Minimizing sets of enzymes to differentiate between species

David Buezas
CENTRIA
FCT/UNL
david.buezas@gmail.com

João Almeida
CREM
FCT/UNL
jmfa@fct.unl.pt

Pedro Barahona
CENTRIA
FCT/UNL
pb@di.fct.unl.pt

Overview

- Introduction
  - Motivation
  - Antecedents
  - Objectives
- Mapping ARDRA into Minimum Set Covering
  - Enzymes
  - Gel electrophoresis
  - Definitions
  - Coverage table
- Alternative models and benchmarks
  - Datasets
  - Greedy
  - Backtrack
  - Boolean CP
  - Finite Domain CP I
  - Finite Domain CP II
  - Summary of results
- Conclusions and further work
  - Conclusions
  - Variants

Non distinguishable species

- A large number of species cannot be distinguished via standard non genetic analysis in the lab.
- Sequencing nucleic acids is still too expensive to be applied to a large number of individuals.
- Less expensive techniques:
  - RFLP, RAPD and MSP-PCR form clusters
    - have many limitations
  - ARDRA goes beyond clustering
    - Enzymes are manually selected
      - Cumbersome to extend
      - Non optimal sets

Overview of ARDRA techniques

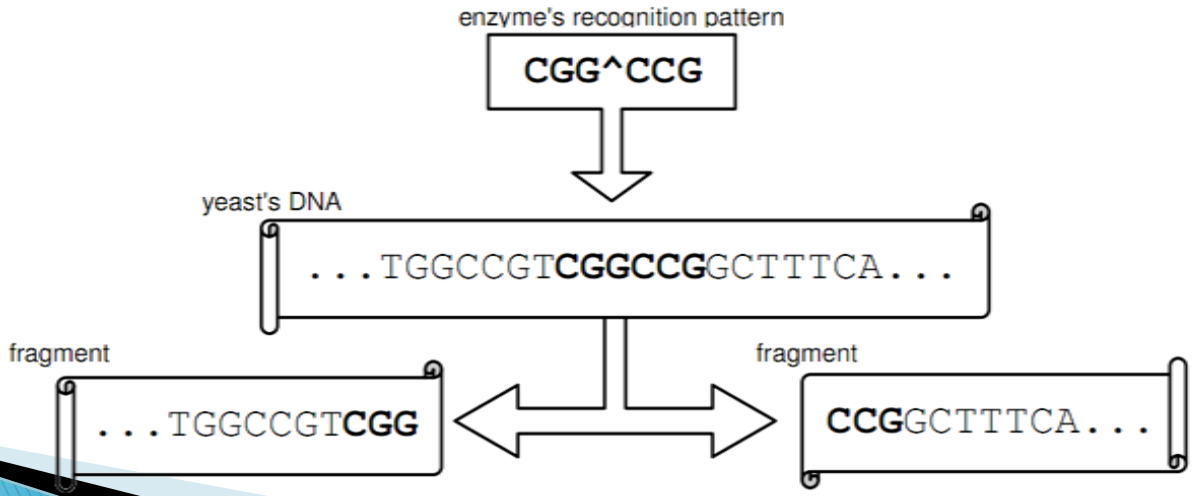
- ARDRA (Amplified Ribosomal DNA Restriction Analysis)
  - Differentiates organisms inside a particular eubacterial family
  - A particular set of enzymes was chosen for that family
  - Sometimes it does not indicate a single species
- ARDRA-ITS
  - Same approach
  - Uses a different DNA region (5.8S-ITS)
  - Aimed to a particular family of fungal species
- ARDRA-ITS variant
  - Aimed to a family of yeast associated with food
  - Very successful
- Non of the above are general techniques
  - They aim to particular families of species

Purpose of this paper

- Inferring the **minimum set of enzymes** required to identify species
  - General→ applicable to any set of organisms
  - Automatic → making it convenient
  - Optimal → minimum sets → reduced costs
- Proposing this problem to be used as benchmark for Constraint Programming methods applied to Bioinformatics

