

I. Belda · X. Llorà · E. Giralt

Evolutionary algorithms and de novo peptide design

Published online: 10 May 2005
© Springer-Verlag 2005

Abstract One of the goals of computational chemistry is the automated de novo design of bioactive molecules. Despite significant progress in computational approaches to ligand design and efficient evaluation of binding energy, novel procedures for ligand design are required. Evolutionary computation provides a new approach to this design issue. This paper presents an automated methodology for computer-aided peptide design based on evolutionary algorithms. It provides an automatic tool for peptide de novo design, based on protein *surface patches* defined by user. Regarding the restrictive constraints of this problem a special emphasis has been made on the design of the evolutionary algorithms implemented.

Keywords Ligand design · Peptide design · Surface patch · Evolutionary algorithms · Genetic algorithms

1 Introduction

A goal of computational chemistry – as well as general drug design – is the automated de novo drug design [8, 26]. Despite significant advances in computational approaches to ligand design [1, 52, 2] efficient and reliable methodologies are still required. This is motivated by the ever increasing number of

protein targets in drug design, which are being functionally and structurally characterized. This situation is the result of major advances in experimental methods for structure determination [13, 12] and high-throughput modeling [44].

Peptides are emerging as promising drugs for several illnesses [30]. Turning peptide into real drugs has been difficult – they have poor absorption, distribution, metabolism, excretion (ADME) properties. However, this handicap is soon to disappear thanks to the development of new methods of drug delivery [31], and to the use of derivatives (for example *D* amino acids) considered metabolically more stable [22]. This article aims to design virtual peptide drugs which serve as effective ligands to the target area of the protein – also known as surface patch [24] – previously defined by the researchers. It should be noticed that, in spite of peptide chemical synthesis methods are well advanced, synthesis purification and biophysical studies of the binding of a peptide to a target protein implies several months of experimental work. One application of such peptide drugs could be to act as inhibitors of some pathological functionalities of the target protein [50].

However, such endeavor requires to effectively dealing with huge combinatorial search spaces. Evolutionary computation is mature enough [21] to play a key role in the creation of such tools. In order to perform this combinatorial search we adopted and compared four evolutionary algorithms: a simple genetic algorithm (GA) [23], a Lamarckian genetic algorithm (LGA) [25], a population-based incremental learning (PBIL) [4], and the Bayesian optimization algorithm (BOA) [37]. In order to obtain the fitness of the evolved peptides, computations based on *docking* [43] measures were used – Sect. 3. We have termed this new approach as evolutionary structure-based de novo peptide design algorithm (ENPDA).

The main advantage of this approach is that novel structures, not contained in any database, can be obtained. Thus, algorithms must address two main tasks. First, a competent search must be conducted to explore such a high-dimensional chemical space – evolutionary algorithms are key players in this search. Second, the combinatorial peptide search space – the set of all algorithmically treatable molecules – must be structured into regions of higher and lower quality to allow

I. Belda · E. Giralt
Institut de Recerca Biomèdica de Barcelona, Parc Científic de Barcelona, Universitat de Barcelona. E 08028 Barcelona, Spain
E-mail: ibelda@pcb.ub.es

X. Llorà
Illinois Genetic Algorithms Laboratory, National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
E-mail: xllora@illgal.ge.uiuc.edu

E. Giralt
Departament de Química Orgànica, Universitat de Barcelona. E 08028 Barcelona, Spain

Present address: E. Giralt (✉)
Josep Samitier, 1-5. E 08028 Barcelona, Spain
Tel.: +34-93-4037125
Fax: +34-93-4037126
E-mail: egiralt@pcb.ub.es

the search and prediction of desired properties. Several evolutionary algorithms were adapted to deal with such complex peptide search. We also developed different estimation methods to obtain the fitness of peptides, based on the docking energy – an estimation of the binding energy between a peptide and a protein.

The developed methodology has been tested in several specific domains. The results obtained are encouraging. Some of the new virtually designed ligands have shown better docking energies than those designed using a purely chemical-knowledge based approach by chemical experts in our research group. A detailed description of such heuristics is beyond the scope of this paper, being presented elsewhere [5].

2 Need for new approaches

Modern approaches to drug design are expensive processes where thousands of molecules are synthesized, screened and iteratively optimized. New drug compounds design require seeds or lead compounds to start such optimization process. Several classical approaches exist in the quest for lead compounds [39]. In some cases lead compounds are designed using chemical knowledge [14]; in other cases, huge compound libraries are screened searching for biological activity [28]. Despite the classical approach used, the cost of the search and the time invested may become unfeasible [10].

New approaches for lead compound identification are needed since the classical methods fail in achieving lead compounds in reasonable time and cost. Nowadays, computational horsepower increases each month at the same time that it becomes cheaper. Therefore, promising general-purpose computational approaches are emerging for lead-compound identification and drug-design approaches [45]. Those computational approaches try to screen virtually – or in silico – thousands of compounds, without the constraints of a real chemical synthesis. In recent years, a large number of computational methods have been developed to check – in silico – the biological activity of the compounds screened [9, 29, 41]. However, those methods are not as precise as the in vitro methods. Nevertheless, the small amount of time used to perform the screening, and the affordable computational costs, encourage the use of these methodologies, at least, in the first stages of the drug design process.

