Quelques questions en analyse des génomes

Alessandra Carbone
Département d'Informatique &
Equipe de Génomique Analytique, INSERM
Université Pierre et Marie Curie, Paris 6
Alessandra.Carbone@lip6.fr

Quelques applications des méthodes d'optimisation

- biais des codons et analyse de l'espace des codons.
Comment peut-on utiliser cette information pour parler d'un espace d'organismes? L'importance de définir des nouvelles distances dans ces espaces formels pour en tirer de l'information biologique.

- alignement des séquences
Alignement par paires et alignement multiple

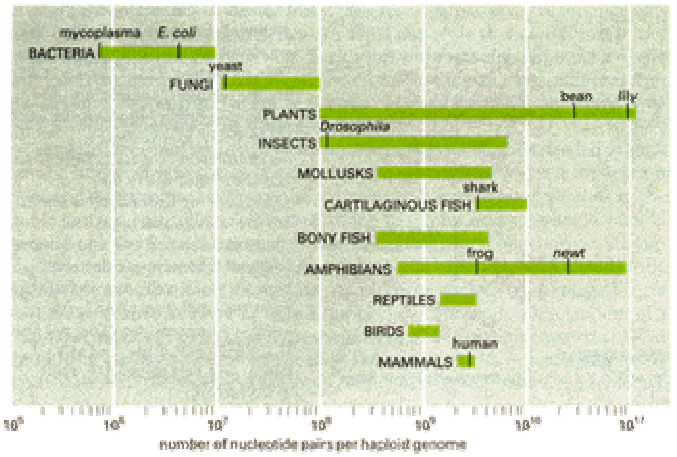
- dynamique des génomes
Algorithmes de reconstruction du processus de réarrangement des génomes

- reconstruction des génomes et des metagénomes.
Des nouvelles questions théoriques s'ouvrent.

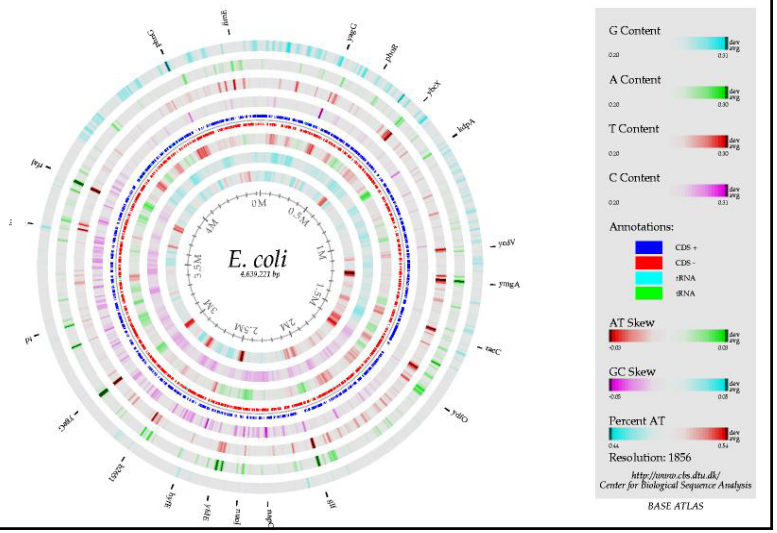
Aujourd'hui, il n'existe pas d'histoire cohérente sur les applications de l'optimisation combinatoire en bioinformatique.

Nous sommes pour le moment simplement très vigilants aux possibles applications.

Background : genomes and lengths

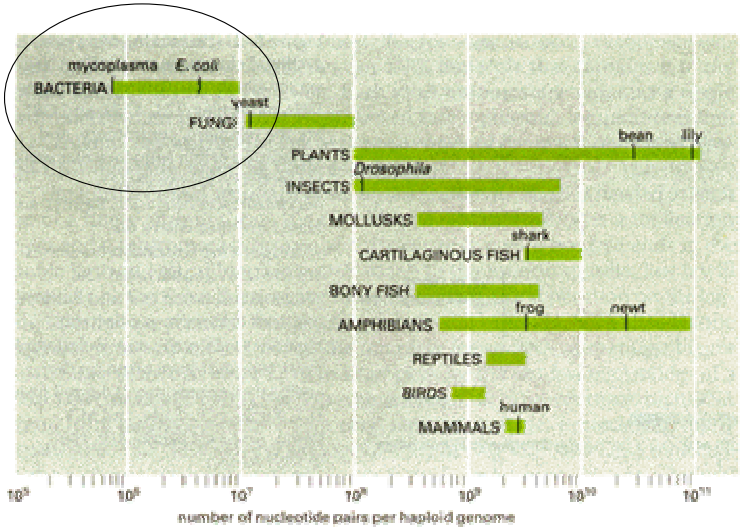


Large scale statistical analysis of a genome



Un nombre de gènes comparable…

Organism	Year	Millions of bases sequenced	Total coverage (%)	Coverage of euchromatin (%)	Predicted number of genes	Number of genes per million bases sequenced
Saccharomyces cerevisiae	1996	12	93	100	5,800	483
Caenorhabditis elegans	1998	97	99	100	19,099	197
Drosophila melanogaster	2000	116	64	97	13,601	117
Arabidopsis thaliana	2000	115	92	100	25,498	221
Human chromosome 21	2000	34	75	100	225	7
Human chromosome 22	1999	34	70	97	545	16
Human genome rough draft (public sequence)	2001	2,693	84	90	31,780	12
Human genome rough draft (Celera sequence)	2001	2,654	83	88–93	39,114	15

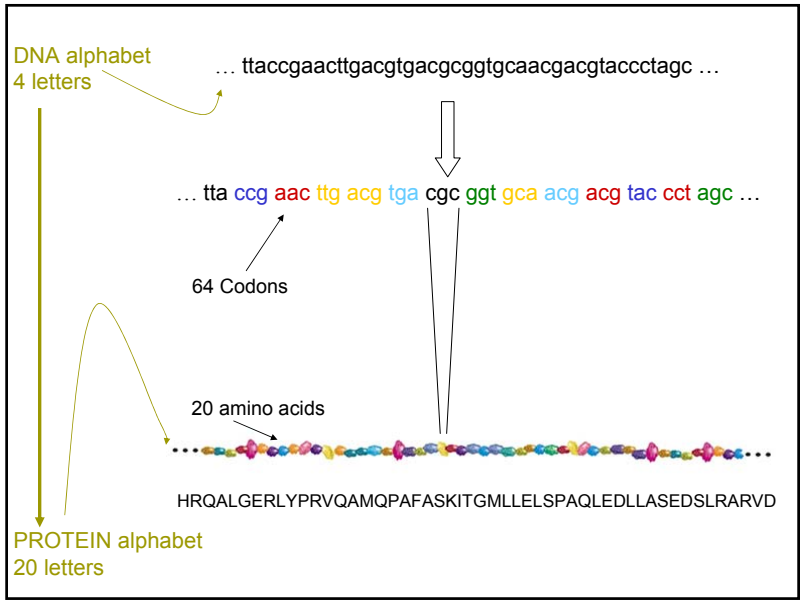


TOPIC 1
Genome analysis and statistical bias

Nucleotides A T C G

Di-nucleotides AT CG AG AC …

Triplets ATG CGG TAG CCT …



g

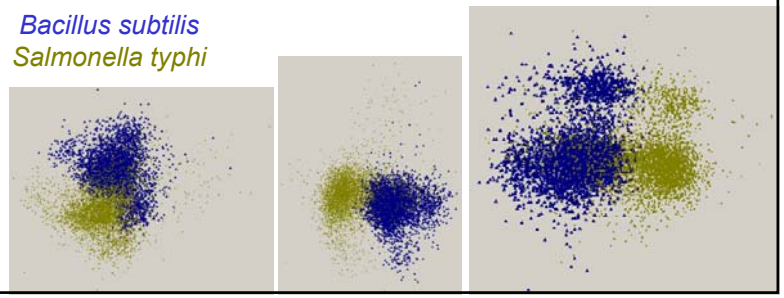
$g = [x_{1,g}\ x_{2,g}\ \dots\ x_{64,g}]$ $x_{i,g}$ relative frequency of codon i in g

Vector normalisation:

$(x_{i,g} - \underline{x}_i) / \sigma_i$ $\underline{x}_i$ mean of frequencies $x_{i,g}$

σ_i standard deviation of $x_{i,g}$

Bacillus subtilis
Salmonella typhi



$g = [x_{1,g}\ x_{2,g}\ \dots\ x_{64,g}]$ $x_{i,g}$ relative frequency of codon i in g

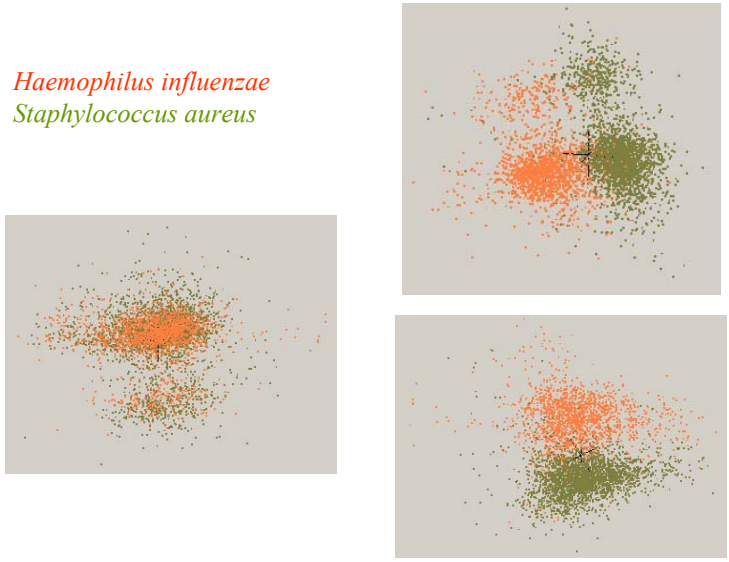
Vector normalisation:

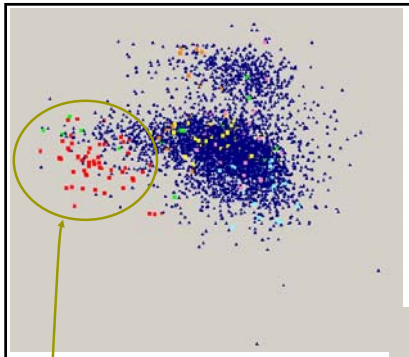
$(x_{i,g} - \underline{x}_i) / \sigma_i$ $\underline{x}_i$ mean of frequencies $x_{i,g}$

σ_i standard deviation of $x_{i,g}$

similar geometry
&
translation + rotation

Haemophilus influenzae
Staphylococcus aureus

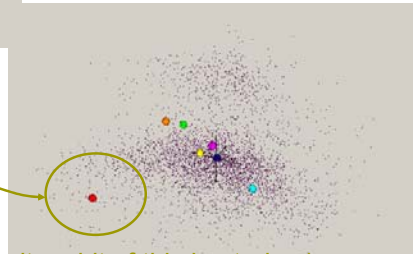




Gene functions:

Ribosomal proteins
ATP binding proteins
IS proteins
NADH proteins
Flagellar biosynthesis proteins
Lipoproteins, membrane proteins, transport proteins

Proteins codifying for “translation”, glycolysis … Roughly speaking, they are the most expressed in *E.coli*

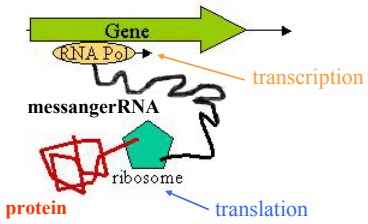


E.coli *(biologists know quite a bit of this bacterium)*

Is there some interesting statistical bias in genomes due to codon usage ?
Can we exploit such bias to extract information of biological interest ?

