Mathematical Programming Approaches in Computational Biology

Giuseppe Lancia

Dipartimento di Matematica e Informatica
Università di Udine

MIP 2006

Outline

Outline

Outline

Some history

What is C.B. ?

(*Quot capita, tot sententiae*. My definition is)

Computational Biology

Study of mathematical and computational problems of modeling biological processes in the cell, removing experimental errors from genomic data, interpreting the data and providing theories about their biological relations.

Born around early 90s

Initially, **mostly computer scientists** dominated the field

(Algorithmic approaches, Computational complexity, String-related problems, Information retrieval, Genomic data base,....)

What is C.B. ?

(*Quot capita, tot sententiae*. My definition is)

Computational Biology

Study of mathematical and computational problems of modeling biological processes in the cell, removing experimental errors from genomic data, interpreting the data and providing theories about their biological relations.

Born around early 90s
Initially, **mostly computer scientists** dominated the field
(Algorithmic approaches, Computational complexity, String-related problems, Information retrieval, Genomic data base,....)

Optimization in CB

cb2mp

1. model “alive” objects (proteins, genes, DNA sequences,...) into mathematical objects (graphs, vectors, strings,...)
2. model the phenomenon with constraints
3. cost of solution $\simeq$ probability of being correct
4. find best solution $\mapsto$ (NP-hard) optimization problem

Optimization in CB

cb2mp

1. model “alive” objects (proteins, genes, DNA sequences,...) into mathematical objects (graphs, vectors, strings,...)
2. model the phenomenon with constraints
3. cost of solution $\simeq$ probability of being correct
4. find best solution $\mapsto$ (NP-hard) optimization problem

...“Mathematical Programming people” entered the field

Optimization in CB

cb2mp

1. model "alive" objects (proteins, genes, DNA sequences,...) into mathematical objects (graphs, vectors, strings,...)
2. model the phenomenon with constraints
3. cost of solution $\simeq$ probability of being correct
4. find best solution $\mapsto$ (NP-hard) optimization problem

The "1st" MP paper in CB (?)

F. Alizadeh, R. Karp, D. Weisser and G. Zweig, **Physical Mapping of Chromosomes Using Unique Probes**, *Proc. Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1994

Problem modeling

A CB problem can be modeled into:

- a new problem, solved with *ad hoc* approach
- a known problem (TSP, SC, MAXCLIQUE,...) solved by *state-of-the-art* program off-the-shelf

The growth of MP in CB

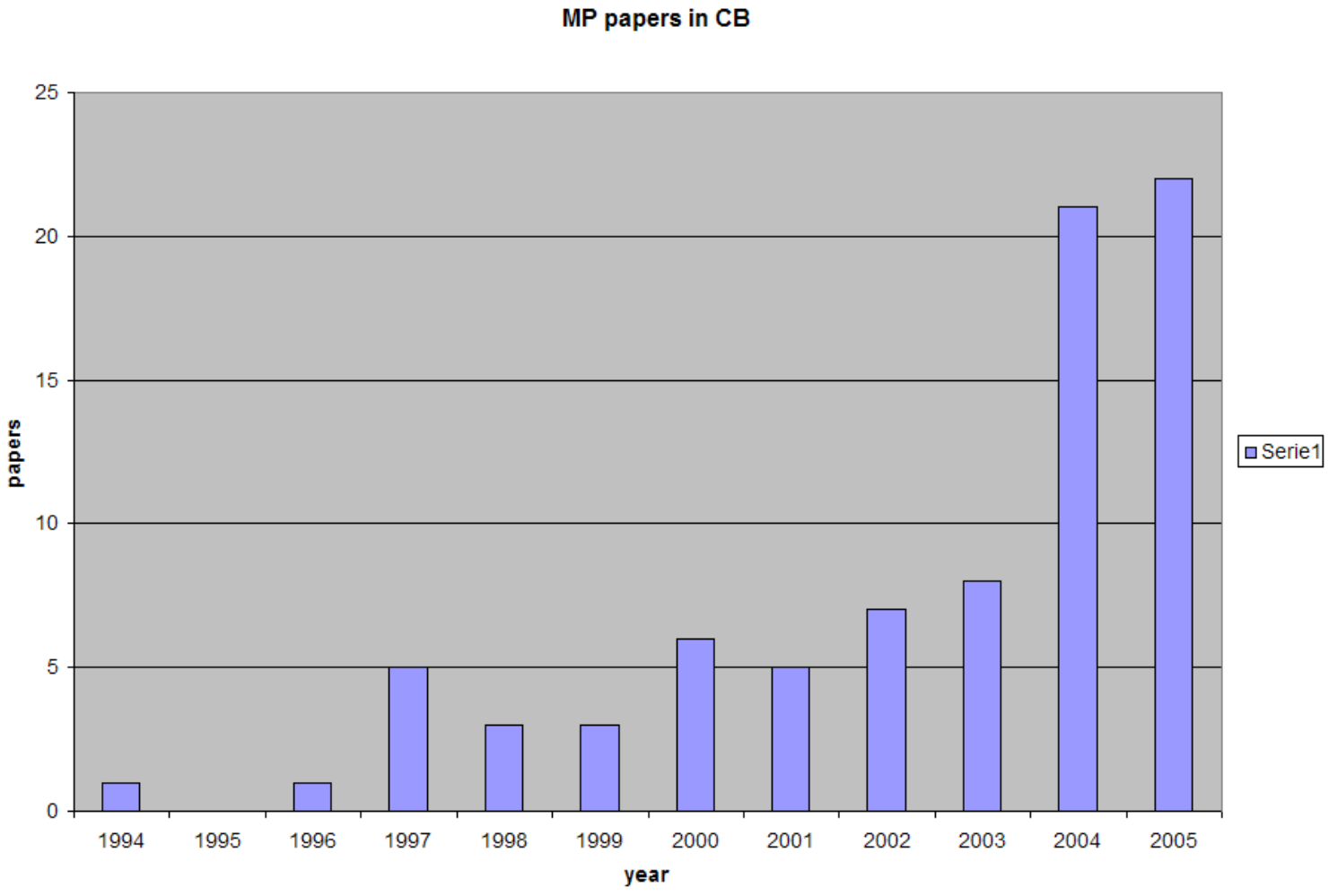
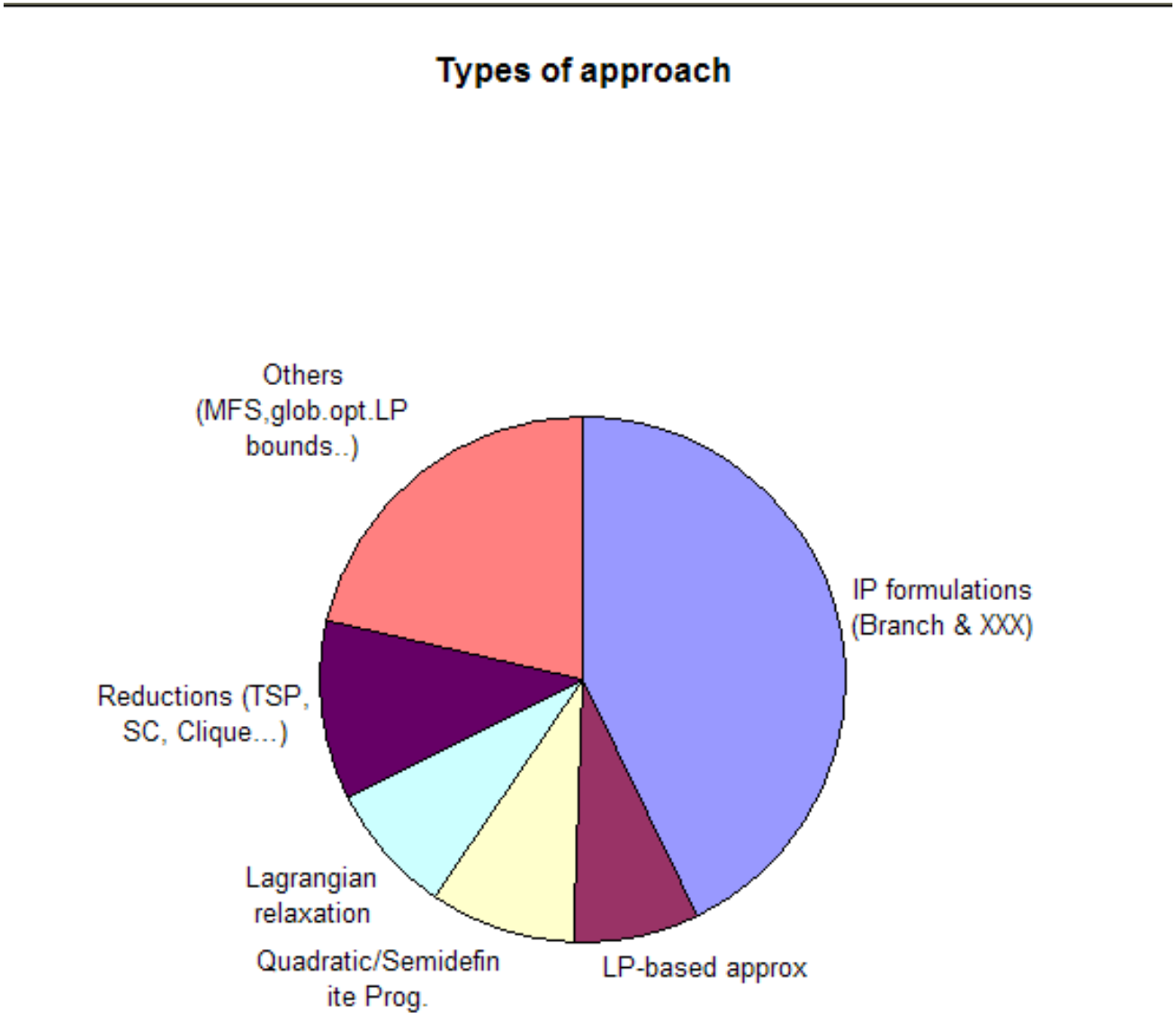