New search algorithms, however, are still needed. Evolutionary algorithms are perfect candidates for applications where deterministic or analytic methods fail or become unfeasible [15, 19, 21]. The underlying binding mathematical model optimized uses docking calculations. AutoDock 3.0.5 – the docking program used to perform the evaluation of the candidates – provides a measure of such energies. Unfortunately, nowadays docking is not a perfect simulation of the reality [42]. Also, the search space is too large to be systematically explored. Each evaluation takes more than 30 min of calculations on a 1.60 GHz Pentium IV. Hence, such constraints favor the usage of evolutionary algorithms, and its inherent parallel nature, to steer the search of lead peptides. These assumptions

have also been previously stated by other researchers [11, 17, 40, 47].

Using such evolutionary-based approaches, huge quantities of compounds can be screened in a fast, cheap, and parallel-guided manner. Although several other approaches using evolutionary computation have been proposed [7, 11, 17, 32, 35, 40, 47], our methodology offers the possibility to evolve peptides as lead protein binding compounds. The software developed also offers the possibility to deal with both, natural amino acids, and D amino acids – metabolically more stable, presenting better ADME values.

3 An evolutionary approach to peptide design

In this work we have developed a tool for de novo peptide design based on evolutionary algorithms – ENPDA. The fitness function is based on estimations of the binding energy between the peptide ligand and the target protein. In order to estimate the binding energy, we have used a molecular mechanics approach known as *docking* [34]. Our calculations rely on Autodock 3.0.5 [33]. Moreover we have developed two heuristics which are able to interact with Autodock. The first heuristics is based on a fast, low quality computation. The second performs a higher quality estimation, computationally more expensive.

The first heuristics developed takes about 30 min in a Pentium IV 1.6 GHz to compute the estimation of the binding energy, whereas the second heuristics takes around 2.5 h. Such time frames – although always much shorter than the in vitro evaluations – steered us to use parallel computation facilities in order to screen a reasonable number of compounds. Evolutionary algorithms can be parallelized in straightforward way. Limitations on the population size are only bounded by the computational resources available. However, we maintain the worst case scenario. Such scenario happens when not enough resources are available, forcing us to deal with small populations. A detailed description of these heuristics is beyond the scope of this paper, being explained in detail elsewhere [21].

The worst-case scenario described above forced us to implement special operators against premature convergence in such small populations. The first action taken was in the initialization step, where we did not allow to continue if a certain diversity degree is not reached in the random initialization – usually based on physico-chemical properties of the compound. The second technique introduced to guarantee the diversity in the population was based on the fitness sharing concept [20]. Other methods to manage diversity in evolutionary algorithms can be found at the literature [16, 51].

Sharing is based on the idea that individuals in a particular niche have to share the available resources. The more individuals located in the neighborhood of a certain individual, the more its fitness value is degraded. The neighborhood is defined in terms of a distance measure d_{ij} and specified by the so-called niche radius σ_{share} . If the distance between two individuals is bigger than σ_{share} , no degradation of the fitness

is carried out. The fitness of the individuals is computed as:

$$F'(i) = \frac{F(i)}{\sum_{j=1}^M \phi(d_{i,j})}, \quad (1)$$

where, $F'(i)$ the new scaled fitness, $F(i)$ is the original fitness of the individual i , M is the number of individuals of the population, and $\phi(d_{ij})$ is defined as:

$$\phi(d_{i,j}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{\text{share}}}\right)^\alpha & \text{if } d_{ij} < \sigma_{\text{share}} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

We have fixed α to 1. Thus, the fitness of an individual is not modified if there are no other individuals in its niche. The distance between individuals is measured using an euclidean distance between physico-chemical properties of the amino acid present in the individual being measured. The analyzed properties are *hydrophobicity* and *charge*. For instance, if we want to compute the distance between peptide EWEWWW and peptide RPRAAA, first we compare the hydrophobicity of the amino acids of the first locus, {E, R} – which is approximately the same – then the charge – which is very different since E is charged negatively and R positively. These approach is repeated for all the amino acids in the sequences.

In order to guarantee a proper diversity supply during the initialization step, the diversity measure presented above is computed among the individuals in the initial population. If a given diversity threshold is not met in the initialization, the individuals are dropped, starting the process again.

We have used four evolutionary algorithms, a Darwinist GA [23], a LGA [25], a PBIL [4], and the BOA [37]. The GA is based on the simple genetic algorithm proposed by Holland [23]. The LGA is a simple genetic algorithm enhanced with a local search procedure. This local search is done by means of a (1+1) evolutionary strategy [3]. That is, each gene of an individual undergoes a Gaussian mutation using some evolvable standard deviations. If the fitness of the perturbed individual is better than the fitness of the original one, the original one is replaced by the generated offspring. The process is repeated until a number of iterations is reached. Back described in detail how the standard deviation are evolved [3].

The PBIL [4] and the Bayesian optimization algorithm [37] – as well as other estimation of distribution algorithms [27] – estimate the distribution of the best individuals in the population to guide the breeding of the new population. PBIL estimates the distribution assuming that there are no dependences among variables – in our case the different locus on the amino acid sequences. Selecting the n best individuals, PBIL infers the probability of each possible value for each gene counting the frequency appearance of each gene value in the selected population, already biased toward the good individuals by selection. Then with this estimation of distribution, new individuals are generated.

Bayesian optimization algorithm, instead, infers relationships between genes using Bayesian networks. Such approach is useful when linkage among genes exist in the problem to be solved. Such situation may be highly likely in ligand design, where the different parts of a sequence may collaborate

to minimize the docking energy. The Bayesian network inferred is sampled in order to create new potentially good individuals.

