Metaheurísticas aplicadas a la resolución del problema de ensamblado de fragmentos de ADN

Gabriela Minetti
Laboratorio de Investigación en Sistemas Inteligentes
Universidad Nacional de La Pampa
Argentina
minettig@ing.unlpam.edu.ar

Guillermo Leguizamón
Universidad Nacional de San Luis
Argentina
legui@unsl.edu.ar

Enrique Alba - Gabriel Luque
Dpto. de Lenguajes y Ciencias de la Computación
Universidad de Málaga
Spain
{eat,gabriel}@lcc.uma.es

1. Introducción

En las últimas décadas, importantes avances en la biología molecular y en la tecnología genética subyacente, han provocado un crecimiento inconmensurable en el volumen y variedad de información generada por una vasta comunidad científica. Por ejemplo, el secuenciamiento genético (*genome and proteome sequences*, en Inglés), la identificación de genes (*gene identification*), la identificación del perfil de la expresión genética (*gene expression profiling*) entre otras áreas genéticas, marcaron y marcan la necesidad de involucrar el conocimento de expertos pertenecientes a otras ciencias tales como las matemáticas, las ciencias de la computación, la física y la biología, a los efectos obtener mejores resultados y en menos tiempo. La bioinformática es, entonces, un campo interdisciplinar dedicado a desarrollar técnicas que permitan: analizar secuencias genéticas, identificar y predecir estructuras moleculares, extraer caracterísitcas de microarrays de datos, etc.

El conjunto de técnicas en bioinformática utilizadas en las distintas áreas de la biología es extenso y de componentes heterogéneos. Podemos distinguir dos grandes grupos de técnicas algorítmicas. Uno de ellos está conformado por algoritmos especialmente diseñados para un uso

bioinformático específico. Por otro lado, el segundo subconjunto está formado por metaheurísticas y técnicas modernas; las cuales pueden aplicarse en diferentes campos.

En el primer caso los algoritmos han sido diseñados y modelados específicamente para manejar información biológica. Es el caso de herramientas como: CAP3 [1], CLUSTALW-pairwise y CLUSTAL-MSA [2], FASTA [3, 4], BLAST y sus variantes [5, 6], Vector de momentos de composición [7], Modelado de la dinámica molecular [8], IRAP [9], entre otras.

En el segundo caso, las técnicas algorítmicas pertenecientes a la inteligencia computacional, han sido intensamente usadas en difirentes campos y se han adaptado a muchos usos bioinformáticos. La razón es que pueden resolver problemas de muy alta dimensión con fuertes restricciones de manera eficiente. Ejemplos de esto son: las redes neuronales artificiales (*Artificial Neural Networks, ANNs*) [10, 11, 12, 13, 14], los algoritmos evolutivos (*Evolutionary Algorithms, EAs*) [15, 16, 17, 18, 19, 20], los métodos de Montecarlo guiado (*MC*) [21, 22, 23, 24], los Optimización Basada en el Comportamiento de Colonias de Hormigas (*Ant Colony Optimization, ACO*) [25]. Todas ellas son metaheurísticas (resolutores con estructura interna y búsqueda no exhaustiva guiada) que se usan en diferentes campos tales como: sistemas industriales, diseño en ingeniería, logística, comunicaciones, etc. Aunque este tipo de algoritmos tiene un uso más generalizado, resultan eficaces y eficientes cuando la complejidad del problema y su respectivo espacio de soluciones son extensos o crecen continuamente. Esta es una característica muy importante y extremadamente necesaria a la hora de manipular enormes cantidades de información biológica. Además, dichas técnicas no necesitan contar con datos precisos y completos para obtener más información (o soluciones) y de muy buena calidad. Por otro lado, estas técnicas inteligentes no son exhaustivas ni determinísticas. Estas características reducen considerablemente el esfuerzo computacional empleado y pueden producir resultados múltiples para una misma situación. Además estas metaheurísticas presentan otra ventaja significativa en el area de la bioinformática ya que resultan sumamente eficientes en la resolución de problemas de optimización combinatoria como por ejemplo el problema de alineamiento de secuencias (*Sequence Alignment Problem*), el problema de ensamblado de fragmentos (*Fragment Assembly Problem, FAP*), el problema de análisis protéico (*Protein Folding Problem*), etc. Tales características y ventajas son difíciles de encontrar o de incorporar en las técnicas del primer grupo.

