such models and the ability to use them to design completely novel chemical classes is far less). The ways that macromolecular structural information may be applied to the problem of screening a virtual library, as well as alternative strategies to follow in the absence of such information, is described in a later section.

A typical VS 'flowchart' is shown in Figure 6. Note that it is applicable to any stage in a project, from follow-up of an initial screening hit to fine tuning the *in vivo* properties of a development candidate. Bear in mind that the more information that is available, the more efficiently the VS process can be.

Box 2. Information that helps direct the strategy for searching a virtual library

Knowledge about compounds that interact with this receptor
- Substrates
- Other known classes of inhibitors, antagonists or agonists
- SAR within various series
- Pharmacophores deduced from compound classes
- Patented compounds

Knowledge about receptor structure and receptor–ligand interactions
- Homology model(s)
- X-ray and/or NMR structures
- Thermodynamics of ligand binding (specific compounds)
- Effect of point mutations
- Dynamic motions of receptor and ligands

Knowledge about drugs in general
- Chemical structures and properties of known drugs
- Medicinal chemistry 'instinct'
- Rules of conformational analysis
- Thermodynamics of receptor–ligand interactions

Current practices in virtual screening

Commercial vs 'home-grown' software

There are several commercial software vendors developing tools for pharmaceutical drug design applications. The commercial software is widely used around the world and it is proper to acknowledge that computational chemists have found this software to be valuable in a wide range of applications. However, it is also fair to say that, in general, commercial modeling software suffers from certain limitations, especially with respect to processing virtual libraries. The software tends to be unstable, slow and poor at 'batch processing' large numbers of molecules. Also, the commercial vendors have not yet determined effective ways to incorporate structural information into the design of compound libraries.

In addition to limitations in software engineering, it must be acknowledged that an understanding of many aspects of drug design is still sadly lacking – for example, the scoring functions for predicting ligand-binding free energy are still relatively primitive[9,10]. The commercial software vendors do not generally have the luxury of tackling these extremely challenging and fundamental scientific problems. For all these reasons, many pharmaceutical companies are developing proprietary software and methods.

How fast is fast?

Before starting to compare VS methods it is reasonable to ask how many compounds need to be processed in one month? If one molecule per minute is processed, then ~44,000 molecules can be processed per month. If 32 processors are assigned to the problem, then ~1.4 million compounds per month can be screened. Obviously, much larger libraries than this need to be handled. Given these speed limitations, VS has generally been used to study a single scaffold or lead class. This of course is useful, but results in a very narrow search; all the ideas examined are closely related to the initial compounds.

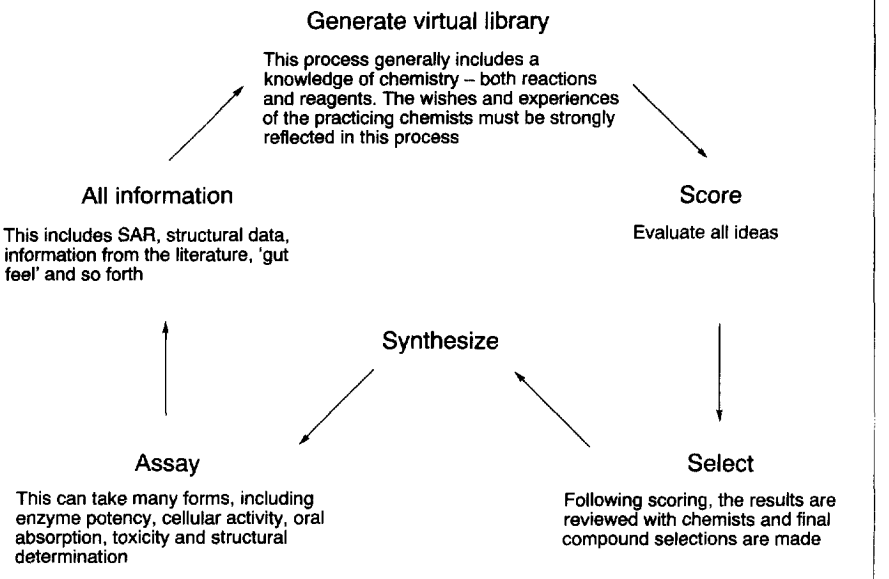

Figure 6. *Schematic overview of the virtual screening process.*

2D similarity

There are two straightforward approaches that can greatly enhance the speed of processing of a virtual library. The first involves the use of 2D-similarity methods. A wide variety of descriptors can be calculated from a 2D representation of a molecule (molecular graph)[11–14]. Most 2D descriptors can be calculated rapidly, allowing hundreds of thousands of structures to be processed in an hour. The rapid growth of combinatorial chemistry and the need to process large numbers of compounds has brought about renewed interest in these methods. Brown and Martin published an analysis of several 2D and 3D descriptors and found the 2D descriptors superior to 3D for clustering compounds into groups displaying similar physical properties and biological activities[15,16]. Because the 2D methods are so rapid, they are often used to select compounds from a virtual library that are similar to an existing lead. This is depicted in Figure 7. The same methods can be used to locate similar compounds from a screening database.

Clustering or 'pooling' reagents

The second approach to speed up processing of a virtual library involves clustering or 'pooling' reagents based on similarity[17]. Recall the second basic assumption that full enumeration of the virtual library is unnecessary. Rather, the use of evolutionary methods or clustering to greatly reduce the number of compounds in the virtual library is desired. Figure 8 shows how reagent pooling might work. Suppose that 1,000 building blocks could be clustered, based on their similarity, down to 50 families. Then only one representative member of each family in the library needs to be evaluated. Thus, if there are three side-chain positions, instead of $1{,}000^3 = 10^9$ compounds, there are now only $50^3 = 125{,}000$ compounds. Using 32 processors, at one molecule per minute per processor, a library of 125,000 compounds could be examined in three days, or 12 libraries in one month. If attention is limited to 36 scaffold classes – perhaps based on the in-house experiences and instincts of the chemists – it would take three months to evaluate all these libraries.

Evolutionary methods

Evolutionary algorithms are probabilistic search techniques based on the principles of biological evolution[18]. In an evolutionary algorithm, possible solutions are encoded in a 'chromosome-like' data structure. A group of (typically random) chromosomes that make up a population of solutions is allowed to 'evolve', thereby producing a superior set of solutions. Some evolutionary methods, including genetic algorithms (GAs), evolutionary programming and evolutionary strategies, have been used in a variety of drug design applications. A brief overview of GAs and their application to the design of combinatorial libraries is provided below, but the interested reader is encouraged to consult a number of excellent reviews[19–21].

Figure 9 provides a schematic description of a GA. The first step in the GA cycle is the generation of the initial

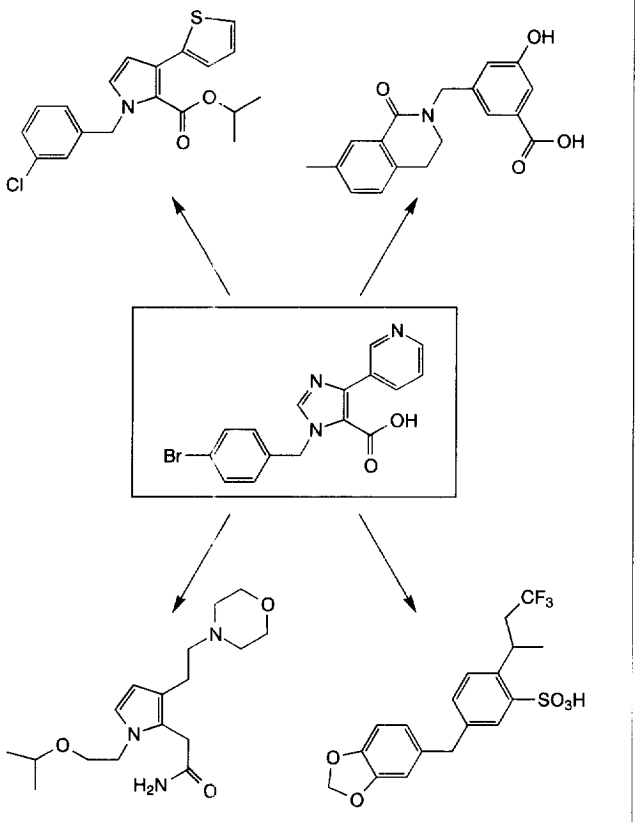

Figure 7. *Identifying similar compounds using simple 2D methods.*

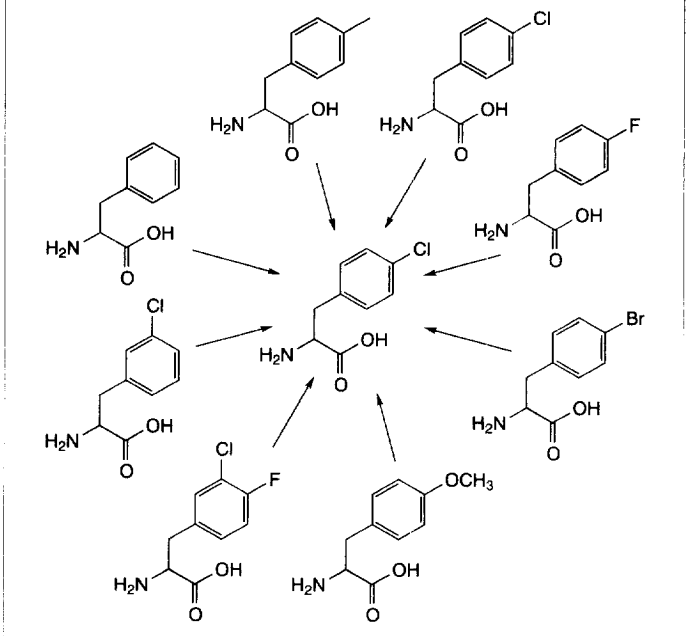

Figure 8. *Reagent pooling.*

population. Once the population is generated, the fitness of each chromosome is evaluated. The most 'fit' members of the population are then chosen to produce the next generation. This is known as the selection phase. In the crossover phase, two of the selected parents are paired and genetic material is exchanged. To avoid being trapped in local minima, a small fraction of the population undergoes point mutations, which effectively increase the gene pool. The new population created by crossover and mutation then completely replaces the current population and the cycle is repeated for a predetermined number of generations.