4 Validating the evolutionary algorithms

Before using the algorithms previously described, we performed some tests in order to validate their implementation. We performed two kind of tests. For the first kind of tests, 20 typical fitness functions, retrieved from the literature [48], were optimized using the Darwinist GA and two different configurations of LGA's (LGA1 and LGA2). In the second set of tests, we made scalability tests using three deceptive trap functions [36], LGA, PBIL, and BOA.

In the first set of experiments, the three algorithms (GA, LGA1, and LGA2) performed the same number of evaluations: 37500. GA had 75 individuals per population and 500 generations. LGA1 had five individuals per population in 300 generations and 25 local optimization steps. And LGA2 had ten individuals per generation, along 250 generations and 15 local optimization steps. It is important to note here that such small populations are the results of the computational resources available for the real drug design problems.

For each function, the algorithms were run 100 times. Figures 1, 2, 3, and 4 show the average of those runs. All the parameter values of the functions being optimized can be found in Yao and Liu [48]. The same nomenclature was maintained here.

Regarding the second set of experiments, we tested LGA, BOA, and PBIL in three deceptive trap functions. These functions require linkage learning capabilities in order to solve them quickly, reliable, and accurately [21]. The tested functions were composed of traps $k = 3$ and $k = 5$, and $k = 6$ bipolar [36]. The population parameters of the algorithms were manually adapted in each experiment in order to optimize them to the problem and the number of genes were the solution is represented. Despite the modifications introduced to adapt PBIL and BOA to be able to deal with peptide sequences, both algorithms display the same scalability trends as the one presented by their authors [37]. However is important to note here that the population requirements of BOA made its usage unfeasible for approaching the peptide design endeavor, due to the large population size requirements, although we used in some experiments.

5 Experiments

Once we verified the implementation of the evolutionary algorithms, as well as the heuristics for computing the docking energy, we tested our approach on real problems of drug design. We designed peptides for three proteins of therapeutic interest. They target proteins were: (1) p53 [46] – an important natural tumor suppressor; (2) prolyl oligopeptidase [49] – an enzyme involved in processing some neuropeptides; and (3) DNA gyrase [18] – a protein involved in bacterial

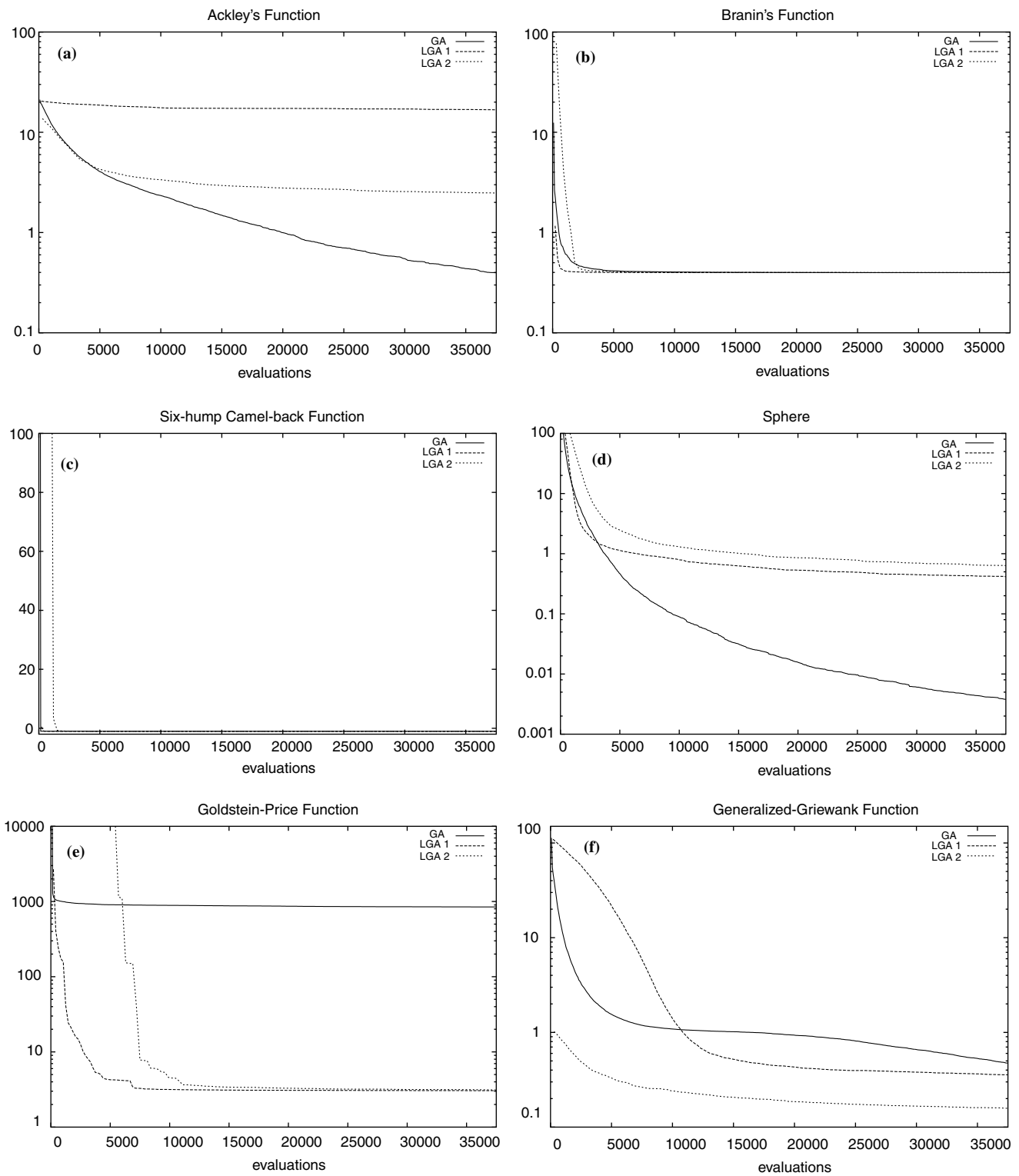


Fig. 1 Evolution of the averaged best individual of 100 runs of each algorithm over the problems: **a** Ackley's Function, **b** Branin Function, **c** Six-hump camel-back function, **d** Sphere model, **e** Goldstein-price function, **f** Generalized Griewank function

