Evolutionary Computation Applications in Current Bioinformatics

Bing Wang[1,2] and Xiang Zhang[2]
*[1]School of Electrical Engineering & Information, Anhui University of Technology,
Ma'anshan, Anhui 243002,
[2]Department of Chemistry, University of Louisville, KY 40292,
[1]China
[2]USA*

1. Introduction

Over the past few decades' rapid development in genomics, proteomics, metabolomics and other types of omics research, a tremendous amount of data related to molecular biology have been produced. Understanding and exploiting these data is now the key to the success of advancing molecular biology, and this requirement has been stimulated the development and expansion of bioinformatics (Altman, 2007; Jones, et al., 2006). As a fast growing interdisciplinary scientific area, bioinformatics can be defined in several ways, but the emphasis is always on the use of information processing methods to manage, analyze and interpret information from biological data, sequences and structures, with promising applications to biomarker discovery and pharmaceutical design. Important sub-disciplines within the field include (Pal, et al., 2006):

a. Development and implementation of tools and databases that enable efficient access, usage and management of various types of information.
b. Analysis and interpretation of various types of data, including nucleotide and amino acid sequences, protein domains, and protein structures.
c. Development of new algorithms to assess relationships among members of large data sets, such as methods for protein family classification, protein structure and function prediction, gene location and correlation networks.
d. Simulation of biological process using computational models to assist experiment design and implementation, such as protein functional site finding, disease biomarker discovery and drug design.

The post-genome era is characterized by a major expansion in the available biological data. Many important bioinformatics problems are so comprehensive that an exhaustive search of all potential solutions is always challenging, and most likely impossible. Yet another approach of using biologists' current library of standard constructive and approximate algorithms is impractical in terms of time, money, and computational power (Fogel & Corne, 2002). The researchers are then either forced to pose a simpler hypothesis which typically leads to wrong understanding of problem, or to attempt to develop computational algorithms which can search large solution spaces in a reasonable time.

Source: New Achievements in Evolutionary Computation, Book edited by: Peter Korosec,
ISBN 978-953-307-053-7, pp. 318, February 2010, INTECH, Croatia, downloaded from SCIYO.COM

Therefore, evolutionary computation algorithms have been gaining the attention of researchers for solving current bioinformatics problems (Fogel, et al., 2008). As a class of randomized search and optimization techniques which inspired by the process of biological evolution, evolutionary computation can be used to search very large and complex spaces effectively and return good solutions in a rapid fashion. The majority of current implementations of evolutionary computation methods descend from three strongly related but independently developed approaches: genetic algorithms (GAs), evolutionary programming (EG) and evolution strategies (ES). In general, an evolutionary computation method first generates an initial population of solutions. It then repeats a simulated natural evolutionary processes which includes reproduction, mutation, recombination, natural selection and survival of the fittest (Eberbach, 2005). In the last decade, evolutionary computation has experienced a tremendous growth in applications for bioinformatics.

The purpose of this chapter is to provide a survey on the role of evolutionary computation methods, especially GAs, in current bioinformatics tasks. Some important bioinformatics topics, such as sequence analysis, protein structure and function prediction, protein-protein interaction prediction and microarray analysis will be explained here. Conclusions and some future research directions are also discussed in this chapter.

2. Sequence analysis

Since the development of high-throughput techniques in biological experimental methods during the last two decades, the rate of addition of new sequences to the database increases continuously. However, such a collection of sequences does not, by itself, help our understanding of the biology of different organisms. Therefore, the multiple sequence alignment (MSA) is of great interest to biologists since it can provide scientific insight of inferring evolutionary history or discovering conserved regions among closely related protein, ribonucleic acid (RNA) or Deoxyribonucleic acid (DNA) sequences (Pei, 2008; Pirovano & Heringa, 2008; Sobel & Martinez, 1986). In many cases, the MSA assumes the target sequences have an evolutionary relationship by which they share a lineage and are descended from a common ancestor. This assumption makes MSA a fundamental and crucial tool in analysis of sequences which come from the same or a close family. MSA is often used to assess sequence conservation of protein domains, secondary and tertiary structures.

MSA is the process of lining up a set of sequences in the "best possible way". Sum-of-pairs score (SP-score) is usually used to determine the "best possible way" to build an alignment. For k sequences of length at most n via dynamic programming, the SP-score can be solved in $\theta(2^k n^k)$ steps. Unfortunately, this method is almost always time-consuming and unpractical even for a small number of sequences. Moreover, MSA is known as NP-hard (Just, 2001; Wang & Jiang, 1994), and hence finding the best solution is intractable. However, it can be solved by treating the MSA problem as an optimization problem and therefore, the evolutionary computation methods can be applied to search the positions of gap for aligning multiple sequences.