Figure: MP papers in CB over years

Problems vs approaches



Problems vs approaches

	IP	LP	QP/SDP	LR	Red.	Oth.
Haplotyping	•	•	•			•
Protein structure comparison	•			•	•	•
Protein folding/threading	•			•		•
Sequence analysis and alignment	•	•				•
Phylogeny/supertree reconstruction		•	•			•
Protein sequence design			•	•		
Protein side-chain positioning	•		•			
Protein docking	•					•
Mapping and assembly	•				•	
Genome Rearrangements	•				•	
Sequencing by Hybridization	•					
Probe selection	•					
Protein encoding	•					
Protein energy minimization			•			
RNA alignment				•		
PCR primer selection					•	
DNA Microarrays					•	•

Conferences

- Australian Comp. Sc. Conference (ACSC)
- Combinatorial Pattern Matching (CPM)
- European Workshop on Evolutionary Bioinformatics (EvoBIO)
- European Symp. on Algorithms (ESA)
- European Conf. on Comp. Biol. (ECCB)
- Intl. Symposium on Algorithms and Computation (ISAAC)
- Intl.Symposium on Comp. Life Science (CompLife)
- Intl. Conf. on Intelligent Systems for Molecular Biology (ISMB)
- Pacific Symp. on Biocomputing (PSB)
- RECOMB
- IEEE Intl. Workshop on High Performance Computational Biology (HiCOMB)
- SIAM Symp. on Discrete Algorithms (SODA)
- Workshop on Algorithms in Bioinformatics (WABI)

Journals

- 4OR
- Bioinformatics
- Discr. Appl. Math.
- Genome Research
- INFORMS J. on Computing
- International J. of Robotics Research
- J. of the ACM
- J. of Bioinformatics and Computational Biology
- J. of Computational Biology
- J. of Combinatorial Optimization
- Mathematical Programming
- Networks
- Operations Research

An ϵ of Biology

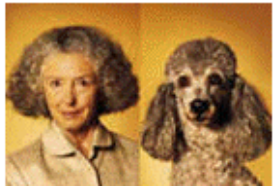
Life is told by genomes



- A genome is “string” in a language over the 4-letter alphabet of DNA {A,T,C,G}
- In human is some 3,000,000,000 letters
- DNA encodes our similaritis and differences

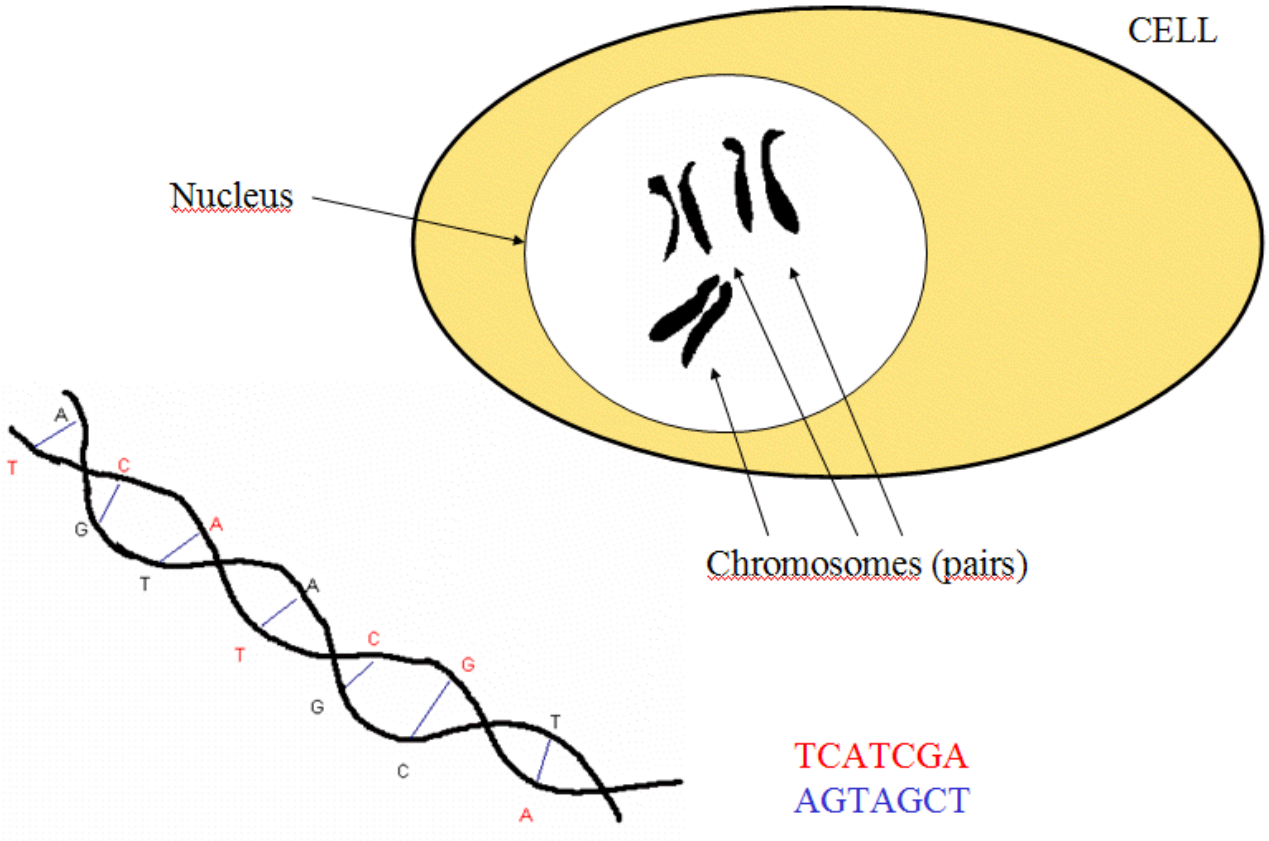
Small genomic changes can make big appearence changes (or can they?)...