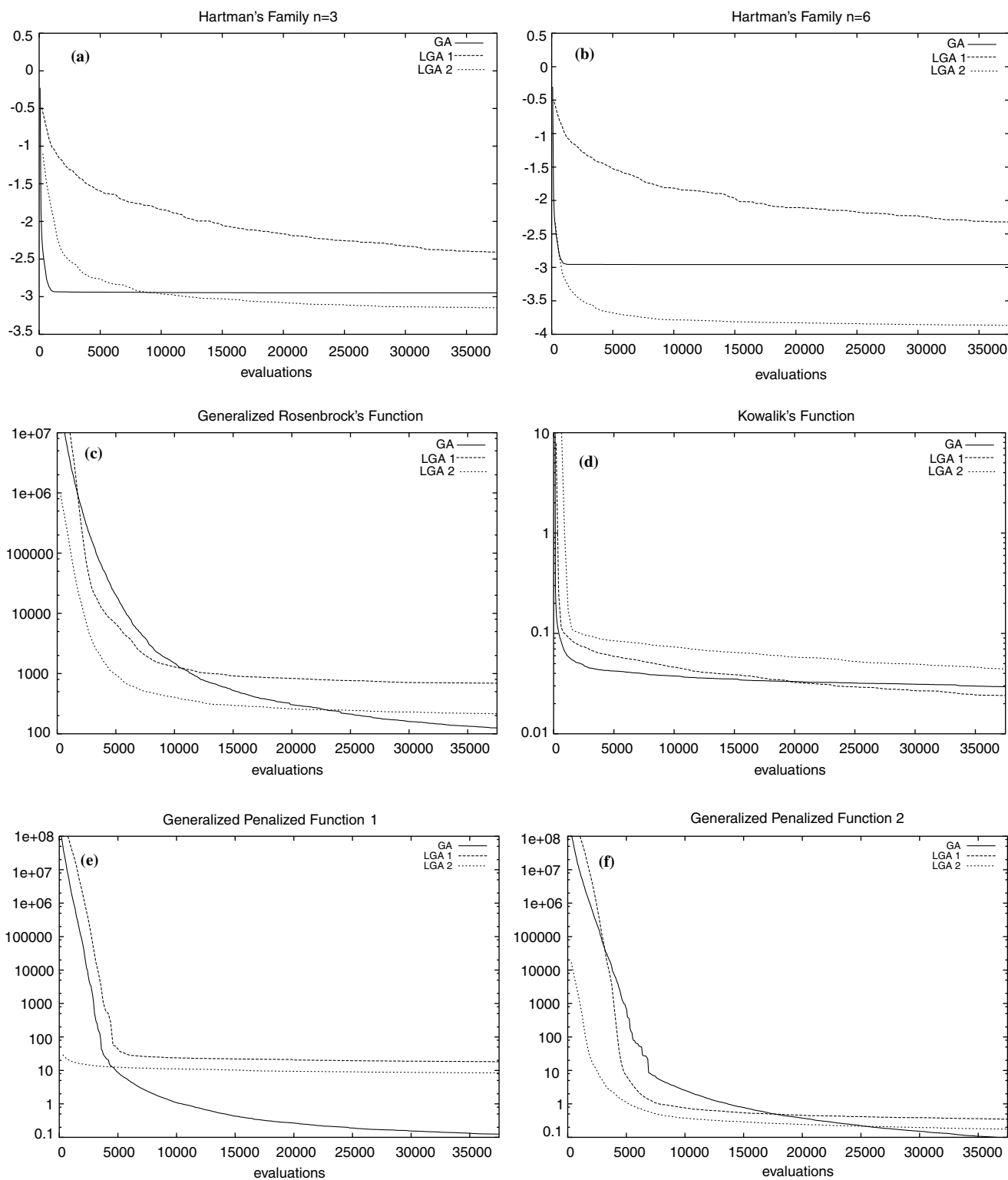


Fig. 2 Evolution of the averaged best individual of 100 runs of each algorithm over the problems: **a** and **b** Hartman's Family, with $n = 3$ and $n = 6$, and **c** generalized Rosenbrock's function, **d** Kowalik's function, **e** and **f** generalized penalized function 1 and 2