Notredame and Higgins proposed a MSA approach based on genetic algorithm and an associated software package called SAGA (sequence alignment by genetic algorithm) (Notredame & Higgins, 1996). This method uses a genetic algorithm to select from an evolving population the alignment which optimizes the COFFEE Objective Function (OF)

(Notredame, et al., 1998). The OF is a measure of the consistency between the multiple alignments and a library of CLUSTALW pairwise alignments (Thompson, et al., 2002). The approach was tested in a set of 13 cases based mainly on alignments of sequences of known tertiary structure. It was claimed by the authors that this method can find globally optimal multiple alignments or very close to it in a reasonable time frame for completely unaligned sequences. But it also has been mentioned that SAGA is still fairly slow for large test cases (e.g. with >20 or so sequences). This genetic algorithm-based method was extended and improved to a new package, named RAGA (Notredame, et al., 1997), for alignment of two homologous RNA sequences whose secondary structure of one of them is known.

A similar work by Zhang et al. (Zhang & Wong, 1997) was described to align sequences in a two-step method. This first step identifies matches whose input is the sequences to be aligned and output is the matched subunits. The matched subunits are organized on a form called pre-alignment. The pre-alignment is the input of the second step, which identifies mismatches (i.e. deletion, insertions and substitutions). The output of the second step is an alignment. In this work, the task of identifying matches is converted into a search problem using a genetic algorithm. To apply GA, each biomolecular sequence was represented by the subunits. The alignment of sequence was converted from characters space to subunits space and therefore, the computational cost was decreased dramatically.

Other relevant researches of solving multiple sequence alignment using evolutionary computation methods can be found in (Fisz, 2006; Gondro & Kinghorn, 2007). Each of these methods relies on the principle similar to SAGA: a population of multiple alignments evolves by selection, combination and mutation. The main difference between these methods and SAGA are the design of better mutation operators that can improve the efficiency and the accuracy of the algorithms.

3. Protein structure prediction

A protein is a chain of amino acid residues that folds into a specific native tertiary structure under certain physiological conditions. There are 20 amino acids which can be divided into several classes based on their size and other physical and chemical properties. Proteins fold into one or more specific spatial conformations that enable the proteins to perform their biological function. In order to understand the functions of proteins at a molecular level, it is often necessary to determine the three dimensional structure of each protein. Protein structure prediction is, therefore, one of the most important research topics in bioinformatics.

A number of studies using the evolutionary computation method for protein structure prediction have been made in the last decades. As the first attempt to predict protein structure using GAs, Dandekar and Argos used a tetrahedral lattice and structural information was encoded as gene (Dandekar & Argos, 1996; Dandekar & Argos, 1997). Each residue has seven possible conformations encoded by three bits. Each gene therefore is encoded with 3×N bits long, where N is the number of residues. The fitness function contains terms that encouraged strand formation and pairing and penalizes steric clashes and nonglobular structures. The function was parameterized on a set of four helix bundle proteins and on one of the β-structure proteins reproduced. The success of this method relies on the correct pre-assignment of secondary structure, and may introduce bias in the potential towards the experimental structure.

Sun et al. (Sun, et al., 1999) reduced the molecular structure of each protein to its backbone atoms and each side chain was approximated by a single virtual united-atom. A statistical potential of mean force derived from known protein structures was used to assess fitness. A conformation library of peptide fragments containing 2-5 amino acid residues was extracted from known protein structures to construct initial conformations. Fragments were selected from the library based on sequence similarity, which appears that it will introduce a strong bias, particularly for the longer fragments. A root mean square error of 1.66Å on average to the crystal structure can be achieved for melittin, a protein of 26 residues. Similar results for avian pancreatic polypeptide inhibitor and apamin were also obtained.

Another earlier work discussing the reduced three-dimensional lattice protein using genetic algorithm was reported by Unger et al. (Unger & Moult, 1993). In this method, each peptide was considered as only single point units without side chains and represented by three bits to encode five degrees of freedom. All residues are divided into two groups: hydrophobic and hydrophilic. The evaluation function scored -1 for each pair of non-bonded hydrophobic neighbors. The algorithm begins with a population of identical unfolded configurations, and the population size is 200. The string of bond angles along the chain was used for describing a conformation. Each generation begins with a series of K mutations being applied to each individual in the population, where K is encoding length. These mutations are filtered using a Monte Carlo step, and mutations resulted in better energy will be accepted. Cross-over sites were selected randomly. Three types of Montel Carlo (MC) methods were applied for comparing the performance of GA. Test data consisted of a series of ten randomly produced 27 length sequences and ten randomly produced 64 length sequences. Experimental results indicted that GA can find the global minimum for all but one sequence. MC also can find the global minimum for short sequences, but it is not for the longer sequences. Although this work demonstrated the potential advantages of GA-base methods for protein structure prediction, this simple model did not test its applicability to real proteins.