Background : translation

g



DNA alphabet
4 letters: ACGT

↓

Protein Alphabet
20 letters

Background : redundant genetic code

2nd base in codon

1st base in codon	U	C	A	G	3rd base in codon
U	Phe	Ser	Tyr	Cys	U
	Phe	Ser	Tyr	Cys	C
	Leu	Ser	STOP	STOP	A
	Leu	Ser	STOP	Trp	G
C	Leu	Pro	His	Arg	U
	Leu	Pro	His	Arg	C
	Leu	Pro	Gln	Arg	A
	Leu	Pro	Gln	Arg	G
A	Ile	Thr	Asn	Ser	U
	Ile	Thr	Asn	Ser	C
	Ile	Thr	Lys	Arg	A
	Met	Thr	Lys	Arg	G
G	Val	Ala	Asp	Gly	U
	Val	Ala	Asp	Gly	C
	Val	Ala	Glu	Gly	A
	Val	Ala	Glu	Gly	G

Preferential codons: codons that appear with higher frequency in most genes

Background : bias in codon usage & preferred codons

high tRNA number
high expression

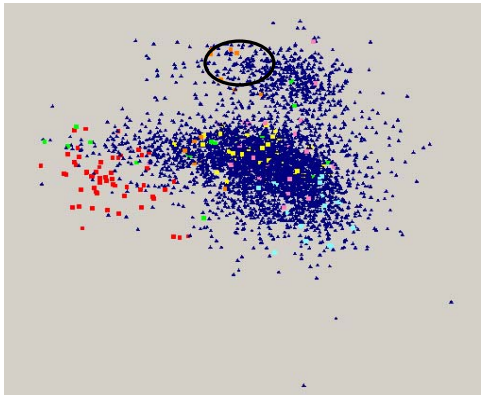
correlated to (experimentally)

codon preference

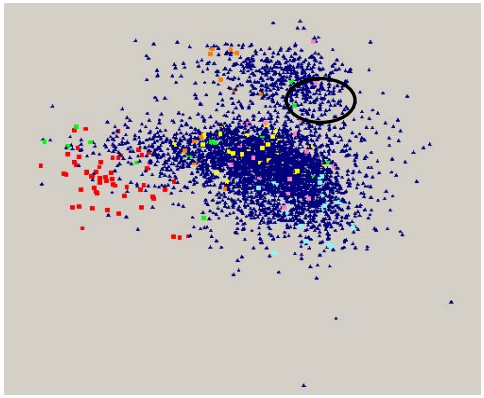
in *E.coli* and other organisms that reproduce rapidly

Codon preference and tRNA : Ikemura, 1985; Bennetzen and Hall, 1982; Bulmer, 1987; Gouy and Gautier, 1982.
tRNA and elongation rate : Varenne *et al.*, 1984.
High expression and codon preference : Grantham *et al.*, 1980; Wada *et al.*, 1990; Sharp and Li, 1987; Sharp *et al.*, 1986; Médigue *et al.*, 1991; Shields and Sharp, 1987; Sharp *et al.*, 1988; Stenico *et al.*, 1994.

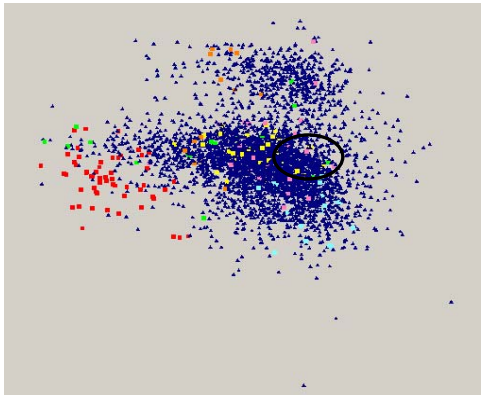
How to define “codon bias” and how to search for highly biased genes in an automatic manner?



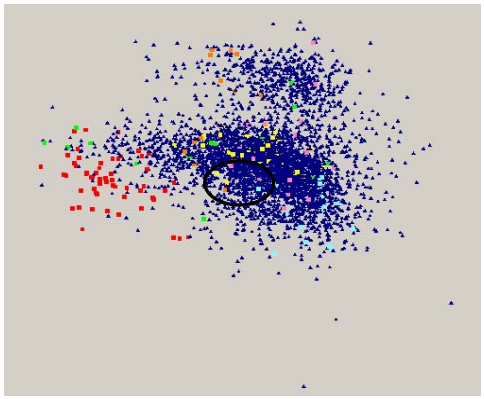
How to define “codon bias” and how to search for highly biased genes in an automatic manner?



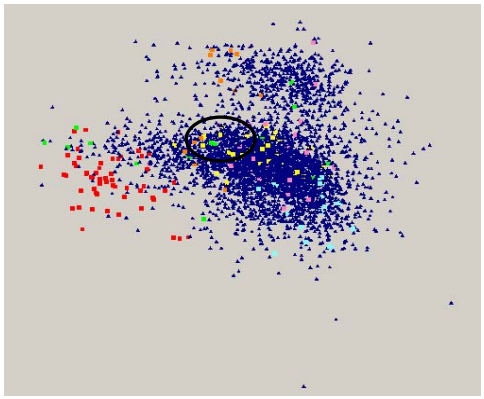
How to define “codon bias” and how to search for highly biased genes in an automatic manner?



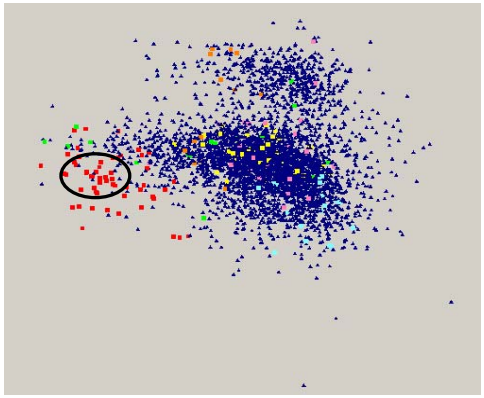
How to define “codon bias” and how to search for highly biased genes in an automatic manner?



How to define “codon bias” and how to search for highly biased genes in an automatic manner?



How to define “codon bias” and how to search for highly biased genes in an automatic manner?



g $(\Pi_{k=1\ldots11}\, w_k)^{1/11}$

w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8 w_9 w_{10} w_{11}

$$CAI(g) = (\Pi_{k=1\ldots L}\, w_k)^{1/L}$$

Codon Adaptation Index

(Sharp & Li, 1987)

L number of codons in g

w_k $\dfrac{\text{frequency of the } k^{th} \text{ codon of g in S}}{\text{frequency of the dominant synonymous codon in S}}$

proteines codifying for “translation”, glycolysis …

Let S be a set of genes and g be a gene

$$CAI(g) = (\Pi_{k=1\ldots L}\, w_k)^{1/L}$$ (Sharp & Li, 1987)

Codon Adaptation Index

L number of codons in g

w_k $\dfrac{\text{frequency of the } k^{th} \text{ codon of g in S}}{\text{frequency of the dominant synonymous codon in S}}$

proteines codifying for “translation”, glycolysis …

Let S be a set of genes and g be a gene

$$CAI(g) = (\Pi_{k=1\ldots L}\, w_k)^{1/L}$$ (Sharp & Li, 1987)

Codon Adaptation Index

L number of codons in g

w_k $\dfrac{\text{frequency of the } k^{th} \text{ codon of g in S}}{\text{frequency of the dominant synonymous codon in S}}$

we compute S

Let S be a set of genes and g be a gene

$$SCCI(g) = (\Pi_{k=1\ldots L}\, w_k)^{1/L}$$

Self Consistent Codon Index

L number of codons in g

w_k $\dfrac{\text{frequency of the } k^{th} \text{ codon of g in S}}{\text{frequency of the dominant synonymous codon in S}}$

we compute S

Let S be a set of genes and g be a gene

$$SCCI(g) = (\Pi_{k=1\ldots L}\, w_k)^{1/L}$$

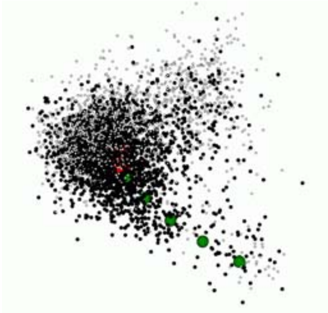
Self Consistent Codon Index

Self consistency condition

SCCI values on genes in S are **maximal** :
$SCCI(G/S) \leq SCCI(S)$, G is the set of all genes

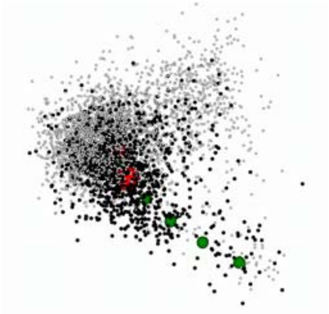
Idea of the algorithm:

- Compute the weight of the codons over the whole genome and compute afterwards SCCI values for all genes
- Select the 50% of genes with the highest SCCI value
- Repeat the iteration and select the 25% of the genes
- and so on… until we arrive to the 1% of genes in the original set.
- … then repeat the iteration on the 1% of genes with highest SCCI until convergence is reached.



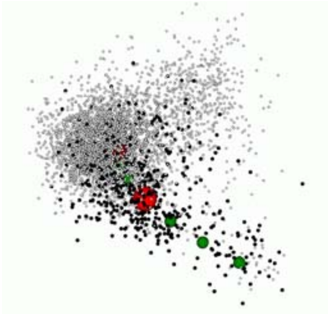
Idea of the algorithm:

- Compute the weight of the codons over the whole genome and compute afterwards SCCI values for all genes
- Select the 50% of genes with the highest SCCI value
- Repeat the iteration and select the 25% of the genes
- and so on… until we arrive to the 1% of genes in the original set.
- … then repeat the iteration on the 1% of genes with highest SCCI until convergence is reached.



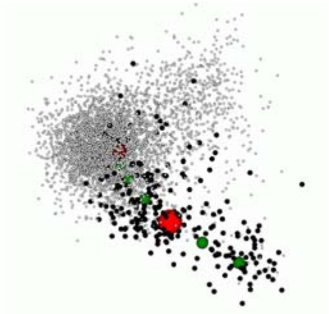
Idea of the algorithm:

- Compute the weight of the codons over the whole genome and compute afterwards SCCI values for all genes
- Select the 50% of genes with the highest SCCI value
- Repeat the iteration and select the 25% of the genes
- and so on… until we arrive to the 1% of genes in the original set.
- … then repeat the iteration on the 1% of genes with highest SCCI until convergence is reached.



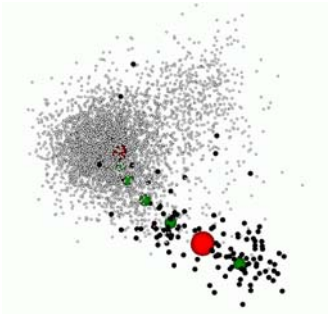
Idea of the algorithm:

- Compute the weight of the codons over the whole genome and compute afterwards SCCI values for all genes
- Select the 50% of genes with the highest SCCI value
- Repeat the iteration and select the 25% of the genes
- and so on… until we arrive to the 1% of genes in the original set.
- … then repeat the iteration on the 1% of genes with highest SCCI until convergence is reached.