Figure 10 illustrates the use of a GA to design a tripeptide library. The chromosome in this GA consists of 3 genes, each of which represents one of the amino acids in the tripeptide. The integers inside the gene represent the individual amino acids (1 = ala, 2 = gly, and so on). If the library is limited to the 20 naturally occurring amino acids then there are 20^3 = 8,000 possible combinations.

The algorithm begins with a random population of tripeptides, each of which is assigned a fitness value based on some metric. This fitness could be based on similarity to a known inhibitor, the ability to fulfill a set of pharmacophoric constraints or some other criteria. The members of the population with the highest fitness are then recombined as shown in Figure 10. Mutation can be introduced by changing a residue in one of the tripeptides (e.g. Ala to Phe). The process of selection, crossover and mutation is repeated until a stopping condition is met. This stopping condition may be based on convergence of the population, a set number of generations or several other factors.

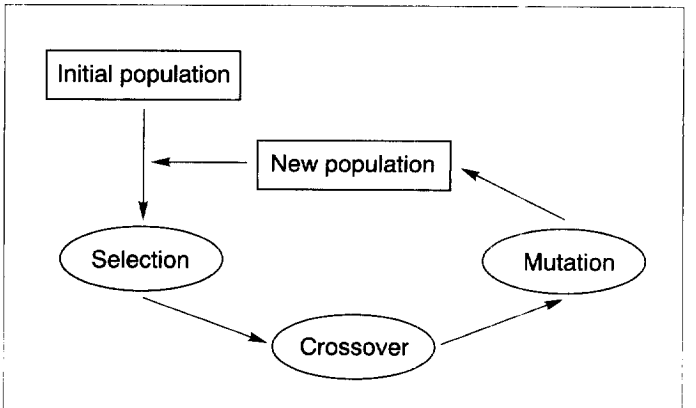

Figure 9. *Flowchart of a genetic algorithm.*

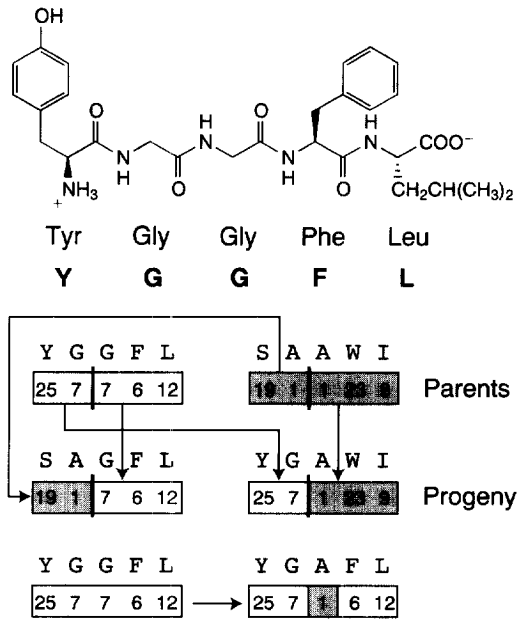

Figure 10. *Genetic algorithm chromosome structure used to evaluate a tripeptide combinatorial library.*

Evolutionary methods have been used in a wide variety of combinatorial chemistry applications. Sheridan and Kearsley have used a GA to suggest a library of peptoids that are similar to a peptide target[22]. Cho has utilized a similar approach to design a library of pentapeptides that were consistent with an existing QSAR model[23]. Genetic algorithms have also been applied in the laboratory. Weber and coworkers at Roche (Basel, Switzerland) utilized a GA to optimize the reagents used in a four-component Ugi reaction to produce thrombin inhibitors[24]. The population in this experiment consisted of 10 isocyanides, 40 aldehydes, 10 amines and 40 carboxylic acids, thus a full enumeration of the library would have required the synthesis of 160,000 compounds. After carrying out 18 generations with a population size of 20 compounds in each generation they were able to produce a 0.28 μm inhibitor. Singh and coworkers at Sterling (Collegeville, PA, USA) carried out similar work using a GA to optimize a library of hexapeptides that acted as substrates for stromelysin[25].

The program CONJURE (Vertex Pharmaceuticals) uses a GA that employs structural information in its scoring function. For each molecule in the population an ensemble of low-energy conformations is produced in the context of the enzyme active site. These conformations are scored using one of several empirical scoring functions and the estimated binding energy of the best conformation is assigned as the fitness of the molecule. Figure 11 shows the reagent population used by CONJURE in the design of a library of thrombin inhibitors. Full enumeration of this library would require the evaluation of more than 2 billion products, but rather than fully enumerating the library, CONJURE uses its GA to select a focused set of compounds that are complementary to the active site. In this case, CONJURE was able to identify a known thrombin inhibitor in less than two hours[26].

Evolutionary methods provide one means of focusing a chemistry effort on a smaller number of compounds. While these methods are rapid and easy to implement there are caveats. The major drawback to evolutionary methods is that they are non-deterministic; successive runs may not necessarily produce the same answer. In addition there is no guarantee that the solution found by an evolutionary method will be the best solution. It has become common practice to carry out a series of runs of an evolutionary program with a variety of starting points.

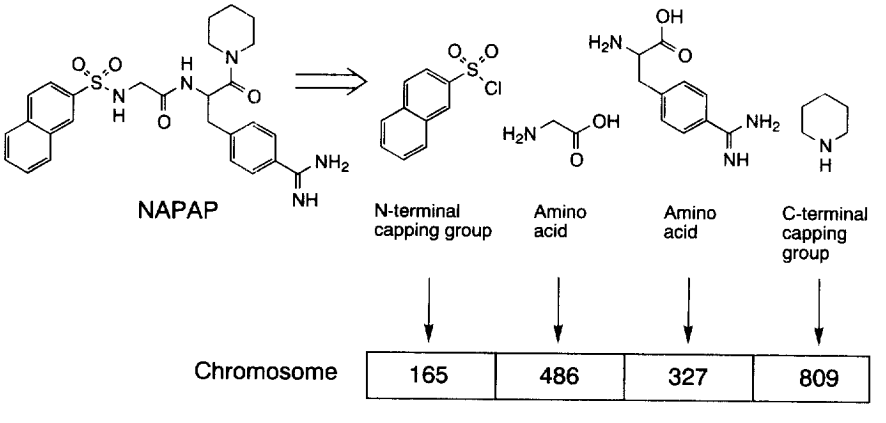

Figure 11. *Each building block used to generate NAPAP is treated as an independent unit. By a combination of mutation and crossover, novel compounds can be generated rapidly.*

Multi-conformer databases

There is one other 'special case' that should be mentioned: the pre-calculated multi-conformer database. A sizable fraction of the CPU time in evaluating a 3D virtual library can be taken up with the generation of conformers. However, with access to a small in-house sample collection database of say 100,000 compounds, if the study is limited to the compounds in this library, then a multi-conformer database with perhaps 10–50 conformers per molecule could be calculated. As the database is static, the conformers need only be generated once, stored and accessed when needed.

Prospects for further improvements in virtual screening

The effect of increasing processing speed

How fast can VS be performed? At one molecule per second per processor approximately 10^8 compounds per month can be evaluated on 32 processors. Recall that there are 10^{15} molecules to examine in the practical virtual library. Also, at any given time there are likely to be ten or more targets of interest within an organization and ideally multiple conformations (say, 10–100) for each target would be considered. So it is clear that even a rate of 10^8 per month is far too slow! Moreover, a speed of one molecule per second is not currently achievable, even with highly approximate methods.

Will faster hardware solve the problem?

Another trap that should be avoided is the belief that 'hardware will save us'. While it is undoubtedly true that hardware will continue to come down in price, the problem cannot be solved simply through improvements in price/performance. In the near future, it may be the case that processors are an order of magnitude faster and ten times as many processors could be dedicated to a given problem. This would increase the throughput to 10^{10} per month, still many orders of magnitude shy of what is needed. Additional improvements in speed will have to come from improvements in algorithms as well as more intelligent treatment of the problems.

> Conclusion #1: To increase the speed of processing a virtual library it is essential to add more intelligence (i.e. filtering) to the process.

The need for filters

Figure 12 shows the previous VS flowchart (Figure 6) with filters added in different locations. Filters can take many forms, for example:

- Does the molecule look like it can be synthesized?
- Does the molecule resemble compounds found in issued patents?
- Does the molecule appear 'drug-like'?
- Does the molecule have the right general 'shape' and 'properties' to bind to the receptor of interest?
- Does the predicted bound conformation of the molecule form key interactions that are known to be valuable from prior experience?

