3D Protein Substructure Identification and Nearest Neighbor Search

Submitted to
Department of Computer Science
in Partial Fulfillment of the Requirements for the Degree of Master of Philosophy

by

YANG Yong

August 2010
Abstract

Identifying the location of binding sites on proteins is of fundamental importance for a wide range of applications including molecular docking, de novo drug design, structure identification and comparison of functional sites. Structural genomics projects are beginning to produce protein structures with unknown functions. Therefore, efficient methods are required if all these structures are to be properly annotated. Identifying the interface between two interacting proteins provides important clues to the function of a protein and can reduce the search space required by docking algorithms to predict the structures of complexes. Here we develop a software package for identifying similar three-dimensional (3D) protein substructures via direct comparison of 3D structures.

We develop an efficient heuristic algorithm for finding protein 3D substructures in a 3D protein structure that are similar to a given 3D protein substructure. This algorithm can also be used for searching a database of 3D protein structures. We also design an algorithm for finding similar 3D substructures from two given 3D protein structures. Our approach is to directly compare the substructures by computing a rigid transformation such that the distance between a pair of corresponding points in the matched substructures is bounded by a given value $d$. We propose an efficient local search heuristic approach for finding an approximate rigid transformation. We implement the algorithms and produce a software package that works well in practice. Experiments show that our software package can find pairs of similar 3D substructures in reasonable time. We also provide graphical interface to allow users to view the pairs of 3D protein substructures from different angles.

Finding the closest objective for a query in a database is a classical problem in com-
puter science. For some modern biological applications, computing the similarity between two objects might be very time consuming. For example, it takes a long time to compute the edit distance between two whole chromosomes and the alignment cost of two 3D protein structures. So we study the nearest neighbor search problem in metric space, where the pair-wise distance between two objects in the database is known and we want to minimize the number of distances computed on-line between the query and objects in the database in order to find the closest object. We have designed two randomized approaches for indexing metric space databases, where objects are purely described by their distances with each other. Analysis and experiments show that our approaches need to compute $O(\log n)$ objects in order to find the closest object, where $n$ is the number of objects in the database.
Contents

Abstract i

Acknowledgement iii

List of Publications vii

List of Tables viii

List of Figures ix

List of Abbreviations xi

1 Finding 3D protein substructures in a 3D protein structure 1

1.1 Motivation 1

1.2 Results 2

1.3 Availability 2

1.4 Introduction 2

1.5 Methods 5

1.5.1 The formal definitions of the problems 5

1.5.2 The algorithm for single substructure search 7

1.5.2.1 A rigid transformation 7

1.5.2.2 $\epsilon$-rigid transformation 8

1.5.2.3 The algorithm for rotation 9

1.5.2.4 A new efficient local search heuristic algorithm for finding a good $\epsilon$-rigid transformation 9

1.5.2.5 The whole algorithm 10
1.5.3 The algorithm for all pairs of substructures .......................... 11
1.5.4 Reducing the running time .............................................. 13

1.6 Implementation ................................................................. 13

1.7 RESULTS ........................................................................... 15
1.7.1 Running Time and Accuracy ............................................. 15
1.7.2 Case Studies ................................................................. 17
  1.7.2.1 Case 1 (Selfloop) ...................................................... 17
  1.7.2.2 Case 2 (Selfloop) ...................................................... 18
  1.7.2.3 Case 3 (self-chain) .................................................. 19
  1.7.2.4 Case 4 (two different proteins) ................................. 19
  1.7.2.5 Substructure Search: Case study 1 ............................. 20
  1.7.2.6 Substructure Search: Case study 2 ......................... 22

1.8 Conclusion ................................................................. 23

2 Nearest Neighbor Search ...................................................... 28
  2.1 Introduction .................................................................. 28
  2.2 Preliminary ................................................................. 30
  2.3 The Basic Approach .......................................................... 32
    2.3.1 Experiments on the Number of Samples Required. ......... 37
  2.4 The Inside-Out Algorithm ............................................. 40
  2.5 Reducing the Space ....................................................... 43

3 Conclusions and Future Work ........................................... 51
  3.1 Conclusions ................................................................. 51
  3.2 Future Work ................................................................. 52
List of Publications