Particularmente nuestra línea de trabajo abarca el desarrollo de metaheurísticas para resolver el problema de ensamblado de fragmentos. Para enunciar este problema, es necesario definir previamente el proceso de secuenciamiento [26]: 1°) El ADN is dividido aleatoriamente en millones de fragmentos, 2°) dichos fragmentos son leídos por una máquina de secuenciamiento de ADN y 3°) un ensamblador une los fragmentos leídos que se superponen, reconstruyendo la secuencia original. Ésta, es una técnica general denominada *shotgun sequencing* e introducida por Sanger et al. [27]. El ensamblado de fragmentos de ADN se divide en tres fases diferentes: fase de superposición (encuentra los fragmentos superpuestos), fase de distribución (encuentra el orden de los fragmentos basado en el puntaje de similitud computado) y por último, la fase de consenso (deriva la secuencia de ADN a partir de la distribución anterior).

El objetivo de este artículo es presentar los desarrollos realizados hasta el momento para resolver el FAP y los trabajos futuros.

2. Desarrollos realizados

Hemos desarrollado diferentes versiones de 2 algoritmos metaheurísticos: Variable Neighborhood Search (VNS) y GAs. VNS es una metaheurística presenteda por Hansen *et al.* en [28], que no sigue una trayectoria sino que explora diferentes vecindarios predefinidos de la solución actual usando un método de búsqueda local. Los GAs [29], son una clase especial de Algoritmos Evolutivos. Un algoritmo genético mantiene una población, de soluciones candidatas (individuo), que evoluciona generación tras generación; siendo la selección, la recombinación y la mutación los principales operadores usados para modificar las características de dichas soluciones.

En ambos casos, hemos usado una representación por permutación para codificar las soluciones y dos funciones de optimización, una de ellas mide la calidad de la solución (maximización) [15] y la otra mide el número de contigs (minimización) [30]. En tanto que el método de búsqueda local utilizado en VNS ha sido el algoritmo 2-opt [31]. Mientras que en las versiones de los algoritmos genéticos [32] hemos implementado diferentes operadores de recombinación: Edge Recombination, Cycle Crossover, Order Crossover y Partial Mapped Crossover. Todos ellos son operadores diseñados para utilizar soluciones reprsentadas por permutación. Además las poblaciones iniciales han sido generadas utilizando tres diferentes estrategias de inicialización: aleatoria, por medio del algoritmo 2-opt y a través de una heúristica que hemos desarrollado especialmente para este problema.

En general estas técnicas metaheurísticas han resuelto de forma óptima instancias de poca y mediana complejidad.

3. Trabajo Futuros

Una de las líneas a seguir es la hibridación de los GAs con los métodos clásicos de resolución del FAP. Por otro lado se estudiará la incorporación de la heurística, por nosotros desarrollada, en un algoritmo de optimización basado en el comportamiento de una colonia de hormigas (ACO) y así resolver el problema antes mencionado.

Referencias

[1] W. Huang and A. Madan. CAP3: A DNA Sequence Assembly Program. *Genome Research*, 9(9):868–877, 1999.

[2] J.D. Thompson, D.G. Higgins, and T.J. Gibson. CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22:4673–4680, 1994.

[3] W.R. Pearson. Comparison of methods for searching protein sequence databases. *Protein Sci.*, 4:1145–1160, 1995.

[4] W.R. Pearson and D.J Lipman. Improved tools for biological sequence analysis. *Proc. Natl Acad. Sci. USA,85*, pages 2444–2448, 1998.

[5] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, (1990):403–410, 1990.

[6] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, (25):3398–3402, 1997.

[7] J. Ruan, K. Wang, J. Yang, L.A. Kurgan, and K. Cios. Highly accurate and consistent method for prediction of helix and strand content from primary protein sequences. *Artificial Intelligence in Medicine*, (35):19–35, 2005.

[8] D.M. York, T.A. Darden, L.G. Pedersen, and M.W. Anderson. Molecular dynamics simulation of hiv-1 protease in a crystalline environment and in solution. *Biochemestry*, 32(6):1143–1153, 1993.

[9] J. Chen, W. Hsu, M. Lee, and S.Ñg. Systematic assessment of high-throughput experimental data for reliable protein interactions using network topology. *16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'04)*, pages 368–372, 2004.

[10] G.G. Towell and J.W. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, (70):119–165, 1994.

[11] N.I. Larsen, J. Engelbrecht, and S. Brunak. Analysis of eukaryotic promoter sequences reveals a systematically occurring CT-signal. *Nucleic Acids Research*, (23):1223–1230, 1995.

[12] Y.D. Cai, H. Yu, and K.C. Chou. Artificial neural network method for predicting HIV protease cleavage sites in protein. *Journal of Protein Chemical*, 17:607–615, 1998.

