Indroduction

A vital goal in drug discovery is identifying novel compounds that can serve a starting point in drug discovery. It is estimated that there are between 10^{60}-10^{100} [1, 2] potential chemical compounds that have a molecular weight < 500 Dalton. By comparison PubChem, the largest database of known chemicals, has a little > 19 million compounds to date[3], covering only a very small percentage of potential chemical space. Even combinatorial libraries, which can range in size up to billions of compounds, do not begin to fully sample the range of all possible compounds. As the full compound space is too vast to search comprehensively, strategies have to be employed to search this space efficiently for discovery of novel lead drug compounds.

De novo design handles this challenge by building compounds from scratch to complement the target receptor. The guiding principal in this approach is that small molecules that are complementary to the target receptor, both in shape and chemical properties, will have the most specific binding. Resulting compounds also need to be "drug-like" and readily synthetically accessible. In theory any molecule of chemical space could be constructed using *de novo* approaches. To reduce the search through chemical space to a manageable problem, strong physical constraints must be taken into account at each step during the generation of the lead drug molecule, limiting the chemical space explored to those regions specifically complementary to the target receptor. The advantage of this approach over a virtual screening strategy to identify these compounds is that the search is directed to the relevant regions in chemical space with a far greater range and diversity of potential lead compounds that can be evaluated.

Also, since compounds are built within the shape constraints of the target receptor, the structures are generated with optimal conformational geometries for binding. In most virtual screening algorithms these conformations must be sampled and can be missed. The main drawback is that the resulting compounds need to be experimentally synthesized and tested, rather than taken from an in-house inventory or ordered commercially. As our ability to predict binding affinities improves, the tradeoff between greater speed of screening and greater diversity of results may drive an increase in use of *de novo* design strategies.

De novo design is inherently combinatoric, as there are many choices are available at each step in molecule generation, leading to a NP-hard problem that cannot be provably solved to the global optimum. Any solution to the problem is going to represent a local optima. And so the success of this approach depends on well-chosen constraints for the problem and an efficient search strategy. The primary constraints are the geometric and chemical constraints derived from the target receptor or target ligand(s), and internal constraints for the geometry and chemistry of the lead compound being constructed. The shape and chemical constraints include both positive and negative requirements. For example receptor site-points or "hot-spots" of interaction must be matched with complementary functional groups in all candidate structures, whereas boundary constraints that define disallowed structural regions must be avoided. These primary constraints are directly handled in the structure generation phase in *de novo* design. The secondary constraints include synthetic accessibility of the final compounds and their predicted ADMET (adsorption, distribution, metabolism, elimination and toxicity) properties. These constraints are handled both by heuristics employed during

structure generation and also as filters on the final set of compounds. While the set of constraints are similar to all *de novo* design algorithms, strategies for generating structures and for searching chemical space to fit these constraints vary considerably among the programs.

One major distinction in *de novo* design programs are whether they are receptor-based or ligand-based. In receptor-based programs the three-dimensional structure of the target receptor is known and provides the primary constraints. Ligand-based programs either generate a 3-D structural pharmacophore model to generate geometric and chemical constraints that are similar to the receptor-based constraints, or they use similarity to a known active ligand or QSAR model as the primary constraint. The features in *de novo* design algorithms can be broken down into the following components, and choices at each of these components can distinguish one program from another:

i. *Site points.* Site points represent "hot spots" of interaction with the target receptor. They define the primary geometric and chemical constraints for the structure generation. Some ligand-based *de novo* design programs use similarity to a molecule template instead of site points.

ii. *Molecule building blocks*. These are the units for constructing the structure. They can be atoms, fragments, or templates (generic fragments).

iii. *Structure generation.* The strategy chosen here is one way used to classify *de novo* design programs. Strategies include: (a) "grow" which starts from a seed point and grows a structure; (b) "fragment-link" where fragments are place in site

points and linked together; and (c) sampling approaches, where molecules are randomly grown. Implicit in each strategy is rules and geometric constraints to generate chemically reasonable structures.

iv. *Search strategy.* The strategy used to search through the combinatoric set of possible (sub)structures, combined with the structure generation strategy, forms the core of the *de novo* design algorithm. Common search strategies include breadth-first search, depth-first search, evolutionary algorithms, and Monte Carlo.

v. *Structure Evaluation.* Evaluates structures based on primary constraints – geometry and chemistry for binding, or similarity to a known active ligand. By evaluating substructures at every step during structure generation, the choices can be pruned during structure generation.

vi. *ADMET and synthetic accessibility.* These are the secondary constraints that can be taken into consideration as a scoring function during structure-generation, as a post-filter after construction. In addition some programs use heuristics during structure-generation to incorporate these constraints.

While there are many variations on the algorithms at each of these component steps, as described below, the this review will focus on the structure

Historical approaches to *de novo* design

The first *de novo* design programs were receptor-based. The field began with programs being developed to describe the binding properties of a target receptor. Initial programs [4-8] devised strategies to identify sites that represented "hot spots" of interaction within

a receptor, placed small molecule fragments or skeletons to interact at these sites and linked them together with generic spacer molecules. Initial ligand-based programs were variations of the receptor-based programs using pharmacophore-derived constraints[9, 10]. Later ligand-based *de novo* programs were developed for cases without site points, and primarily used evolutionary methods to optimize molecules to a QSAR model of activity[11], or by similarity to a known active molecule[12].

Many choices and variations have been tried for each of the different components in *de novo* design as described below.

Identify Interaction Sites

In studying known inhibitors, it is found that there are certain ligand-receptor interactions, deemed "hot spots" that are important for binding and inhibition. These interaction site-points can be generated from analyzing the 3-D coordinates of a target receptor, or from a three-dimensional pharmacophore model based on a superposition of bioactive ligands. These provide positive primary constraints that are positive (must be matched) during structure generation. The coordinates of the receptor, and excluded volume regions from a pharmacophore model, provide primary constraints that are negative (regions to be avoided) during structure generation. For cases where the three-dimensional structure of the receptor is known, the interaction sites can be taken from a known pharmacophore for that protein, or predicted. The first program to predict "hot spots" was the Goodford GRID[13] program, which created a grid inside a target receptor and calculated the energy of probe atoms placed at each point in the grid to create contours of interaction for different probe types. The peak of which would represent an

interaction site. While several early *de novo* programs used GRID[7, 14, 15] to identify site points, the first approach to identify site points that was used in a *de novo* design program was HSITE[4], a rule-based method to search for hydrogen bonding sites based on ideal hydrogen bond geometries derived from crystal data. Other early programs expanded the rule-based method by adding lipophilic regions[16]. HIPPO[17] was the first to add covalent bonds, metal binding sites, and complex hydrogen bonding patterns. MCSS[18] took a unique approach and used a modified molecular dynamics code to identify not spots by simulating probe molecules which simultaneously interacted with the protein but not with each other. The lowest energy probes are retained as starting fragments for *de novo* design. Once the interaction sites are identified, most initial programs used a rule-based scheme to place small molecule fragments that interacted with the site points. Other programs use docking codes on fragments to provide initial placements for initial placement of fragments[7, 19].