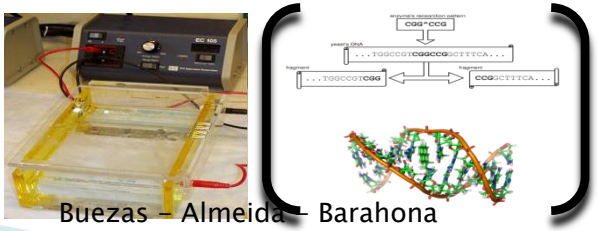
Restriction enzymes

- They cut DNA at specific nucleotide sequences, known as restriction sites.
- If that precise sequence is present in different positions in two different organisms, the digestion will produce DNA segments of different size on each.



Gel electrophoresis

- The size of the resulting fragments can be approximated by following the steps:
  1. the fragments are marked with a fluorescent dye and put in the gel,
  2. an electric current is circulated through the gel,
  3. the fragments migrate from the cathode to the anode at different speeds depending on their molecular weight,
  4. the sample is illuminated with a special light to make the dye fluoresce,
  5. a picture of the pattern formed is taken,
  6. the approximate sizes of the fragments is calculated from the picture.



Definitions

- Two patterns P and Q are **distinct** *iff*
  - there is one band in P not present in Q (considering the error present in gel electrophoresis):

$$\text{distinct}(P,Q) =_{def} (\exists u \in P)(\forall v \in Q)(u \not\approx v) \vee (\exists u \in Q)(\forall v \in P)(u \not\approx v)$$

- A restriction enzyme E_k is said to **differentiate** two yeast specimens y_1 and y_2 *iff:*
  - the digested y_1 and y_2 present distinct patterns

$$\text{differentiate}(i,j,k) =_{def} \text{distinct}(P(i,k),P(j,k))$$

- A set S of enzymes is called **discriminating** *w.r.t.* Y *iff:*
  - for any pair of yeast in Y there is an enzyme in S that **differentiates it**

$$\text{discriminate}(S,Y) =_{def} \forall (i\text{-}j) \in Y \ \exists k \in S : \ \text{differentiate}(i,j,k)$$

- A **discriminating** set S is **minimal** *iff:*
  - It has minimal cardinality:

$$\text{min\_disc}(S,Y) =_{def} \ \text{discriminate}(S,Y) \wedge (\forall R \ \text{discriminate}(R,Y) \rightarrow \#S \leq \#R)$$

The Boolean coverage table

- With all the previous tools we can compute in polynomial time a coverage table *D s.t.*
  - $D(Y_i-Y_j,E_k)=1$ *iff* **differentiate(I,j,k)**

	E_1	E_2	...	E_{331}
Y_1-Y_2	0	1	...	1
Y_1-Y_3	1	0	...	0
...	...	...	...	...
Y_{22}-Y_{23}	0	0	...	0

Datasets used to test the system

- Enzymes
  - 3500 elements reduced to 331 unique restriction sites.
- Yeasts
  - 5.8S-ITS region of the operons of 23 simmilar species.
- Computer
  - Intel(R) Core(TM)2 Duo T5670@1.80GHz (2 CPUs) with 3 GB of RAM.
- Constraint solving system
  - SICStus 4 CLP.

Greedy model

- Iterative accumulation of the most covering enzyme until all yeast pairs are covered.
  - Only presented for comparison with CP models.
  - Does not find minimum solutions

Algorithm 1 Greedy model

1: $S \leftarrow \emptyset$
2: $Y \leftarrow \{i\text{-}j : i \in 1..N_y, j \in 1..N_y, i < j\}$ ▷ the set of all yeast pairs to cover
3: $E \leftarrow \{k : k \in 1..N_e\}$ ▷ the set of all enzymes available
4: **while** $Y \neq \emptyset$ **do**
5: $\quad e \leftarrow \underset{k \in E}{\text{argmax}} \left(\sum_{i\text{-}j \in Y} \text{differentiate}(i,j,k) \right)$ ▷ select the most covering enzyme
6: $\quad S \leftarrow S \cup \{e\}$
7: $\quad Y \leftarrow Y \setminus \{i\text{-}j : \text{differentiate}(i,j,e)\}$ ▷ subtract the covered yeast pairs
8: **end while**
9: **return** S