Other investigations on protein structure prediction are available in (Arunachalam, et al., 2006; Contreras-Moreira, et al., 2003; Cooper, et al., 2003). These studies show that GAs is superior to MC and other search methods for protein structure prediction.

4. Protein-protein recognition and docking

Protein-protein recognition represents a fundamental aspect of biological function. Although the protein structures are now routinely determined by experimental methods, it is much more difficult to ascertain the structure of protein complexes. When two molecules are in close proximity, it can be energetically favorable for them to bind together tightly. The molecular docking study focuses on the prediction of energy and physical configuration of binding between two molecules. The success of docking and the resulting docked configuration can refine the design of drug molecules. Methods for protein docking, such as DOCK (Kuntz, et al., 1982), FLOG (Miller, et al., 1994), and GOLD (Jones, et al., 1997), are widely used in drug-discovery programs. The principal techniques currently available for protein docking are: molecular dynamics, MC method, genetic algorithms, fragment-based methods, complementarity methods and distance geometry. Here, we will focus on the EC-based protein docking approaches.

GOLD (Genetic Optimisation for Ligand Docking) is a docking program that uses a GA search strategy and includes rotational flexibility for selected receptor hydrogens along with full ligand flexibility (Jones, et al., 1997). For searching the space of available binding modes efficiently, hydrogen bond motifs have been directly encoded into the GA. The fitness function is the sum of a hydrogen bond term, a 4-8 inter-molecular dispersion potential and a 6-12 intra-molecular dispersion potential for the internal energy of the ligand. Each complex was run using an initial population of 500 individuals into five sub-populations, and migration of individual chromosomes between sub-populations was permitted. The GOLD validation test set is one of the most comprehensive docking methods. It comprises of 100 different protein complexes. This program achieved a 71% of success rate based primarily on a visual inspection of the docked structures. An extension to GOLD can be found in another work (Verdonk, et al., 2003) which included an addition of hydrophobic fitting points used in the least squares fitting algorithm to generate the ligand orientation.

Gardinaer et al. (Gardiner, et al., 2001) described a GA for protein-protein docking method, in which the proteins were represented by dot surfaces calculated using the Connolly program (Connolly, 1986). The GA was used to move the surfaces of one protein relative to the other to locate the area of greatest surface to complementarity between the two. Surface dots were deemed complementary if their normals are opposed, their Connolly shape type is complementary, and their hydrogen bonding or hydrophobic potential is fulfilled. For a possible orientation of the query with respect to the target, the number of matching dots and the number of clashes between query dots and target interior points were counted. If any dots matched, penalty was determined by the number of clashes; otherwise, penalty was set as a very big value (100,000 in the paper). The fitness function was then given by the number of matches subtracted penalty. The algorithm was tested on 34 large protein-protein complexes where one or both proteins had been crystallized separately. Parameters were established for 30 of the complexes that have at least one near-native solution ranked in the top 100.

AutoDock software (Goodsell, et al., 1996) uses a genetic algorithm as a global optimizer combined with energy minimization as a local search method. In this implementation, the ligand is flexible and the receptor is rigid. The ligand-receptor was represented as a grid. The genetic algorithm uses two point crossover and mutation operators. The fitness function comprises five terms: a directional 12-10 hydrogen bond term; a coulombic electrostatic potential; a term proportional to the number of sp^3 bonds in the ligand to represent unfavourable entropy of ligand binding due to the restriction of conformational degrees of freedom; and a desolvation term. This scoring function is based loosely around the AMBER force field from which protein and ligand parameters are taken. The desolvation term is an inter-molecular pairwise summation combining an empirical desolvation weight for ligand carbon atoms, and a pre-calculated volume term for the protein grid. Each of the five terms are weighted using an empirical scaling factor determined using linear regression analysis from a set o 30 protein-ligand complexes with known binding constants. Now the software has been updated to version 4.0.

A number of other investigations can be found in (Gardiner, et al., 2001; Gardiner, et al., 2003; Kang, et al., 2009; Po & Laine, 2008).