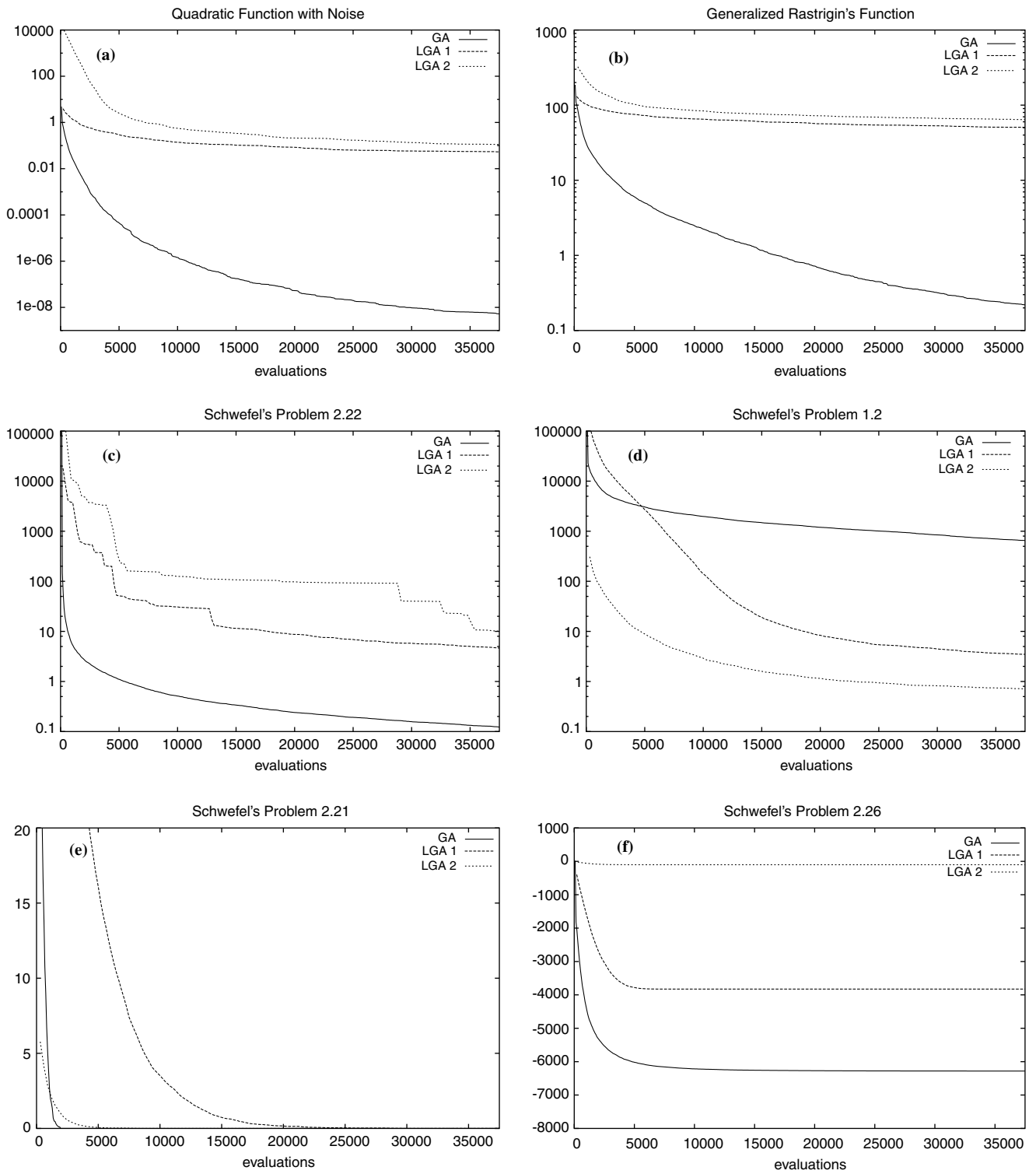


Fig. 3 Evolution of the averaged best individual of 100 runs of each algorithm over the problems: **a** Quadratic function with noise, **b** Generalized Rastrigin's function, and **c, d, e,** and **f**) Schwefel problems 2.22, 1.2, 2.21, and 2.26

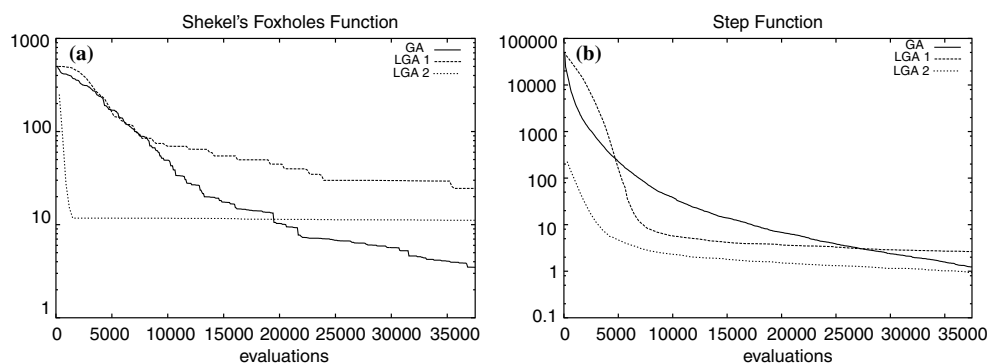


Fig. 4 Evolution of the averaged best individual of 100 runs of each algorithm over the problems: **a** Shekel's Foxholes Function and **b** Step function

replication processes. We designed binding peptides to the surface patches defined by drug design users.

The evolved peptides are currently being synthesized in the wet laboratory in order to test them *in vitro*. Figures 5, 6, and 7 show the best peptides proposed for p53, prolyl oligopeptidase, and DNA gyrase, respectively. We have carried out some *in silico* comparisons between those peptides and others designed by the drug design experts of our research group. The results are shown elsewhere [6].