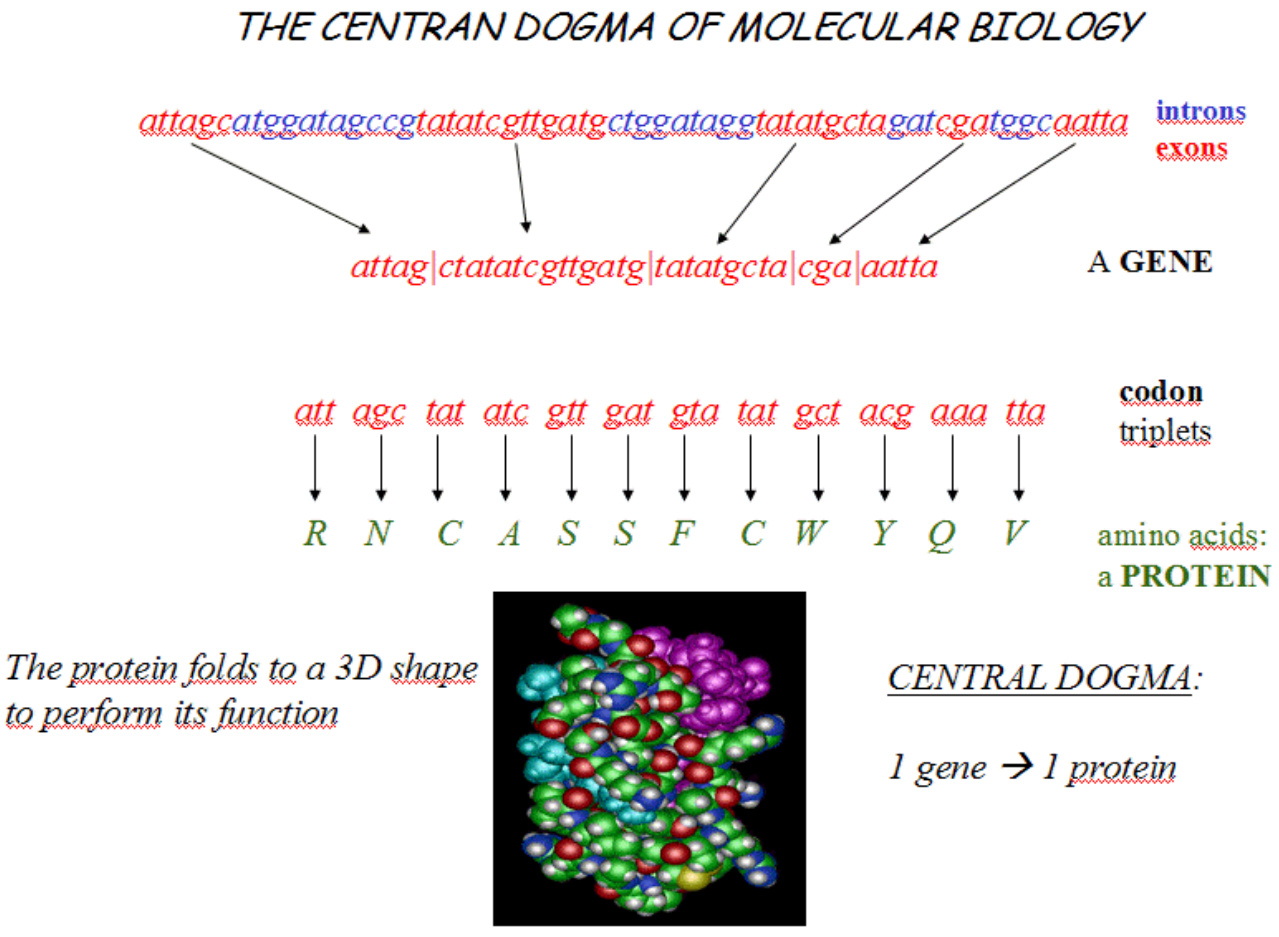

The cell

Eukariotic diploid organisms



The central dogma of molecular biology

1 gene, 1 protein



Genome rearrangements

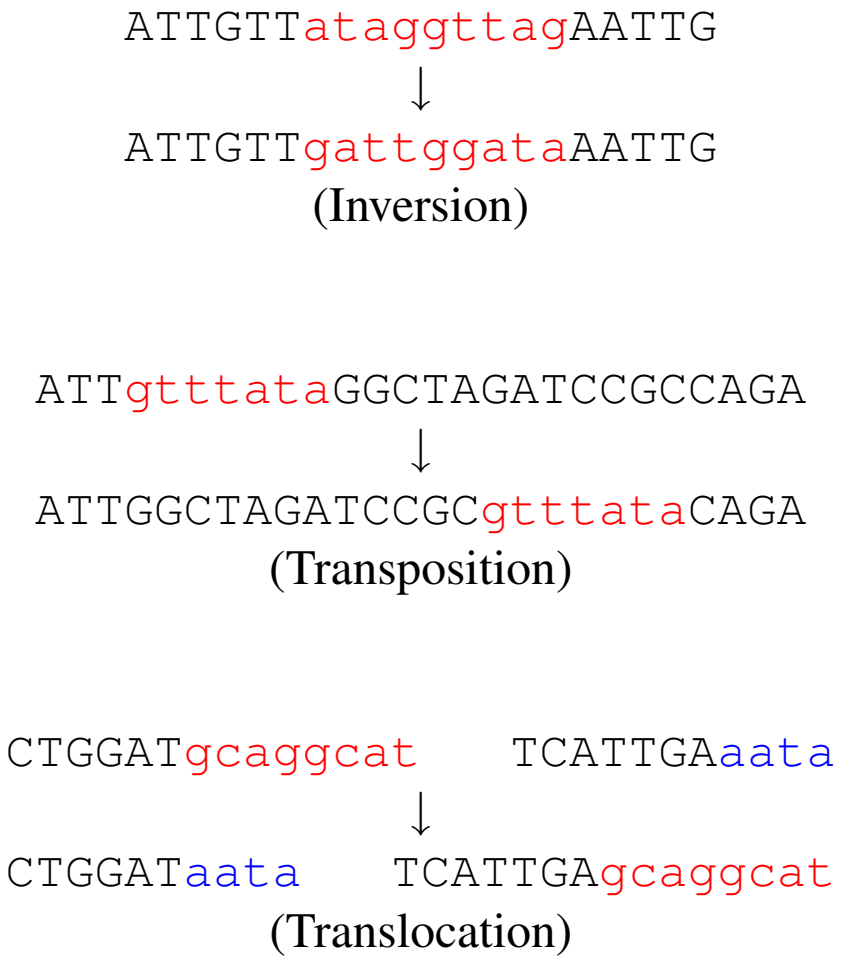


Figure: Evolutionary events

Genomes evolve by means of

- Inversions
- Transpositions
- Translocations

of DNA regions.

- Given two genomes, fix n common genes.

Combinatorial problem

Given permutations π and σ and operators in set $\mathcal{F}$, find shortest sequence of operators $f_1, \ldots, f_k$ s.t. $\sigma = f_k(f_{k-1}(\ldots(f_1(\pi))\ldots)$

- Very difficult! Focus on operators of same type (e.g. inversions). Still difficult.
- Wlog take $\sigma = (12\ldots n)$. Hence we talk of sorting by (inversions, transpositions,...)
- Reversals (inversions) are the most important rearrangement

Example

Sorting by reversals

1. 5 6 4 8 3 2 1 9 7
2. 1 2 3 8 4 6 5 9 7
3. 1 2 3 8 4 5 6 9 7
4. 1 2 3 6 5 4 8 9 7
5. 1 2 3 6 5 4 8 7 9
6. 1 2 3 4 5 6 8 7 9
7. 1 2 3 4 5 6 7 8 9

Sorting by reversals is NP hard (Caprara '96)
Complexity of sorting by transpositions is unknown

Example

Sorting by reversals

1. 5 6 4 8 3 2 1 9 7
2. 1 2 3 8 4 6 5 9 7
3. 1 2 3 8 4 5 6 9 7
4. 1 2 3 6 5 4 8 9 7
5. 1 2 3 6 5 4 8 7 9
6. 1 2 3 4 5 6 8 7 9
7. 1 2 3 4 5 6 7 8 9