Molecule Building Blocks

The first *de novo* design programs used molecule templates [5, 6] as the building blocks, along with programs still in current use[9, 20]. Templates are joined to create a 3-D molecular graph, termed a skeleton, whose vertices are labeled by solely by hybridization state and edges by bond type. This approach divides structure generation into two steps: primary structure generation of generating a skeleton that fits all geometric constraints, and secondary structure generation[21] of substituting atoms into the graph to fit the chemical constraints such as hydrophobicity and electrostatic properties. This approach collapses the search space by looking at structures with the

same geometry simultaneously. In contrast, the atom-based approach starts with real atoms and builds up molecules. It has the theoretical advantage that it allows more diversity in the results, with the corresponding challenge of finding efficient strategies to search through the larger chemical space. Atom-based building blocks have been used successfully in early programs[8, 22] but are harder to constrain to reasonable, synthetically accessible and "drug-like" structures, and require larger combinatorial sampling. Atom-based building blocks have become less common in recent algorithms.

Another development was to the choice of fragments and building rules to incorporate synthetic accessibility and "drug-like" heuristics into the structure generation. The first step in this approach was the RECAP[23] procedure, which broke down existing drugs from the Derwent World Drug Index (WDI), according to common retrosynthetic pathways (i.e. to produce a library of reactants). TOPAS[12, 24] was the first program to use a library generated in such a way, and incorporate the same reaction chemistry into the structure generation, creating 25,000 unique fragments from 11 retrosynthetic pathways.

Structure Generation and Search Strategies

Historically *de novo* design programs have been categorized by their search strategy. The three main categories are (a) fragment-link (b) grow and (c) sampling. This section will briefly describe these algorithms as they are further elaborated in the below. The first *de novo* programs used the *fragment-link* approach, where appropriate fragments were placed at key interaction sites and linked together. There were many strategies on how to link the fragments. One was to join fragments with pre-defined

linkers such as spacer skeletons[6] or fragments from a database[16, 25]. Another was to generate a lattice and perform either depth-first search or breadth-first search along the lattice from one fragment to the other to generate linker fragments. Regular diamond lattices[8], irregular lattices from docked fragments[7] or random lattices [19] were tried for this strategy. Other programs employed an iterative build-up procedure, similar to the *grow* strategy, until all site points were connected. FlexNovo[26] uses a k-greedy search for its buildup procedure. LigBuilder[27] used an evolutionary algorithm to guide the build-up procedure.

The *grow* strategy starts with a seed point or fragment and builds up a molecule. Two of the seminal programs employing the *grow* strategy was the GROW[28] program and LEGEND[22]. GROW generated peptides from amino acid fragments in multiple conformations using a tree-search pruned by predicted binding affinity at each step to guide the growth. LEGEND took the opposite tack and used an atom-based growth strategy with random selection at every decision point (i.e. selection of growth point, selection of next atom, selection of join type) to guide the search process. Many other approaches have been tried to efficiently search combinatorial space during the build-up procedure including: random selection combined with depth-first search[15, 29], Metropolis Monte Carlo[30], and various tree search strategies [9, 20, 31].

The last category for structure generation can be termed sampling approaches, which use sampling and optimization processes to control molecule generation, rather than using site points to direct them in a specific direction such as to grow outwards or to link fragments. Several strategies of this type have been tried including molecular

dynamics [32], Monte Carlo [33], simulated annealing[21, 34], particle-swarm[35] and evolutionary algorithms (EA) [11, 24, 36-39], which is the most common algorithm in recent ligand-based programs. Ligand-based programs that lack site points, such as those with a template molecule or QSAR as the primary constraint, all use a sampling-based method to generate structures.

Each of these strategies requires a connection scheme to join building blocks. With atoms the rules are usually defined by the individual atom valences. Some atom-based programs have linear chains in growing molecule or link between fragment, and look for rings either on the fly[29] during structure generation, by opening, closing, expanding and contracting rings during sampling[40], or as a post-processing step after structure generation to search for rings[41]. With fragment-based methods building blocks can be joined together using a single bond, rings can be fused or spiro-joined. Recently, reaction-based connection rules have been used[24, 38], as a heuristic to incorporate synthetic accessibility into the structure generation stage. Programs that use molecular templates as building graphs have an additional search step after generation of a molecular skeleton to replace vertices with atom type identities to match chemical constraints such as hydrophobicity and electrostatics[9, 21]

Structure evaluation

Receptor-based *de novo* programs use an estimation of binding energy for primary structure evaluation. However, predicting binding affinity accurately continues to be one of the biggest hurdles with *de novo* design programs. Early programs focused mainly on steric constraints and hydrogen bonding[5, 7, 8]. LEGEND[22] was the first

to use a molecular mechanics force-field for scoring. Force-field scores have many shortcomings due to the approximations in the force-field in applying it to ligands, and most notably in the neglect of desolvation and entropy terms, and can be computationally demanding. LUDI[42, 43] developed the first empirical scoring function by defining a set of ligand-receptor interaction types such as hydrogen bonding electrostatic and lipophilic interactions, as well as penalty terms such as number of rotatable bonds. It derived weightings for these terms from a least squares regression on a series of ligands with known binding constants and crystal structures. The challenges here were the small size of the available data set at that time, which limits accuracy to proteins and ligands similar to those used in the regression set. Knowledge-based scoring, first implemented in SMoG[44, 45], uses atom-based ligand-receptor interaction terms with weights derived from a large statistical study of ligand-receptor complexes and the frequencies of various ligand-receptor pairs in these complexes. The advantage of this approach is that there are a larger number of ligand-receptor complexes than those with known binding energies, and so more diversity went into the set, resulting in a less biased scoring function. A common problem with all receptor-based scoring schemes is that they only take into account a static protein. Skelgen is the first program to handle receptor flexibility[46, 47], which was shown to improve the diversity of results in conformational and chemical space, and activity of designed ligands. Many programs that used a receptor-based scoring function also had features to score ligands based on the 3-D pharmacophore model[9, 10, 48, 49] either by deriving receptor-based constraints from the model directly, or by scoring by similarity to the model.

Ligand-based de novo design programs that do not use a pharmacophore model have fundamentally different scoring functions than above. One approach is to derive a scoring function from a QSAR model[11, 40, 50]. This has the disadvantage the scoring parameters have to be re-input for every receptor target. Another common approach is to use the similarity to an active template [24, 37, 51, 52] as the scoring function. This is easier to code up, but reduces the diversity of the resulting of molecules.

Synthetic accessibility and ADMET.

Synthetic accessibility continues to be the second major hurdle with de novo drug design programs. It is evaluated along with prediction of ADMET properties as part of the secondary scoring. Initial de novo design programs performed this evaluation on the final set of structures. CAESA[17] was the first program developed to predict synthetic accessibility, and was based on retrospective analysis. SEEDS compares core substructures both to reaction databases for synthesis pathways and compound database to identify derivatives [53]. More recent approaches are based on similarity to available reactants and heuristics for molecular complexity[54]. A recent survey showed that these two latter approaches have superior success in predicting synthetic accessibility[55].