Population-based incremental learning designed peptides for the protein p53. PBIL used 50 individuals per generation, 20 generations, 20 selected individuals to estimate the distribution, elitism, and a learning rate of 0.5. In the prolyl oligopeptidase problem we used a LGA with 50 individuals, ten generations, two local optimization steps, and elitism. Finally, in the DNA gyrase protein we used BOA with 100 individuals, 40 generations, 50 individuals selected to infer the Bayesian network, and elitism. BOA's population sizing requires thousands of individuals to infer an accurate Bayesian network. Unfortunately, we were not able to work with such population size due to the expensive computational cost of the heuristics and the limited computational resources available. The results presented in Fig. 7 were obtained using an Origin 3400 with eight process elements.

It should be said that for p53 and prolyl oligopeptidase we used the accurate heuristics, whereas for DNA gyrase we used the quick heuristics. The reason was that BOA needed more individuals to properly infer the Bayesian network. It was not possible to work with populations of 100 individuals when using the accurate heuristics – and obtaining results in a reasonable time. For such reason it was necessary to employ the quick heuristic. Please refer to Belda et al. [6] for a detailed explanation of the heuristic adjustment.

6 Discussion

Recent advances in evolutionary computational methods have allowed us to develop a new approach to *in silico* design of peptide ligands. We have implemented four evolutionary algorithms to steer this combinatorial search. Such endeavor requires great amounts of computational resources. Due to

the computational constraints it is not possible to assert, with a reliable statistical confidence, which of them is the better suited for the drug design problem addressed. However, qualitative comparisons can be made at this point, and theoretical analysis have already been started.

Some runs presented in the previous section took almost three weeks on a parallel machine, bounding the amount of data available for a proper comparison. However, we can make some comments about the general behavior of each algorithm and the quality of the evolved peptides in the particular problem domain explored. For instance, LGA performance is better than the Darwinistic GA, since the local search of LGA helps building an appropriate sequence in the small population scenario. Also, we have observed that BOA needs a large population – when compared to the available computational resources – in order to infer a proper Bayesian network [36]. Hence, since the evaluation methods are computationally expensive, we can only use BOA with the quick heuristic. Current efforts done by Pelikan and Sastry [38] in *fitness inheritance* measures may help palliate the number of individuals that need to be evaluated in each generation. PBIL results were competitive with the ones achieved using other techniques. PBIL, even using a naive estimation of distribution method maintained a proper population diversity. Such diversity helped PBIL not to converge prematurely. Finally, when compared in equal conditions and same domain, LGA reached peptide ligands with higher estimated binding affinity.

We tested this new design approach in three different protein domains. Elsewhere [5, 6] we compared *in silico* the evolved peptides and the ones designed by our research group. In general, the peptides designed *in silico* performed better with *in silico* comparisons. When compared with randomly generated peptides, the evolved peptides greatly outperform them, requiring one order of magnitude less evaluations.

7 Conclusions and further work

This paper has presented the development of a ENPDA. This new computational evolutionary tool has proved to be able to design peptides – with a high affinity – to protein surfaces in

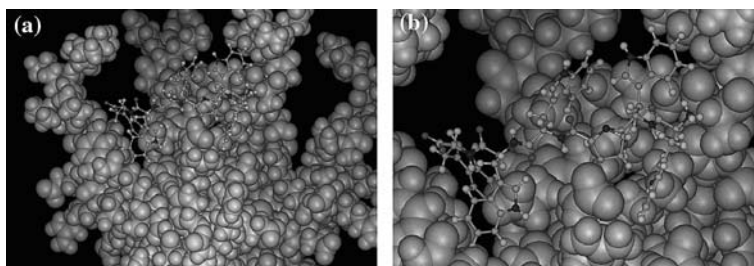


Fig. 5 Proposed peptide, $WWPWWW$, docked in the user-defined surface patch of the protein p53. High resolution color plates and 3D models of the figures displayed above can be found at <http://www-illigal.ge.uiuc.edu/~xllora/Pool/Papers/ SI-Soft-Computing/>

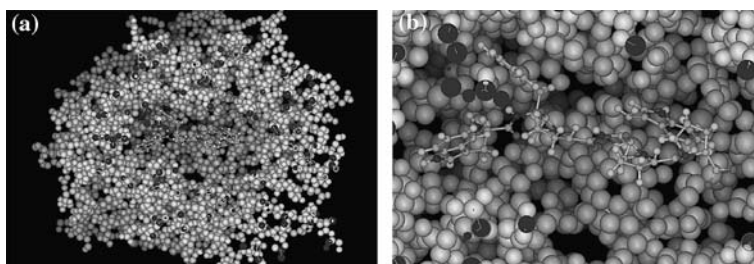


Fig. 6 Proposed peptide, $WWPWPP$, docked in the user-defined surface patch of the protein prolyl oligopeptidase. High resolution color plates and 3D models of the figures displayed above can be found at <http://www-illigal.ge.uiuc.edu/~xllora/Pool/Papers/ SI-Soft-Computing/>

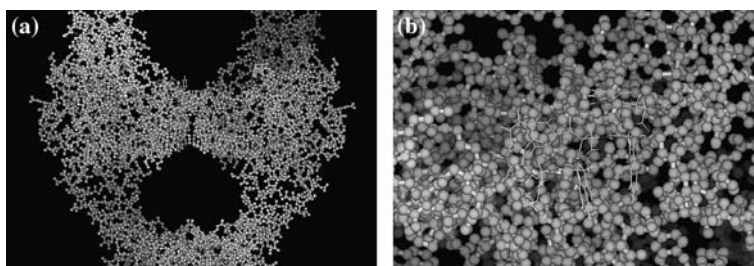


Fig. 7 Proposed peptide, $WWWWS$, docked in the user-defined surface patch of the protein DNA gyrase. High resolution color plates and 3D models of the figures displayed above can be found at <http://www-illigal.ge.uiuc.edu/~xllora/Pool/Papers/ SI-Soft-Computing/>

silico. However, it is important to mention that the success of such approach depends on the quality of the docking mechanism used. Therefore, our future work focuses on improving our evolutionary algorithms, as well as optimizing of the docking procedure for the particular case of flexible peptide ligands. This effort includes the evolution of variable length peptide ligands. Thus, the peptide length will adapt to the size of the surface patch defined by the user over the protein surface.