Filters allow many more compounds to be processed because the vast majority of compounds can be dismissed very quickly by consideration of the 2D molecular graph. This avoids the slower steps of generating 3D conformations, docking and scoring.

Detailed analysis of virtual screening: 2D information

In this section each step in the VS process will be described in more detail (Figure 12).

Molecular construction ('synthesis')

The initial problem of VS analysis is construction or 'synthesis' of molecules on the computer. In general, there are two choices: either the kinds of compounds that are constructed can be limited or retrosynthetic analysis can be performed on a larger number of compounds to test their practicality.

Working in the forward direction – limiting the compounds constructed – seems to be the more straightforward approach to implement and provides a natural fit with current thinking about the assembly of real combinatorial libraries. Construction of a library on the computer uses a set of 'allowed' building blocks and reactions, such as is given in Figure 13. A limit can be placed on the number of building blocks that are allowed in any molecule – for example, no more than four building blocks assembled from three reactions. The library of building blocks can also be limited to those that are either commercially available or readily and cheaply synthesized. Ideally, such molecule-building programs should apply detailed knowledge of cross-reactivity (i.e. functional groups that are not compatible) to help eliminate molecules that could not be readily synthesized in practice. These programs, such as CONJURE[26] are capable of processing hundreds of thousands of compounds per hour on a single workstation.

The other approach to this problem is to perform an unrestricted generation of a large number of compounds and have a computer-aided organic synthesis (CAOS) program

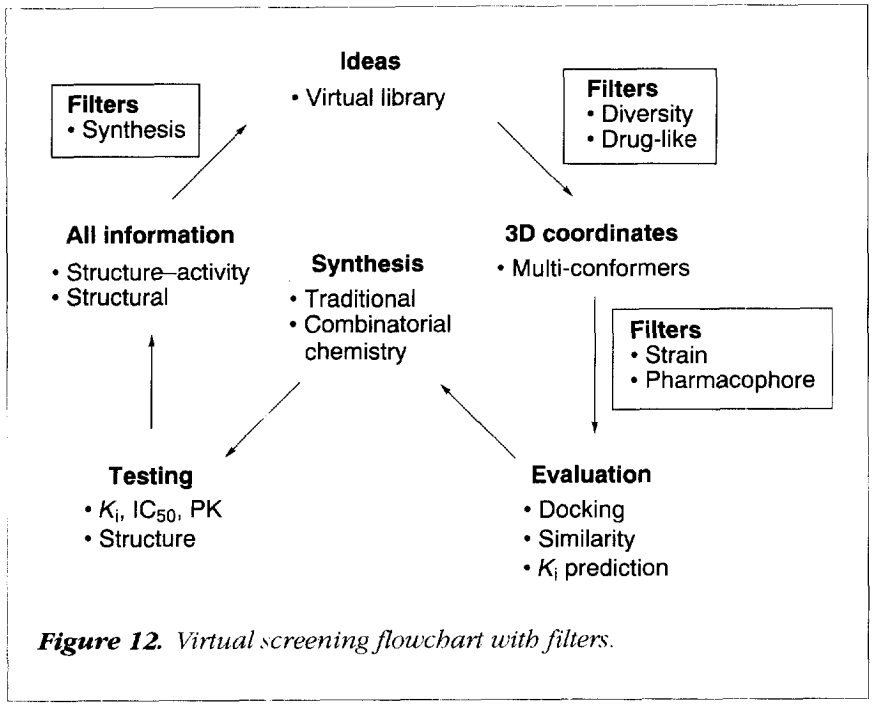

Figure 12. *Virtual screening flowchart with filters.*

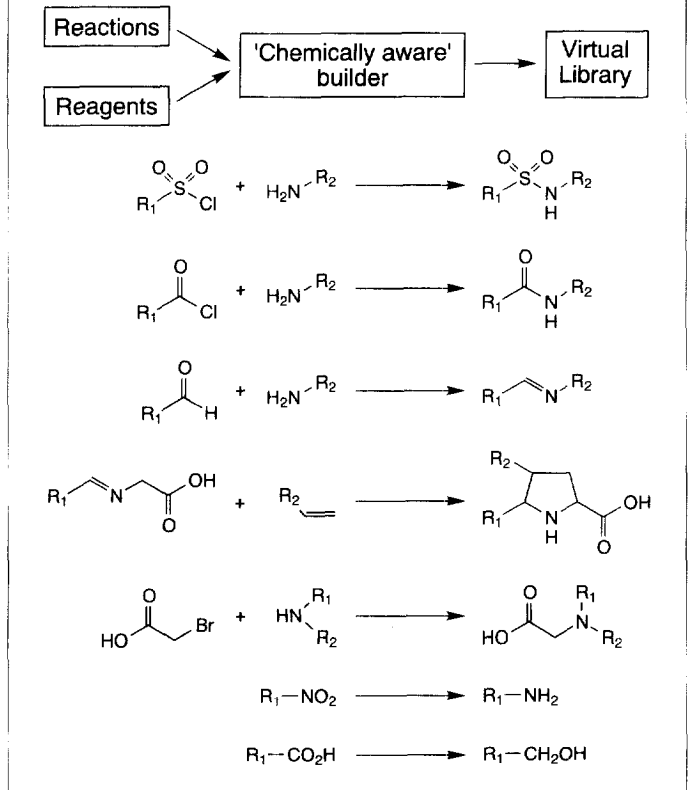

Figure 13. *Examples of some 'elementary' reactions that a synthesis module might use.*

determine which compounds could be readily synthesized. These programs are generally more complex and slow, but in principle should have a richer set of rules to guide analysis. An entire issue of the journal *Recueil des Travaux Chimiques des Pays-Bas* was dedicated to the field of CAOS and provides an excellent review of the topic[27]. Johnson and coworkers have adapted the CAOS paradigm into the high-throughput world with a program called CAESA[28] (Computer-Aided Estimation of Synthetic Accessibility) that is designed to rank the synthesizability of a series of candidate molecules generated by a *de novo* design program[29,30]. CAESA uses a library of generalized synthetic transformations in conjunction with an analysis of features such as stereocenters to determine which molecules can be easily synthesized. Although this approach appears promising, the field of CAOS is extremely complex; despite tremendous improvements having been made over the past 30 years, synthesis programs still cannot reproduce the intuition of an experienced organic chemist.

Application of similarity and diversity

Similarity and diversity analysis provides another way to filter a virtual library. As discussed above, the size of a library can be reduced by selecting a subset of compounds that are similar to a lead. Similarity can be assessed using topological descriptors[11,31], putative pharmacophores[32,33], molecular fields[34–36] or several other methods. Often in the early stages of a project there is no lead molecule. In these cases it has become common practice to utilize computational methods to select a diverse set of compounds from a large virtual library. If a compound from the diverse set exhibits activity, then other similar compounds from the library are synthesized and tested.

As mentioned earlier, with the practice of 'pooling reagents' it is possible to effect a dramatic reduction in the number of compounds to be considered. However, one potential drawback to this strategy is that reagent diversity is not necessarily equivalent to product diversity. Gillet and coworkers have demonstrated that a diverse set of reagents does not lead to the most diverse set of products[37]. In an analysis of three published combinatorial libraries they used dissimilarity-based compound selection (DBCS)[38] to

generate a maximally diverse set of products from a fully elaborated library. The diversity of this set of compounds was then compared with the diversity of a set of compounds generated from a maximally diverse set of reagents. In each case the library generated from the maximally diverse set of reagents was noticeably less diverse than the library selected by DBCS. The authors go on to point out that the synthesis of the set of compounds selected by DBCS can require large numbers of reagents and may not be practical for combinatorial synthesis. They then present a GA-based method that attempts to maximize product diversity while utilizing a practical number of reagents. Libraries selected by this method appear to be significantly more diverse than those selected using reagent-based techniques.

As illustrated in Figure 14, another approach is to select a diverse set of scaffolds as opposed to picking diverse analogs within a single series. Diverse scaffolds can be chosen on the basis of 2D or 3D similarity metrics. In addition, several groups have also developed methods for evaluating the diversity of scaffolds based on the spatial placement of side-chains[39,40].

One drawback with all similarity methods is that in the laboratory, SAR are often found to be 'tight' – that is, tiny changes in chemical structure can make the difference between success and failure. A common concern of using similarity to pare down a database is the risk of missing a lot of potential candidates. This risk is an unavoidable fact of database reduction, but given the huge numbers of compounds that have to be analyzed, it seems a reasonable method to ensure that the greatest range of compounds is sampled. Of course, any molecules that score well in the VS exercise can be followed up by examining close neighbors – in effect, by 'filling in the holes' in regions of chemical space that appear the most promising.