Backtrack model

- Iteration over the size of the solution.
  - First set obtained is an optimal discriminating set.
  - Only presented for comparison with CP models.
  - First solution in 1 minute (varies wildly depending on enzymes order)
  - Too slow to find all solutions (should check 6.000.000 triplets)

Algorithm 2 Backtrack model

```
1:  Y ← {1,...,N_y}                          ▷ the set of all yeast identifiers
2:  p ← 0                                    ▷ p stands for the size of the solution
3:  while p ≤ N_e do
4:      p ← p + 1        ▷ the size of the optimal solution is searched incrementally
5:      for all k_1..k_p ∈ 1..N_e : k_1 < ... < k_p do
6:          S ← {k_1,...,k_p}
7:          if discriminate(S,Y) then
8:              return S
9:          end if
10:     end for
11: end while
```

Boolean CP model

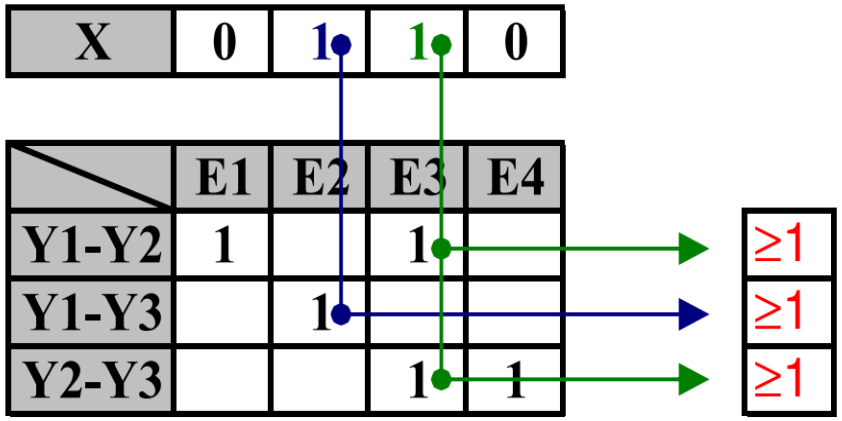
- Labeling of the *X* vector representing enzyme selection, while minimizing *sum(X)*
- The covering of each pair is asserted by a sum-product constraint over *X* and each row of *D*

Algorithm 3 Boolean CP model

1: $X \leftarrow [x_1, ..., x_{N_e}]$ ▷ one boolean variable for each enzyme identifier
2: **for all** $x \in 1..N_e$ **do**
3: $x_k \in 0..1$
4: **for all** $i, j \in 1..N_y : i < j$ **do**
5: $\sum_{k \in 1..N_e} x_k * \text{differentiate}(i, j, k) \geq 1$ ▷ constraints are imposed
6: **end for**
7: **end for**
8: label(X): minimising $\left(\sum_{k \in 1..N_e} x_k \right)$

Boolean CP model

- Intuition:



- First optimal solution: **10 sec.**
- All 300 optimal solutions: **15 min.**

Finite Domain model I

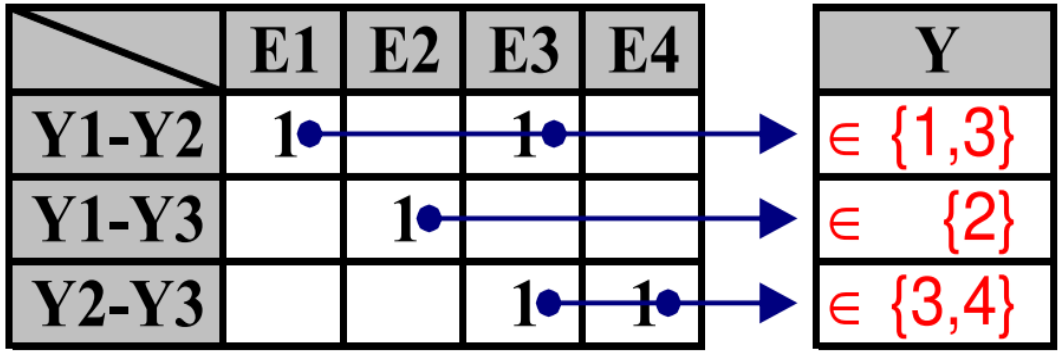
- Labeling of the Y vector representing which enzyme will be used to cover each yeast pair, while minimizing number of different values on Y