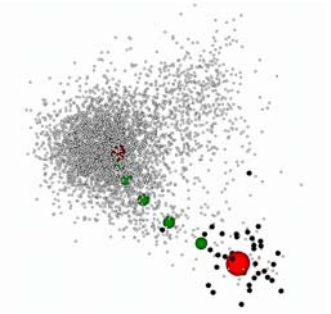
Idea of the algorithm:

- Compute the weight of the codons over the whole genome and compute afterwards SCCI values for all genes
- Select the 50% of genes with the highest SCCI value
- Repeat the iteration and select the 25% of the genes
- and so on… until we arrive to the 1% of genes in the original set.
- … then repeat the iteration on the 1% of genes with highest SCCI until convergence is reached.



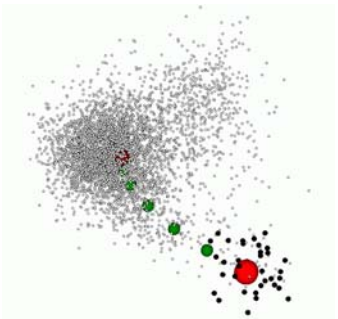
Idea of the algorithm:

- Compute the weight of the codons over the whole genome and compute afterwards SCCI values for all genes
- Select the 50% of genes with the highest SCCI value
- Repeat the iteration and select the 25% of the genes
- and so on… until we arrive to the 1% of genes in the original set.
- … then repeat the iteration on the 1% of genes with highest SCCI until convergence is reached.



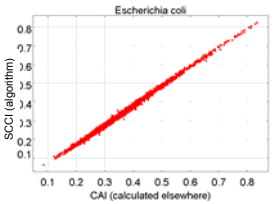
Idea of the algorithm:

- Compute the weight of the codons over the whole genome and compute afterwards SCCI values for all genes
- Select the 50% of genes with the highest SCCI value
- Repeat the iteration and select the 25% of the genes
- and so on… until we arrive to the 1% of genes in the original set.
- … then repeat the iteration on the 1% of genes with highest SCCI until convergence is reached.



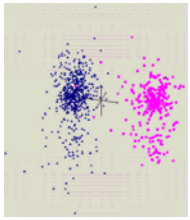
S found by the algorithm: *E.coli*

(*E.coli* reproduce rapidly)



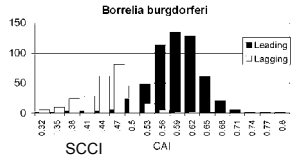
Gene	Annotation
tufA	protein chain elongation factor EF-Tu
tufB	protein chain elongation factor EF-Tu
tsf	protein chain elongation factor EF-Ts
fusA	GTP-binding protein chain elongation factor EF-G
mopA	chaperonin GroEL
dnaK	heat shock protein DnaK
cspA	cold shock protein 7.4
tig	trigger factor
ompA	outer membrane protein
ompX	outer membrane protein
ompC	outer membrane protein
lpp	murein lipoprotein
pal	peptidoglycan-associated lipoprotein
yaiU	putative flagellin structural protein
yfiD	putative formate acetyltransferase
eno	diadenosine tetraphosphatase
tpiA	triosephosphate isomerase
pgk	phosphoglycerate kinase
gapA	glyceraldehyde-3-phosphate dehydrogenase A
fba	fructose-bisphosphate aldolase class II
pykF	pyruvate kinase I
pflB	formate acetyltransferase 1
ahpC	alkyl hydroperoxide reductase C22 subunit
sodA	superoxide dismutase SodA
tktA	transketolase 1/2 isozyme
rpoC	RNA polymerase beta prime subunit
rpsI	30S ribosomal subunit protein S9
rpsA	30S ribosomal subunit protein S1
rpsB	30S ribosomal subunit protein S2
rpsC	30S ribosomal subunit protein S3
rpsU	30S ribosomal subunit protein S21
rplA	50S ribosomal subunit protein L1
rplY	50S ribosomal subunit protein L25
rplI	50S ribosomal subunit protein L9
rplL	50S ribosomal subunit protein L7/L12
rplC	50S ribosomal subunit protein L3
rpmE	50S ribosomal subunit protein L31
rplB	50S ribosomal subunit protein L2
rplK	50S ribosomal subunit protein L11
rpmI	50S ribosomal subunit protein A
rpmA	50S ribosomal subunit protein L27
rplD	50S ribosomal subunit protein L4, regulates expression of S10 operon

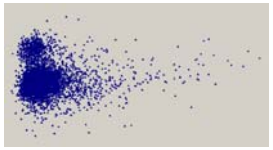
SCCI : a universal measure



Borrelia burgdorferi

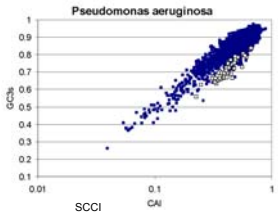
Strand bias





Pseudomonas aeruginosa

GC3 bias



The set of biased genes

- is **unique** (for the organisms we checked, ~210)
- **exists** also for organisms that do not have an evolutionary tendency explained with translational pressure.

For **any** bacteria we can compute:

\+ dominant bias: strand bias, GC3, AT, ...
\+ numerical criteria to determine the strength of translational bias

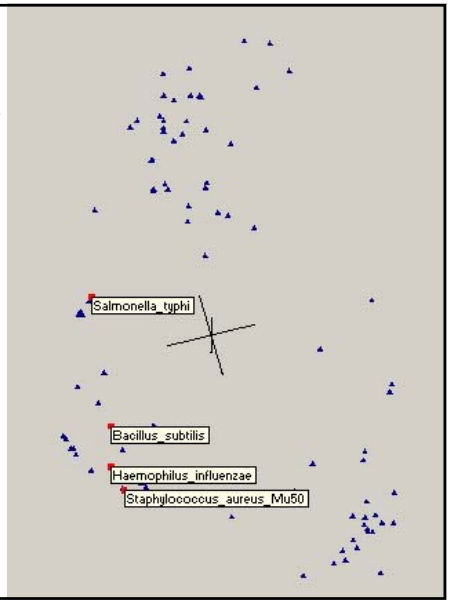
Randomised version

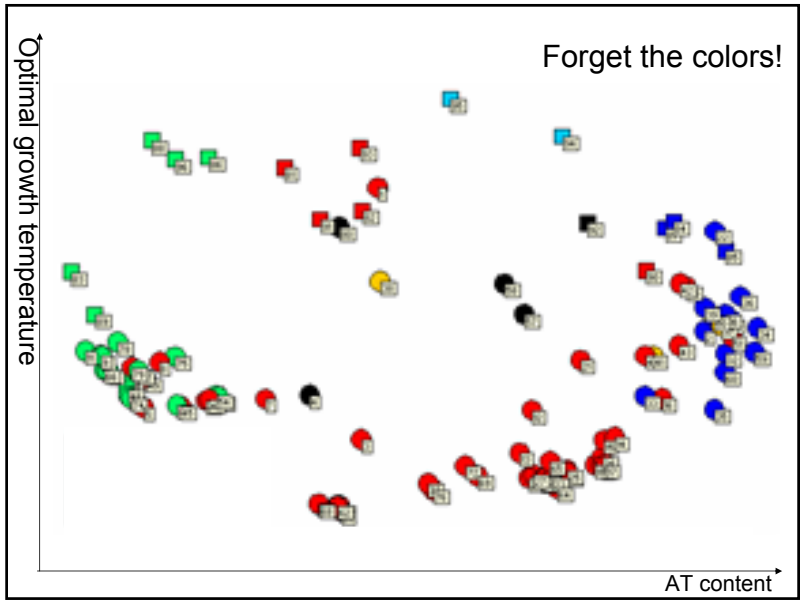
- Randomly choose the 1% of genes in S
- Compute the weights and the SCCI values
- Select the 1% of genes with highest SCCI value
- Repeat the iteration until the algorithm converges

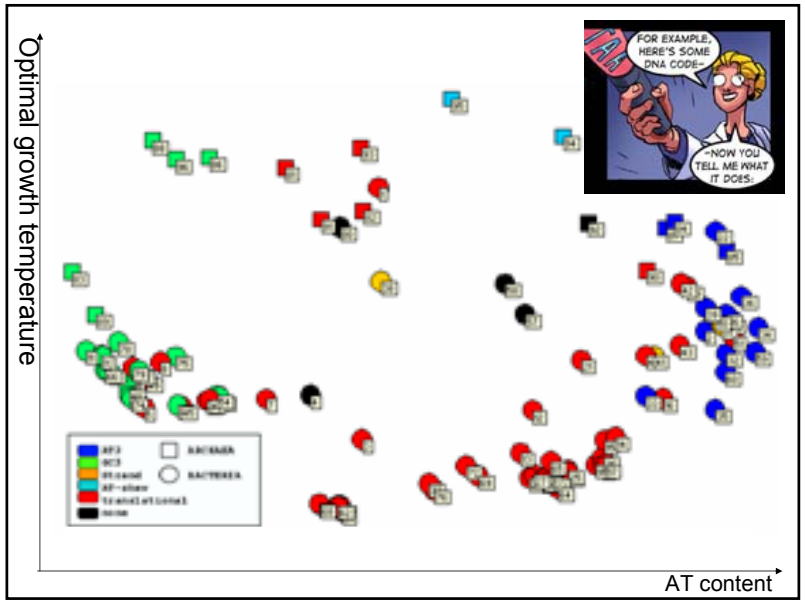
Bacteria and Archaea in SCCI codon space

An organism is a 64-dim vector where

coordinate
=
SCCI codon weight

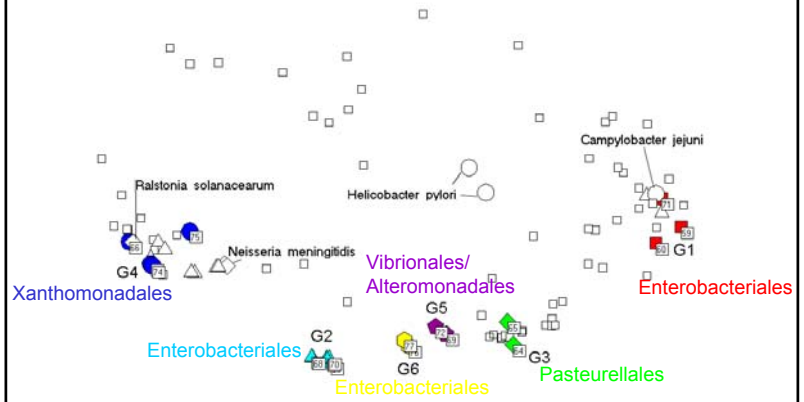






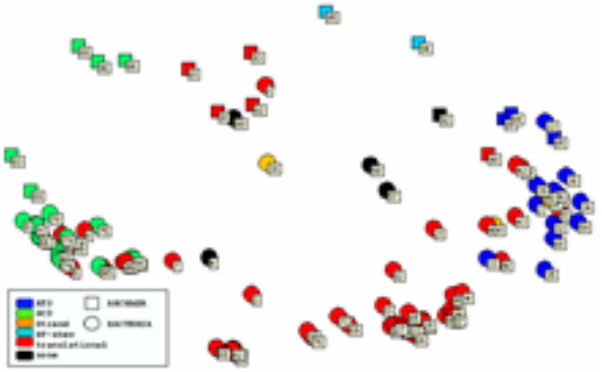
Can we exploit the geometry of the space to derive functional characteristics of groups of organisms?