Some programs include synthetic accessibility and ADMET as heuristics during structure generation. One way to do this is to include a user interaction step where an organic chemist evaluates structures during the build-up process, for example evaluating the initial fragments prior to linking[7, 19, 56], or as a scoring function during an evolutionary algorithm[7, 19, 56]. Another approach was to generate building blocks from fragmenting known drug molecules. This has the heuristic that the building blocks

are “drug-like” [24, 26, 37, 38]. In addition, if the fragmentation is based on retrosynthetic analysis, and regenerated using reaction-based joining rules, then this can also serve as a heuristic for synthetic accessibility, such as in TOPAS[12] and FLUX[37]. Similarly, SYNOPSIS[38] chose available molecules (i.e. reactants) as fragments and used a build-up based on synthetic reactions. Another type of heuristic is to use a substructure lookup during structure generation to filter out substructures that are not drug-like or are synthetically intractable.

Finally some programs include ADMET predictions in a scoring function. For algorithms that build up a structure this score is usually performed after the set of structures has been generated. For algorithms that sample the chemical space of full-size structures, such as the evolutionary algorithms, it can be included in the scoring function during structure generation. This score can range from simple filters using Lipinski’s rule of 5[57] for drug-like compounds, to more complicated of physicochemical properties, or predictions of hERG activity[58].

Several other drug-design methodologies have their roots in *de novo* design. For example fragment-based design approaches are similar to the fragment-link *de novo* strategies, except these take the extra step of validating the fragment positions experimentally prior to linking. The first combinatorial Library design programs started from variations in *de novo* programs - PRO_SELECT[31] evolved from PRO_LIGAND[10] and CombiBUILD[59] from BUILDER[19] (section 8.4).

Common Algorithms in *de novo* structure generation

De novo design algorithms are usually classified according to their structure generation strategy. The three main strategies are: (1) grow (2) fragment-link and (3) sampling (see figure 1).

(1) *GROW*. The *grow* strategy grows a molecule to complement the target receptor (or pharmacophore model) in a sequential build-up procedure. It starts by identifying site interaction points in the target receptor. A site point is chosen as a seed point to start the structure generation. An initial building block is placed in the site to complement its chemical functionality (i.e. electrostatic properties, h-bonding, liophilicity). Growth points are identified on the initial building block. From here, the molecule "grows" through a cycle of adding a building block to the growth points at the end of the partial structure according to connection rules, followed by scoring to evaluate whether to retain the new building block. Growth continues until termination conditions are reached, such as if the molecule extends to all site points or exceeds maximum size. How building blocks are added depends on the search strategy.

(a) In the Metropolis Monte Carlo strategy the acceptance of new building blocks to the growing molecule are biased based on predicted binding affinity according to Boltzmann statistics. A growth point and building block are randomly selected. The new building block is scored by a measure of predicted binding affinity. The Boltzmann factor BF=exp(-affinity_score/RT) is calculated and a random number generated. If BF is greater than the random number the building block is retained and growth points are updated to the newest building block, otherwise it is removed and a new growth point and building block are randomly selected. Note that the BF for building blocks with scores <=

0 is always >= 1 (i.e. always retained). This continues until termination conditions exist. The procedure is re-run from the starting seed until the desired number of structures have been generated.

```
a.  Scheme I: Pseudocode for grow strategy with a Metropolis Monte
Carlo search
Input: receptor or pharmacophore, building block library
      assign sitepoints
      WHILE (large number of structures to grow)DO
            ##generate a structure
            place seed building block (b) in starting sitepoint
            assign growth points to building block
            WHILE (NOT (End=(all sitepoints fit? or > max # atoms?)))
            DO
                  randomly select a growth point (g) partial structure
                  randomly select a building block (b) and add to
                     growth point using connection rules.
                  Prune by primary and secondary constraints
                  IF (pruned)THEN CONTINUE
                  calculate binding affinity score S for (b,g)
                  select or discard according to metropolis search
                        criteria
                  IF (selected)THEN update growth points to g2 on (g,b)
            END DO
            save structure(s) to list
      END DO
      evaluate final structures for predicted binding affinity
      evaluate final structures for synthetic accessibility and ADMET
      prioritize final structures

      (i) Metopolis criteria
          calculate Boltzman Factor BF=exp(-affinity_score/RT)
          generate random number from 0 to 1
          retain building block= TRUE IF BF > random
```

(b) In the various tree-search strategies the inner WHILE loop above is replaced with a tree-search algorithm. Building blocks are tried at each growth point, scored, and added as nodes on a search graph. The nodes may be pruned using primary constraints (e.g. boundary violations) and secondary constraints (e.g. matching to a list of synthetically intractable substructures), and may be prioritized by a scheme such as score or distance to an interaction “hot spot”. In depth-first search the top-scoring node in the

graph is selected for expansion (i.e. to examine for growing) whereas in breadth-first search all nodes at each level are selected for expansion before going to the next level. In this way, Depth-first search completes a solution (ie generates a structure) before examining the next partial solution, whereas breadth-first search expands all the partial solutions simultaneously until all solutions are found. Depth-first search may find a single solution faster, but may not be the best overall solution, whereas breath-first exhaustively searches for solutions. Functions can be added to estimate the costs of continuing along a partial solution to prioritize nodes in "best-first" searches such as A*.

Programs in current use that implement *grow*

The *grow* algorithm is implemented in several *de novo* design programs in current use and that have had success in identifying lead compounds in prospective studies (See table 1.a). AlleGrow, the successor of GrowMol[60] uses the Metropolis Monte Carlo selection criteria. It is available commercially at http://bostondenovo.com/Allegrow.htm. Legend[22] uses random selection at ever choice point. SPROUT[9] takes the other approach and uses a tree-search algorithm, which can be run in a modified "best-search" algorithm, or to completion. It directs growth by prioritizing growth points based on closeness to unsatisfied site points and pruning templates that prevent reaching site points by being too close but not satisfying site point. SPROUT is commercially available at SimBioSys, Inc (http://www.simbiosys.ca/sprout/index.html). FlexNovo[26] uses FlexX to dock initial fragments and a build-up procedure based on a k-greedy algorithm. It is commercially available at BioSolveIt (http://www.biosolveit.de/FlexNovo/).

Advantages, limitations, computational complexity?

The tree-searches are deterministic algorithms. Run to completion, they will find all solutions. The time complexity for most tree searches is $O(b^d)$, where b=branching factor and d= depth, although proper heuristics in best-first search can greatly reduce this. The branching factor in this case is a product of number of attachment points times number of building blocks, whereas depth is the number of building blocks in a final structure. A quick back-of the hand comparison of atom-based versus fragment based approaches would have b for atom-based methods as ~12 atoms/functional groups X 2 attachment points on each on average (3 for sp3 atoms, 2 for sp2, 1 for sp) for atom-based and depth d (~50) leading to $b^d = 24^{50} \approx 10^{70}$, or roughly all of chemical space. For fragment-based with a small fragment library b is 30 fragmentsX 4 average attachment points each (larger since rings included) and depth approximately 8 is $b^d = 120^8 \approx 10^{16}$. We can see why smaller fragment libraries are usually chosen for tree-search methods, whereas Monte Carlo is chosen for both atom-based and fragment-based methods. This also shows the advantage of using generic templates in tree-search approaches such as SPROUT, which reduces the complexity by greatly reducing size of the template library. Note that the diversity covered in this approach is far greater than 10^{16} because atom types are placed into the generic fragments, but it does not approach the full diversity from an atom-based approach.

