CHAPTER I

INTRODUCTION

1.1 BACKGROUND

Bioinformatics can be defined as a collection of mathematical, statistical, and computational methods for analyzing biological sequences including DNA, RNA, and amino acid (protein) sequences. Numerous developments for sequencing the DNA of particular organisms continuously supply new amounts of data on huge scale. Biologists will have difficulties to understand it without the help from more quantitative disciplines.

Many studies in molecular biology require performing specific computational procedures on given sequence data, for example, organizing the data, and hence analysis of biological sequences is often viewed as a part of computational science. However, understanding biological sequences now increasingly needs profound ideas past computational science, in particular, mathematical ideas. In the present, one of the dominant problems in computational molecular biology is multiple sequence alignment (MSA).

New biological sequence develops from pre-existing sequences rather than being created by nature from scratch. This fact is the foundation of biological sequence analysis. If we manage to relate a newly discovered sequence to a sequence about which somewhat (e.g., structure or function) is already known, then there are chances that the known information applies, at least to some extent, to the new sequence as well. Any related sequences that arose from a common ancestral sequence are said to be homologous (see Isaev A. [3]). In practice, two sequences are called homologous if one can establish their relatedness by currently available methods. It is the sensitivity of the methods that produces a borderline between sequences called homologous and ones that are not. For example, two DNA sequences can be called homologous if one can show experimentally that their functions in the respective organisms are related. Hence, sequence homology is a dynamic concept and families of homologous sequences known at that present may change as the sensitivity of the methods improves.

The first step towards inferring homology is to look for sequence similarity. It is not easy to decide whether a given collection of sequences are similar or not if they are very long sequences. In order to see if they are similar, one has to align them properly. A way of how to align a collection of more than two sequences properly is known as MSA.

Many methods have been proposed to solve MSA problem such as dynamic programming (e.g. Smith-Waterman algorithm, see Isaev A. [3]) and heuristic (e.g BLASTA, see Isaev A. [3]). A method has been proposed by Althaus *et al*. [1] to solve the MSA problem which is based on integer linear programming (ILP). The general ILP formulation of the MSA problem is derived from the graph representation of the MSA problem (see Althaus *et al*. [1]; Reinert K. [6]). The gapped trace obtained from the graph representation of the MSA problem, which is the gapped extended alignment graph, corresponds to the alignment of the MSA problem. Although we have the general ILP formulation of the MSA, constructing the ILP model of the MSA problem that can be solved by an ILP solver directly is not straightforward since different MSA problem will have different set of alignment edges and gap arcs involved in the ILP model as well. Hence, we will construct a program that can generate and solve the ILP model of any given MSA problem using MATLAB. The method used to solve the ILP model is branch-and-bound.

1.2 PROBLEM STATEMENT

How to solve a multiple sequence alignment problem utilizing integer linear programming?

1.3 THE OBJECTIVE OF THE *SKRIPSI*

The objective of the *skripsi* is to solve a multiple sequence alignment problem utilizing integer linear programming. We also construct a program that can generate and solve the ILP model of any given MSA problem using MATLAB R2008a.

1.4 SCOPE OF DISCUSSION

In this *skripsi*, we discuss how to generate an ILP model of an MSA problem of DNA sequence for more than two sequences. We also give a brief introduction of the graph representation of the MSA problem since the general ILP formulation of the MSA problem is derived from it. The method used to solve the ILP model is branch-and-bound.

1.5 THE ORGANIZATION OF THE *SKRIPSI*

The *skripsi* is divided into five chapters. Chapter II presents the definitions and basic concepts of MSA, scoring scheme of an alignment, definitions and basic concepts in graph theory, graph representation of the MSA problem, definitions and basic concepts in integer linear programming and branch-and-bound method. Chapter III presents the general ILP formulation and steps of generating an ILP of an MSA problem. Chapter IV presents program implementation and simulation. Chapter V presents conclusions.