Phylogenetically related families : γ-proteobacteria



Similar physiology and habitat

Organisms at small distance: similar physiology and habitat



Environmental clusters :

soil bacteria
enterics
symbions
spore formers
small intercellular pathogens
small extracellular pathogens

New methods to classify organisms are demanded.

Most existing organisms are unicellular organisms and most of them are going to be studied through novel approaches that demand for distinguishing the diversity within the environment.

New classifications can help organising the huge space of organisms.

Coherence in the organisms space based on SCCI

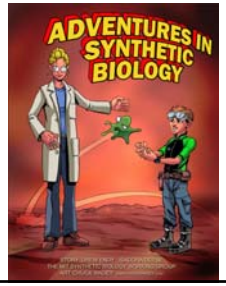
Can we use this signal to deduce some more biological information ?

This analysis helps to identify genes essential for bacterial life
(**minimal gene sets)**

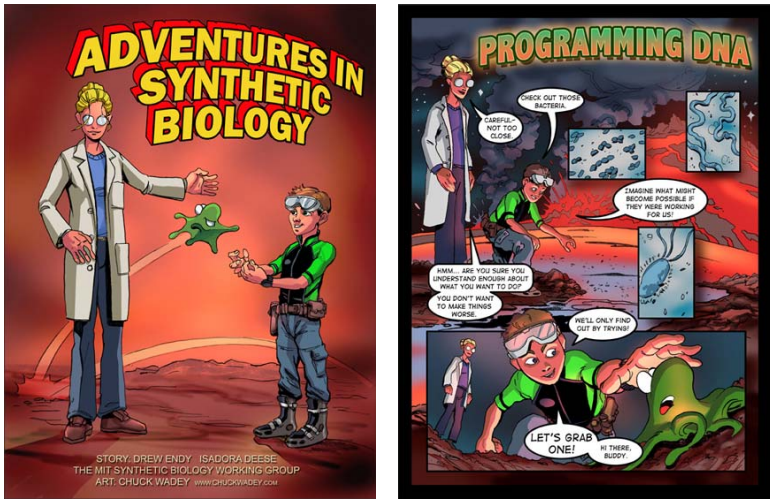
Why to search for a minimal genome ?

Add genes to transform *Mycoplasma* in a “useful” bacteria.

Remedy against environmental pollution, new industrial chemical substances production, insuline production…



Towards a genome synthesis and a genome programming



Craig Venter, November 2002

Synthesis of a bacterial genome

the chromosome will be inserted in a living cell (whose genetic material has been removed) to verify if it can direct normal functional activities of the organism.

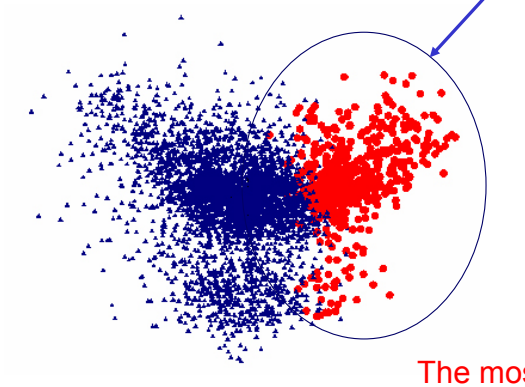
Clyde Hutchison, 1999 (*Science* 286, 2165-2169):

Gene knock out (517) of *Mycoplasma genitalium* (580kb), and estimation of how many genes are necessary to life over 517: about 300 to survive.

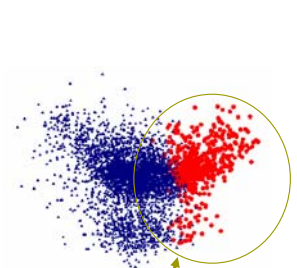
Eckard Wimmer, 2002 (*Science* 297, 1016-1018):

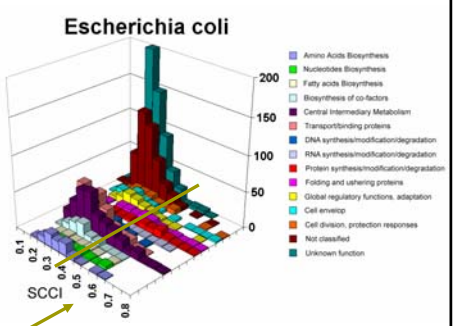
Synthesis of a poliovirus that infects cells! (~7500b)

Are they the **most essential** genes for bacterial life ?



The most expressed



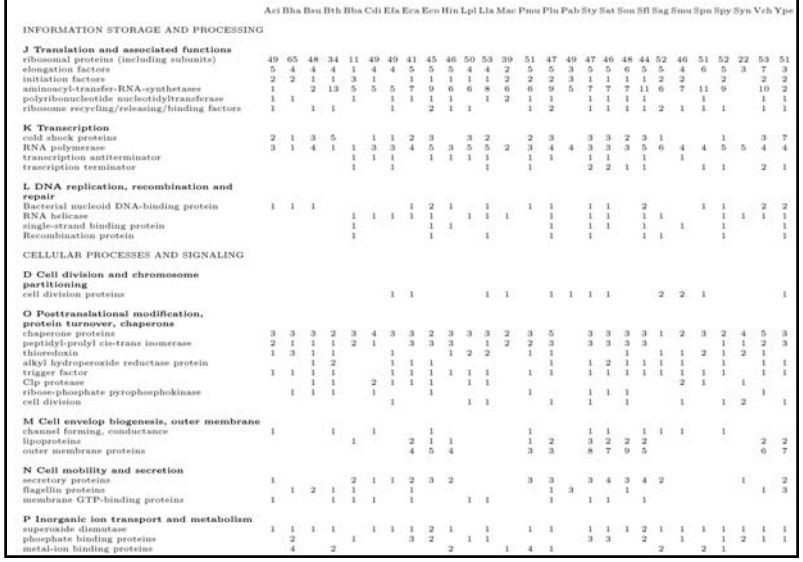


Core genes

We look at the tail

SCCI(g) > $\mu+\sigma$ = 0.42

Map of core genes of 27 organisms (based on 200 most biased genes)



Genes with **specific metabolic functions** are in the tail

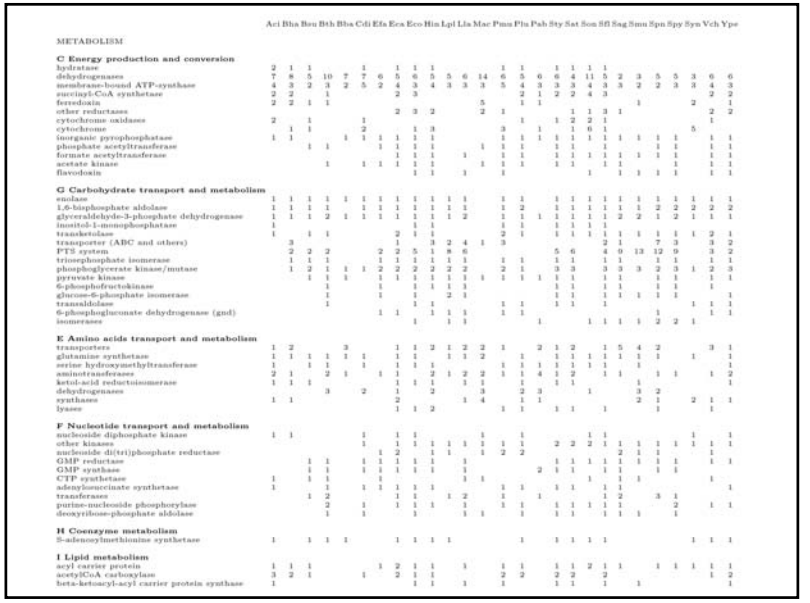
Photosynthesis metabolism : *Synechocystis*
- Phycobilisome proteins
- Photosystem I and II
- Fructose-1,6-bisphosphate-aldolase

Methan metabolism : *Methanosarcina acetivorans*
- Methanol-5 hydroxybenzimidazolylcobamideco methyltransferase
- Methyl coenzyme M reductase
- Methylcobamide methyltransferase isozyme M
- Corrinoid proteins
- Ack, Pta, cdhA

Ferrodoxin metabolism : *Pyrococcus abyssi*
- Ferredoxin
- Ferredoxin oxidoreductase
- Keto-valine-ferredoxin oxidoreductase γ-chain

Carbohydrates metabolism : *Streptococcus mutans*
- Transport and metabolism of cellobiose, sucrose, beta-glucoside
- Metabolism of mannitol
- Genes for metabolism of glucose, fructose, mannose, maltose/maltodextrin

Stress response genes are in the tail



“Fitting” with…

Experiments : transposomal mutagenesis & RNA silencing

B.subtilis	*M.genitalium*	*H.influenzae*	*E.coli*
300 genes/~4000 (Itaya, 1995)	**265** genes / 517 (Hutchison et al., 1999)	**670** genes/ ~1272 (Akerley et al. 2002)	**620** genes / 3746 (Gerdes et al. 2003)
248 genes/~4100 (Kobayashi, 2003)	**382** genes / 482 (Hutchison et al., 2006)		**234** genes / 2994 (Hashimoto et al. 2005)

S.cerevisiae	*C.elegans*	*S.aureus*	*S.pneumoniae*
1105 genes/ 5916 (Giaever et al. 2002)	**1722** genes/ 19427 (Kamath et al. 2003)	**150** genes (Yi et al. 2001)	**110** genes (Thanassi et al. 2002)

Comparative genomics

2 genomes	34 genomes	100 genomes	147 genomes
256 genes (Mushegian & Koonin 1996)	**80** genes (Harris et al 2003)	**60** genes (Koonin et al. 2003)	**35** genes (Charlebois & Doolittle 2004)

Number of genes in the minimal set depends on

Experiments:

- life/environmental conditions of the organism during the experiment

  ⟶ bacteria live in very good lab conditions

Computational detection of sequence homology:

- parameters and tools to detect homologies

  ⟶ there are genomes with more than 60% of genes with unknown function

Genes relevant to **environmental conditions** are missing

Stress response genes are missing

Genes with **uncharacterized functions** are missing

What we learn from this first story

1. algorithms + statistics to analyse microbial genomes without using biological information — robust biological phenomena
2. space of genomes : from phylogeny to environmental classification — new mathematical measures are needed
3. essential genes — looking at data from sequences & difficult comparison with experimental data
4. essential metabolic pathways
5. host-phage co-evolution

Est-ce qu'on peut exploiter

la géometrie de l'espace des organismes
la géometrie de l'espace des gènes

pour reduire l'espace de recherche des fonctions ?

PROBLEME CLE: **ANNOTATION**

séquences similaires ont une origine commune et souvant une fonction similaire

Topic 2
Alignement des séquences

Annotation des genomes: recherche d'**homologie** de séquences (les séquences sont issues d'un ancêtre commun)

Une séquence ACGTACGT a pu évoluer pour donner :

– – ACG–T– A– – –CG– T– – – –
ACACGGTCCTAATAATGGCC

– – – AC–GTA –C —G –T —
CAG –GAAGATCTTAGTTC

probabilités de
délétion: 0.0001
insertion: 0.001
substitution A/G, T/C: 0.00008
substitution A/C, T/G: 0.00002

Ces deux séquences ont le même ancêtre et on veut connaître leur alignement qui reflète leur origine commune (c'est-à-dire ACGTACGT)

Par exemple, l'alignement pourrait être

–AC AC– GGTCCTAAT – –AATGGCC
CAG –GAA –G–AT– – CTTAGTTC – –

Définition: un alignement des séquences S et T est une paire de séquences S' et T' (avec espaces) t.q.