Overall, the *grow* algorithms have been successful in finding new drug candidates. However, they tend to behave poorly in situations where the receptor site consists of 2 or more subpockets separated by a large gap, whereas the fragment-link (next section) performs better in these situations.

(2) ***Fragment-link.*** The fragment-link strategy also starts out by identifying site-points in a target receptor or pharmacophore model. In this case complementary fragments are placed in all of these "hot-spots" to maximize interaction. Results at this point can be pruned by visual inspection. Linking groups are then generated or chosen from a link library and fit to the fragments. Linking groups that do not match the primary constraints (shape & chemistry) or make substructures that violate secondary constraints can be discarded. The final structures are evaluated by predicted binding affinity and secondary scoring characteristics such as synthetic accessibility and ADMET, and prioritized. (see figure 1b and table 1b)

```
Scheme II.  Pseudocode for fragment-link strategy
Input: receptor or pharmacophore, library of initial building blocks,
library of bridging building blocks

      assign sitepoints
      place building block(s) in sitepoint(s) according to rules
      prune by criteria (visual inspection and/or score)
      WHILE (NOT all fragments joined) DO
            identify 2 closest fragments
            identify link points between fragments(closest atoms)
            place bridging group(s) to join at these points by matching
                  distances and angles to bridge library.
      END DO
      evaluate final structures for binding affinity
      evaluate final structures for synthetic accessibility and ADMET
      prioritize final structures
```

Programs in current use that implement *fragment-link*

Several programs have successfully applied the *fragment-link* algorithm to identify lead compounds (see table 1.b.). Best-known in is the Ludi[14, 16] program, available commercially at Accelrys (http://accelrys.com/). The MCSS[61] program has been combined with several others for the bridging step including HOOK[25], Leapfrog[62], LUDI, and by visual inspection. Ligbuilder is a hybrid algorithm which

includes *grow* and *fragment-link*, both using an evolutionary algorithm to generate structures. Ligbuilder is freely available at (http://www.chem.ac.ru/Chemistry/Soft/LIGBUILD.en.html) .

Advantages, limitations, computational complexity?

This approach has the advantage in that it maximizes interactions in the key interaction sites in the target protein. It has the computational advantage that the search for bridge points is an O(n) lookup through a fragment database. The challenge is identifying linking groups of the proper chemistry and geometry that do not greatly alter the orientation of the fragments binding to these sites, and which do not have artificially strained bonds, angles and torsions. In terms of amount of chemical space sampled, it covers roughly the same chemical space as other fragment-based methods using the *grow* strategy.

(3) Sampling strategies

Sampling strategies differ from *grow* and *fragment-link* in that they sample structures without directing generation in a particular direction (outward or explicitly linking interaction sites). Evolutionary algorithms (EA) are the most common chosen for this purpose. There are many types of EAs: genetic algorithms (GA) which encode the molecular structure in a "chromosome" of fixed length that is operated on and transformed into the molecular structure for fitness evaluations; genetic programming, where the chromosomes are trees to allow them to have variable length; and evolutionary strategies which operate directly on the phenotype, which is the molecular structure. A basic evolutionary strategy (μ, λ) is shown below[63], where λ is size of the child population and μ is the size of the parent population in each generation.

The algorithm starts with a population of λ chemical structures (the initial child population) generated from putting a random selection of building blocks together according to the building rules for the building blocks. Each structure in this population is evaluated for "fitness". The fitness is the scoring function that can combine primary and secondary constraints. In receptor-based *de novo* methods the primary score may be the interaction score such as from a docking calculation[64], minus any boundary violations. For ligand-based employing a single template ligand, the primary score may be a similarity score. Secondary scoring considerations, such as requirements for Lipinski's[57] rules, or other ADMET considerations, can be added to the fitness function here, along with other molecule properties such as surface area or radius of gyration. The most fit μ structures are selected to be parents for the next generation. Mutation and crossover operators can be performed on the parents. Mutation in this case is to take a parent structure and remove a building block and replace it according to the joining rules. Crossover is to remove building blocks from each parent and swap them again according to joining rules. Some algorithms have only mutation[24] or only crossover[52]. A total of λ child structures are generated. This cycle is repeated until it reaches a maximum generation of children or a termination condition is reached such as convergence. One feature found with this algorithm was that since building blocks could vary greatly in size, the parent p could grow and shrink as well, while still retaining the same number of building blocks.

```
Scheme III.  Pseudocode for a basic ES (μ, λ)
Generate λ random structures Sc
DO
      Evaluate fitness Fc of each structure Sc in population
      Choose μ most fit (Fc) structures as parents Sp
      Mutate and Crossover of parents Sp to generate population λ
           children Sc
UNTIL (> maximum generations or termination condition)
```

Programs in current use that implement EA

Evolutionary algorithms are especially common in ligand-based design programs, although several receptor-based programs also employ this approach. See Table 1.c for some examples of lead compounds identified using EA. One successful implementation of this algorithm is in the TOPAS [24] ligand-based *de novo* program, which uses pairwise similarity to a molecular template as the fitness function. It sets $\lambda = 100$ and $\mu = 1$ (i.e. 1 parent) with no crossover operation, so all variation is through mutation. It uses 25,000 fragments from the WDI using 11 retrosynthetic pathways. The variance in each new child structure can be controlled by how similar a new building block is to the original building block being mutated, and is controlled by a parameter("step-size") that is a Gaussian distribution of random numbers, resulting a child population that is bell-shaped distribution of variations with the parent at the center. It was found 100 generations was sufficient to explore chemical space in this program. TOPAS is at Hoffmann-La Roche and is not generally available.

Flux[37] was developed based on TOPAS. It finds optimal results with a 50:50 ratio between crossover and mutation, and typically sets maximum generations to 75 (50 was found to converge in most cases). The other main difference from TOPAS is a modified similarity descriptor that is weighted. It is being used at the Goethe University in Germany but is currently not generally available.

LEA3D uses fragments as building blocks generating from fragmenting "drug-like" database of over 8000 fragments into single rings, single rings, fused rings, and acyclic parts. It allows both 1 and 2 point crossover and mutation. It is not generally

available but an in-house version is in use at the Centre De Biochimie Structurale, Montepelier, France.

De novo programs incorporating EA that are commercially available include AutoLudi and LeapFrog[48] and the Molecule Evoluator[56, 65]. AutoLudi is en extension of LUDI that uses EA to modify an existing lead compound by adding on small fragments. LeapFrog commercially available at Tripos (http://www.tripos.com), which evolves a population of molecules in an atom-based method. The Molecular Evoluator[56] uses an unusual fitness function - the user working interactively with the program. It is available at Cidrux Pharmaceuticals [66]

Advantages, limitations, computational complexity?