Algorithm 4 Finite Domain CP model I

1: $Y = [y_{1\text{-}2}, \ldots, y_{i\text{-}j}, \ldots, y_{(N_y-1)\text{-}N_y}] : i < j$ ▷ one Finite Domain variable for each yeast pair
2: **for all** $i, j \in 1..N_y : i < j$ **do**
3: $y_{i\text{-}j} \in \{k \in 1..Ne :$ differentiate$(i, j, k)\}$
▷ the domain of $y_{i\text{-}j}$ is the set of enzyme identifiers that cover the i-j pair
4: **end for**
5: nvalues(Y,N)
6: label(Y): minimizing(N)

Finite Domain model I

- Intuition:



- First solution: **1 sec.**
- All solutions: **n/a**
  - massive symmetries due to constraint used to count number of different values in *Y*

Finite Domain model II

- Somewhat the dual model of the other
- Variables in the vector are associated to enzymes instead of yeast pairs.

Algorithm 5 Finite Domain CP model II

1: $p \leftarrow \#S$ ▷ where #S is the minimum solution size
2: $S \leftarrow [k_1, ..., k_p]$
3: $k_1 < ... < k_p$ ▷ imposing an order avoids repeated solutions
4: **for all** $i, j \in 1..N_y : i < j$ **do**
5: $E \leftarrow \{e : \text{differentiate}(i, j, e)\}$
6: $\bigvee_{i \in 1..p} (k_i \in E)$ ▷ a disjunctive constraint is posed
7: **end for**
8: label(X)

Finite Domain model II

- Intuition:

	E1	E2	E3	E4
Y1-Y2	1		1	
Y1-Y3		1		
Y2-Y3			1	1

S=<k1,k2>
k1 < k2
k1 ∈ {1,3} v k2 ∈ {1,3}
k1 ∈ {2} v k2 ∈ {2}
k1 ∈ {3,4} v k2 ∈ {3,4}

- First solution: **n/a**
  - Requires knowing the size of a minimum solution
- All solutions: **50 sec.**

Summary of results

Algorithm	First solution	All solutions
Greedy	n/a (not optimal)	n/a
Backtrack	1 min	n/a
Boolean CP	10 sec	15 min
Finite domain I	**1 sec**	n/a (symmetries)
Finite domain II	n/a (requires #S)	**50 sec**

- Minimum solution: **3 enzymes**
- Total number of min. solutions: **~300**

Scope of the method

- Taxonomical levels
  - This method could be in principle be applied to *any* taxonomical level
  - Higher level → less enzymes → cheaper
- Types of organisms
  - ARDRA was used to identify bacteria
  - This method should be applicable to other kinds of organisms too

Conclusions

- In this paper we explore several potential models to a Bioinformatics problem, raised by the ARDRA-ITS experimental technique.
- The technique we used mapped the problem into a set covering problem.
- The various models show the advantage of constraint programming over backtracking or purely heuristic search.
- We achieved minimization of the number of enzymes that must be used in to unequivocally tell a yeast within a set of related yeasts.
- Furthermore we found *all* minimum solutions, giving alternatives to the user.
  - Some enzymes might be easier to get, less expensive, or produce more robust results.

Variants

- We assumed that bands in electrophoresis experiments are distinguishable if their lengths differ by 5%
  - that relative difference could be used as a parameter to be maximized,
    - the most *reliable* solution would be found
- Mutant strands
  - A quantified version of the problem could find solutions, even when some nucleotides of the organism had changed.
- We plan to
  - address both variants of this problem,
  - provide a more comprehensive set of benchmarks,
  - and more as experimental results,
  - try different approaches
    - Integer Programming