(1) |S'|=|T'|
(2) en éliminant tous les espaces on obtient S et T

Matches, mismatches et indels

Deux caractères identiques et alignés dans un alignement forment un **match**

Deux caractères alignés différents forment un **mismatch**

Un caractère aligné avec un espace, représente un **indel** (insertion/délétion)

–AC AC– GGTCCTAAT – –AATGGCC
CAG –GAA –G–AT– – CTTAGTTC – –

Une barre montre un mismatch

4 matches, 9 mismatches, 11 indels

Problème algorithmique de base

Trouver un alignement de deux chaînes de caractères S et T que maximise
#matches - #mismatches - #indels

La valeur maximale définit la similarité entre deux séquences

On parle d' «alignement globale optimale»

Biologiquement, sous l'hypothèse d'un modèle de mutation ponctuelle, un mismatch représente une mutation, et un indel représente une insertion ou délétion d'un seul caractère.

Alignement optimale en O(nm) en utilisant la programmation dynamique

- Entrées: S, T |S|=n, |T|=m
- Sortie: la valeur V(n,m) d'un alignement optimale

On va calculer toutes les valeurs
V(i,j)= valeur de l'alignement optimale de S[1],…,S[i] avec T[1]…T[j]
pour tous $0 \le i \le n$, $0 \le j \le m$

Solution par Programmation Dynamique

On pose
M(I, J) = -1 si S(I) "mismatches" S'(J)
M(I, J) = 1 si S(I) "matches" S'(J)

Les formules de recurrence sont:

$$V(I,0) = -I,$$
$$V(0,J) = -J,$$

Cas de base

$$V(I,J) = Max \begin{bmatrix} V(I-1,J-1) + M(I,J) \\ V(I-1,J) - 1 \\ V(I,J-1) - 1 \end{bmatrix}$$

Cas généraux:
I et J alignés
I en face à un gap
J en face à un gap

DP Solution

The DP recurrences are:

$$V(I,0) = -I,$$
$$V(0,J) = -J,$$

Base Cases

$$V(I,J) = Max \begin{bmatrix} V(I-1,J-1) + M(I,J) \\ V(I-1,J) - 1 \\ V(I,J-1) - 1 \end{bmatrix}$$

General Cases
I and J aligned
I opposite a space
J opposite a space

They can be evaluated in O(NM) operations.

Variante : solution PD pour l'alignement locale

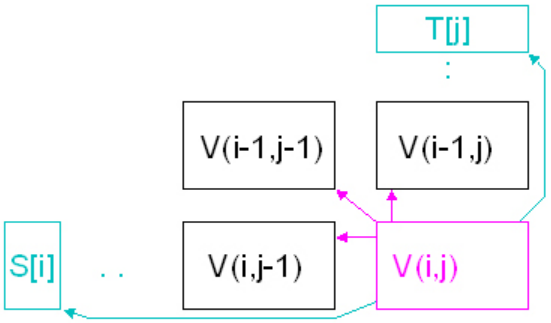
$$V(I,0) = V(0,J) = 0$$
$$V(I,J) = MAX \begin{bmatrix} V(I-1,J-1) + M(I,J) \\ V(I-1,J) - 1 \\ V(I,J-1) - 1 \\ 0 \end{bmatrix}$$

La valeur optimale n'est plus V(N, M) comme dans l'alignement globale, mais plutôt la plus grande valeur V sur toutes les nm valeurs calculées. Le temps est O(nm).
Cette variante est connue comme l'algorithme de Smith-Waterman.

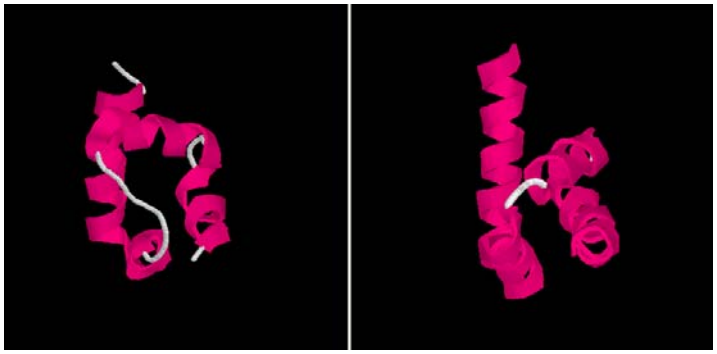
Variante: prise en compte des probabilités de substitution entre resides σ(x,y)

- Un score d'alignement de (charactères ou espace) x et y est σ(x,y)
- La valeur de l'alignement = $\Sigma i=1^{|S'|}\sigma(S'[i],T'[i])$
- Un alignement optimale est un alignement de valeur maximale.

$$V(i,j) = \max \begin{cases} V(i\text{-}1,j\text{-}1) + \sigma(S[i],T[j]) \\ V(i\text{-}1,j) + \sigma(S[i], -) \\ V(i,j\text{-}1) + \sigma(- , T[j]) \end{cases}$$



Recherche de protéines ayant structure similaire



protéine ribosomale L20
Aquifex aeolicus

protéine
polyA binding Homo sapiens

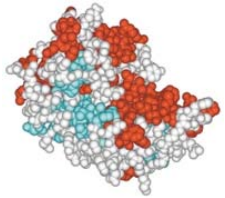
Echec des programmes d'alignement

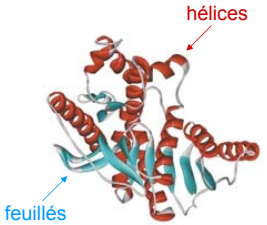
Programme d'alignement de séquences CLUSTALW

```
bacterie: |-WIARINAAVRAYGLNYSTFINGLKKAGIELDRKILADMAVRDPQAFEQVVNKVKEALQVQ
   homme: |HRQALGERLYPRVQAMQPAFASKITGMLLELSPAQLLLLLASEDSLRARVDEAMELIIAHG
Identite: |   #               #         ##    #             #
```

Nombre d'identité = 6

Protéines: structures et clusters hydrophobes

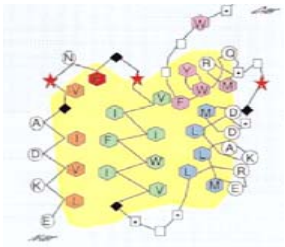




Acides aminés

HRQALGERLYPRVQAMQPAFASKITGMLLELSPAQLLLLLASEDSLRARVD

Clusters hydrophobes dans la détection de protéines à faible homologie



Impliqués dans la **stabilité** du repliement

Conservation

Les clusters ont evolués

insertion

délétion

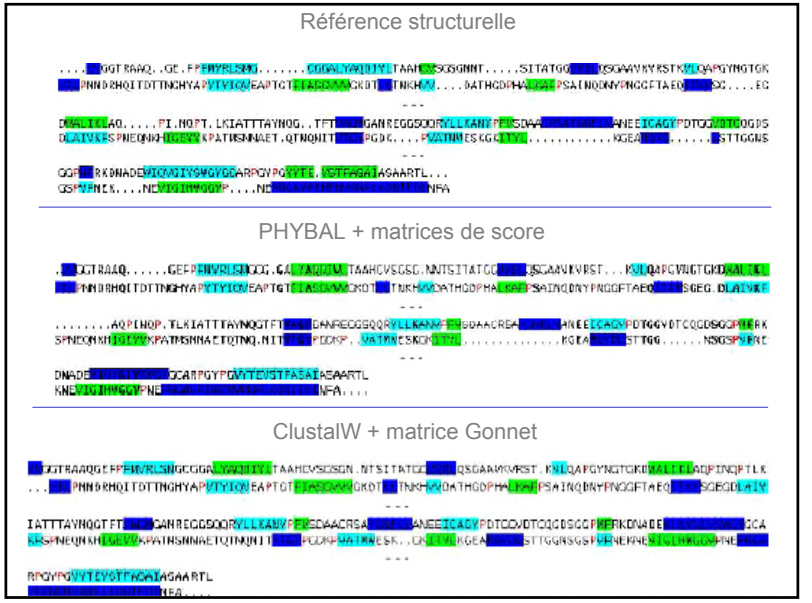
Les clusters sont plus conservés que les séquences et ils deviennent utiles pour l'alignement de protéines à faible homologie

Problème

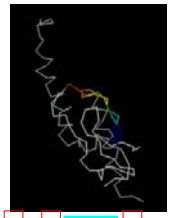
Recherche d'un alignement optimale de clusters hydrophobes pour deux séquences données.

Entrées: soit S, T deux séquences avec description des clusters hydrophobes (position et longueur), |S|=n, |T|=m

Sortie: la valeur d'un alignement optimale



Homologie faible : quelques cas difficiles



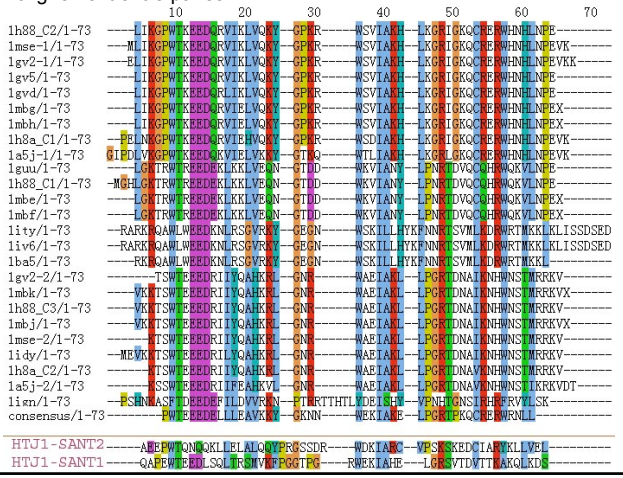
```
1fleI   .AQEPVKGPVSTKPGSCPLILIRCAMLNPPNRCLKD.TDCPGIKKCCEGSCGMACFVPQ....
1a0aA   MKRESHKHAEQARRNRLAVILHELASLIPAEWKQQNVSAAPSKATTVEAACRYIRHLQQNGST
```

Petits bloques d'identités

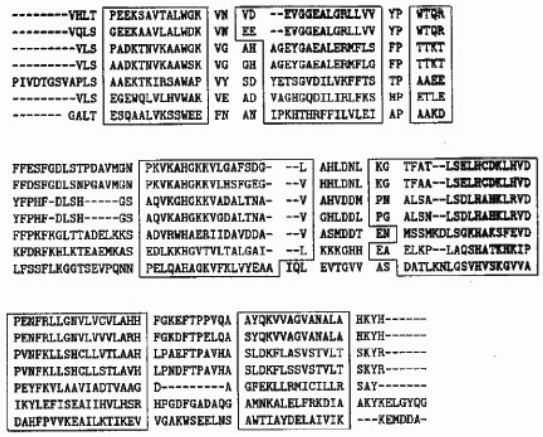
```
1fleI   AQEPVKGPVSTKPGSCPLILIRCAMLNPPN....RCLKDTDCPGIKKCCEGSCG.MACFVP.Q
1udkA   .........NEKSGSCPDMSM...PIPPLGICKTLCNSDSGCPNVQKCCKNGCGFMTCTTPVP
```

Alignement multiple

Dans l'alignement multiple on retrouve plus d'information que dans l'alignement entre paires :



Les 7 hélices α des globines sont encadrées :



Probleme: Étant donné un ensemble de séquences

$S_1, S_2, \ldots S_n$

un alignement multiple est une tuple

$S_1', S_2', \ldots S_n'$

telle que

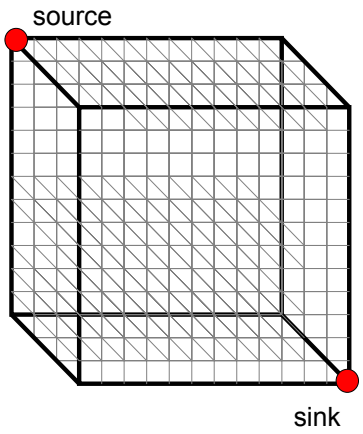
1. $|S_1'| = |S_2'| = \ldots |S_{n'}|$
2. S_i' est obtenu à partir de S_i par l'insertion d'occurrences d'espaces.