A general challenge for EAs is the molecule representation. SMILES strings such as in LEA[11] have the problem that invalid molecules result during crossover and mutation, and also more steps are required to build up a molecule. TreeSmiles, a variation of SMILES with all hydrogens explicitly shown, helps avoid unreasonable structures[56]. The LEA3D successor of LEA uses fragments instead of atoms as the building block, with numbers representing fragments for genetic operations[67]. Other approaches operate directly on the 3-D structures leading to additional translational and rotational operators[36].

The theoretical chemical space available for a fragment-based approach is $(nb)^{ns}$, where nb=number of building blocks in fragment library, and ns is average number of building blocks is the final structure. For TOPAS which has 25,000 building blocks and approximately 4 building blocks in a final structure the total is $(25000)^4 \approx 10^{18}$. For an atom-based method this would approach all of chemical space ($\approx 10^{60}$). However, the

number of structures actually evaluated in an EA run is much smaller as it is given by the function λ X ng, where λ the population each generation and ng the number of generations. For example, TOPAS has population size 100 X 100 generations ≈ 10^4. Similarly, LEA3D has a population size of 40 X 100 generations is 4000. In practice this seems to be sufficient to generate enough reasonable solutions to find interesting leads. Compared to the *grow* and *fragment-link* EA algorithms have the advantage that, since they do not target interaction site points, the output structures are not strained (ie have low intramolecular energies). The corresponding disadvantage is they may not bind to known important interaction sites.

Summary

In examining all *grow*, *fragment-link*, and sampling-based algorithms, one aspect in common is the use of a random operator of some sort during the structure generation process. This is important for two reasons: first because the scoring functions are not perfect the best-scoring atom or fragment may not represent the best binder. Second, and more importantly, because the path to construct a *de novo* structure is not a linear function of the scoring function (i.e. higher scoring final structures are not a linear result of the highest scoring pre-cursors, a structure often needs to go through a lower energy construction pathway to get to the final structure).

Ligand-based programs that use similarity to a molecular template or QSAR for scoring require a sampling approach, and EA is the most commonly chosen one for these programs for its simplicity to program up and its effectiveness in these cases. With receptor-based approaches, the *grow* and *fragment-link* algorithms, which include pharmacophore data in the form of site points, are historically favored. Pure sampling

approaches are most commonly seen as lead-optimization once a core has been designed using a *grow* or *fragment-link* approach.

The major hurdles for *de novo* design to overcome to be an effective tool in drug discovery are the same today as when the field began: how to accurately predicting receptor-based affinity and predict synthetic accessibility. Without these, it could be a costly effort to synthesize complex molecules which may not even bind to the target receptor. Newer heuristics for synthetic accessibility, using reaction-based fragment libraries, and heuristics based on molecular complexity, have improved the quality of structures resulting from de novo design. Several *de novo* design strategies have now been shown to be successful in prospective studies.

Table 1 *De novo* design programs with recent results. See review articles[68, 69] for a more comprehensive list of *de novo* design programs using (a) Grow strategy (b) fragment-link (c) sampling.

a. Grow strategy

Name	*Ligand or Receptor sites*	*Building blocks*	*Search type**	*Scoring type**	*Prospective studies*
AlleGrow (GrowMol[30])	Receptor	Atoms/ fragments	MC	Empirical	Aspartic protease[60] xWNT8 & hWNT8 [70];
Legend[71]	Receptor	Atom	Random	FF	CDK4 inhibitors[53] Aldose reductase[72]
Sprout[9, 20, 54]	Receptor	Skeleton/ fragments	A* or Exhaus-tive[54]	Empirical	dihydroorotate dehydrogenase[73, 74] Nk(2) antagonists[73, 74]

b. Fragment-Link

Name	*Ligand or Receptor sites*	*Building blocks*	*Search type**	*Scoring type*	*Prospective studies*
Ludi[16, 43]	Receptor	Fragments		Empirical	CYP51(w/ MCSS)[75] Leucine aminopeptidase[76, 77]
MCSS[61]	Receptor	Fragments	MD	FF	CYP51(w/LUDI) [75] PPARγ (w/LeapFrog) [62]

c. Sampling

Name	*Ligand or Receptor sites*	*Building blocks*	*Search type**	*Scoring type*	*Prospective studies*
LEA3D[11, 67]	Receptor/ Ligand	Fragments	EA	Empirical	thymidine monophosphate kinase[67]
TOPAS[12, 24]	Ligand	Fragments	EA	similarity	cannabinoid-1 receptor [78, 79] Kv1.5 [12]
SkelGen[80]	Receptor/ Ligand	Fragments	Simulated annealing	empirical	cannabinoid-1 receptor [78, 79] Kv1.5 [12]
Flux1/ Flux2[37, 81]	Ligand	Frag/recap	EA	similarity	TAR RNA[82]
LeapFrog[48]**	Both	Fragments	EA	Empirical	PPARγ (w/MCSS) [62] Link-function only
SYNOPSIS[38]	Receptor	Frag	EA	Target-specific	HIV protease[38]

*FF=force-field,EA=evolutionary algorithm,BFS=breadth-first search,MC=monte carlo.