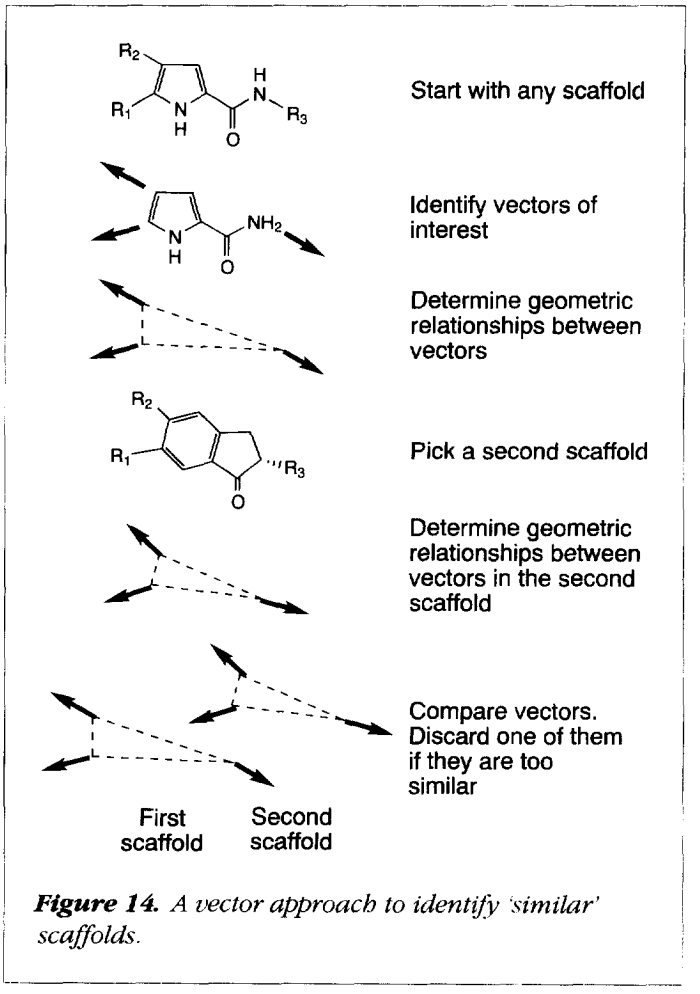

Figure 14. *A vector approach to identify 'similar' scaffolds.*

Filters on chemical structure: the REOS concept

As molecules are constructed, a variety of filters can be applied to 'weed out' compounds that do not meet certain criteria. These criteria can include:

- The presence of certain functional groups that are not desirable, such as reactive moieties and known toxicophores. A representative (partial) list of such undesirable groups is shown in Figure 15.
- The overall 'feel' of the molecule – how big it is. How lipophilic it is. How many rotatable bonds there are. – A molecule with a molecular weight of 900 and 17 rotatable bonds is probably not going to excite any chemists.
- Does the molecule appear to be drug-like? It is clear that the precise 'chemical fingerprint' of a drug is not known, but there are certain themes that are repeatedly found when databases of known drugs and other molecules that have advanced into clinical trials are analyzed[8,41]. The 32 most popular scaffolds that encode half of all known drugs are shown in Figure 16.

By utilizing these criteria in library design efforts, an attempt is made to bias the database towards compounds that have a higher probability of being not just potent but 'good' leads – that is, molecules with appropriate functional groups and physical properties, and the potential to lead to compounds with good *in vivo* activity rapidly.

> Conclusion #2: Most molecules are not appropriate for consideration, so very fast methods that identify and remove them are necessary.

This concept is referred to as REOS – 'Rapid Elimination of Swill' (Figure 17). Most molecules that a VS method might

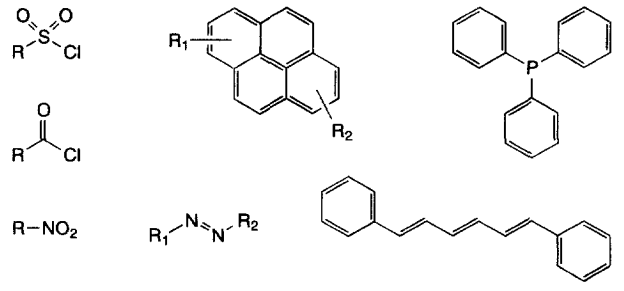

Figure 15. *Examples of undesirable functional groups that might be eliminated from a virtual library.*

construct are probably not worthy of serious consideration. The trick is to identify these molecules rapidly so that no time-consuming analysis takes place (e.g. 3D conformation generation or docking).

There are two caveats to the REOS concept. First, some researchers distinguish between 'initial hits' and 'good leads'. Initial hits might not appear drug-like but, perhaps, can be quickly converted into good leads that will have all the desirable properties of a drug. Of course, converting an imperfect lead into a clinical candidate (e.g. a pentapeptide into a low molecular weight, orally bioavailable peptidomimetic) can be quite challenging.

The second caveat is that because filters are imperfect, some molecules are necessarily missed. This is as true for REOS as it is for any other selection process, but the goal of the REOS approach is to 'enrich' the pool of compounds for synthesis so that they are more likely to contain drug-like molecules.

Detailed analysis : getting to the 3D world

Once a virtual library has been created and the undesirable compounds have been removed, the next step is to generate one or more 3D conformations for each molecule. Several commercial programs are available for converting a 2D connection table into a 3D structure[42,43]. However, few of these methods provide means for dealing with conformational flexibility. Several recent studies have shown that multi-conformation databases are more effective for docking and 3D searching[44–46] (see also The Generation and Use of Large 3D Databases in Drug Discovery; http://www.netsci.org/Science/Cheminform/feature03.html). If the VS method is to evaluate flexible molecules, then the generation of multiple conformers is essential.

The conformational search methods found in most commercial software packages can take from several minutes to

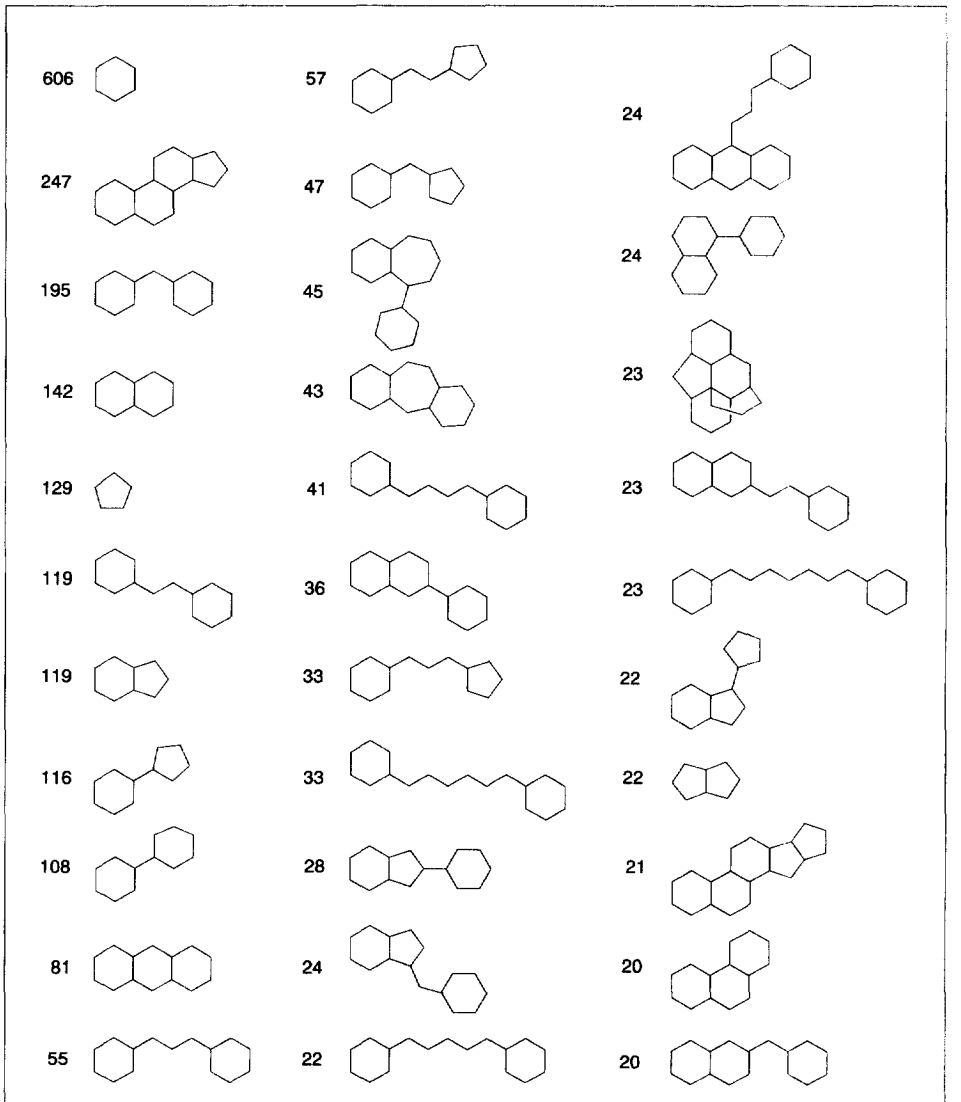

Figure 16. *The 32 frameworks most commonly found in drugs. The numbers show how many drugs have been based on each framework in a total of 2548 drugs.*

several days to produce an ensemble of conformations for a single molecule[47,48]. Obviously this is far too time consuming for even the smallest combinatorial library. One approach to overcoming the combinatorial nature of conformational search is to use a model building algorithm[49–51]. Model building (also termed fragment assembly) algorithms take advantage of known propensities of molecular fragments to adopt energetically favorable configurations. Substructural fragments can be joined together in 3D by model building programs to generate conformations. Conformations of decalin, for example, can be built up from cyclohexane conformations. By utilizing a knowledge base consisting of conformational units, the search is reduced to a problem of finding combinations of units that produce low energy conformations. Fragment assembly methods have the capability to search conformational space rapidly because the search space is limited to reasonable values of bond lengths, angles and torsions. The MONGOOSE (manuscript in preparation) program developed in our laboratory uses a library of experimentally observed torsion angles in conjunction with a fragment joining approach to produce ensembles of conformations rapidly. MONGOOSE is capable of processing ~20,000 structures per day on a single workstation. While this represents a considerable advance in speed relative to other published programs, it is still far too slow to allow analysis of the huge virtual libraries being considered here.