1. Fei Guo, Lusheng Wang, Yong Yang. Efficient Algorithms for 3D Protein Substructure Identification. The 4th International Conference on Bioinformatics and Biomedical Engineering (iCBBE 2010), Chengdu, China, 2010

2. Lusheng Wang, Yong Yang and Guohui Lin. Randomized Approaches for Nearest Neighbor Search in Metric Space When Computing the Pairwise Distance Is Extremely Expensive. The Sixth International Conference on Algorithmic Aspects in Information and Management (AAIM 2010), Weihai, China, 2010
List of Tables

1.1 The running time to search the whole PDB (divided) using substructures of different lengths .................................................. 17

2.1 The constant $C_m$. ................................................................. 37

2.2 Number of samples required for 10000 objects. .................. 44

2.3 Number of samples required for 20000 objects. .................. 45

2.4 Number of samples required for 30000 objects. .................. 46

2.5 Number of samples required for 40000 objects. .................. 47

2.6 Number of samples required for 50000 objects. .................. 48

2.7 Experiment results for 200000 objects divided into 10 groups. Original Ave. No. is the average number of samples required to get the closest object when the 200,000 objects are stored in one group. Ave. No. (divided) is the average number of samples required to get the closest object when the 200,000 objects are divided into 10 groups. Global samples is the number of samples in $S_g$ for each fixed $m$ (dimension) . 50
List of Figures

1.1 A translation translates the cube in 3D. The red cube represents the translated version of the blue one. ........................................ 7
1.2 A rotation rotates around an axis. The red cube represents the rotated version of the blue one. ........................................ 8
1.3 Algorithm 1: the algorithm for the single substructure search problem. . 11
1.4 Algorithm 2: the algorithm for all pairs of substructures. ............. 12
1.5 The interface of the package. ............................................. 14
1.6 Showing a pair of substructures from different angles. ............... 15
1.7 The average running time for a fixed length of protein structures in the database......................................................... 16
1.8 The substructure from atom 659 of residue 87 to atom 685 of residue 90 in chain A matches the substructure from atom 1984 of residue 65 to atom 2010 of residue 68 in chain B. ................................. 18
1.9 The substructure from atom 1254 of residue 167 to atom 1280 of residue 170 in chain A matches the substructure from atom 1775 of residue 33 to atom 1801 of residue 36 in chain B. ................................. 19
1.10 The substructure from atom 365 of residue 51 to atom 590 of residue 81 in chain A matches the substructure from atom 2767 of residue 51 to atom 2992 of residue 81 in chain B. ................................ 20
1.11 The matched 3D substructures: from atom 28 of residue 4 to atom 67 of residue 9 and from atom 203 of residue 26 to atom 242 of residue 30.

1.12 The matched 3D substructures: from atom 306 of residue 63 to atom 357 of residue 69 in 2fwk and from atom 319 of residue 48 to atom 370 of residue 54 in 1mgq.

1.13 The 3D coordinates for substructure of residues 46–50 in 1m5qA.

1.14 The matches of substructure of residues 46–50 in 1m5qA with the three substructures in 1clq, 1cqn, and 1yuw.

1.15 The 3D coordinates for substructure of residues 893–900 in 1w9cA: part 1.

1.16 The 3D coordinates for substructure of residues 893–900 in 1w9cA: part 2.

1.17 The matches of substructure of residues 892–801 in 1w9cA with the three substructures in 1c3g, 1z4n, and 3bm3.

2.1 The regular \( n_e \) polygon, for \( n_e = 8 \).

2.2 The Outside-In Algorithm.

2.3 The relationship between number of samples and the number of objects in \( U \).

2.4 The relationship between number of samples and the number of objects in \( U \).

2.5 The relationship between number of samples and the number of objects in \( U \).

2.6 The Inside-Out Algorithm.
List of Abbreviations

3D three-dimensional
SVM Support Vector Machine
PPI Protein-Protein Interaction
RMSD Root Mean Square Deviation
LWPS the Largest ’Well-Predicted’ Subset
PDB Protein Data Bank
AESA Approximating and Eliminating Search Algorithm
$k$-NN $k$-Nearest Neighbor