Sorting by reversals is NP hard (Caprara '96)
Complexity of sorting by transpositions is unknown

Example

Sorting by reversals

1. 5 6 4 8 3 2 1 9 7
2. 1 2 3 8 4 6 5 9 7
3. 1 2 3 8 4 5 6 9 7
4. 1 2 3 6 5 4 8 9 7
5. 1 2 3 6 5 4 8 7 9
6. 1 2 3 4 5 6 8 7 9
7. 1 2 3 4 5 6 7 8 9

Sorting by reversals is NP hard (Caprara '96)
Complexity of sorting by transpositions is unknown

Example

Sorting by reversals

1. 5 6 4 8 3 2 1 9 7
2. 1 2 3 8 4 6 5 9 7
3. 1 2 3 8 4 5 6 9 7
4. 1 2 3 6 5 4 8 9 7
5. 1 2 3 6 5 4 8 7 9
6. 1 2 3 4 5 6 8 7 9
7. 1 2 3 4 5 6 7 8 9

Sorting by reversals is NP hard (Caprara '96)
Complexity of sorting by transpositions is unknown

Example

Sorting by reversals

1. 5 6 4 8 3 2 1 9 7
2. 1 2 3 8 4 6 5 9 7
3. 1 2 3 8 4 5 6 9 7
4. 1 2 3 6 5 4 8 9 7
5. 1 2 3 6 5 4 8 7 9
6. 1 2 3 4 5 6 8 7 9
7. 1 2 3 4 5 6 7 8 9

Sorting by reversals is NP hard (Caprara '96)
Complexity of sorting by transpositions is unknown

Example

Sorting by reversals

1. 5 6 4 8 3 2 1 9 7
2. 1 2 3 8 4 6 5 9 7
3. 1 2 3 8 4 5 6 9 7
4. 1 2 3 6 5 4 8 9 7
5. 1 2 3 6 5 4 8 7 9
6. 1 2 3 4 5 6 8 7 9
7. 1 2 3 4 5 6 7 8 9

Sorting by reversals is NP hard (Caprara '96)
Complexity of sorting by transpositions is unknown

Example

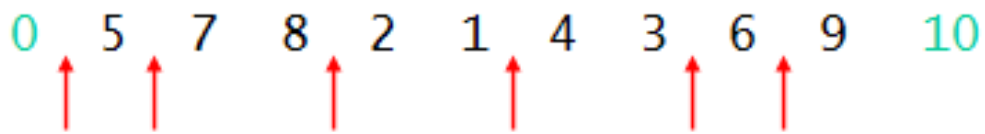
Sorting by reversals

1. 5 6 4 8 3 2 1 9 7
2. 1 2 3 8 4 6 5 9 7
3. 1 2 3 8 4 5 6 9 7
4. 1 2 3 6 5 4 8 9 7
5. 1 2 3 6 5 4 8 7 9
6. 1 2 3 4 5 6 8 7 9
7. 1 2 3 4 5 6 7 8 9

Sorting by reversals is NP hard (Caprara '96)
Complexity of sorting by transpositions is unknown

The concept of breakpoint

Breakpoint at position i if $|\pi(i) - \pi(i+1)| > 1$

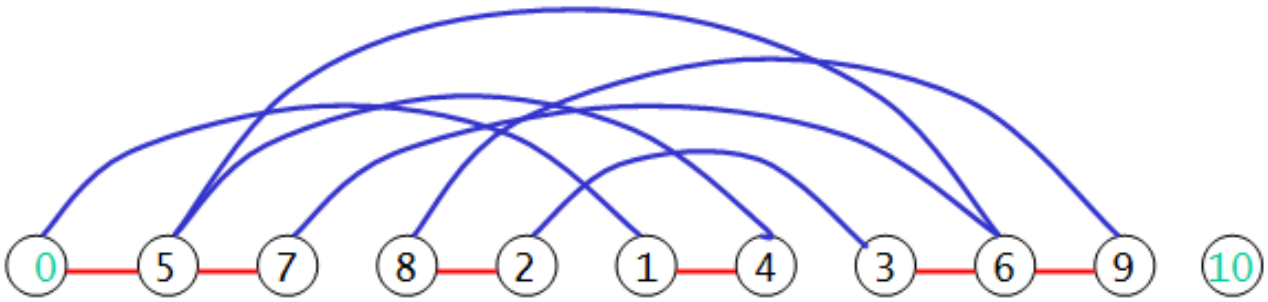


$d(\pi)$ = inversion distance
$b(\pi)$ = n. breakpoints

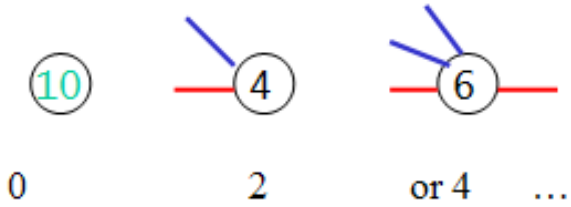
Trivial (weak) bound: $d(\pi) \geq b(\pi)/2$

Example $d(\pi) \geq 6/2 = 3$

The breakpoint graph



Each node has degree



hence the graph can be decomposed in (alternating) cycles!

Alternating cycle decomposition

Theorem

(Bafna,Pevzner'95) Let $c(\pi)$ = max # cycles in alternating cycle decomposition. Then $d(\pi) \geq b(\pi) - c(\pi)$.

Very strong bound!

Example: $c(\pi) = 2$ and $d(\pi) \geq 6 - 2 = 4$

Computational results

- good IP formulation of max cycle decomposition (not of SBR directly)
- pricing is general matching
- Pseudo alternating cycles: can re-use an edge
- Decomposition in pseudo-alt. cycles gives weaker (but not much) L.B.
- Pricing is bipartite matching, much faster
- Can solve up to $n = 200$ in seconds/minutes (Caprara,Ng,L,'01)
- Combinatorial approaches (Kececioglu,Sankoff'95) up to $n = 40$ in hours/days

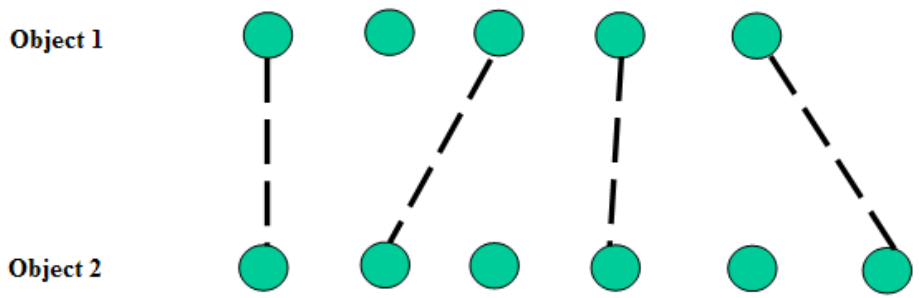
Alignments and non-crossing matchings

Alignments and non-crossing matchings

For two objects, each an ordered list of units, an alignment

1. maps (part of one) into (part of) the second
2. the map respects the order.

Alignment is non-crossing matching



to align = to compare (**obj:** highlight similarities)
Alignable bio-objects:

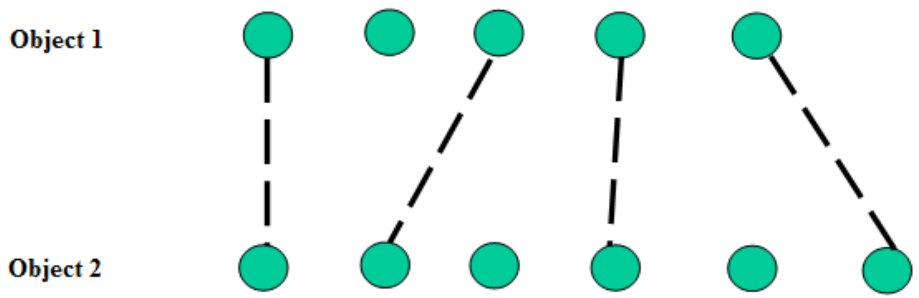
- genomic sequences (list of letters)
- protein structures (list of residues)

Alignments and non-crossing matchings

For two objects, each an ordered list of units, an alignment

1. maps (part of one) into (part of) the second
2. the map respects the order.