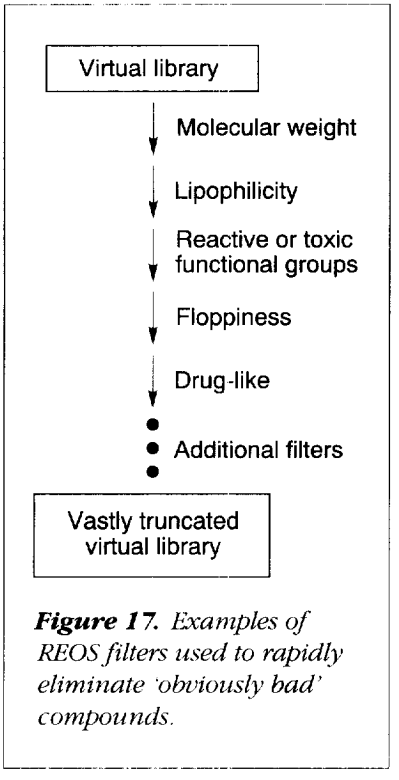

Figure 17. *Examples of REOS filters used to rapidly eliminate 'obviously bad' compounds.*

> Conclusion #3: Before going to the trouble of 3D generation, it is essential to use very rapid 2D 'shape' and 'distance' information to remove molecules that cannot possibly match the active site (Figure 18).

It is well known that a chemical graph encodes much of the 3D information within a molecule[11]. Thus, going directly from 2D information to selection of compounds for screening or synthesis can be a powerful approach. Information present in a molecular graph can also be used to determine whether a molecule can meet a set of pharmacophoric constraints. Leach and coworkers have developed a method that uses a neural network to determine the maximal and minimal possible geometric distances between two atoms in a molecular graph[52]. The molecular graph can then be rapidly analyzed to determine whether the more time-consuming process of 3D-structure generation should be performed. Topological descriptors can also give an approximation of the shape and dimensions of a ligand molecule. With respect to docking, these approximate dimensions can be compared with the dimensions of a cavity or binding site, and used to determine which ligands will be converted to 3D structures and docked.

Docking and scoring: the heart of the 3D problem

Docking

Methods for docking ligands in active sites have a long history, and the topic has been well and frequently reviewed[53,54]. Early efforts relied on docking a single conformation of each ligand[55], but it is now widely recognized that this is not an ideal practice because most molecules are quite flexible and there is no single conformer that adequately describes the set of possible bound conformations. As a consequence, more recent efforts have relied on either multi-conformation docking or flexible docking.

In multi-conformer docking, a set of conformations (typically, 10–50) is generated and then each conformer is docked as a rigid molecule into the enzyme, which is held fixed throughout. It is a good strategy for examining an in-house sample collection database to select compounds for HTS. Multi-conformer docking tends to have much better hit rates than single-conformer docking[56]. An example is shown in Figure 19. By generating, docking and scoring a set of 30 conformers of mycophenolic acid (MPA, a 7 nM inhibitor of the enzyme inosine monophosphate dehydrogenase), it was found that the lowest-energy docking conformation predicted the experimental bound X-ray conformation with an RMSD (root mean squared deviation) of 1.1 Å (Ref. 57).

Recently, it has become possible to add information about the active site to docking procedures, making them much faster and more effective. For example, with DOCK it is now possible to 'color' the site-point spheres to ensure that only functional groups with certain properties (e.g. hydrogen bond donors) will occupy those locations[58]. It is also possible

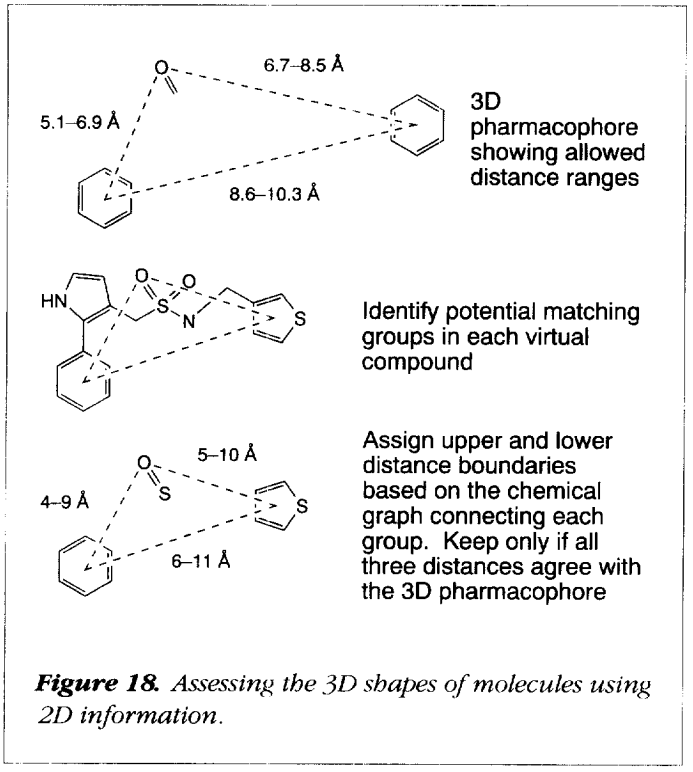

Figure 18. *Assessing the 3D shapes of molecules using 2D information.*

to run pharmacophore searches before docking to make sure that certain geometric patterns of functional groups are present. For example, each conformation can be inspected to make sure that there are two hydrogen bond donors separated by 7 Å, and only dock those conformations that satisfy this requirement. This of course begs the question 'How do we know we want to have two hydrogen bond donors separated by 7 Å?' This information can be obtained from experimental structures of protein–ligand complexes, or from the SAR of active compounds. Or, an attempt can be made to generate this information on the computer, as follows: First, a program such as GRID[59–61] or MCSS[62–64] is used to produce a list of 'site points', which are regions in the active site where prototype fragments (e.g. benzene, water or formaldehyde) are favored. There can be many copies of each fragment type – 63 benzenes, 37 formaldehydes and so forth. Second, each of the docked conformations is filtered to ensure that some minimum number of these site points have been hit.

Several docking methods have been developed that allow ligand flexibility. Some of these programs use a GA that encodes rigid body translations and rotations as well as rotations about dihedral angles in a chromosome. The programs begin with a random set of ligand orientations and 'evolve' a set of orientations that optimize a scoring function[65–68]. Other programs employ a similar strategy with another optimization algorithm in the place of the GA[69,70]. An alternative approach to flexible docking is to dock pieces of a ligand and either reconnect the pieces or use one of the pieces as an 'anchor' for performing a conformational search in the active site[71,72]. Although flexible docking shows great promise, most of the methods currently in existence require several minutes to dock a single ligand. Due to these time requirements flexible docking is typically used to evaluate small sets of compounds.

A few groups have recently developed docking programs that are specifically designed to evaluate combinatorial libraries[73–75]. These programs operate by attaching a series of side-chain conformers to a scaffold that has been pre-positioned according to computational studies or crystallographic data. After each side-chain is attached, its affinity for the receptor is evaluated using an empirical scoring function. The use of a common scaffold drastically reduces the complexity of the docking problem and allows hundreds of thousands of compounds to be evaluated in a day. Kick and coworkers have used this VS approach to discover a 73 nM inhibitor of cathepsin D. A different approach to computational screening that employs multicopy sampling to identify amino acids that bind to specific regions of a binding site has been implemented by Zheng. While this approach has the advantage of allowing the protein to be considered flexible, it is also computationally expensive. The program is only capable of evaluating 48 ligands per day, several orders of magnitude less than what is required for most library design projects.

Finally, it is important to note that the conformer generation and docking steps are generally still rate limiting in VS. Multi-conformer rigid docking, especially when enhanced with sphere coloring and pharmacophore matching, can be very quick – about one second per conformer – but the conformers must first be generated, and this can be time-consuming to do properly. Flexible docking programs avoid the need to generate conformers beforehand, but the docking process itself can be quite slow. Whichever approach is selected, evaluating even a few molecules per minute is considered quite fast. Recall too that in the majority of these methods, a single static conformer of the receptor is used.

Pharmacophores

At this point it is worth describing in more detail the ways in which pharmacophores may be used in VS. Obviously, in those cases where structural information for the target is lacking, a good pharmacophore model can provide a powerful filter[32,76]. Indeed, even in cases where structural information

is available, it may be worthwhile to apply pharmacophore filters because it greatly reduces the amount of docking required.

Pharmacophore analysis can be very fast. For example, to check a library of 1,000,000 conformers against a three-point pharmacophore model can be done on a typical workstation in a few minutes. Of course, the 3D structures of each conformation must first be generated, which can be very slow.