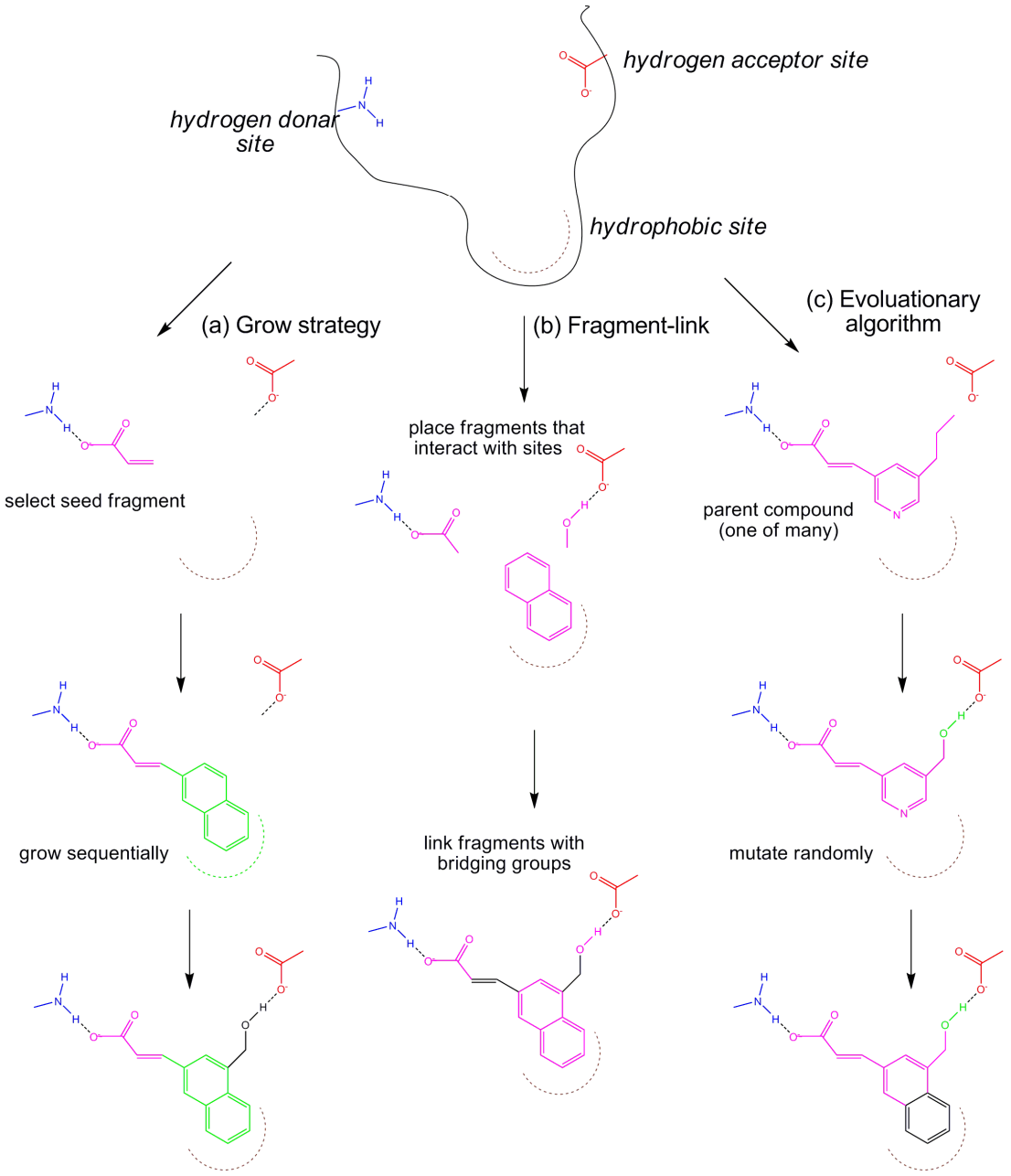

Figure 1. Comparison of 3 different categories of structure generation algorithms. All start with identifying site points in the target receptor. (a) Grow strategy – an initial fragment is placed at one site point and grown sequentially(magenta,green, then black fragment). (b) fragments are placed in all site points (magenta) and linked together (black). (c) complete initial structure (magenta) is mutated in random locations (green then black).

1. Bohacek, R. S.; McMartin, C.; Guida, W. C., The art and practice of structure-based drug design: a molecular modeling perspective. *Med Res Rev* **1996,** 16, (1), 3-50.
2. The numbers game. *Nat Rev Drug Discov* **2002,** 1, (12), 929-929.
3. The PubChem Project. In.
4. Danziger, D. J.; Dean, P. M., Automated site-directed drug design: a general algorithm for knowledge acquisition about hydrogen-bonding regions at protein surfaces. *Proc R Soc Lond B Biol Sci* **1989,** 236, (1283), 101-13.
5. Lewis, R. A.; Dean, P. M., Automated site-directed drug design: the formation of molecular templates in primary structure generation. *Proc R Soc Lond B Biol Sci* **1989,** 236, (1283), 141-62.
6. Lewis, R. A.; Dean, P. M., Automated site-directed drug design: the concept of spacer skeletons for primary structure generation. *Proc R Soc Lond B Biol Sci* **1989,** 236, (1283), 125-40.
7. Lewis, R. A.; Roe, D. C.; Huang, C.; Ferrin, T. E.; Langridge, R.; Kuntz, I. D., Automated site-directed drug design using molecular lattices. *J Mol Graph* **1992,** 10, (2), 66-78, 106.
8. Lewis, R. A., Automated site-directed drug design: approaches to the formation of 3D molecular graphs. *J Comput Aided Mol Des* **1990,** 4, (2), 205-10.
9. Gillet, V. J.; Newell, W.; Mata, P.; Myatt, G.; Sike, S.; Zsoldos, Z.; Johnson, A. P., SPROUT: recent developments in the de novo design of molecules. *J Chem Inf Comput Sci* **1994,** 34, (1), 207-17.
10. Clark, D. E.; Frenkel, D.; Levy, S. A.; Li, J.; Murray, C. W.; Robson, B.; Waszkowycz, B.; Westhead, D. R., PRO-LIGAND: an approach to de novo molecular design. 1. Application to the design of organic molecules. *J Comput Aided Mol Des* **1995,** 9, (1), 13-32.
11. Douguet, D.; Thoreau, E.; Grassy, G., A genetic algorithm for the automated generation of small organic molecules: drug design using an evolutionary algorithm. *J Comput Aided Mol Des* **2000,** 14, (5), 449-66.
12. Schneider, G.; Clement-Chomienne, O.; Hilfiger, L.; Schneider, P.; Kirsch, S.; Bohm, H. J.; Neidhart, W., Virtual Screening for Bioactive Molecules by Evolutionary De Novo Design Special thanks to Neil R. Taylor for his help in preparation of the manuscript. *Angew Chem Int Ed Engl* **2000,** 39, (22), 4130-4133.
13. Goodford, P. J., A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. *J Med Chem* **1985,** 28, (7), 849-57.
14. Bohm, H. J., The computer program LUDI: a new method for the de novo design of enzyme inhibitors. *J Comput Aided Mol Des* **1992,** 6, (1), 61-78.
15. Rotstein, S. H.; Murcko, M. A., GroupBuild: a fragment-based method for de novo drug design. *J Med Chem* **1993,** 36, (12), 1700-10.
16. Bohm, H. J., LUDI: rule-based automatic design of new substituents for enzyme inhibitor leads. *J Comput Aided Mol Des* **1992,** 6, (6), 593-606.
17. Gillet, V.; Myatt, G.; Zsoldos, Z.; Johnson, A., SPROUT, HIPPO and CAESA: Tools for de novo structure generation and estimation of synthetic accessibility. *Perspectives in Drug Discovery and Design* **1995,** 3, (1), 34-50.
18. Miranker, A.; Karplus, M., Functionality maps of binding sites: a multiple copy simultaneous search method. *Proteins* **1991,** 11, (1), 29-34.