Alignment is non-crossing matching

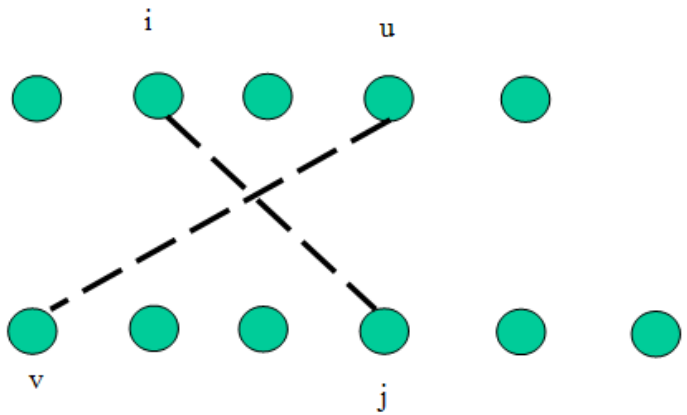


to align = to compare (**obj:** highlight similarities)
Alignable bio-objects:

- genomic sequences (list of letters)
- protein structures (list of residues)

non-crossing matching

Variables x_{ij}. Lines ij and uv cross if $(i-u)(j-v) < 0$.



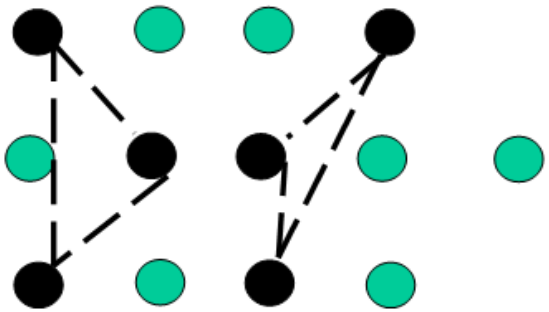
[Clique inequalities] For each set Q of mutually crossing lines Q, valid inequality

$$x(Q) \leq 1$$

Separation can be solved as shortest path in acyclic directed grid, cost $O(n^2)$
(Lenhof,Reinert,Vingron,'98; Carr,L,Istrail,Walenz'00)

IP models for alignment problems always embed clique inequalities

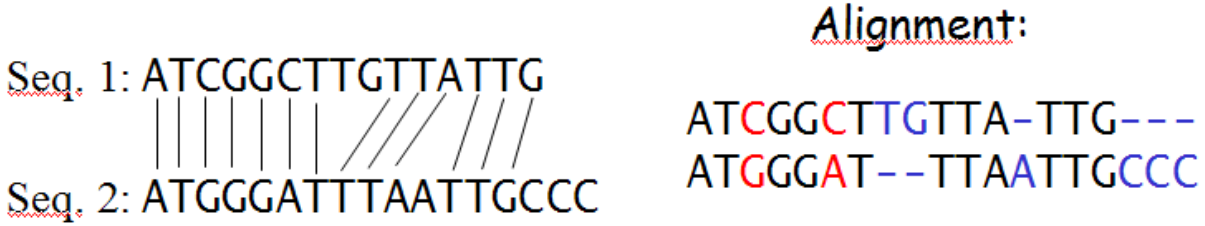
Multiple alignment



- In a multiple alignment we have k objects, each a list.
- Model with layered graph.
- Non crossing lines are no longer enough, need mixed-cycle inequalities (cycles that mix alignment edges and precedence constraints)
- Can be separated in polytime (Kececioglu, Lenhof, Reinert,Mutzel,Vingron'00)

Alignment of genomic sequences

Genomic sequence alignment



We are given

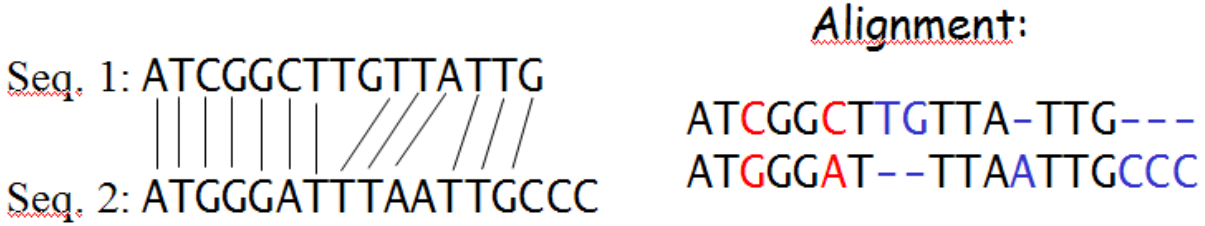
1. a cost matrix τ (4×4 for DNA, 20×20 for a.a.) $\tau(\mathtt{a}, \mathtt{g})$ cost of aligning $\mathtt{a}$ with $\mathtt{g}$
2. indel cost δ (cost of not aligning a symbol)

Find best alignment (min/max non-crossing matching)

complexity

- two sequences: polynomial, Dynamic Programming $O(n^2)$
- k sequences: NP-hard (Wang,Jiang'94; Bonizzoni,Della Vedova'01)

Genomic sequence alignment



We are given

1. a cost matrix τ (4×4 for DNA, 20×20 for a.a.) $\tau(\mathtt{a}, \mathtt{g})$ cost of aligning $\mathtt{a}$ with $\mathtt{g}$
2. indel cost δ (cost of not aligning a symbol)

Find best alignment (min/max non-crossing matching)

complexity

- two sequences: polynomial, Dynamic Programming $O(n^2)$
- k sequences: NP-hard (Wang,Jiang'94; Bonizzoni,Della Vedova'01)

Multiple Sequence Alignment

```
MAT-ER-
MOTHER-
MAD--RE
MAT--RE
MUTTER-
```

```
ACT-GG-
ACTCGG-
AGT--CT
CCT--GT
A-TTCG-
```

Motivations

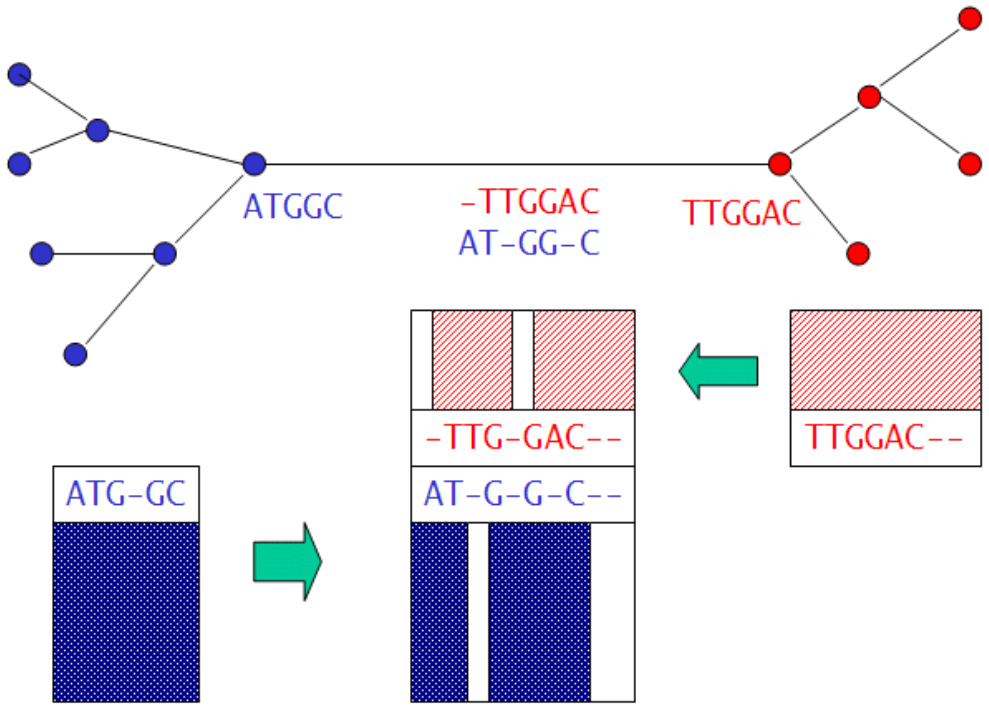
- Finding conserved patterns (functionally relevant)
- Clustering genomic sequences (e.g. protein families)
- Evolutionary studies (e.g. intra-species comparisons)
- ...