Another important topic is '3D similarity'. There are very exciting methods being developed that allow assessment of the 3D similarity of each conformer against a stored pharmacophore model (or, indeed, any other molecule)[77,78]. These methods can study hundreds of thousands of molecules per hour and allow attention to be concentrated on the tiny subset that overlaps well with the pharmacophore or lead molecule. The same techniques can be combined with macromolecular structural information, using site points as described earlier.

Finally, scaffolds can be classified based on the 3D diversity of their side-chain vectors. By combining the same side-chain-vector information with data about the active site, it is possible to select only those compounds that have appropriate vectors to fill the pockets of interest. This can greatly reduce the number of compounds that need to be docked.

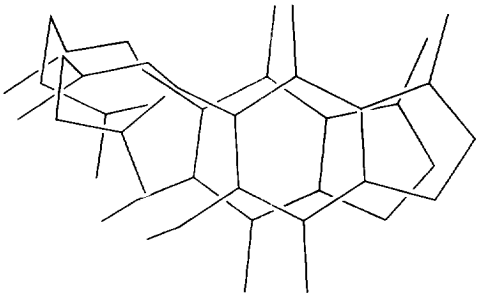

Figure 19. *A comparison of the crystal structure of mycophenolic acid in inosine monophosphate dehydrogenase with a structure produced using multi-conformer docking.*

Caveats about structural information

While there is tremendous value in structural information for helping to dock and score a virtual library, it is important to note a few caveats about using this information. These points will be worth keeping in mind in the following discussion on scoring functions.

Proteins are not rigid. This may a be a truism, but the best ways to account for dynamic motions in proteins are not obvious. In principle, there are two ways to accomplish this: a truly dynamic treatment of the system or a series of static models. The dynamic approach would be best, although extremely expensive. The static models can be considered a reasonable alternative if the number of discrete static 'snapshots' is small enough. The snapshots might be obtained from various sources: a series of molecular dynamics trajectories; a selection from several structures in an NMR ensemble; multiple X-ray structures; or by using a protein side-chain conformational library[79], a set of low-energy conformations within the active site could be generated. With the latter approach, it is not unreasonable to expect that 10–100 separate conformations may be needed; nevertheless, there may be ways to treat multiple side-chain conformations simultaneously. At present, none of these approaches is widely used because of the extra CPU time required.

Water. It is obvious that water plays a critical role in ligand binding[80–82] but, as with protein dynamic motions, it is not clear how to handle this. Various approaches are possible: using waters shown by X-ray or NMR[83]; using a continuum dielectric model[84]; employing a rule-based method for adding discrete waters as needed[85,86]; or using a surface-area-based approximate treatment such as GB/SA (generalized born surface area)[87]. Complicating matters is the fact that docking methods are generally quite good at reproducing bound X-ray structures even when waters known to be important are not included in the simulations[65,66].

Homology models. When are homology models accurate enough for docking? This is profoundly unclear at present, but with respect to selecting compounds from a virtual library, it seems likely that no homology model will have nearly the same degree of 'resolution' as an X-ray or NMR structure. Perhaps a better way to view using such a model is shown by Figure 20. By docking representative members of a wide range of compound classes against the model, it should be possible to tell which kinds of molecules have the greatest chances of success. This provides useful guidance to chemistry about which scaffold classes to pursue, but is less helpful for selecting specific compounds.

Scoring functions

Once docking has occurred, how are the hits scored? This is perhaps the most daunting problem in the entire VS process.

In principle, free-energy methods such as FEP are the most accurate developed to date[88,89]. However, such CPU-intensive

methods are not going to be useful in the future for screening billions of molecules on the computer. At present, on 32 processors it might be possible to perform FEP on ten molecules per week. Perhaps, five years from now, with a tenfold increase in CPU speed and a tenfold increase in the number of processors, it would be possible to study 4,000 molecules per month. This might be useful as a 'final check step' in a VS exercise, especially where synthesis is time-consuming.

> Conclusion #4: For the foreseeable future, there will be continued dependence on highly approximate methods.

This conclusion has a profound impact on the way a virtual library is scored. If the only way to evaluate billions of molecules is rapidly and approximately, the accuracy of our scoring functions per unit of CPU time has to be maximized. First, a distinction should be made between 'tailored' and 'general-purpose' scoring functions. When the binding affinity of a series of homologous inhibitors into a particular site is known, it is often possible to tailor a specific scoring function to fit the data. For example, Bohacek was able to predict the binding affinity of a series of thermolysin inhibitors using a simple count of the number of hydrogen bonds and the number of hydrophobic contacts[90]. Holloway was able to correlate the potency of a series of HIV protease inhibitors to the molecular mechanics interaction energy between the enzyme and the inhibitor[91]. In both cases the scoring function was then used in the design of new inhibitors. Although these scoring functions are locally powerful, the methods used are typically not transferable between systems.

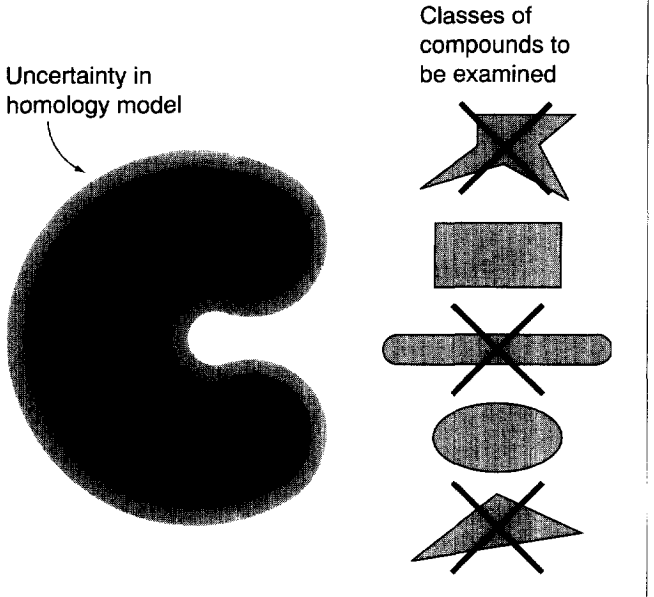

Figure 20. *A crude homology model is used as a 'first pass' structural filter to remove classes of compounds with inappropriate shapes.*

In order to evaluate libraries in the early stages of a project it is necessary to have a scoring function that is not dependent on a knowledge of the types of molecules that bind to the site. Several groups have developed general purpose scoring functions derived from careful examination of a varied series of enzyme–inhibitor complexes[92–94]. Probably the best known of these is the scoring function developed by Boehm[95,96]. It includes terms for surface area, hydrogen bonds and ionic interactions, as well as a penalty for the number of rotatable bonds. Using a database of 45 enzyme–inhibitor complexes, the Boehm function is able to predict binding affinity with an RMS (root mean squared) error of ~2 kcal.

To further improve the accuracy, additional terms for a scoring function may be considered. The trade-off is that each new term will increase the time required to evaluate each conformation or molecule. Examples include:

Solvent effects. The desolvation component of binding is clearly important. Detailed desolvation calculations are quite slow, but approximate methods based on surface area, such as atomic solvation parameter (ASP) models may be worth considering[87,97]. It also may be possible to simply penalize conformers that place lipophilic groups in contact with solvent[98,99]. Recently several groups have included desolvation terms in the scoring function utilized by docking programs[100,101]; however, comprehensive studies demonstrating the effectiveness of these terms have yet to be published.

Strain energy. If a ligand binds in a high-energy conformation the overall free-energy change that accompanies binding will be less favorable. Forcefield or molecular orbital methods may be used to estimate the conformational energy difference between the bound and free state. In highly optimized compounds, the strain energy is generally low, so K_i prediction methods that have been 'trained' using only potent inhibitors may not reflect the importance of this term. However, for evaluating a large number of non-optimized compounds, the inclusion of a strain-energy term can greatly reduce the occurrence of false hits.

Entropy. A molecule with many rotatable bonds has a huge number of conformations available to it – in other words, it

has many degrees of freedom. As a consequence, such a molecule must pay a large 'entropic penalty' upon binding. In Boehm's scoring function this is approximated by simply multiplying the number of rotatable bonds by a constant[102], which is a very fast, approximate way to treat the problem. Generating a complete set of conformations for each molecule of interest may not be tractable and it is not clear whether generating a representative subset of conformations will suffice. It may also be possible to establish a more precise set of rules based on the principles of conformational analysis.

Overlap with pre-defined site points. As discussed earlier, there may be experimental or computational data suggesting that certain kinds of functional groups belong in certain places. A check for each docked conformation can be made both to make sure that a minimum number of site points has been hit and to eliminate conformers that contain mismatches.

Themes. Often, certain structural motifs appear quite frequently in a set of high-scoring docking results. For example, many molecules even though they are structurally unrelated, may all put a hydrogen bond donor at a certain site. Or, molecules of a certain general shape might have the highest scores. The occurrence of themes can be used to either include or exclude compounds, depending on the criteria selected[103,104].

Diversity. At the end of the scoring process, there may be far too many molecules that appear to 'score well'. Under these circumstances, one reasonable strategy might be to select a diverse set of compounds. Errors in the scoring functions are not likely to be uniformly distributed; for example, errors that result from imperfections in the strain-energy term will affect some compounds more than others. By picking the most diverse set of 'hits' the chances of success may be increased.

