Università degli Studi di Milano Bicocca
**Scuola di Scienze**
**Dipartimento di Informatica, Sistemistica e Comunicazione**
**Corso di laurea in Informatica**

# Integer Linear Programming approaches on the DNA recombination problem

**Relatore**: *Bonizzoni Paola*

**Co-relatore**: *Della Vedova Gianluca*

**Relazione della prova finale di:**

*Antonio Vivace*

*Matricola 793509*

**Anno Accademico 2016-2017**

**Abstract**

We introduce the *Computational Biology* field and familiarise with *Integer Linear Programming*, defining its inception, uses and approach, and how ILP-based approaches have become a standard optimization technique in bioinformatics, reviewing an application.

Then, we formalise the "DNA Recombination and Rearrangement" problem based on the what is observed in some species of ciliates, followed by an analysis and report of some of existent approaches and their central ideas, limitations and reductions applied.

Finally, an ILP formulation of the DNA Recombination problem is given, describing the implementative tools used and the main encountered difficulties.

# Contents

# 1 — Introduction

## 1.1 Computational Biology

*Computational Biology* is defined as the development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques to the study of biological, behavioral, and social systems[1].

The field is now thirty years old and it's covered by many conferences and journals publishing papers. It features graph theory, network flows, combinatorics, integer and linear programming problems, statistical approaches, probabilistic methods, hidden Markov models, neural networks as its tools.

Some of the most important challenges are[2]:

- Protein structure prediction;
- Homology searches;
- Multiple alignment and phylogeny construction;
- Genomic sequence analysis and gene-finding.

In particular, *Computational Molecular Biology* (bioinformatics) focuses on studying existing and emerging approaches, techniques and algorithms for string computation (sequences) providing a significant intersection between computer science and molecular biology [3].

For these reasons, the field is inherently multidisciplinary: it's appealing to the Mathematical Programming and Operations Research community. Today, computational biology papers are written by computer scientists, biologists, statisticians, physicists and mathematicians, pure and applied.

Concretely, the problem areas are [4]:

- Sequence Analysis. Comparison of genomic sequences within individuals of a same species, or intra-species, in order to highlight their differences and similarities. Reconstruction of long sequences by assembly of shorter sequence fragments. Error correction for sequencing machines.

- Hybridization and microarrays. Use of hybridization for sequencing. Use of microarrays for tissue identification, clustering and feature selection discriminating healthy from diseased samples. Design of optimal primers for PCR experiments. Physical mapping (ordering) of probes by hybridization experiments.

- Protein Structures. Protein fold prediction from aminoacid sequence (ab-initio), or from sequence + other known structures (threading of sequences to structures). Alignment of RNA sequences depending on their structure. Protein fold comparison and alignment of protein structures. Study of protein docking and synthesis of molecules of given 3D structure.

- Evolution. Comparison of whole genomes to highlight evolutionary macro events (inversions, transpositions, translocations). Computation of evolutionary distances between genomes. Computation of common evolutionary subtrees or of evolutionary supertrees.

Note that the term *bioinformatics* is used also as an umbrella term for the (wider) body of biological studies using computer programming as part of their methodology, as well as a reference to specific analysis "pipelines" that are repeatedly used, particularly in the field of genomics.

# 2 — Integer Programming

## 2.1 Definition

*Linear programming* (ILP) is a technique for the mathematical optimization of a linear objective function, subject to linear equality and linear inequality constraints.

Linear programs are problems that can be expressed in canonical form as:

$$\begin{array}{ll} \text{maximize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b} \\ \text{and} & \mathbf{x} \geq \mathbf{0} \\ & (\mathbf{x} \in \mathbb{Z}^n) \end{array}$$

If the variables are forcibly constrained to be integers, we call the program *Integer* or *Integer Linear* (ILP).

0-1 integer programming or binary integer programming (BIP) is the special case of integer programming where variables are required to be 0 or 1 ($\mathbf{x} \in \{0, 1\}$).

In contrast to linear programming, which can be solved efficiently in the worst case, integer programming problems are in many practical situations (bounded variables) NP-hard. BIP are classified as NP-hard too ("01 integer programming" is one of the *Karp's 21 NP-complete problems*).

## 2.2 In Computational Biology

At its inception, the focus of Computational Biology was on the development of efficient algorithms and data structures that were able to deal with the data being introduced in life science applications. Lately, the introduction of high throughput methods for biomedical data analysis and the rise of Systems Biology (the study of systems of biological components) made Statistical Learning approaches a standard [5].

Furthermore, new and accessible sequencing methods caused an exponential growth of the available genomic data.

This element and the fact that biological processes are usually reduced and studied as simulations (because the actual nature of them is still being investigated, as in the case of our problem) lead to the introduction of a lot new optimization problems in the field.

In most cases, these optimisazion problems are discrete ones: hence the success of ILP-based approaches.