19. Roe, D. C.; Kuntz, I. D., BUILDER v.2: improving the chemistry of a de novo design strategy. *J Comput Aided Mol Des* **1995,** 9, (3), 269-82.
20. Gillet, V.; Johnson, A. P.; Mata, P.; Sike, S.; Williams, P., SPROUT: a program for structure generation. *J Comput Aided Mol Des* **1993,** 7, (2), 127-53.
21. Todorov, N. P.; Dean, P. M., A branch-and-bound method for optimal atom-type assignment in de novo ligand design. *J Comput Aided Mol Des* **1998,** 12, (4), 335-49.
22. Nishibata, Y.; Itai, A., Automatic creation of drug candidate structures based on receptor structure. Starting point for artificial lead generation. . *Tetrahedron* **1991,** 47, (43), 8985-8990.
23. Lewell, X. Q.; Judd, D. B.; Watson, S. P.; Hann, M. M., RECAP--retrosynthetic combinatorial analysis procedure: a powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. *J Chem Inf Comput Sci* **1998,** 38, (3), 511-22.
24. Schneider, G.; Lee, M. L.; Stahl, M.; Schneider, P., De novo design of molecular architectures by evolutionary assembly of drug-derived building blocks. *J Comput Aided Mol Des* **2000,** 14, (5), 487-94.
25. Eisen, M. B.; Wiley, D. C.; Karplus, M.; Hubbard, R. E., HOOK: a program for finding novel molecular architectures that satisfy the chemical and steric requirements of a macromolecule binding site. *Proteins* **1994,** 19, (3), 199-221.
26. Degen, J.; Rarey, M., FlexNovo: structure-based searching in large fragment spaces. *ChemMedChem* **2006,** 1, (8), 854-68.
27. Wang, R.; Gao, Y.; Lai, L., LigBuilder: A Multi-Purpose Program for Structure-Based Drug Design. *Journal of Molecular Modeling* **2000,** 6, (7), 498-516.
28. Moon, J. B.; Howe, W. J., Computer design of bioactive molecules: a method for receptor-based de novo ligand design. *Proteins* **1991,** 11, (4), 314-28.
29. Rotstein, S. H.; Murcko, M. A., GenStar: a method for de novo drug design. *J Comput Aided Mol Des* **1993,** 7, (1), 23-43.
30. Bohacek, R. S.; McMartin, C., Multiple Highly Diverse Structures Complementary to Enzyme Binding Sites: Results of Extensive Application of a de Novo Design Method Incorporating Combinatorial Growth. *J. Am. Chem. Soc.* **1994,** 116, (13), 5560-5571.
31. Murray, C. W.; Clark, D. E.; Auton, T. R.; Firth, M. A.; Li, J.; Sykes, R. A.; Waszkowycz, B.; Westhead, D. R.; Young, S. C., PRO_SELECT: combining structure-based drug design and combinatorial chemistry for rapid lead discovery. 1. Technology. *J Comput Aided Mol Des* **1997,** 11, (2), 193-207.
32. David A. Pearlman, M. A. M., CONCEPTS: New dynamic algorithm for <I>de novo</I> drug suggestion. *Journal of Computational Chemistry* **1993,** 14, (10), 1184-1193.
33. Gehlhaar, D. K.; Moerder, K. E.; Zichi, D.; Sherman, C. J.; Ogden, R. C.; Freer, S. T., De novo design of enzyme inhibitors by Monte Carlo ligand generation. *J Med Chem* **1995,** 38, (3), 466-72.
34. Todorov, N. P.; Dean, P. M., Evaluation of a method for controlling molecular scaffold diversity in de novo ligand design. *J Comput Aided Mol Des* **1997,** 11, (2), 175-92.

35. Hartenfeller, M.; Proschak, E.; Schuller, A.; Schneider, G., Concept of combinatorial de novo design of drug-like molecules by particle swarm optimization. *Chem Biol Drug Des* **2008,** 72, (1), 16-26.
36. Glen, R. C.; Payne, A. W., A genetic algorithm for the automated generation of molecules within constraints. *J Comput Aided Mol Des* **1995,** 9, (2), 181-202.
37. Fechner, U.; Schneider, G., Flux (2): comparison of molecular mutation and crossover operators for ligand-based de novo design. *J Chem Inf Model* **2007,** 47, (2), 656-67.
38. Vinkers, H. M.; de Jonge, M. R.; Daeyaert, F. F.; Heeres, J.; Koymans, L. M.; van Lenthe, J. H.; Lewi, P. J.; Timmerman, H.; Van Aken, K.; Janssen, P. A., SYNOPSIS: SYNthesize and OPtimize System in Silico. *J Med Chem* **2003,** 46, (13), 2765-73.
39. Pierce, A. C.; Rao, G.; Bemis, G. W., BREED: Generating novel inhibitors through hybridization of known ligands. Application to CDK2, p38, and HIV protease. *J Med Chem* **2004,** 47, (11), 2768-75.
40. Nachbar, R. B., **2000**.
41. Andrew R. Leach, R. A. L., A ring-bracing approach to computer-assisted ligand design. *Journal of Computational Chemistry* **1994,** 15, (2), 233-240.
42. Bohm, H. J., Prediction of binding constants of protein ligands: a fast method for the prioritization of hits obtained from de novo design or 3D database search programs. *J Comput Aided Mol Des* **1998,** 12, (4), 309-23.
43. Bohm, H. J., The development of a simple empirical scoring function to estimate the binding constant for a protein-ligand complex of known three-dimensional structure. *J Comput Aided Mol Des* **1994,** 8, (3), 243-56.
44. Ishchenko, A. V.; Shakhnovich, E. I., SMall Molecule Growth 2001 (SMoG2001): an improved knowledge-based scoring function for protein-ligand interactions. *J Med Chem* **2002,** 45, (13), 2770-80.
45. DeWitte, R. S.; Shakhnovich, E. I., SMoG: de Novo Design Method Based on Simple, Fast, and Accurate Free Energy Estimates. 1. Methodology and Supporting Evidence. *J. Am. Chem. Soc.* **1996,** 118, (47), 11733-11744.
46. Todorov, N. P.; Buenemann, C. L.; Alberts, I. L., De novo ligand design to an ensemble of protein structures. *Proteins* **2006,** 64, (1), 43-59.
47. Alberts, I. L.; Todorov, N. P.; Dean, P. M., Receptor flexibility in de novo ligand design and docking. *J Med Chem* **2005,** 48, (21), 6585-96.
48. *Leapfrog, SYBYL 7.1*, TRIPOS: St. Louis, MO.
49. Waszkowycz, B.; Clark, D. E.; Frenkel, D.; Li, J.; Murray, C. W.; Robson, B.; Westhead, D. R., PRO_LIGAND: an approach to de novo molecular design. 2. Design of novel molecules from molecular field analysis (MFA) models and pharmacophores. *J Med Chem* **1994,** 37, (23), 3994-4002.
50. Pellegrini, E.; Field, M. J., Development and testing of a de novo drug-design algorithm. *J Comput Aided Mol Des* **2003,** 17, (10), 621-41.
51. Brown, N.; McKay, B.; Gilardoni, F.; Gasteiger, J., A Graph-Based Genetic Algorithm and Its Application to the Multiobjective Evolution of Median Molecules. *J. Chem. Inf. Comput. Sci.* **2004,** 44, (3), 1079-1087.
52. Globus, A.; Lawton, J.; Wipke, T., Automatic molecular design using evolutionary techniques. *Nanotechnology* **1999,** 10, (3), 290-299.