Exact multiple alignment

- Dynamic Programming approach $O(2^k n^k)$ for k sequences of length n
- In real-life it can be $k = 10$, $n = 1000$
- DP breaks down at $k = 4, n = 40$
- IP can go to $k = 6, n = 200$ (Kececioglu et al.'00, branch-and-cut)
- Still problem too difficult to solve exactly

Approximate multiple alignment

Trees can be used to guide alignments



Given a tree T, define $\mathcal{A}(T)$ the alignment it yields

Approximate multiple alignment

Routing cost of a tree

The pairwise distances induce a metric

Definition: $r(T)$, the routing cost of tree T, is the sum of all pair distances in the tree.

Theorem

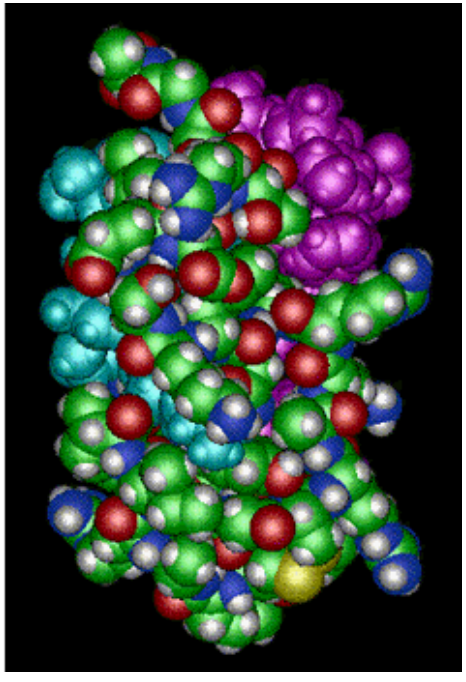
let T be any tree. Then the alignment $\mathcal{A}(S)$ has value $v(\mathcal{A}(s)) \leq r(T)$

Then, use T^* such that $r(T^*)$ is minimum

- Finding T^* still difficult, but good IP formulation (branch-and-price) exists (Fischetti,L,Serafini'02))
- Solution at least a 2-approx
- By comparison to LB: Min RC tree within 6% on avg. Applied to $k = 30$, $n = 400$

Alignment of protein structures

Alignment of Protein structures



- A Protein is a complex molecule with primary, linear structure (sequence of aminoacids) and 3-Dimensional structure (protein fold).
- Protein STRUCTURE determines its FUNCTION
- Drug Design calls for constructing peptides with a 3D shape complementary to a protein, so as to dock onto it.

Alignment of Protein structures

Motivation

- Discovery of protein function (shape determines function)
- Search in 3D data bases
- Protein classification and evolutionary studies
- ....

Contact maps

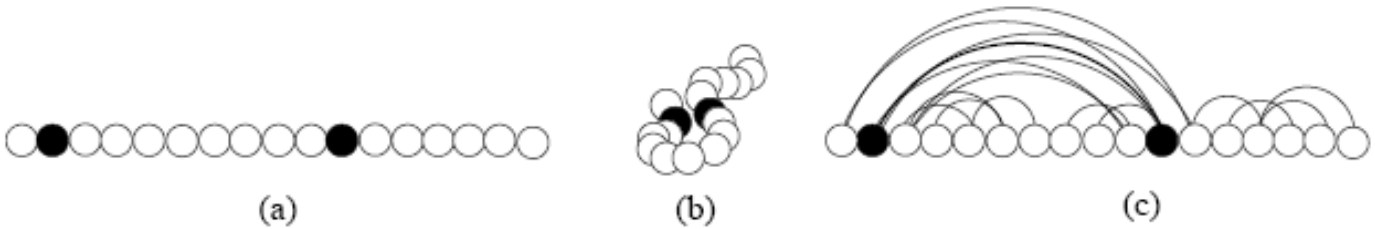


Fig. 1. (a) An unfolded protein. (b) After folding. (c) The contact map graph.

A contact map is a graph

- A node for each amino acid
- An edge (contact) between close aa (d < 5Å) in fold

Contact map alignment

Similar contact maps = similar 3D folds

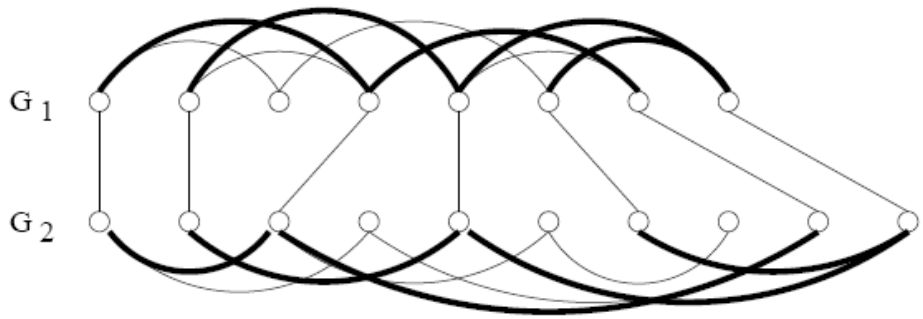


Fig. 2. An alignment of value 5.

Definition

CMO problem: find alignment maximizing shared contacts (overlap)

Problem is NP-hard (Goldman,Istrail,Papadimitriou,'99)

CMO

- IP formulation with x_{ij} (node-to-node) and $y_{iu,jv}$ (contact-to-contact) variables (L,Carr,Istrail,Walenz'00)
- Solved by Branch-and-Cut (clique inequalities)
- Compact optimization can replace Branch-and-Cut (speedup 10$\times$) (Carr,L'03)
- Lagrangian relaxation of QP formulation (subproblem weighted non-crossing matching) (Caprara,L'02)
- Reduction to maxclique on "structured" graphs (Barnes,Sokol,Strickland'05; Andonov,Yanev'03)

Computational results

- IP B&C: 1st time optimal alignment for PDB proteins
- Best approach: Lagrangian relaxation
- avg # nodes 100 # edges 200 time secs:mins
  - Up to 500 nodes, 800 edges for similar proteins
  - Trouble with 50 nodes, 100 edges for very dissimilar proteins