On veut déterminer un alignement optimal entre toutes les séquences, parce que la similarité multiple suggère une structure, une fonction et une origine (dans le sens de l'évolution) commune entre les différentes protéines.

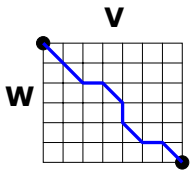
Le problème est NP-difficile (kn, où k est la longueur maximale des séquences).

Alignement de trois séquences

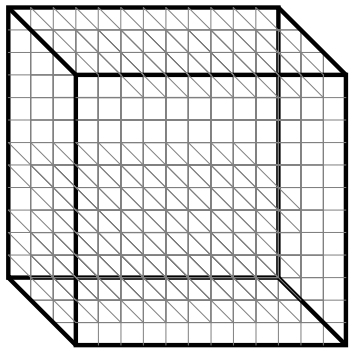
- Même stratégie que pour aligner 2 séquences
- Utiliser un cube en 3-D, “Manhattan Cube”, avec chaque axe qui représente une séquence a aligner
- Pour les alignements globales partir de la “source” jusqu’au “sink”



Alignement 2-D vs 3-D

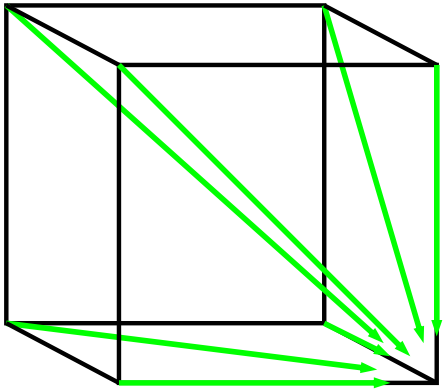


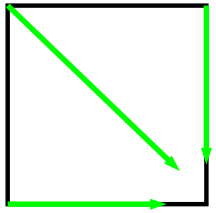
2-D edit graph



3-D edit graph

Cellule 3-D vs cellules 2-D

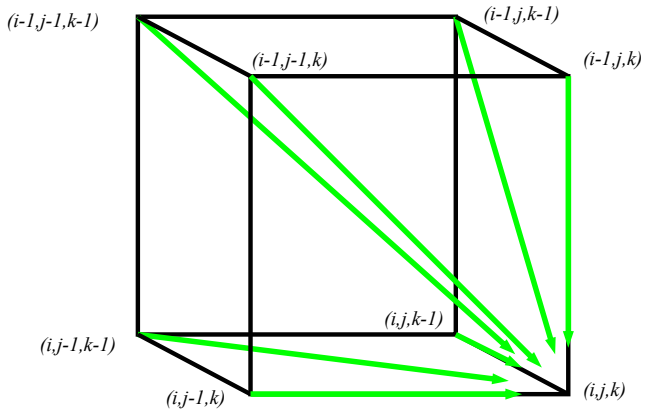


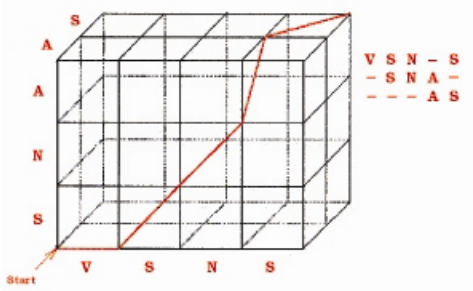


En **2-D**, 3 arcs dans chaque carre

En **3-D**, 7 arcs dans chaque cube

Architecture d’une cellule d’alignement en 3-D





Chemin d'alignement pour 3 séquences

Construction de l'alignement multiple a partir de l'alignement par paires

Etant donne l'alignement de 3 séquences arbitraires :

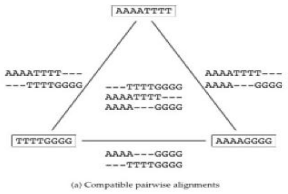
```
x: ACGCTGG-C;   x: AC-GCTGG-C;   y: AC-GC-GAG
y: ACGC--GAC;   z: GCCGCA-GAG;   z: GCCGCAGAG
```

Peut-on reconstruire un alignement multiple que les induits ?

PAS TOUJOURS

Combinaisons d'alignements par paires optimaux dans un alignement multiple

Combinaison des alignements par paires possible

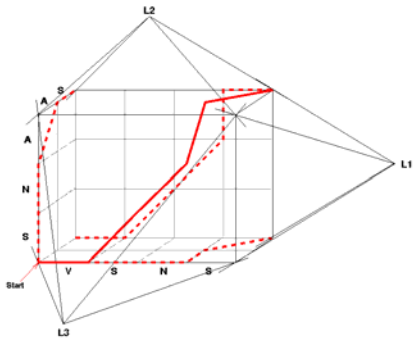


(a) Compatible pairwise alignments

Combinaison des alignements par paires impossible

(b) Incompatible pairwise alignments

Projection de l'alignement multiple



Toutes les 3 projections par paires de l'alignement multiple

Un alignement 3-D peut être projeté en 2-D pour représenter un alignement entre paires de séquences.

Méthode gourmande : les profils

- Considérer ces 4 séquences

s1 GATTCA
s2 GTCTGA
s3 GATATT
s4 GTCAGC

Approache gourmand :profils

- Il y a $\binom{4}{2}$ = 6 possibles alignements

```
s2  GTCTGA                  s1  GATTCA--
s4  GTCAGC (score = 2)      s4  G-T-CAGC(score = 0)

s1  GAT-TCA                 s2  G-TCTGA
s2  G-TCTGA (score = 1)     s3  GATAT-T (score = -1)

s1  GAT-TCA                 s3  GAT-ATT
s3  GATAT-T (score  = 1)    s4  G-TCAGC (score = -1)
```

s_2 *et* s_4 sont proches; combiner:

s2 GTCTGA
s4 GTCAGC
} $s_{2,4}$ (profile) GTCt/aGa/cA

Nouveau ensemble de 3 sequences:

s_1 GATTCA
s_3 GATATT
$s_{2,4}$ GTCt/aGa/c

Alignement basé sur la consistance

La prévention est le meilleur médicament

Pour chaque alignement multiple, les alignements par paires induits sont nécessairement consistant, cad que étant donné un alignement multiple de x,y,z, si la position x_i aligne avec la position z_k et z_k aligne avec y_j dans les projections x-z et z-y des alignements en multiple, alors x_i doit aligner avec y_j dans l'alignement x-y projeté.

Les alignements basés sur la consistance appliquent ce principe mais renversé, en utilisant des alignements a des séquences intermédiaires comme évidence pour guider l'alignement des paires x et y.

Problème ouvert:
concevoir un algorithme d'alignement multiple qui préserve la consistance et qui tiens en compte les bloques hydrophobes

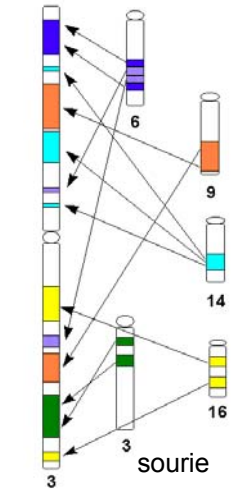
Topic 3 :
le réarrangement des génomes

Un exemple : l'homme et la sourie

- La sourie a 2.1 x10^9 bp vs 2.9 x 10^9bp chez l'homme.
- A peu près 95% du matériel génétique est partagé.
- 99% des gènes partagés sur un totale de 30,000.
- Les 300 gènes sans homologues dans les deux espèces concernent surtout l'immunité, la détoxification, l'odeur et la sexualité.

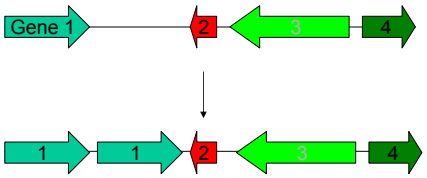
Homme et sourie

- Il existe une quantité significative de réarrangement des génomes entre homme et sourie.
- Ici on voit la carte du chromosome 3 chez l'homme.
- Il contient des séquences homologues a au moins 5 chromosomes de la sourie.



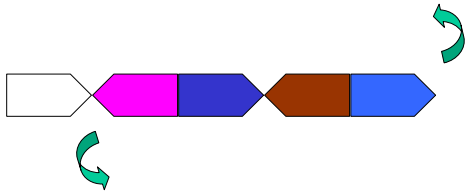
Mécanismes:

Duplication de gènes (un ou plusieurs à la fois), insertions



et inversion de gènes…

Comparaison de l'ordre des gènes :



Comparaison de l'ordre des gènes :

Comparaison de l'ordre des gènes :

Comparaison de l'ordre des gènes :

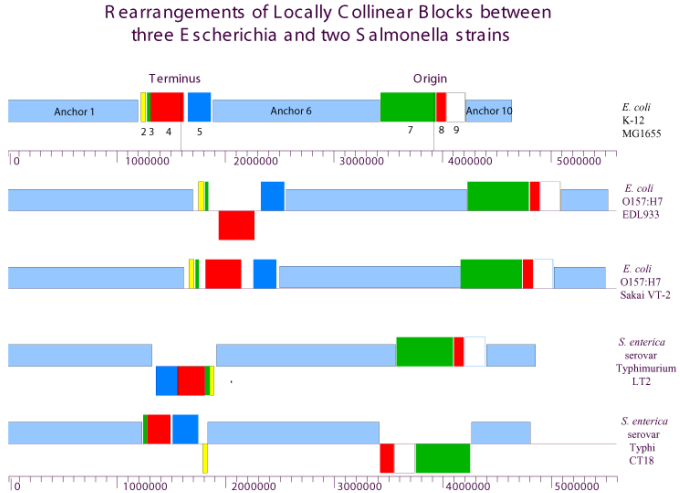
Comparaison de l'ordre des gènes :

Avant?

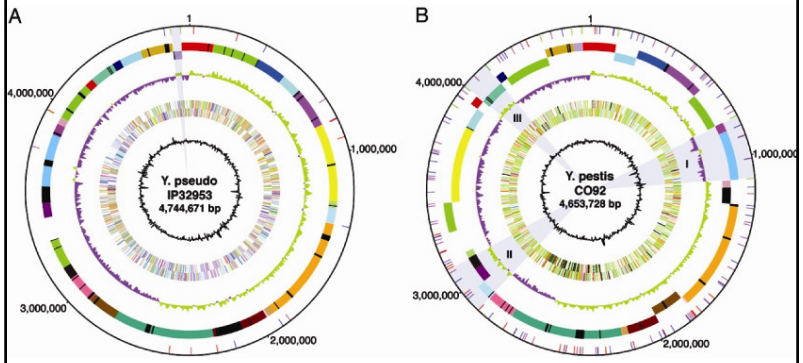
Apres?