Another important concept to be developed is a *dynamic quality fitness estimation*. We have developed several ways of estimating the binding energy – heuristics. A further work direction is to develop new binding energy estimation tools with different computational costs and different quality. Cheap and poor quality estimators can be provided by machine learning systems – mainly neural networks and support vector machines – trained to carry out binding energy predictions. The other extreme of the affinity estimators are represented by the chemical synthesis of the compounds and the in vitro evaluation, obtaining accurate methods to estimate binding energies. The idea of dynamic quality fitness estimation, besides looking for increasing the population size without

increasing, is to increase the quality of the evaluation along the run. The ultimate goal is to start with cheap estimators, finishing with the measures obtained in the wet lab – introducing the researchers in the final stages of the design loop.

Acknowledgements We thank Ivan Traus for useful discussions about the computational aspects of this work, and Professor Xavier Vilasis for his mathematical support in the implementation of Bayesian network learning algorithms. The authors would also like to thank the Parc Científic de Barcelona for providing the computational resources.

This work was partially supported by grants from Fundació BBVA, Fundació Marató TV3, Ministerio de Ciencia y Tecnología FEDER (BIO 2002-2301 and EET2001-4813), the Air Force Office of Scientific Research, Air Force Materiel Command, USAF (F49620-03-1-0129), and by the Technology Research, Education, and Commercialization Center (TRECC), at University of Illinois at Urbana-Champaign, administered by the National Center for Supercomputing Applications (NCSA) and funded by the Office of Naval Research (N00014-01-1-0175). The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the Technology Research, Education, and

Commercialization Center, the Office of Naval Research, or the U.S. Government.