[13] H.C. Wang, J. Dopazo, L.G. de la Fraga, Y.P. Zhu, and J.M. Carazo. Self-organizing tree-growing network for the classification of protein sequences. *Protein Sci*, 7:2613–2622, 1998.

[14] A. Mills, B. Yurke, and P. Platzman. Error-tolerant massive DNA neural-network computation. In *4th Int. Meeting on DNA-Based Computing*, Baltimore, Penns., 1998.

[15] R. Parsons, S. Forrest, and C. Burks. Genetic Algorithms, Operators, and DNA Fragment Assembly, 1993.

[16] C.Ñotredame and D.G. Higgins. SAGA: sequence alignment by genetic algorithm. *Nucleic Acids Research*, 24(8):1515–1524, 1996.

[17] C.Ñotredame, L. Holm, and D.G. Higgins. COFFEE: an objective function for multiple sequence alignments. *Bioinformatics*, 14(5):407–422, 1998.

[18] K. Kim and C.K. Mohan. Parallel hierarchical adaptive genetic algorithm for fragment assembly. In IEEE, editor, *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, volume 1, pages 600–607, 2003.

[19] L. Li and S. Khuri. A Comparison of DNA Fragment Assembly Algorithms. In *Proceedings of the 2004 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences*, pages 329–335, Las Vegas, 2004.

[20] G. Luque, E. Alba Torres, and S. Khuri. *Parallel Algorithms for Bioinformatics*, chapter Chapter 16: Assembling DNA Fragments with a Distributed Genetic Algorithm. Wiley, New York, 2005.

[21] A.P. Lyubartsev, A.A. Martsinovski, and P.N. Vorontosov-Veñuaminov. New Approach to Monte Carlo Calculation of the Free Energy: Method of Expanded Ensembles. *Journal of Chemical Physics*, 96:1776–1783, 1992.

[22] E. Marinari and G. Parisi. Simulated Tempering: A new Monte Carlo Scheme. *Europhys. Lett.*, 19:451–458, 1992.

[23] G. Churchill, C. Burks, M. Eggert, M.L. Engle, and M.S. Waterman. Assembling DNA Sequence Fragments by Shuffling and Simulated Annealing. Technical Report LA-UR-93-2287, Los Alamos National Laboratory, Los Alamos, NM, 1993.

[24] C. Burks, R.J. Parsons, and M.L. Engle. Integration of competing ancillary assertions in genome assembly. In R. Altman, D. Brutlag, P. Karp, R. Lathrop, and D. Searls, editors, *Proceedings Second International Conference on Intelligent Systems for Molecular Biology*, pages 62–69, Menlo Park, CA, 1994. AAA1 Press.

[25] P. Meksangsouy and N. Chaiyaratana. DNA fragment assembly using an ant colony system algorithm. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03*, volume 3, pages 1756– 1763. IEEE. ISBN: 0-7803-7804-0, 2003.

[26] M. Pop, S.L. Salzberg, and M. Shumway. Genome sequence assembly: Algorithms and issues. *Computer*, 35(7):47–54, 2002.

[27] F. Sanger, A.R. Coulson, G.F. Hong, D.F. Hill, and G.B. Petersen. Nucleotide Sequence of Bacteriophage Lambda DNA. *Journal of Molecular Biology*, 162(4):729–773, 1982.

[28] P. Hansen, N. Mladenovic, and J.A. Moreno Pérez. Variable neighbourhood search. *Revista Iberoamericana de Inteligencia Artificial*, (19):77–92, 2003. ISSN: 1137-3601.

[29] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The MIT Press, Cambridge, Massachusetts, first edition, 1975.

[30] E. Alba and G. Luque. A New Local Search Algorithm for the DNA Fragment Assembly Problem. In *Evolutionary Computation in Combinatorial Optimization, EvoCOP'07*, volume 4446 of *Lecture Notes in Computer Science*, pages 1–12, Valencia, Spain, 2007. Springer.

[31] E. Alba Torres, G. Luque, and G. Minetti. Variable neighborhood search for solving the dna fragment assembly problem. In *Anales del XIII Congreso Argentino de Ciencias de la Computación (CACIC)*, pages 1359 – 1370, Corrientes y Resistencia, Argentina, October 2007.

[32] E. Alba Torres, G. Luque, and G. Minetti. Seeding strategies and recombination operators for solving the dna fragment assembly problem. *INFORMATION PROCESSING LETTERS ELSEVIER. En etapa de evaluación*, 2008.