L'évolution est présente dans la divergence de l'ordre des gènes

Problème : étant données deux permutations d'un ensemble de segments génomiques, trouver l'ensemble minimale d'opérations pour transformer une permutation dans l'autre.



Réarrangement chez les bactéries *:*
Yersinia pseudotuberculosis versus *Yersinia pestis*



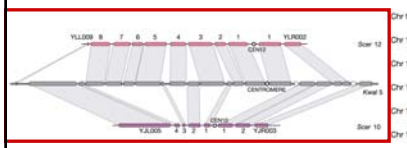
Grande virulence

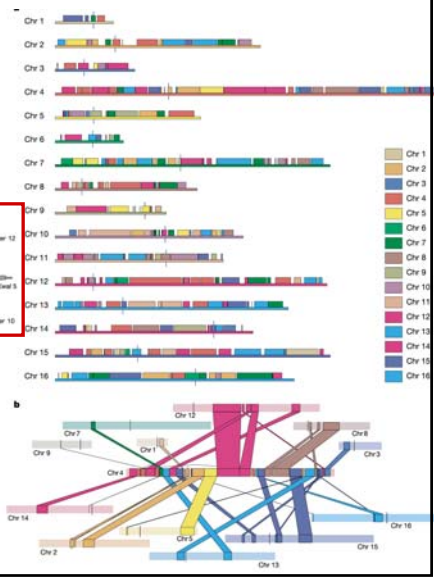
Les **réarrangements des génomes** sont rares par rapport aux **mutations ponctuelles**:

10 substitutions par génération d'un organisme
1 réarrangement non fatale chaque 5-10 millions d'années

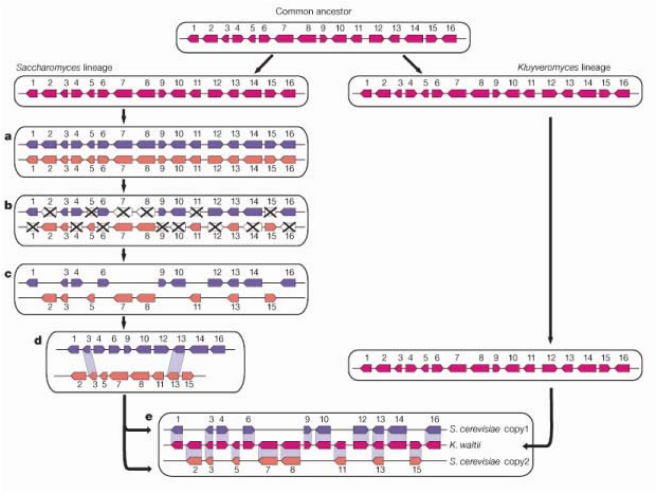
La rareté des événements de réarrangement nous permet d'établir l'existence de processus évolutionnaires parce que la chance d'un renversement est minuscule. En conséquence, par la découverte de tous réarrangements, nous pouvons reconstruire des hypothèses évolutionnaires.

Duplication du génome de la levure





… et réarrangement



Whole Genome Duplications in Evolution

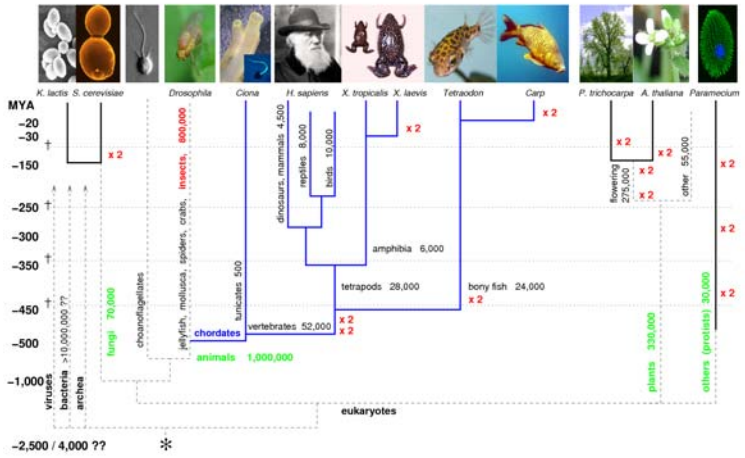


Image de Herve Isambert

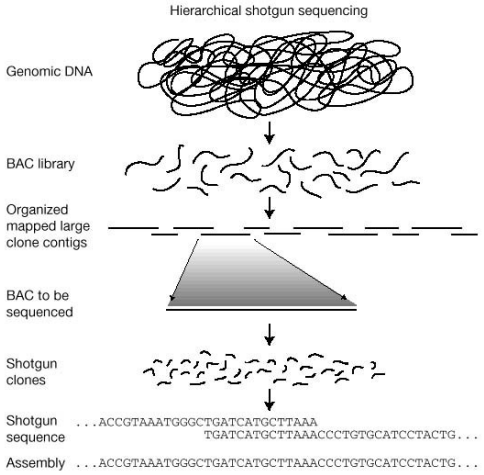
Topic 4: l'assemblage des séquences

Pour séquencer des larges portions d'ADN, il est possible de les diviser dans des petits fragments et effectuer le séquençage sur ces morceaux. Le problème que se présent est la reconstruction de la chaîne a partir des sous-séquences.

Etant donné un ensemble de séquences, trouver la chaîne de longueur minimale qui contient tous les membres de l'ensemble de sous-chaînes.

Ce problème est NP-complet. Il y a des algorithmes gourmands qui réalisent le réassemblage de façon satisfaisante en temps raisonnable. Le problème est difficile a cause des nombreuses séquences répétées.

Séquençage des génomes



Jusqu'a la fin des années 1990 l'"assemblage de fragments lus" (shotgun fragment assembly) du génome humain était vu comme un problème intraitable.

Shortest Superstring Problem

Problème: étant donne un ensemble de chaînes, trouver la chaîne la plus courte qui les contient toutes

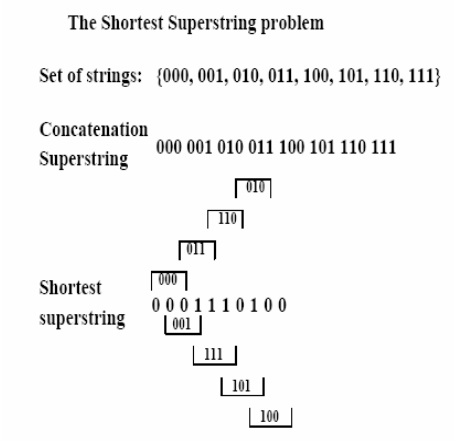
Entrée: Chaînes $s_1, s_2, \ldots, s_n$

Sortie: Une chaîne s qui contient toutes les chaînes $s_1, s_2, \ldots, s_n$ comme sous-chaînes, tel que la longueur de s est minimisée

Complexité: NP – complet

Note: cette formulation ne considère pas les erreurs de séquençage

Shortest Superstring Problem: Example



Réduction de SSP à TSP

- Définir *overlap* (s_i, s_j) comme la longueur du préfix le plus long de s_j qui s'apparie avec un suffixe de s_i (sans erreurs).

aaaggcatcaaatctaaaggcatcaaa
aaaggcatcaaatctaaaggcatcaaa

Qu'est-ce que *overlap* (s_i, s_j) ***pour ces chaînes ?***

Réduction de SSP à TSP

- Définir *overlap* (s_i, s_j) comme la longueur du préfix le plus long de s_j qui s'apparie avec un suffixe de s_i.

aaaggcatcaaatctaaaggcatcaaa
aaaggcatcaaatctaaaggcatcaaa
aaaggcatcaaatctaaaggcatcaaa

overlap=12

Réduction de SSP à TSP

- Définir *overlap (s_i, s_j)* comme la longueur du préfix le plus long de s_j qui s'apparie avec un suffixe de s_i.

  aaaggcatcaaatctaaaggcatcaaa
  aaaggcatcaaatctaaaggcatcaaa
  aaaggcatcaaatctaaaggcatcaaa

- Construire un graphe avec *n* noeuds qui représente les *n* chaînes $s_1, s_2, \ldots, s_n$.
- Insérer des arcs pondérés de longueur (poids) *overlap (s_i, s_j)* entre les noeuds s_i et s_j.
- Chercher le plus court chemin qui visite chaque noeud exactement une fois. Il s'agit du **Problème du Voyageur de Commerce** (**Traveling Salesman Problem** - TSP), qui on sait être NP – complet.

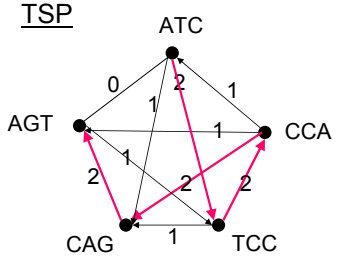
Réduction de SSP à TSP: un exemple

$S = \{$ ATC, CCA, CAG, TCC, AGT $\}$

SSP

AGT
CCA
ATC
ATCCAGT
TCC
CAG

TSP



ATCCAGT

La technologie permet de déterminer tous les l-mers, pour un l fixé, qui sont contenus dans une chaîne S d'ADN

Le but est de reconstruire la chaîne S a partir de ses l-mers.

On peut formuler le problème comme un problème du chemin Hamiltonien sur un graphe, mais c'est plus productif de le voir comme un problème de chemin Eulérien.