53. Honma, T.; Hayashi, K.; Aoyama, T.; Hashimoto, N.; Machida, T.; Fukasawa, K.; Iwama, T.; Ikeura, C.; Ikuta, M.; Suzuki-Takahashi, I.; Iwasawa, Y.; Hayama, T.; Nishimura, S.; Morishima, H., Structure-based generation of a new class of potent Cdk4 inhibitors: new de novo design strategy and library design. *J Med Chem* **2001,** 44, (26), 4615-27.
54. Boda, K.; Johnson, A. P., Molecular complexity analysis of de novo designed ligands. *J Med Chem* **2006,** 49, (20), 5869-79.
55. Gasteiger, J., De novo design and synthetic accessibility. *J Comput Aided Mol Des* **2007,** 21, (6), 307-9.
56. Lameijer, E. W.; Kok, J. N.; Back, T.; Ijzerman, A. P., The molecule evoluator. An interactive evolutionary algorithm for the design of drug-like molecules. *J Chem Inf Model* **2006,** 46, (2), 545-52.
57. Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J., Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews* **2001,** 46, (1-3), 3-26.
58. Aronov, A. M., Predictive in silico modeling for hERG channel blockers. *Drug Discov Today* **2005,** 10, (2), 149-55.
59. Kick, E. K.; Roe, D. C.; Skillman, A. G.; Liu, G.; Ewing, T. J.; Sun, Y.; Kuntz, I. D.; Ellman, J. A., Structure-based design and combinatorial chemistry yield low nanomolar inhibitors of cathepsin D. *Chem Biol* **1997,** 4, (4), 297-307.
60. Ripka, A. S.; Satyshur, K. A.; Bohacek, R. S.; Rich, D. H., Aspartic protease inhibitors designed from computer-generated templates bind as predicted. *Org Lett* **2001,** 3, (15), 2309-12.
61. Caflisch, A.; Miranker, A.; Karplus, M., Multiple copy simultaneous search and construction of ligands in binding sites: application to inhibitors of HIV-1 aspartic proteinase. *J Med Chem* **1993,** 36, (15), 2142-67.
62. Dong, X.; Zhang, Z.; Wen, R.; Shen, J.; Shen, X.; Jiang, H., Structure-based de novo design, synthesis, and biological evaluation of the indole-based PPARgamma ligands (I). *Bioorg Med Chem Lett* **2006,** 16, (22), 5913-6.
63. Schwefel, H. P., Deep insight from simple models of evolution. *Biosystems* **2002,** 64, (1-3), 189-98.
64. Pegg, S. C.; Haresco, J. J.; Kuntz, I. D., A genetic algorithm for structure-based de novo design. *J Comput Aided Mol Des* **2001,** 15, (10), 911-33.
65. Lameijer, E. W.; Tromp, R. A.; Spanjersberg, R. F.; Brussee, J.; Ijzerman, A. P., Designing active template molecules by combining computational de novo design and human chemist's expertise. *J Med Chem* **2007,** 50, (8), 1925-32.
66. *The Molecular Evoluator*, Cidrux Pharminformatics: Haarlem, The Netherlands.
67. Douguet, D.; Munier-Lehmann, H.; Labesse, G.; Pochet, S., LEA3D: a computer-aided ligand design for structure-based drug design. *J Med Chem* **2005,** 48, (7), 2457-68.
68. Schneider, G.; Fechner, U., Computer-based de novo design of drug-like molecules. *Nat Rev Drug Discov* **2005,** 4, (8), 649-663.
69. Mauser, H.; Guba, W., Recent developments in de novo design and scaffold hopping. *Curr Opin Drug Discov Devel* **2008,** 11, (3), 365-74.
70. Voronkov, A. E.; Baskin, II; Palyulin, V. A.; Zefirov, N. S., Molecular modeling of modified peptides, potent inhibitors of the xWNT8 and hWNT8 proteins. *J Mol Graph Model* **2008,** 26, (7), 1179-87.

71. Nishibata, Y.; Itai, A., Confirmation of usefulness of a structure construction program based on three-dimensional receptor structure for rational lead generation. *J Med Chem* **1993,** 36, (20), 2921-8.
72. Iwata, Y.; Naito, S.; Itai, A.; Miyamoto, S., Protein structure-based de novo design and synthesis of aldose reductase inhibitors. *Drug Des Discov* **2001,** 17, (4), 349-59.
73. Heikkila, T.; Thirumalairajan, S.; Davies, M.; Parsons, M. R.; McConkey, A. G.; Fishwick, C. W.; Johnson, A. P., The first de novo designed inhibitors of Plasmodium falciparum dihydroorotate dehydrogenase. *Bioorg Med Chem Lett* **2006,** 16, (1), 88-92.
74. Ali, M. A.; Bhogal, N.; Findlay, J. B.; Fishwick, C. W., The first de novo-designed antagonists of the human NK(2) receptor. *J Med Chem* **2005,** 48, (18), 5655-8.
75. Ji, H.; Zhang, W.; Zhang, M.; Kudo, M.; Aoyama, Y.; Yoshida, Y.; Sheng, C.; Song, Y.; Yang, S.; Zhou, Y.; Lu, J.; Zhu, J., Structure-based de novo design, synthesis, and biological evaluation of non-azole inhibitors specific for lanosterol 14alpha-demethylase of fungi. *J Med Chem* **2003,** 46, (4), 474-85.
76. Grembecka, J.; Sokalski, W. A.; Kafarski, P., Computer-aided design and activity prediction of leucine aminopeptidase inhibitors. *J Comput Aided Mol Des* **2000,** 14, (6), 531-44.
77. Grembecka, J.; Mucha, A.; Cierpicki, T.; Kafarski, P., The most potent organophosphorus inhibitors of leucine aminopeptidase. Structure-based design, chemistry, and activity. *J Med Chem* **2003,** 46, (13), 2641-55.
78. Alig, L.; Alsenz, J.; Andjelkovic, M.; Bendels, S.; Benardeau, A.; Bleicher, K.; Bourson, A.; David-Pierson, P.; Guba, W.; Hildbrand, S.; Kube, D.; Lubbers, T.; Mayweg, A. V.; Narquizian, R.; Neidhart, W.; Nettekoven, M.; Plancher, J. M.; Rocha, C.; Rogers-Evans, M.; Rover, S.; Schneider, G.; Taylor, S.; Waldmeier, P., Benzodioxoles: novel cannabinoid-1 receptor inverse agonists for the treatment of obesity. *J Med Chem* **2008,** 51, (7), 2115-27.
79. Mark Rogers-Evans, Alexander I. A. Konrad H. B. D. K. G. S., Identification of novel cannabinoid receptor ligands <I>via</I> evolutionary de novo design and rapid parallel synthesis. *QSAR & Combinatorial Science* **2004,** 23, (6), 426-430.
80. Stahl, M.; Todorov, N. P.; James, T.; Mauser, H.; Boehm, H. J.; Dean, P. M., A validation study on the practical use of automated de novo design. *J Comput Aided Mol Des* **2002,** 16, (7), 459-78.
81. Fechner, U.; Schneider, G., Flux (1): a virtual synthesis scheme for fragment-based de novo design. *J Chem Inf Model* **2006,** 46, (2), 699-707.
82. Schuller, A.; Suhartono, M.; Fechner, U.; Tanrikulu, Y.; Breitung, S.; Scheffer, U.; Gobel, M. W.; Schneider, G., The concept of template-based de novo design from drug-derived molecular fragments and its application to TAR RNA. *J Comput Aided Mol Des* **2008,** 22, (2), 59-68.