SNPs and haplotyping

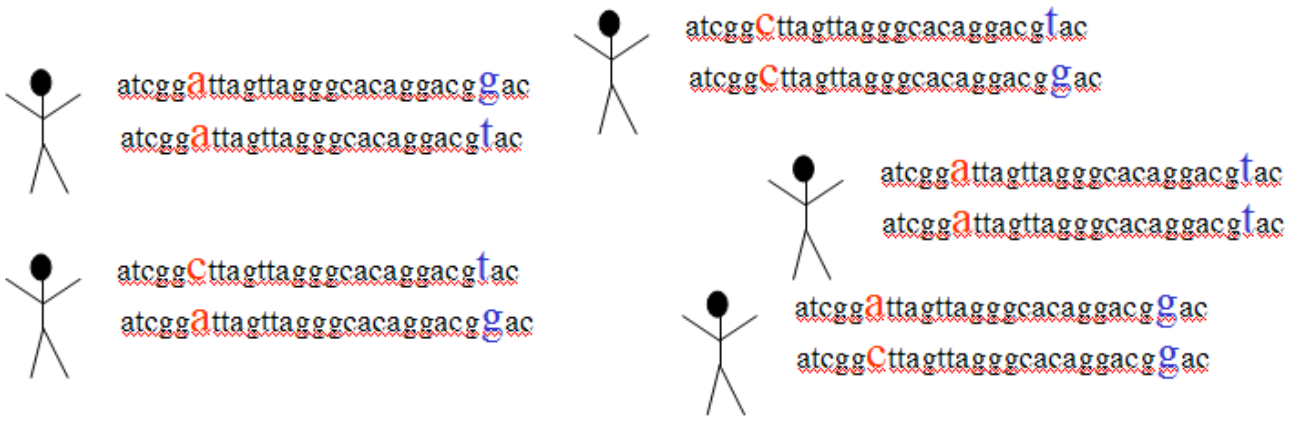
Single Nucleotide Polymorhism

Definition

Polymorphism: A feature that

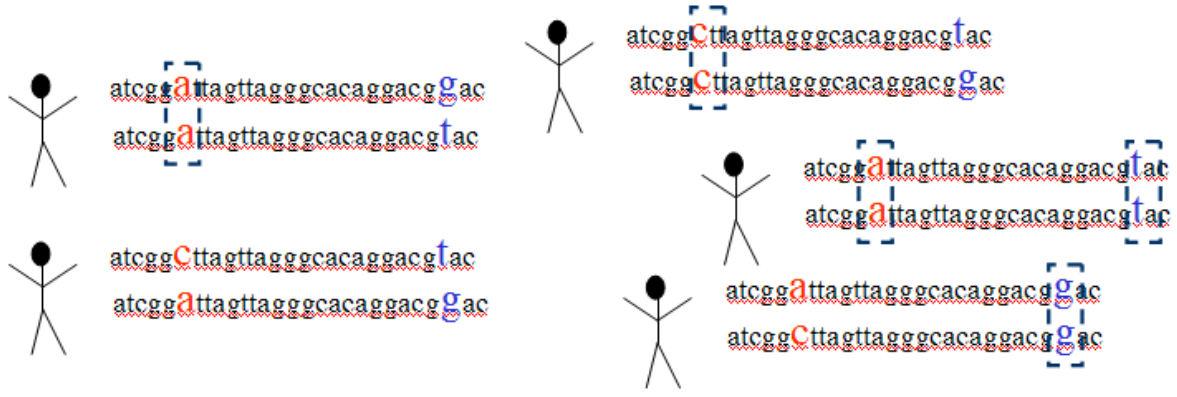
- each one possesses
- not identical for everyone

E.g., eye-color, blood type...
Typically, variants (alleles) are just few
Smallest polymorphism is content of specific base : SNP

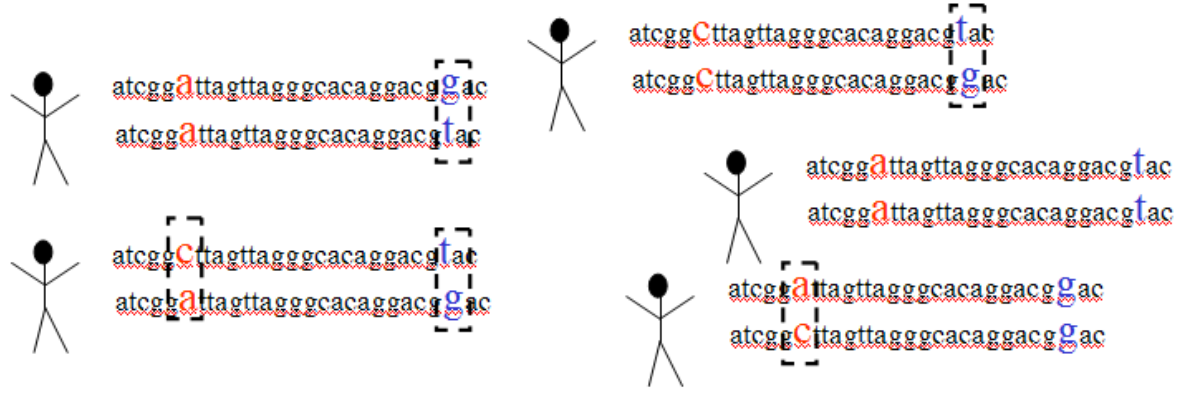


Humans are diploid

At each SNP, one can be homozygous



or heterozygous



Haplotypes

- **Haplotype:** string of SNPs alleles on a chromosome copy

Example

p.chr.	tgatTgtgaTccgaaAggTcctC	p. hapl.	TTATC
m.chr.	tgatAgtgaTccgaaGggTcctA	m. hapl.	ATGTA

Haplotypes are useful for

- Diagnostics
- Forensics
- Population genetics
- ....

Haplotypes are expensive to obtain in wet-lab

Haplotypes

- **Haplotype:** string of SNPs alleles on a chromosome copy

Example

p.chr.	tgatTgtgaTccgaaAggTcctC	p. hapl.	TTATC
m.chr.	tgatAgtgaTccgaaGggTcctA	m. hapl.	ATGTA

Haplotypes are useful for

- Diagnostics
- Forensics
- Population genetics
- ....

  Haplotypes are expensive to obtain in wet-lab

Haplotypes and genotypes

- **Haplotype:** string of SNPs alleles on a chromosome copy
- **Genotype:** conflation of both haplotypes

Example

haplotype	A	G	G	T	A	G
haplotype	T	G	A	A	A	G
genotype	A-T	G	A-G	A-T	A	G

genotype does not specify alleles origin

- Haplotypes most informative but expensive
- Genotypes ambiguous but cheap

Solution: retrieve haplotypes from genotypes

Haplotypes and genotypes

- **Haplotype:** string of SNPs alleles on a chromosome copy
- **Genotype:** conflation of both haplotypes

Example

haplotype	A	G	G	T	A	G
haplotype	T	G	A	A	A	G
genotype	A-T	G	A-G	A-T	A	G

genotype does not specify alleles origin

- Haplotypes most informative but expensive
- Genotypes ambiguous but cheap

Solution: retrieve haplotypes from genotypes

Haplotypes and genotypes

- **Haplotype:** string of SNPs alleles on a chromosome copy
- **Genotype:** conflation of both haplotypes

Example

haplotype	A	G	G	T	A	G
haplotype	T	G	A	A	A	G
genotype	A-T	G	A-G	A-T	A	G

genotype does not specify alleles origin

- Haplotypes most informative but expensive
- Genotypes ambiguous but cheap

Solution: retrieve haplotypes from genotypes

Haplotypes and genotypes

- **Haplotype:** string of SNPs alleles on a chromosome copy
- **Genotype:** conflation of both haplotypes