Graphe de reconstruction: composition des *l-mer*

- Une chaîne *s* de longueur *n.*
- ***Spectre(s,l)*** – multi-ensemble non ordonné de tous les possibles *(n – l + 1) l*-mers
- L'ordre des éléments individuels dans le *Spectre(s,l)* n'a pas d'importance. Pour *s* = TATGGTGC tous les *Spectre(s,3)* suivants sont équivalents

  {TAT, ATG, TGG, GGT, GTG, TGC}
  {ATG, GGT, GTG, TAT, TGC, TGG}
  {TGG, TGC, TAT, GTG, GGT, ATG}

- Des séquences différentes peuvent avoir le même spectre:

  Spectre(GTATCT,2)=

  Spectre(GTCTAT,2)=

  {AT, CT, GT, TA, TC}

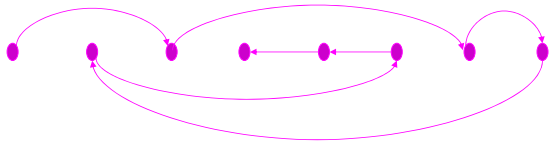
- Problème: Reconstruire une chaîne a partir de ses *l*-mers
- Entrée: Un ensemble *S*, représentant tous les *l*-mers d'une chaîne *s* inconnue
- Sortie: la chaîne *s* tel que *Spectre(s,l)* = *S*

Approche du chemin hamiltonien

Chaque nœud représente un k-mer et un arc est dirigé d'un nœud a l'autre si le deuxième k-mer peut être obtenu du premier en supprimant le dernier caractère et en ajoutant un caractère a la droite

S = { ATG AGG TGC TCC GTC GGT GCA CAG }

H ATG AGG TGC TCC GTC GGT GCA CAG



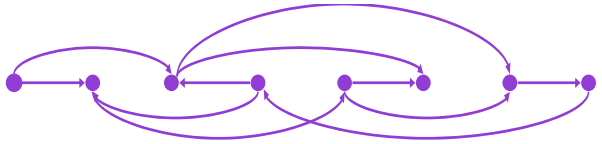
ATGCAGGTCC

Le chemin a visité chaque NOEUD une fois

Un graphe plus compliqué:

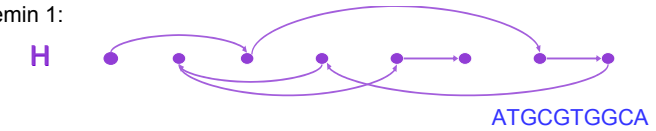
S = { ATG TGG TGC GTG GGC GCA GCG CGT }

H

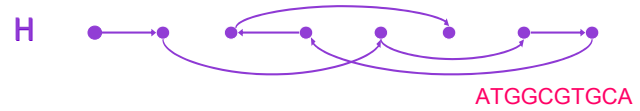


S = { ATG TGG TGC GTG GGC GCA GCG CGT }

Chemin 1:



Chemin 2:

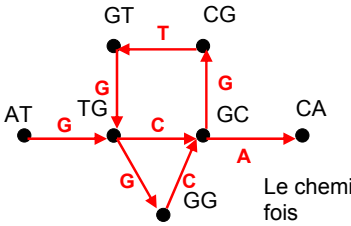


Approche du chemin Eulérien

S = { ATG, TGC, GTG, GGC, GCA, GCG, CGT }

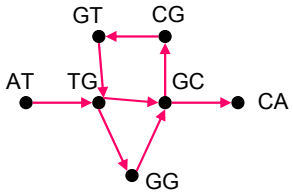
Noeuds correspondent a (k – 1) – mers : { AT, TG, GC, GG, GT, CA, CG }

Arcs correspondent a k – mers observés

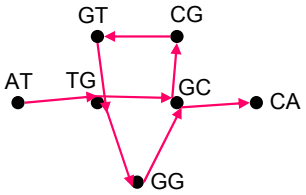


Le chemin visite chaque ARC une fois

S = { AT, TG, GC, GG, GT, CA, CG } corresponde a deux différents chemins :



ATGGCGTGCA



ATGCGTGGCA

D'habitude on a plusieurs chemins eulériens.

Comment choisir un chemin eulérien qui est le plus prometteur, cad celui qui correspond le mieux a S ?

D'habitude nous avons de l'information additionnelle sur les k-mers, par exemple sur la distance approximative entre k-mers dans S. Cette information est représentée en donnant un poids a chaque paire d'arcs successifs dans G qui reflète grosso modo la probabilité que deux arcs apparaissent l'un après l'autre dans un chemin eulérien pour S.

Revenons a l'approche Hamiltonien…

Chaque nœud est une k-mer de S, et chaque arc a un poids qui reflet la génération appropriée d'une k+1-mer dans un chemin Hamiltonien. On veut trouver le chemin Hamiltonien de distance maximale.

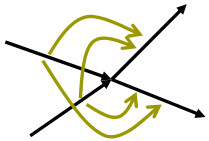
Il semble un problème difficile mais la structure particulière des graphes rende le problème résoluble.

La structure des graphes

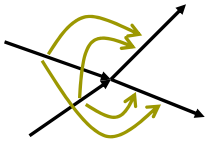
Le graphe Hamiltonien associé au données, où
- noeuds = k-mers
- arcs = k+1-mers dans une solution S'

est le di-graphe du graphe eulérien où
- nœuds = k-1-mers
- arcs = k-mers

Dualement:
étant donné un graphe eulérien G, le graphe Hamiltonien L(G) est le di-graphe:



noeuds de L(G) = arc de G
arc de L(G) = paire d'arcs consecutifs de G



noeuds de L(G) = arc de G
arc de L(G) = paire d'arcs consecutifs de G

Chaque arc vert a la valeur d'une paire d'arcs consecutifs de G.

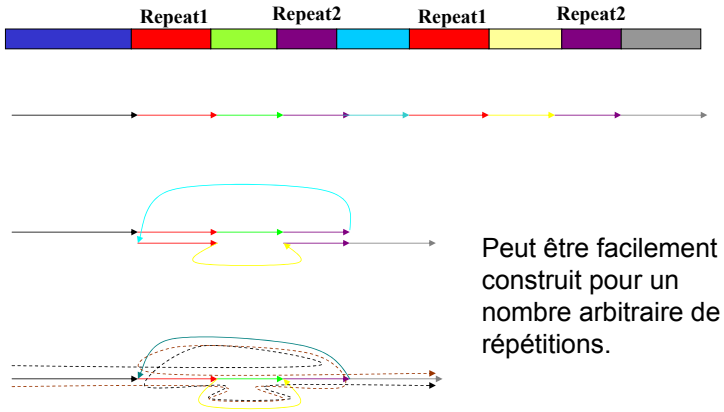
Le problème est de trouver un TSPath de valeur maximale.

Si le graphe a seulement 2 arcs en entrée et 2 arcs en sortie de chaque nœud, le TSP a une solution polynomiale. En particulier, même pour une distance très large entre nœuds, TSP a une solution en temps O(nlogn).

Si chaque valeur des arcs est distincte, alors la solution est unique.

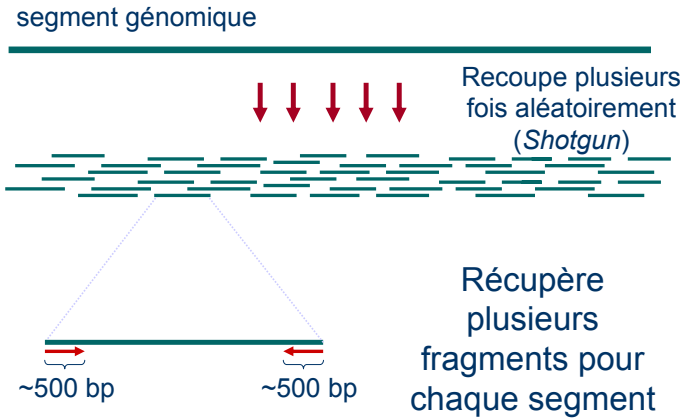
Quand les degrés sont plus que 2, il y a une procédure branch-and-bound qui réduit le problème au cas du degré 2 que peut être résolu de façon optimale.

Répétitions multiples



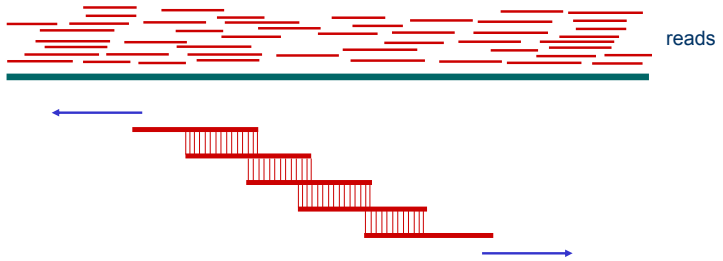
Peut être facilement construit pour un nombre arbitraire de répétitions.

On revient sur le « Shotgun Sequencing »



Récupère plusieurs fragments pour chaque segment

Assemblage des fragments



Recouvrir la région avec une redondance de ~7-fold

Chevaucher les reads et les étendre pour reconstruire la région génomique originale

Recouvrement des Reads

Longueur du segment génomique: ***L***
Nombre de reads: ***n***
Longueur de chaque read: ***l***

Recouvrement ***C = n l / L***

Combien de recouvrements sont nécessaires ?

Modèle de **Lander-Waterman**

Scénario théorétique, modèle de Lander-Waterman

Taille du génome : 1million de paires de bases (~25 longueurs de clone)

Longueur moyenne d'un clone: 40kb
Couverture des clones = 10x

Sondes : 500
Longueur des sondes : 10-40bp.

La première couverture du maïs FPC, 2001

Taille du génome: 2500 Mb

Longueur moyenne d'un clone : 150kb
Couvertures de clones : 17x

414 contigs ont été assignés aux chromosomes
Les contigs couvrent approximativement 2148 MB,
Les contigs assignées aux chromosomes couvrent approximativement 1839 MB.

Metagénomique: reconstruction des génomes et défis algorithmiques

Métagénomique = étude des métagénomes
ou un métagénome est l'ensemble des séquences d'ADN extraites de communautés multi-espèces prélevées dans l'environnement.

Ces communautés sont constituées d'organismes non cultivables. Il a été estimé que seul 1% des procaryotes de la plupart des environnements peuvent être cultivés nécessitant pour la plupart des organismes que le clonage soit effectué directement sur les échantillons prélevés dans l'environnement.

Le problème d'assemblage des fragments d'ADN en métagenomique

En métagénomique le problème est difficile puisqu'il est nécessaire de pouvoir associer le fragment de séquence à l'espèce duquel il provient.

Les programmes d'assemblage comme Phred/Phrap et Sequencher ont été mis au point pour combiner des séquences issues d'un même génome et sont fondés sur des hypothèses fausses lorsqu'elles sont appliquées à un métagénome.

Exemple: une différence de base entre 2 séquences est considérée comme une erreur de séquençage par ces programmes, alors que cela pourrait être le signe d'une différence d'origine en métagénomique.

Le grand nombre de génomes viraux (jusqu'à plusieurs millions) prélevés dans l'environnement ainsi que les répétitions de séquences de type transposons rendent la discrimination entre les organismes et l'assemblage des séquences d'autant plus difficile.

Ces problèmes nécessitent des approches algorithmiques fines, des fragments d'ADN plus longs, ainsi que des taux de couverture plus importants.

Récemment, la différence de nombre de répétitions de tétranucléotides entre génomes a pu être utilisée pour discriminer ces génomes, mais cette approche nécessite un échantillon de faible complexité et une répartition équitable des génomes.

Le polymorphisme des séquences nucléotidiques, la duplication des gènes et le transfert horizontal de gène sont tous des obstacles à la fiabilité de l'assemblage des génomes.

www.ihes.fr/~carbone/teaching a la fin des transparents de cours sur les différents sujets voir références bibliographiques

References : biais des codons et espace d'organismes

Algorithm and microbial SCCI codon space :

- **F.Képès,** CNRS & génopole Evry
- **A.Zinovyev**, IHÉS & Institut Curie (Paris)

A. Carbone, A. Zinovyev, F. Képès, Codon adaptation index as a measure of dominating codon bias, *Bioinformatics*, **19**, 2005–2015, 2003.

A. Carbone, F. Képès, A. Zinovyev , Codon Bias Signatures, Organization of Microorganisms in Codon Space, and Lifestyle, *Molecular Biology and Evolution*, **22**, 547–561, 2004.

Metabolic networks comparison :

- **D.Madden**, IHÉS & IGI (USA)

A. Carbone, R. Madden, Insights on the Evolution of Metabolic Networks of Unicellular Translationally Biased Organisms from Transcriptomic Data and Sequence Analysis, *Journal of Molecular Evolution*, **59**, 1–25, 2005.

Minimal gene sets :

A.Carbone, Computational prediction of genomic functional cores specific to different microbes, *Journal of Molecular Evolution*, **63,** 733-746, 2006.

Host-phage co-evolution:

A.Carbone, Codon bias is a major factor explaining phage evolution in translationally biased hosts, *Journal of Molecular Evolution*, 2007. In press.