Consensus scoring. After scoring a huge virtual library, it is likely that a very large list will remain. This is because scoring functions are imperfect and it is not possible to safely discriminate between small differences in score. One way to get around this problem is to perform consensus scoring, for example, by only including compounds that obtain a certain threshold score in at least three different scoring functions. Of course, some interesting compounds will be lost as a result, but there will be a higher likelihood that the compounds selected will be interesting.

Integration

This review has mentioned a wide array of computational tools – programs for calculating properties, descriptors, generating conformations, docking and scoring – but one factor that has not yet been considered is the integration of these tools. Software tools typically come from several different commercial and academic sources and are not easily integrated. The user typically resorts to running a series of programs in a serial fashion, with each program reading a set of output files produced by the previous program. In many cases the input and output files may not be compatible so a file translation utility such as Babel[105] must be used. Operating in this fashion can introduce severe bottlenecks. For example, a program that processes one million molecules can easily produce several gigabytes of intermediate results. The time required to read and write the intermediate files can have a major impact on performance. To ensure that our system is truly high throughput a modular software architecture was created that is capable of smoothly integrating a variety of in-house, commercial and academic codes. A schematic view of the system is shown in Figure 21. A central hub uses one of several network protocols to communicate with software components that perform a variety of molecular design and analysis functions. The system is capable of coordinating the activities of a large number of processes running on multiple computers with minimal intervention from the end user.

Summary and future prospects

In consideration of the various stages of VS that have been described in this review, it is clear that the entire process

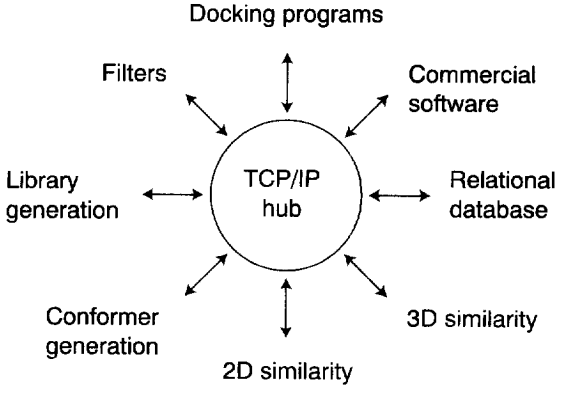

Figure 21. *A modular software architecture for molecular design. TCP/IP, transmission control protocol/interface protocol.*

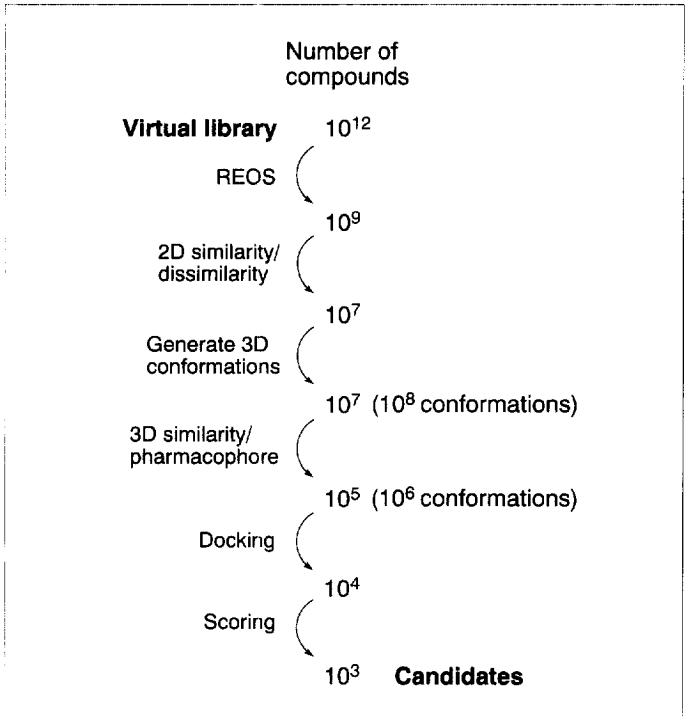

Figure 22. *State of the art in virtual screening.*

relies on the combination of many different kinds of analysis, information and filters.

> Conclusion #5: The ability to combine many filters will allow us to process huge numbers of virtual compounds quickly.

Conceptually, this process is depicted in Figure 22. Only those compounds that 'pass' all the filters (i.e. meet all of the criteria) are of interest. At each stage, a large percentage of compounds is removed. Vast numbers of compounds are processed in the early stages, which requires extremely efficient algorithms and approximate methods. The number of compounds that get to the final stages is only a tiny fraction of the total; as a consequence more time can be afforded on the analysis of each molecule (although still not a lot of time).

Estimates of the current state of the art in VS tend not to have lasting value, but some rough estimates that are expected to be achievable in the year 2000 are also given in Figure 22. The number of compounds that can, in principle, be processed on 32 state-of-the-art processors (e.g. an SGI Origin 2000) is shown along each step in the VS process.

Given the data in Figure 22 and assuming reasonable further improvements in hardware and algorithms, it seems likely that in the near future, all the necessary components will be in place for processing virtual libraries with as many as 10^{15} virtual compounds. But there is much work remaining and tremendous improvements to make before this goal is realized. The following problems remain:

- Missing or limited structural information;
- Poor scoring functions;
- Imprecise understanding of the properties of drug-like molecules;
- Inability to map 3D properties onto 2D structures;
- Incorrect assessment of existing SAR data;
- Poor docking strategies;
- Incorrect synthetic assessment.

Clearly, further improvements in these areas would be of value, and they are likely to be widely pursued by academic and industrial groups.

In addition, there are many other reasons, in any given project, why the entire VS approach may fail. These are true for any drug design approach, and are:

- Poor pharmacokinetics (e.g. absorption, metabolism and half-life);
- Specificity issues;
- Cost of development (e.g. scale-up and formulation);
- Lack of patent coverage;
- Poorly selected therapeutic target.

While it is clear that VS is already possible on a limited scale, and it will become increasingly accessible and valuable over the next decade, it is equally clear that it will be many decades before anyone will be able to reduce the entire process of drug design and development to an automated or 'simple' process that is efficient and reliable.

Acknowledgements

We thank Paul Charifson and Guy Bemis for helpful discussions during the preparation of this manuscript.

REFERENCES

1 Gallop, M.A. *et al.* (1994) *J. Med. Chem.* 37, 1233–1251
2 Gordon, E.M. *et al.* (1994) *J. Med. Chem.* 37, 1385–1399
3 Selway, C.N. and Terrett, N.K. (1996) *Bioorg. Med. Chem.* 645–654
4 Ellman, J., Stoddard, B. and Wells, J. (1997) *Chem. Biol.* 94, 2779–2782
5 Bunin, B.A., Plunkett, M.J. and Ellman, J.A. (1994) *Proc. Natl. Acad. Sci. U. S. A.* 91, 4708–4712
6 Maclean, D. *et al.* (1997) *Proc. Natl. Acad. Sci. U. S. A.* 94, 2805–2810
7 Burbaum, J.J. *et al.* (1995) *Proc. Natl. Acad. Sci. U. S. A.* 92, 6027–6031