Example

haplotype	A	G	G	T	A	G
haplotype	T	G	A	A	A	G
genotype	A-T	G	A-G	A-T	A	G

genotype does not specify alleles origin

- Haplotypes most informative but expensive
- Genotypes ambiguous but cheap

Solution: retrieve haplotypes from genotypes

Haplotyping problem

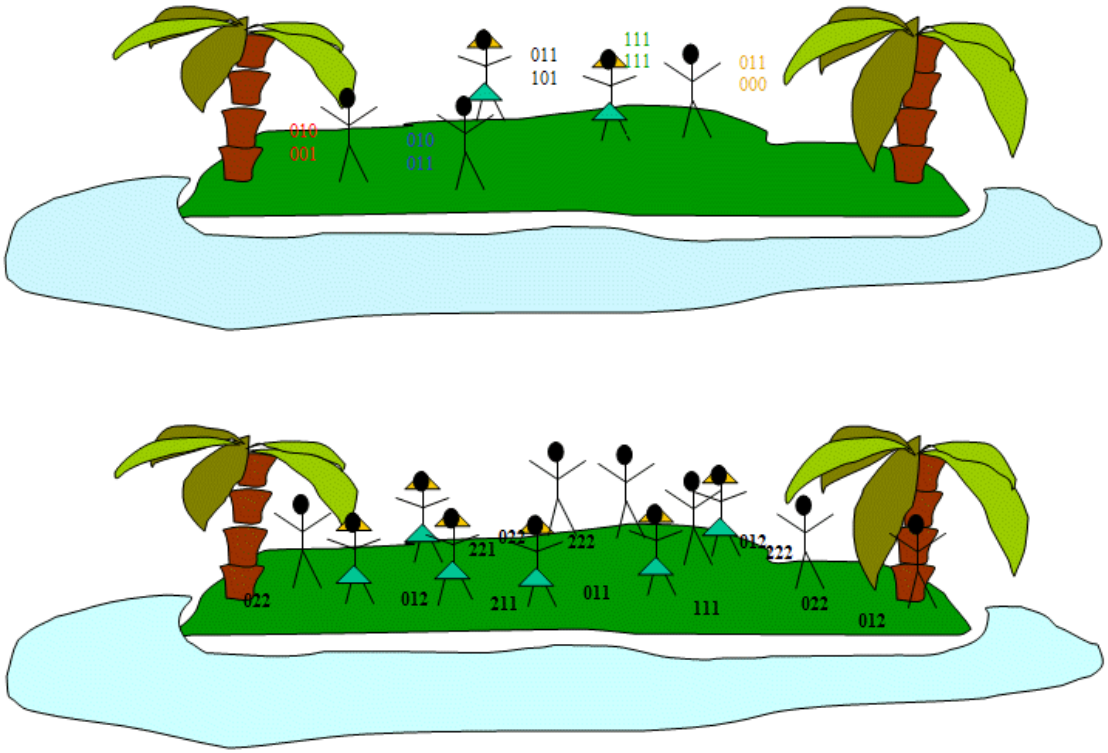
Problem statement

Given G (genotypes) find H (haplotypes) such that each $g \in G$ has a resolution $\{h', h''\} \subset H$

The problem would be trivial unless we introduce (biology-driven) constraints and/or objective function

Haplotyping problem

Biology reasons point to reuse of haplotypes



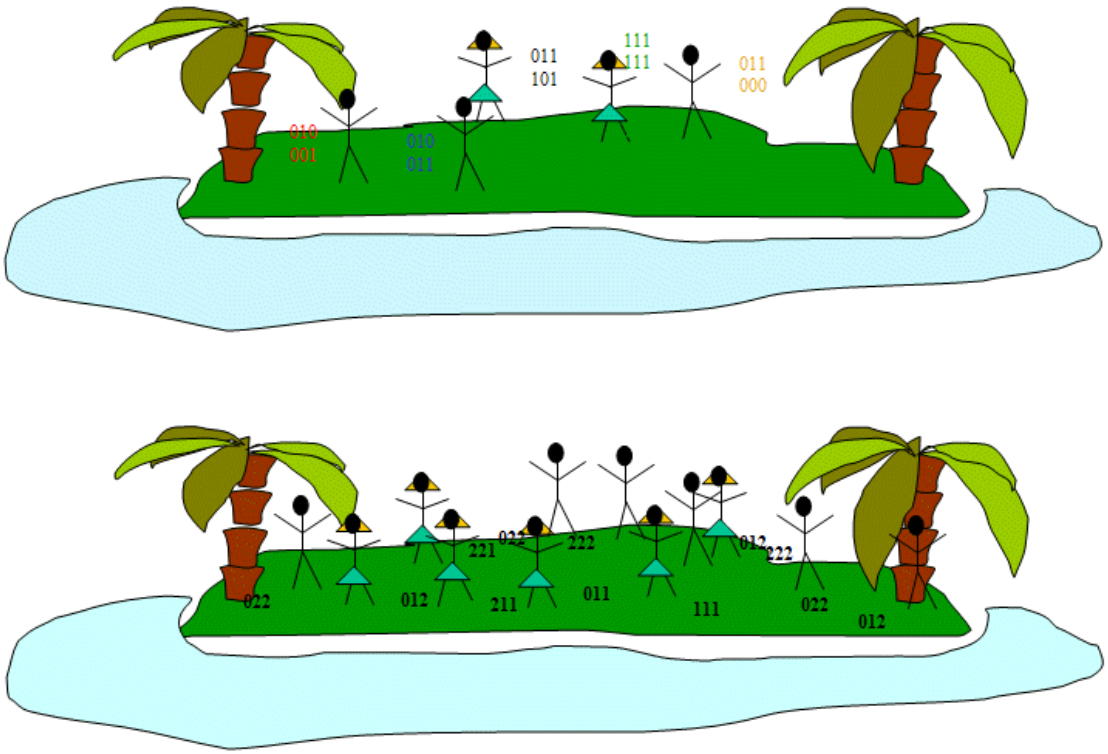
Objective function (parsimony):

Find H that resolves G and s.t. $|H|$ is minimum

Theorem (L,Pinotti,Rizzi'03) The problem is APX-hard

Haplotyping problem

Biology reasons point to reuse of haplotypes



Objective function (parsimony):

Find H that resolves G and s.t. $|H|$ is minimum

Theorem (L,Pinotti,Rizzi'03) The problem is APX-hard

Ambiguous genotypes

- g is ambiguous if $\#\text{het}(g) > 1$
- If $\#\text{het}(g) = k$, then it has 2^{k-1} resolutions

$$g = HatHH = \begin{pmatrix} catcc+ \\ gatgg \end{pmatrix} \vee \begin{pmatrix} catcg+ \\ gatgc \end{pmatrix} \vee \begin{pmatrix} catgc+ \\ gatcg \end{pmatrix} \vee \begin{pmatrix} catgg+ \\ gatgg \end{pmatrix}$$

IP formulation (Gusfield'03) w/binary variables

- x_h for each possible haplotype h
- $y_{h',h''}$ for each possible resolution $\{h', h''\}$

Ambiguous genotypes

- g is ambiguous if $\#\mathrm{het}(g) > 1$
- If $\#\mathrm{het}(g) = k$, then it has 2^{k-1} resolutions

$$g = HatHH = \begin{pmatrix} catcc+ \\ gatgg \end{pmatrix} \vee \begin{pmatrix} catcg+ \\ gatgc \end{pmatrix} \vee \begin{pmatrix} catgc+ \\ gatcg \end{pmatrix} \vee \begin{pmatrix} catgg+ \\ gatgg \end{pmatrix}$$

IP formulation (Gusfield'03) w/binary variables

- x_h for each possible haplotype h
- $y_{h',h''}$ for each possible resolution $\{h', h''\}$

Computational results

- IP formulation has exponential # of both vars and constraints
- Some tricks are applied to # number of variables (no col. generation)
- Gusfield solves small instances by B& B (up to 50 individuals on 30 SNPs of which few, $\leq$ 15 ambiguous)
- Running times range from seconds to hours
- Approach breaks down even for small (but very ambigous) instances (e.g. 40 $\times$ 40 over 25 ambiguous)

Alternative formulations

- Poly-size ILP (Brown,Harrower'03; Bafna et al.'03)
- These models yield much weaker bounds than (IP1)
- (Brown,Harrower'04) propose cuts to improve bound. Results comparable with Gusfield's=>applicable only to small problems
- Semidefinite Programming approach to this problem has also been proposed (Kalpakis,Namjoshi'05)
- New ideas needed to move up to next level. Col generation approach via Set Covering under investigation

Polynomial cases and approximation

From IP formulation:

Theorem

(Cilibrasi,van Iersel,Kelk,Tromp'05; L,Rizzi,'05) If $\#het(g) \leq 2$ for all g, then problem is polynomial.

Theorem

(L,Pinotti,Rizzi'04) If $\#het(g) \leq k$ for all g, then there is polynomial 2^{k-1}-approximation.

Recently an $O(\log m)$-approx, semidefinite prog-based (Huang,Chao,Chen'05)

Several other imporant problems:

- haplotyping from genotype fragments
- haplotyping for disease association
- (im)perfect phylogeny haplotyping

Moreover,

- Protein folding and docking
- Virus barcoding and feature selection
- Phylogeny reconstruction
- ....

But (luckily) time's up

For Further Reading (www.dimi.uniud.it/lancia)

G. Lancia.

Applications to Computational Molecular Biology.

in “Handbook on Modeling for Discrete Optimization”, (G. Appa, P. Williams, P. Leonidas and H. Paul eds),

Kluwer International Series in Operations Research and Management Science, Vol. 88, 2006.

G. Lancia.

Integer Programming Models for Computational Biology Problems.

Journal of Computer Science and Technology, 19(1):60–77, 2004.

H. Greenberg, W. Hart and G. Lancia.

Opportunities for Combinatorial Optimization in Computational Biology.

Informs Journal on Computing, 16(3):1–22, 2004.