### 2.2.1 Advantages

There are a number of additional reasons why ILP should be taken into consideration, even when the problems seems to not require it or the advantage of introducing an ILP formulation isn't initally clear[6]:

- Commercial ILP *solvers* are available (with academic licenses);
- The progress of those solvers has been spectacular: benchmark ILP problems can be solved *200-bilion* times faster than twenty-years ago;
- Even for a problem where a worst-case efficient general algorithm might be possible, the time and effort needed to find it, implement it as a computer program, is typically much greater than the time and effort needed to formulate and implement an ILP solution to the problem.
- Some problems can be modeled in a much more efficient with ILP.

### 2.2.2 A typical approach

Usually,

## 2.3 Design of an ILP formulation

### 2.3.1 Idioms

Here's how many logic expressions can be expressed as linear disequalities without side effects or uncovered cases [6].

Suppose $L$ is an integer linear function of binary variables with $M$ being its upper limit and $b$ a positive integer. Many of these idioms can be reduced if some or all variables are binary, strictly positive, or bounded.

**If-Then**

$$L \geq b \rightarrow z$$

Linearly:

$$L - (M \times z) \leq b - 1$$

**Only-If**

$$z = 1 \text{ only if } L \geq b$$

Let $s$ be the smallest value that L can achieve and set $m = s - b$. Linearly:

$$L + m \times z \geq m + b$$

These two idioms can be used as building blocks for many more:

**NAND**

Let $L_1$ and $L_2$ be linear functions whose variables are bounded, and $L_1 \geq b_1$ and $L_2 \geq b_2$ . We require that at *most* one of the two linear inequalities is satisfied.

$$z_1 + z_2 \leq 1$$

Where $z_1 = 1$ if $L_1 \geq b_1$ and $z_2 = 1$ if $L_2 \geq b_2$. We use the *If-Then* twice idiom to express these two conditions.

**OR**

Here we require that at *least* one of the two linear inequalities is satisfied.

$$z_1 + z_2 \geq 1$$

Followed by two *Only-If* idioms to express $z_1 = 1$ *only* if $L_1 \geq b_1$ and $z_2 = 1$ *only* if $L_2 \geq b_2$.

**XOR**

If we want *exactly* one of the inequalities to be satisfied:

$$z_1 + z_2 = 1$$

Again, followed by two *Only-If* idioms to express $z_1 = 1$ *only* if $L_1 \geq b_1$ and $z_2 = 1$ *only* if $L_2 \geq b_2$ and two *If-Then* (if and only if).

**Implied Satisfaction**

To express

$$L_1 \geq b_1 \rightarrow L_2 \geq b_2$$

We need an *If-Then* idiom for the first equality, an *Only-If* idiom for the second and

$$z_1 \leq z_2$$

**Not-Equal**

Suppose $Z_1$ and $Z_2$ are linear functions of integer variables whose values are bounded. Then $Z_1 - Z_2$ and $Z_2 - Z_1$ are bounded to integer values, too. Then, we can express

$$Z_1 \neq Z_2$$

as

$$(Z_1 - Z_2 \geq 1) \text{ OR } (Z_2 - Z_1 \geq 1)$$

Using our *OR* idiom previously explained: let $s_1$ the lower bound for $Z_1 - Z_2$ and $s_2$ the lower bound for $Z_2 - Z_1$. Set $m_1 = s_1 - 1$ and $m_2 = s_2 - 1$. The final inequalities will be

$$(Z_1 - Z_2) + m_1 \times l_1 \geq m_1 + 1$$
$$(Z_2 - Z_1) + m_2 \times l_2 \geq m_2 + 1$$
$$l_1 + l_2 \geq 1$$

# 3 — The Problem

## 3.1 Biological Background

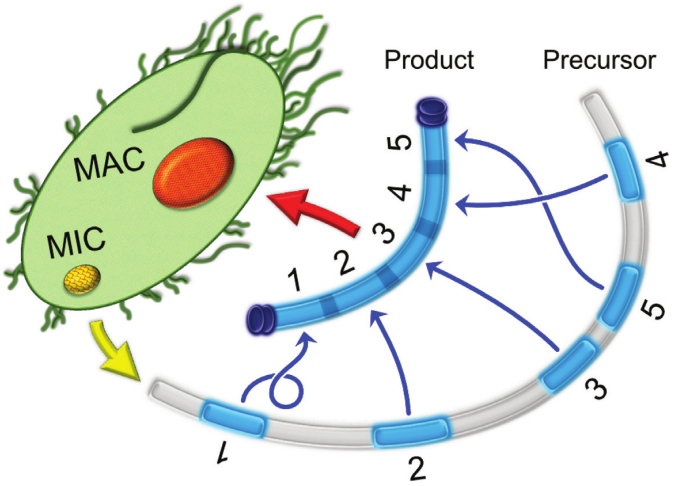


**Figure 3.1:** In the somatic macronucleus (MAC), chromosomes assemble from precursor MDS building blocks (blue), which may be scrambled in some species. In the germline micronucleus (MIC), the Macronuclear Destined Sequences (MDSs) for all somatic chromosomes are dispersed over the long chromosome, and interrupted by Internally-Eliminated Sequences (IESs) and other noncoding DNA (gray). In some cases, an MDS may appear in a permuted order, or inverted.[7].

