Practical Project for the Constraint Processing and Programming course Fall 2015

Mario Martin

November 18, 2015

Abstract

In this project you will use a constraint programming tool (Minizinc or Gecode) to solve a medium size constraint problem in a realistic context that involves pre-processing of data, declarative modelization of the problem, and finding the optimal search scheme for your problem.

1 Introduction

One well known technique to obtain the genome sequence of a living organism consists in applying some enzymes to cells of the organism (to break the nucleus) and the centrifugation of the resulting compose in order to separate the DNA. Unfortunately, this process of extraction and purification ends with several pieces of DNA that should be assembled together in order to recover the original sequence.

For each piece of DNA, it is possible to obtain the sequence of nucleotides that compose it. A filtering process allows us to remove pieces that are overlapping or repeated. After this filtering, we end with some sequences of DNA that are mostly non overlapping and that we suspect that could be used to rebuild the complete original sequence of DNA. We suspect that because they are different and because the weight of the different pieces of DNA together is very similar to the estimated molecular weight of the complete molecule. The goal of the project is to rebuild the original sequence of DNA from these pieces.

2 Data

We are provided with a set of text files, one for each molecule of DNA that we want to rebuild. Each text file contains a list of pieces of DNA recovered. Each line describe one sequence of nucleotides that compose one piece of DNA. For instance, the file with following lines:

```
TCCGCGTTTGCTAAGTACTGTATGA
GTATGACGCTTCTGAATGGTCCACGGTT
GGATCAATAGTGCCCTACCATCT
ACTCGGCTGATCGATGAACACC
```

```
ATGTAAGCTGACGAGTAG
AACACGATCGCTCAAGGTGTTTACG
CGTCTTACAGGGT
AAAACGATGGAGCTCTTGGAGT
CCATCTCGTCGATCAA
TACCCCGA
GAAATGCACC
AGGGAAGTGGGCATAAGAAAC
TACCGCTGTAACGAGAGAAG
GGTGACCGGAGATCAGT
GAGCCTACAGCACGTGTTCTTATGATC
```

represents a simple problem with only 15 pieces of DNA that should be assembled in order to rebuild the complete genome. The pieces are randomly sorted and have different lengths.

3 The rules

We have to recover the complete sequence of DNA fulfilling the following constraints:

- Each piece of DNA appears only once in the sequence.
- One piece of DNA d_1 can follow another piece of DNA d_2 only when the last $k > 2$ nucleotides of d_1 coincide with the first k nucleotides of d_2. The number k is a parameter that has to be found with your program. The higher the number for your solution, the higher the probability of a good reconstruction of the DNA sequence.
- We know from the molecular weight of the complete DNA molecule how many nucleotides form the sequence. The number of nucleotides is in the name of the file. So a file with name `D2000-1.txt` contains a list of pieces of DNA that has to be reassembled to obtain a sequence of exactly 2.000 nucleotides in length.

4 The solving process

In the web page of the course you will find a compressed file with name "Benchmarks.zip" that contains the files for this project. You should rebuild as much as possible sequences of DNA of this file. Some of files consist in problems that cannot be solved. For the problems where reconstruction is not possible, you should say that.

To solve the program you will need to follow the steps described:

1. Write a program that is feed with the pieces of DNA and return a data file suitable for Minizinc (a `dzn` file) or Gecode.
2. Model your problem in the chosen tool
3. Find the fastest strategy to solve your problem

5 Documentation

I will require two kinds of solution for each problem:

1. A solution that satisfy the constraint of the problem
2. The best solution in terms of the higher k number

The documentation required for your project is the programs that you wrote to solve the problem, a description in natural language of the way you modeled the problem, the solution for each DNA sequence, and a written discussion of the search strategies used and why you think they worked better.

6 Bonus points

You can increase your overall grade for the curse if you extend your program for both:

- Dealing with *foreing* pieces of DNA, that is, pieces of DNA that should not be used in the rebuilding of your DNA sequence
- Dealing with two ways pieces of DNA sequences, that is, pieces of DNA can be considered as they are written in the file or reversed.

I will provide you a file with problems of this kind if you decide to accept the challenge.