5. Conclusions

This chapter provides an overview of some bioinformatics tasks and the relevance of the evolutionary computation methods, especially GAs. There are two advantages of GA-based approaches. One is that GAs are easier to run in parallel than single trajectory search procedures, and therefore allow groups of processors to be utilized for a search. The other is that GAs appear to be more efficient in finding acceptable solutions than other semi-random move methods such as MC (Pedersen & Moult, 1996).

Although the current GA-based methods are very useful and can produce elegant solutions for bioinformatics tasks, there are some general characteristics that might limit the effectiveness of GAs. First, the basic selection, crossover, and mutation operators are common to all applications. Second, a GA requires extensive experimentation for the specification of several parameters so that appropriate values can be identified. Third, GAs involves a large degree of randomness and different runs may produce different results. So it is necessary to incorporate problem specific domain knowledge into GAs to reduce randomness and computational time and current research is going on in this direction also.

However, as an optimization algorithm and an effective searching tool, GAs can be used in other bioinformatics tasks, such as gene expression and microarray data, gene regulatory network identification, construction of phylogenetic trees, protein functional site prediction, characterization of metabolic pathways, and so on.

6. Acknowledgement

This work has been partially supported by National Science Foundation of China (project No. 60803107) and the startup fund of University of Louisville.

7. References

Altman, R.B. (2007) Current progress in bioinformatics 2007, Brief Bioinform, 8, 277-278.

Arunachalam, J., Kanagasabai, V. and Gautham, N. (2006) Protein structure prediction using mutually orthogonal Latin squares and a genetic algorithm, Biochem Biophys Res Commun, 342, 424-433.

Connolly, M.L. (1986) Shape complementarity at the hemoglobin alpha 1 beta 1 subunit interface, Biopolymers, 25, 1229-1247.

Contreras-Moreira, B., Fitzjohn, P.W., Offman, M., Smith, G.R. and Bates, P.A. (2003) Novel use of a genetic algorithm for protein structure prediction: searching template and sequence alignment space, Proteins, 53 Suppl 6, 424-429.

Cooper, L.R., Corne, D.W. and Crabbe, M.J. (2003) Use of a novel Hill-climbing genetic algorithm in protein folding simulations, Comput Biol Chem, 27, 575-580.

Dandekar, T. and Argos, P. (1996) Ab initio tertiary-fold prediction of helical and non-helical protein chains using a genetic algorithm, Int J Biol Macromol, 18, 1-4.

Dandekar, T. and Argos, P. (1997) Applying experimental data to protein fold prediction with the genetic algorithm, Protein Eng, 10, 877-893.

Eberbach, E. (2005) Toward a theory of evolutionary computation, Biosystems, 82, 1-19.

Fisz, J.J. (2006) Combined genetic algorithm and multiple linear regression (GA-MLR) optimizer: Application to multi-exponential fluorescence decay surface, J Phys Chem A, 110, 12977-12985.

Fogel, G. and Corne, D. (2002) Evolutionary computation in bioinformatics. Morgan Kaufmann ; Elsevier Science, San Francisco, Calif. Oxford.

Fogel, G.B., Porto, V.W., Varga, G., Dow, E.R., Craven, A.M., Powers, D.M., Harlow, H.B., Su, E.W., Onyia, J.E. and Su, C. (2008) Evolutionary computation for discovery of composite transcription factor binding sites, Nucleic Acids Res, 36, e142.

Gardiner, E.J., Willett, P. and Artymiuk, P.J. (2001) Protein docking using a genetic algorithm, Proteins, 44, 44-56.

Gardiner, E.J., Willett, P. and Artymiuk, P.J. (2003) GAPDOCK: a Genetic Algorithm Approach to Protein Docking in CAPRI round 1, Proteins, 52, 10-14.

Gondro, C. and Kinghorn, B.P. (2007) A simple genetic algorithm for multiple sequence alignment, Genet Mol Res, 6, 964-982.

Goodsell, D.S., Morris, G.M. and Olson, A.J. (1996) Automated docking of flexible ligands: applications of AutoDock, J Mol Recognit, 9, 1-5.

Jones, D.T., Sternberg, M.J. and Thornton, J.M. (2006) Introduction. Bioinformatics: from molecules to systems, Philos Trans R Soc Lond B Biol Sci, 361, 389-391.

Jones, G., Willett, P., Glen, R.C., Leach, A.R. and Taylor, R. (1997) Development and validation of a genetic algorithm for flexible docking, J Mol Biol, 267, 727-748.