Ciliated protists (microbial eukaryotes using cilia for locomotion) exhibit nuclear dimorphism through the presence of separate germline and somatic nuclei. The somatic macronucleus (MAC) provides templates for the transcription of all genes required for asexual growth while the germline micronucleus (MIC) is used for the exchange of meiotic products during sexual reproduction [7].

Several species of ciliates, such as Stylonychia or Oxytricha, go through extensive gene rearrangement while differentiating somatic macronuclei from germline micronuclei: an extensive fragmentation, elimination and sometimes broader rearrangement of the germline DNA, coupled to DNA amplification and telomere addition [8] forms the somatic macronuclei.

The ciliate Oxytricha is a microbial eukaryote with two genomes, one of which experiences extensive genome remodeling during development. Each round of conjugation initiates a cascade of events that construct a transcriptionally active somatic genome from a scrambled germline genome, with considerable help from both long and small noncoding RNAs.

## 3.2 Biological Motivation

[9]

## 3.3 Formalisation

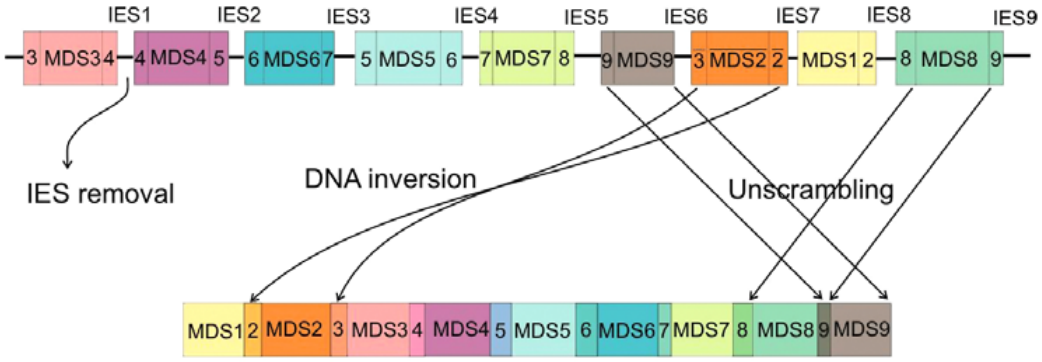


**Figure 3.2:** Schematic representation of the scrambled Actin I micronuclear germline gene in Oxytricha nova (top) and the correctly assembled macronuclear gene (bottom). Each block represents an MDS, and each line between blocks is an IES. The numbers at the beginning and at the end of each segment represent the pointer sequences. Note that MDS3 MDS8 require permutation and inversion to assemble into the orthodox, linear order MDS1 MDS9 in the macronucleus. The bars above MDS2 and its pointers indicate that this block is inverted relative to the others, i.e., this sequence is the Watson - Crick reverse complement of the version in the macronucleus; from[10].

## 3.4 Existent Approaches/Solutions

## 3.5 ILP formulation

# Bibliography


[1] NIH Biomedical Information Science and Technology Initiative Consortium. NIH working definition of bioinformatics and computational biology. `https://web.archive.org/web/20120905155331/http://www.bisti.nih.gov/docs/CompuBioDef.pdf`, 2000.

[2] David B. Searls. Chapter 1 - grand challenges in computational biology. In David B. Searls Steven L. Salzberg and Simon Kasif, editors, *Computational Methods in Molecular Biology*, volume 32 of *New Comprehensive Biochemistry*, pages 3 – 10. Elsevier, 1998.

[3] Dan Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, New York, NY, USA, 1997.

[4] Giuseppe Lancia. Mathematical programming in computational biology: an annotated bibliography. *Algorithms*, 1(2):100–129, 2008.

[5] Ernst Althaus, Gunnar W. Klau, Oliver Kohlbacher, Hans-Peter Lenhof, Knut Reinert. Integer Linear Programming in Computational Biology. In: Lecture Notes in CS 5760.

[6] Dan Gusfield. Integer linear programming in computational biology tutorial. In *Integer Linear Programming in Computational Biology: An entry-level course for biologists (and other friends)*. Cambridge Press, 2018.

[7] Jonathan Burns, Denys Kukushkin, Kelsi Lindblad, Xiao Chen, Nataa Jonoska, and Laura F. Landweber. mds ies db: a database of ciliate genome rearrangements. *Nucleic Acids Research*, 44(D1):D703–D709, 2016.

[8] D.M. Prescott. *The DNA of Ciliated Protozoa. Microbiol.* 1994.

[9] Angela Angeleska, Nataa Jonoska, and Masahico Saito. Dna recombination through assembly graphs. *Discrete Applied Mathematics*, 157(14):3020 – 3037, 2009.

[10] D.M. Prescott, A.F. Greslin. Scrambled actin I gene in the micronucleus of Oxytricha nova. *Developmental Genetics*, (13):6674, 1992.