8 Bemis, G.W. and Murcko, M.A. (1996) *J. Med. Chem.* 39, 2887–2893
9 Ajay and Murcko, M.A. (1995) *J. Med. Chem.* 38, 4953–4967
10 Ajay, Murcko, M.A. and Stouten, P.F.W. (1997) in *Practical Application of Computer-Aided Drug Design* (Charifson, P.S., ed.), pp. 355–410, Marcel Dekker
11 Hall, L.W. and Kier, L.B. (1991) in *Reviews in Computational Chemistry* (Vol. 2) (Lipkowitz, K.B. and Boyd, D.B., eds), pp. 367–422, VCH
12 Kearsley, S.K. *et al.* (1996) *J. Chem. Inf. Comput. Sci.* 36, 118–127
13 Carhart, R.E., Smith, D.H. and Venkataraghavan, R. (1985) *J. Chem. Inf. Comput. Sci.*, 64–73
14 Nilakantan, R. *et al.* (1987) *J. Chem. Inf. Comput. Sci.* 27, 82–85
15 Brown, R.D. and Martin, Y.C. (1997) *J. Chem. Inf. Comput. Sci.* 37, 1–9
16 Brown, R.D. and Martin, Y.C. (1996) *J. Chem. Inf. Comput. Sci.* 36, 572–584
17 Martin, E.J. *et al.* (1995) *J. Med. Chem.* 38, 1431–1436
18 Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley
19 Clark, D.E. and Westhead, D.R. (1996) *J. Comput.-Aided Mol. Design* 10, 337–358
20 Willett, P. (1995) *Trends Biotechnol.* 13, 516–521
21 Parrill, A.L. (1996) *Drug Discovery Today* 1, 514–521
22 Sheridan, R.P. and Kearsley, S.K. (1995) *J. Chem. Inf. Comput. Sci.* 35, 310–310
23 Cho, S.J., Zheng, W. and Tropsha, A. (1996) in *American Chemical Society Meeting*, New Orleans, LA
24 Weber, L. *et al.* (1995) *Angew. Chem., Int. Ed. Engl.* 34, 2280–2282
25 Singh, J. *et al.* (1996) *J. Am. Chem. Soc.* 118, 1669–1676
26 Stahl, M.T., Walters, W.P. and Murcko, M.A. (1996) in *American Chemical Society National Meeting*, Orlando, FL
27 *Recueil des Travaux Chimiques des Pays-Bays* (1992) 111, 239–334
28 Gillet, V.J. *et al.* (1995) *Perspect. Drug Des. Discovery* 34–50
29 Murcko, M.A. (1997) in *Practical Applications Computer-Aided Drug Design* (Charifson, P.S., ed.), pp. 305–354, Marcel Dekker
30 Boehm, H.J. (1993) in *3D QSAR in Drug Design – Theory Methods and Applications* (Kubinyi, H., ed.), pp. 386–405, ESCOM
31 Galvez, J. *et al.* (1995) *J. Chem. Inf. Comput. Sci.* 35, 272–284
32 van Drie, J.H., Weininger, D. and Martin, Y.C. (1989) *J. Comput.-Aided Mol. Design* 3, 225–251
33 Sprague, P.W. (1995) *Perspect. Drug Des. Discovery* 3, 1–20
34 Cramer, R.D. and Bunce, J.D. (1988) *J. Am. Chem. Soc.* 110, 5959–5967
35 Kellogg, G.E., Semus, S.F. and Abraham, D.J. (1991) *J. Comput.-Aided Mol. Design* 5, 545–552
36 Thorner, D.A. *et al.* (1997) *J. Comput.-Aided Mol. Design* 11, 163–174
37 Gillet, V.J., Willett, P. and Bradshaw, J. (1997) *J. Chem. Inf. Comput. Sci.* 37, 731–740
38 Holliday, J.D., Ranade, S.S. and Willett, P. (1995) *Quant. Struct.-Act. Relatsh.* 14, 501–506
39 Chapman, D. (1996) *J. Comput.-Aided Mol. Design* 10, 501–512
40 Hruby, V.J. *et al.* (1996) *Mol. Diversity* 2, 46–56
41 Cummins, D.J. *et al.* (1996) *J. Chem. Inf. Comput. Sci.* 36, 750–763
42 Pearlman, R.S. (1993) in *3D QSAR in Drug Design* (Kubinyi, H., ed.), pp. 41–79, ESCOM
43 Sadowski, J. and Gasteiger, (1993) *J. Chem. Rev.* 93, 2567–2581
44 Kearsley, S.K. *et al.* (1994) *J. Comput.-Aided Mol. Design* 8, 565–582
45 Van Drie, J.H. (1996) *J. Comput.-Aided Mol. Design* 10, 623–630
46 Smellie, A., Kahn, S.D. and Teig, S.L. (1995) *J. Chem. Inf. Comput. Sci.* 35, 285–294
47 Ghose, A.K. *et al.* (1993) *J. Comput. Chem.* 14, 1050–1065
48 Treasurywala, A.M., Jaeger, E.P. and Peterson, M.L. (1996) *J. Computat. Chem.* 17, 1171–1182
49 Leach, A.R., Prout, K. and Dolata, D.P. (1990) *J. Comput.-Aided. Mol. Design* 4, 271–282
50 Dolata, D.P., Leach, A.R. and Prout, K. (1987) *J. Comput.-Aided Mol. Design* 1, 73–85
51 Wipke, W.T. and Hahn, M.A. (1988) *Tetrahedron Comput. Methodol.* 1, 141
52 Jordan, S.N., Leach, A.R. and Bradshaw, J. (1995) *J. Chem. Inf. Comput. Sci.* 35, 640–650
53 Blaney, J.M. and Dixon, J.S. (1993) *Perspect. Drug Des. Discovery* 1, 310–319
54 Jones, G. and Willett, P. (1995) *Curr. Opin. Biotechnol.* 6, 652–656
55 DesJarlais, R.L. *et al.* (1986) *J. Med. Chem.* 29, 2149–2153
56 Miller, M.D. *et al.* (1994) *J. Comput.-Aided Mol. Design* 8, 153–174
57 Sintchak, M.D. *et al.* (1996) *Cell* 85, 921–930
58 Shoichet, B.K. and Kuntz, I.D. (1993) *Protein Eng.* 6, 723–732
59 Goodford, P.J. (1985) *J. Med. Chem.* 28, 849–857
60 Boobbyer, D.N.A. *et al.* (1989) *J. Med. Chem.* 32, 1083–1094
61 Wade, R.C., Clark, K.J. and Goodford, P.J. (1993) *J. Med. Chem.* 36, 140–147
62 Miranker, A. and Karplus, M. (1991) *Protein Struct. Funct. Genet.* 11, 29–34
63 Caflisch, A., Miranker, A. and Karplus, M. (1993) *J. Med. Chem.* 36, 2142–2167
64 Miranker, A. and Karplus, M. (1995) *Protein Struct. Funct. Genet.* 23, 472–490
65 Clark, K.P. and Ajay (1995) *J. Comput. Chem.* 16, 1210–1226
66 Judson, R.S. *et al.* (1995) *J. Comput. Chem.* 16, 1405–1419
67 Oshiro, C.M., Kuntz, I.D. and Dixon, J.S. (1995) *J. Comput -Aided Mol. Design* 9, 113–130
68 Jones, G. *et al.* (1997) *J. Mol. Biol.* 7, 727–748
69 Morris, G.M. *et al.* (1996) *J. Comput.-Aided Mol. Design* 10, 293–304
70 Gehlhaar, D.K. *et al.* (1995) *Chem. Biol.* 2, 317–324
71 Rarey, M. *et al.* (1996) *J. Mol. Biol.* 261, 470–489
72 Welch, W., Ruppert, J. and Jain, A.N. (1996) *J. Comput.-Aided Mol. Design* 3, 449–462
73 Murray, C.W. *et al.* (1997) *J. Comput.-Aided Mol. Design* 11, 193–207
74 Kick, E.K. *et al.* (1997) *Chem. Biol.* 4, 297–307
75 Zheng, Q. and Kyle, D.J. (1996) *Bioorg. Med. Chem.* 4, 631–638
76 Sheridan, R.P. *et al.* (1989) *Proc. Natl. Acad. Sci. U. S. A* 86, 8165–8169
77 Bemis, G.W. and Kuntz, I.D. (1992) *J. Comput.-Aided Mol. Design* 6, 607–628
78 Good, A.C. *et al.* (1995) *J. Comput.-Aided Mol. Design* 9, 1–12
79 Ponder, J.W. and Richards, F.M. (1987) *J. Mol. Biol.* 193, 775–791
80 Poornima, C.S. and Dean, P.M. (1995) *J. Comput.-Aided Mol. Design* 9, 500–512
81 Poornima, C.S. and Dean, P.M. (1995) *J. Comput.-Aided Mol. Design* 9, 513–520
82 Poornima, C.S. and Dean, P.M. (1995) *J. Comput.-Aided Mol. Design* 9, 521–531
83 Karplus, P.A. and Faerman, C. (1994) *Curr. Opin. Struct. Biol.* 4, 770–776
84 Sitkoff, D., Sharp, K.A. and Honig, B. (1994) *J. Phys. Chem* 98, 1978–1988
85 Danziger, D.J. and Dean, P.M. (1989) *Proc. R. Soc. London* 236, 101–113
86 Baker, E.N. and Hubbard, R.E. (1984) *Progr. Biophys. Mol. Biol.* 44, 97–179
87 Still, W.C. *et al.* (1990) *J. Am. Chem. Soc.* 112, 6127–6129
88 Mark, A.E. and Gunsteren, W.F.v. (1995) in *New Perspectives in Drug Design* (Dean, P.M., Jolles, G. and Newton, C.G., eds), pp. 185–200, Academic Press
89 Kollman, P. (1993) *Chem. Rev.* 93, 2395–2417
90 Bohacek, R.S. and McMartin, C. (1994) *J. Am. Chem. Soc.* 116, 5560–5571
91 Holloway, M.K. *et al.* (1995) *J. Med. Chem.* 38, 305–317
92 Jain, A. (1996) *J. Comput.-Aided Mol. Design* 10, 427–440
93 DeWitte, R.S. and Shakhnovich, E.I. (1996) *J. Am. Chem. Soc.* 118, 11733
94 Head, R.D. *et al.* (1996) *J. Am. Chem. Soc.* 118, 3959–3969
95 Boehm, H-J. (1992) *J. Comput.-Aided Mol. Design* 6, 61–78
96 Boehm, H-J. (1994) *J. Comput.-Aided Mol. Design* 8, 243–256
97 Wesson, L. and Eisenberg, D. (1992) *Protein Sci.* 1, 227–235
98 Stouten, P.F.W. *et al.* (1993) *Mol. Simulation* 10, 97–120
99 Hahn, M.A. (1995) *J. Med. Chem.* 38, 2080
100 Luty, B.A. *et al.* (1995) *J. Comp. Chem.* 16, 454–464
101 Jones, G., Willett, P. and Glen, R.C. (1995) *J. Mol. Biol.* 245, 43–53
102 Klebe, G. and Boehm, H.J. (1997) *J. Recept. Signal Transduct. Res.* 17, 459–473
103 Gillet, V.J. *et al.* (1994) *J. Chem. Inf. Comput. Sci.* 34, 207–217
104 Clark, D.E. and Murray, C.W. (1995) *J. Chem. Inf. Comput. Sci.* 37, 914–923
105 Shah, A.V. *et al.* in *Computerized Chemical Data Standards: Databases, Data Interchange, and Information Systems, ASTM STP1214* (Lysakowski, R. and Gragg, C.E., eds), American Society for Testing and Materials: Philadelphia