References

1. Ajay A, Walters WP, Murko MA (1998) Can we learn to distinguish between "drug-like" and "non-drug-like" molecules? *J Med Chem* 41(18):3314–3324
2. Apostolakis J, Caflish A (1999) Computational ligand design. *Combi Chem High Throughput Screen* 2:91–104
3. Back T (1997) *Evolutionary algorithms in theory and practice*. Oxford University Press, New York
4. Baluja S, Caruana R (1995) Removing the genetics from standard genetic algorithm. In: Prieditis A, Russell S (eds) *Proceedings of the international conference on machine learning*. Morgan Kaufmann, San Francisco, pp 112–128
5. Belda I, Llorà X, Martinell M, Tarragó T, Giralt E (2004) Computer-aided peptide evolution for virtual drug design. In: Deb K et al (ed) *Genetic and evolutionary computation conference-GECCO 2004*. Also available as IlliGAL technical report, No. 2004015. *Lecture Notes in Computer Science* No. 3103, Springer, Berlin Heidelberg New York, pp 321–332
6. Belda I, Llorà X, Martinell M, Tarragó T, Piqueras M-G, Nicolás E, Giralt E Evolutionary algorithms and de novo peptide design. *J Comput Aided Mol Des* (to submitted)
7. Böhm HJ (1996) Computational tools for structure-based ligand design. *Program Biophys Mol Biol* 3:197–210
8. Böhm HJ (1998) 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* 12(4):309–323
9. Böhm HJ, Schneider G (ed) (2000) *Virtual screening for bioactive molecules*. Wiley, New York
10. Bolten BM, DeGregorio T (2002) Trends in development cycles. *Nat Drug Discover* 1:551–555
11. Budin N, Majeux N, Tenette C, Caflisch A (2001) Structure-based ligand design by a build-up approach and genetic algorithm search in conformational space. *J Comput Chem* 22(16):1956–1970
12. Chiva C, Barthe P, Codina A, Gairí M, Molina F, Granier C, Pugnieri M, Inui T, Nishi H, Nishiuchi Y, Kimura T, Sakakibara S, Albericio F, Giralt E (2003) Synthesis and NMR structure of p41icf, a protein inhibitor of human cathepsin L. *J Am Chem Soc* 125(6):1508–1517
13. Codina A, Gairí M, Tarragó T, Viguera AR, Feliz M, Ludevid D, Giralt E (2002) 1h(n), 15n, 13co, 13ca, 13b assignment and secondary structure of a 20 kda a-l-fucosidase from pea using TROSY. *J Biomol NMR* 22:295–296
14. Corey DR (1999) Design and engineering of proteins as therapeutic agents. In Wu-Pong S, Rojanasakul (ed) *Biopharmaceutical Drug Design Development*. Humana Press, Inc.:187–203
15. Davis L (1996) *Handbook of genetic algorithms*. International Thomson Computer Press
16. Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Schoenauer M, Deb K, Rudolph G, Yao X, Lutton E, Merelo JJ, Schwefel H-P (eds) *Proceedings of the parallel problem solving from nature VI conference*, *Lecture Notes in Computer Science* No. 1917, Springer, Berlin Heidelberg New York pp 849–858
17. Dougllet D, Thoreau E, Grassy G (2000) A genetic algorithm for the automated generation of small organic molecules: drug design using an evolutionary algorithm. *J Comput Aided Mol Des* 14:449–466
18. Gellert M (1981) DNA topoisomerases. *Ann Rev Biochem* 50:879–910
19. Goldberg DE (1989) *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, Reading
20. Goldberg DE, Richardson J (1987) Genetic algorithms with sharing for multimodal function optimisation. In: Grefenstette JJ (ed) *Proceedings of the second international conference on genetic algorithms*. Lawrence Erlbaum, pp 41–49
21. Goldberg DE (2002) *The design of innovation: lessons from and for competent genetic algorithms*. Kluwer Dordrecht
22. Haack T, González MJ, Sánchez Y, Giralt E (1997) D-Amino acids in protein de novo design. II. Protein-diastereomerism versus protein-enantiomerism. *Lett Pep Sci* 4:377–386
23. Holland JH (1975) *Adaptation in natural and artificial systems*. MIT, Cambridge
24. Jones S, Thornton JM (1997) Analysis of protein-protein interaction sites using surface patches. *J Mol Biol* 272:121–132
25. Krasnogor N (2002) *Studies on the theory and design space of memetic algorithms*. PhD Thesis, University of the West England, Bristol
26. Kubinyi H (1999) Chance favors the prepared mind. From serendipity to rational drug design. *J Recept Signal Transduct Res* 19:15–39
27. Larrañaga P, Lozano JA (2002) *Estimation of distribution algorithms*. GENA 2. Kluwer, Dordrecht
28. Lebl M (2002) *Combinatorial chemistry: the history and the basics*. In Mei HY, Czarnik AW. *Integrated Drug Discovery Technologies*. Marcel Dekker:395–405
29. Lengauer T, Lemmen C, Rarey M, Zimmermann M (2004) Novel technologies for virtual screening. *Drug Discov Today* 9(1):27–34
30. Loffet A (2002) Peptides as drugs: is there a market? *J Peptide Sci* 8:1–7
31. Malmsten M (2002) *Surfactants and polymers in drug delivery*. Marcel Dekker, New York
32. Mandell A, Selz K, Shlesinger M (2002) Algorithmic design of peptides for binding and/or modulation of the functions of receptors and/or other proteins
33. Morris GM, Goodsell DS, Huey R, Hart WE, Halliday S, Belew R, Olson AJ (2003) *Userguide Autodock 3.0.5*
34. Morris GM, Olson AJ, Goodsel DS (2000) Protein-ligand docking. *Methods Princip Med Chem* 8:31–48
35. Pegg S, Haresco J, Kuntz I (2001) A genetic algorithm for structure-based de novo design. *J Comput Aided Mol Des* 15:911–933
36. Pelikan M, Goldberg DE (1999) BOA: The bayesian optimization algorithm. Technical Report 99003, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign
37. Pelikan M, Goldberg DE, Cantú-Paz E (1999) BOA: the bayesian optimization algorithm. In: *Proceedings of the genetic and evolutionary computation conference-GECCO 1999*. Morgan Kaufmann, San Francisco
38. Pelikan M, Sastry K (2004) Fitness inheritance in the bayesian optimization algorithm. Technical Report 2004009, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign
39. Sastry VH (1995) Strategies for drug design. *Eastern Pharmacist* 38:455
40. Scheider G, Lee M, Stahl M, Schneider P (2000) De novo design of molecular architectures by evolutionary assembly of drug-derived building blocks. *J Comput Aided Mol Des* 14:487–494
41. Seifert MHJ, Wolf K, Vitt D (2003) Virtual high-throughput in silico screening. *Biosilico* 1(4):143–149
42. Shoichet BK, McGovern SL, Wei B, Irwin JJ (2002) Lead discovery using molecular docking. *Curr Opin Chem Biol* 6(4):439–446
43. Smith GR, Sternberg MJE (2002) Prediction of protein-protein interactions by docking methods. *Curr Opin Struct Biol* 12:28–35
44. Thormann M, Pons M (2001) Massive docking of flexible ligands using environmental niches in parallelized genetic algorithms. *J Comput Chem* 22(16):1971–1982
45. Veselovsky AV, Ivanov AS (2003) Strategy of computer-aided drug design. *Curr Drug Targets Infect Disord* 3(1):33–40
46. Vogelstein B, Lane D, Levine AJ (2000) Surfing the p53 network. *Nature* 408:307–310

-
47. Wang R, Gao Y, Lai L (2000) Ligbuilder: a multi-purpose program for structure-based drug design. *J Mol Model* 6:498–516
 48. Yao X, Liu Y (1997) Fast evolution strategies. In: Peter J. Angeline, Robert G. Reynolds, John R. McDonnell, Russ Eberhart (eds) *Evolutionary programming VI*. Springer, Berlin Heidelberg New York, pp 151–161
 49. Yoshimoto T, Fischl M, Orłowski R, Walter R (1978) Post-proline cleaving enzyme and post-proline dipeptidyl aminopeptidase. Comparison of two peptidases with high specificity for proline residues. *J Biol Chem* 253(10):3708–3716
 50. Zeng J (2000) Mini-review: computational structure-based design of inhibitors that target proteins surfaces. *Comb Chem High Throughput Screen* 3:355–362
 51. Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm. Technical Report 103, Gloriastrasse 35, CH-8092 Zurich, Switzerland
 52. Zou X, Sun Y, Kuntz ID (1999) Inclusion of solvation in ligand binding free energy calculations using the generalized-born model. *J Am Chem Soc* 121(35):8033–8043