Just, W. (2001) Computational complexity of multiple sequence alignment with SP-score, J Comput Biol, 8, 615-623.

Kang, L., Li, H., Jiang, H. and Wang, X. (2009) An improved adaptive genetic algorithm for protein-ligand docking, J Comput Aided Mol Des, 23, 1-12.

Kuntz, I.D., Blaney, J.M., Oatley, S.J., Langridge, R. and Ferrin, T.E. (1982) A geometric approach to macromolecule-ligand interactions, J Mol Biol, 161, 269-288.

Miller, M.D., Kearsley, S.K., Underwood, D.J. and Sheridan, R.P. (1994) FLOG: a system to select 'quasi-flexible' ligands complementary to a receptor of known three-dimensional structure, J Comput Aided Mol Des, 8, 153-174.

Notredame, C. and Higgins, D.G. (1996) SAGA: sequence alignment by genetic algorithm, Nucleic Acids Res, 24, 1515-1524.

Notredame, C., Holm, L. and Higgins, D.G. (1998) COFFEE: an objective function for multiple sequence alignments, Bioinformatics, 14, 407-422.

Notredame, C., O'Brien, E.A. and Higgins, D.G. (1997) RAGA: RNA sequence alignment by genetic algorithm, Nucleic Acids Res, 25, 4570-4580.

Pal, S., Bandyopadhyay, S. and Ray, S.S. (2006) Evolutionary computation in bioinformatics: a review, IEEE/ACM Trans on Systems, Man, and Cybernetics - Part C, 36, 601-615.

Pedersen, J.T. and Moult, J. (1996) Genetic algorithms for protein structure prediction, Curr Opin Struct Biol, 6, 227-231.

Pei, J. (2008) Multiple protein sequence alignment, Curr Opin Struct Biol, 18, 382-386.

Pirovano, W. and Heringa, J. (2008) Multiple sequence alignment, Methods Mol Biol, 452, 143-161.

Po, M.J. and Laine, A.F. (2008) Leveraging genetic algorithm and neural network in automated protein crystal recognition, Conf Proc IEEE Eng Med Biol Soc, 2008, 1926-1929.

Sobel, E. and Martinez, H.M. (1986) A multiple sequence alignment program, Nucleic Acids Res, 14, 363-374.

Sun, Z., Xia, X., Guo, Q. and Xu, D. (1999) Protein structure prediction in a 210-type lattice model: parameter optimization in the genetic algorithm using orthogonal array, J Protein Chem, 18, 39-46.

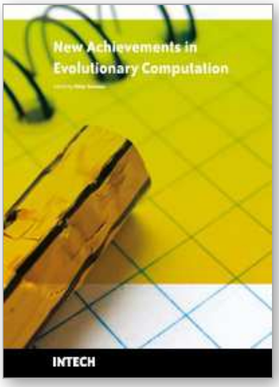
Thompson, J.D., Gibson, T.J. and Higgins, D.G. (2002) Multiple sequence alignment using ClustalW and ClustalX, Curr Protoc Bioinformatics, Chapter 2, Unit 2 3.

Unger, R. and Moult, J. (1993) Genetic algorithms for protein folding simulations, J Mol Biol, 231, 75-81.

Verdonk, M.L., Cole, J.C., Hartshorn, M.J., Murray, C.W. and Taylor, R.D. (2003) Improved protein-ligand docking using GOLD, Proteins, 52, 609-623.

Wang, L. and Jiang, T. (1994) On the complexity of multiple sequence alignment, J Comput Biol, 1, 337-348.

Zhang, C. and Wong, A.K. (1997) A genetic algorithm for multiple molecular sequence alignment, Comput Appl Biosci, 13, 565-581.



New Achievements in Evolutionary Computation
Edited by Peter Korosec

ISBN 978-953-307-053-7
Hard cover, 318 pages
Publisher InTech
Published online 01, February, 2010
Published in print edition February, 2010

Evolutionary computation has been widely used in computer science for decades. Even though it started as far back as the 1960s with simulated evolution, the subject is still evolving. During this time, new metaheuristic optimization approaches, like evolutionary algorithms, genetic algorithms, swarm intelligence, etc., were being developed and new fields of usage in artificial intelligence, machine learning, combinatorial and numerical optimization, etc., were being explored. However, even with so much work done, novel research into new techniques and new areas of usage is far from over. This book presents some new theoretical as well as practical aspects of evolutionary computation. This book will be of great value to undergraduates, graduate students, researchers in computer science, and anyone else with an interest in learning about the latest developments in evolutionary computation.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821