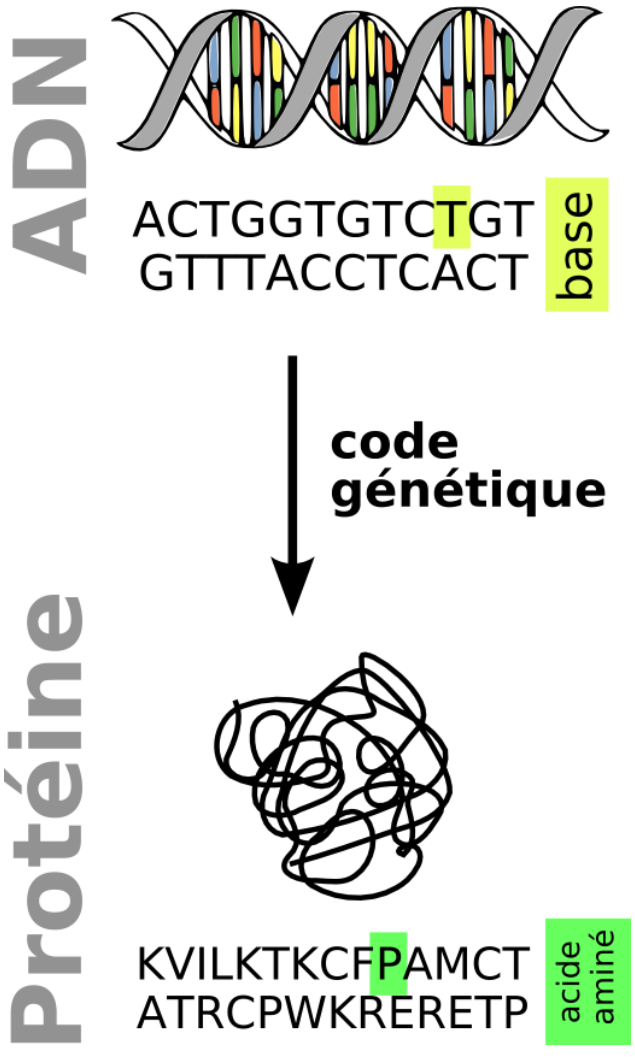
Bioinformatique et calcul haute-performance

Mathieu Giraud
mathieu.giraud@lifl.fr

CNRS, LIFL, Université Lille 1
INRIA Lille Nord-Europe

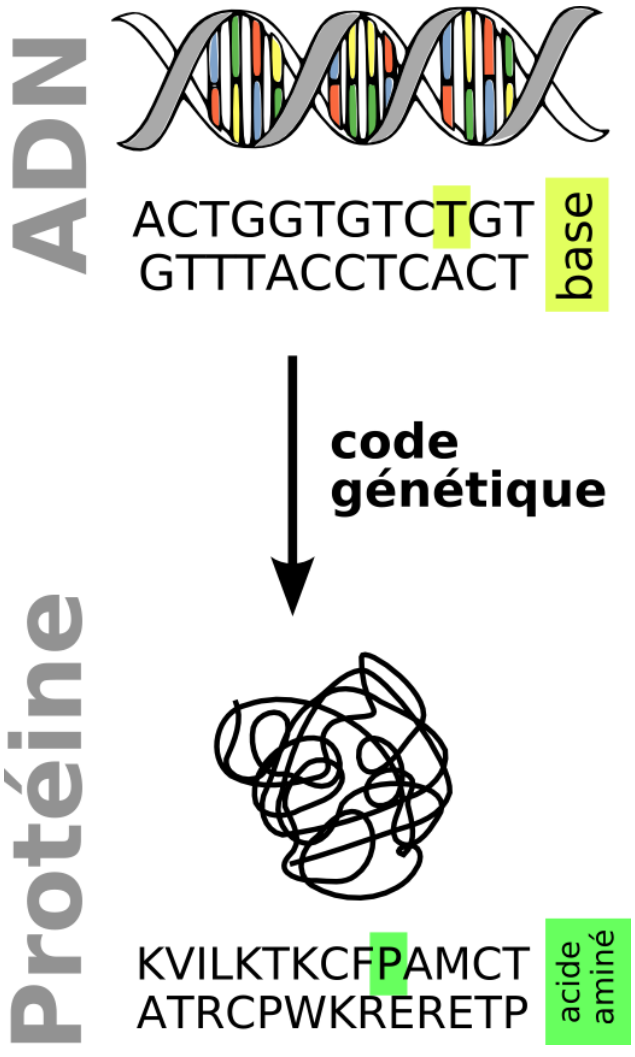
Séminaire Aristote, École Polytechnique, 16 octobre 2008

Séquences génomiques : ADN et protéines

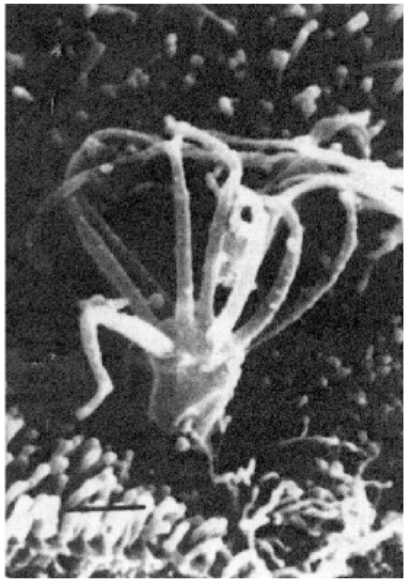


Séquences génomiques : ADN et protéines

- ADN : $\Sigma_4 = \{A, C, G, T\}$
- Protéines :
  $\Sigma_{20} = \{A, C, D, E, F, G, H, I, K,$
  $L, M, N, P, Q, R, S, T, V, W, Y\}$
- Code génétique :
  $\Sigma_4 \times \Sigma_4 \times \Sigma_4 \longrightarrow \Sigma_{20}$
- Bases de données : EMBL (octobre 2008)
  - $233 \cdot 10^9$ bases
  - $155 \cdot 10^6$ séquences



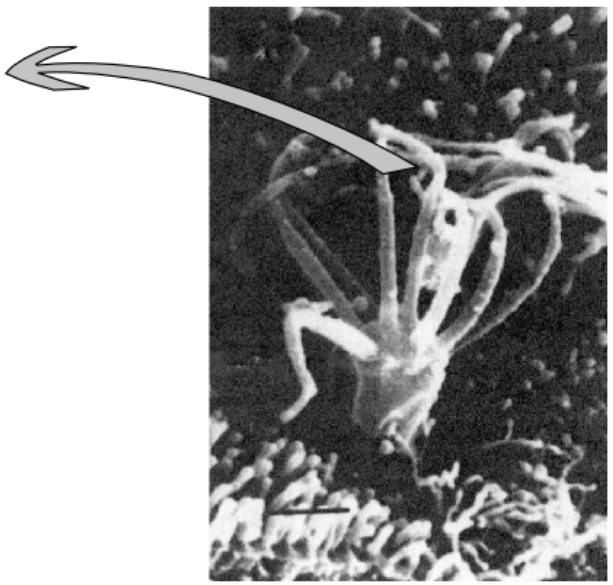
Recherches dans les séquences génomiques



- recherches par annotations

Recherches dans les séquences génomiques

>CfOR194 | Canis Olfactory Receptor
PMYKVILTPIMAYDRYL
KLMNIPLMNPLMSATT
TLMWNIPLMN...

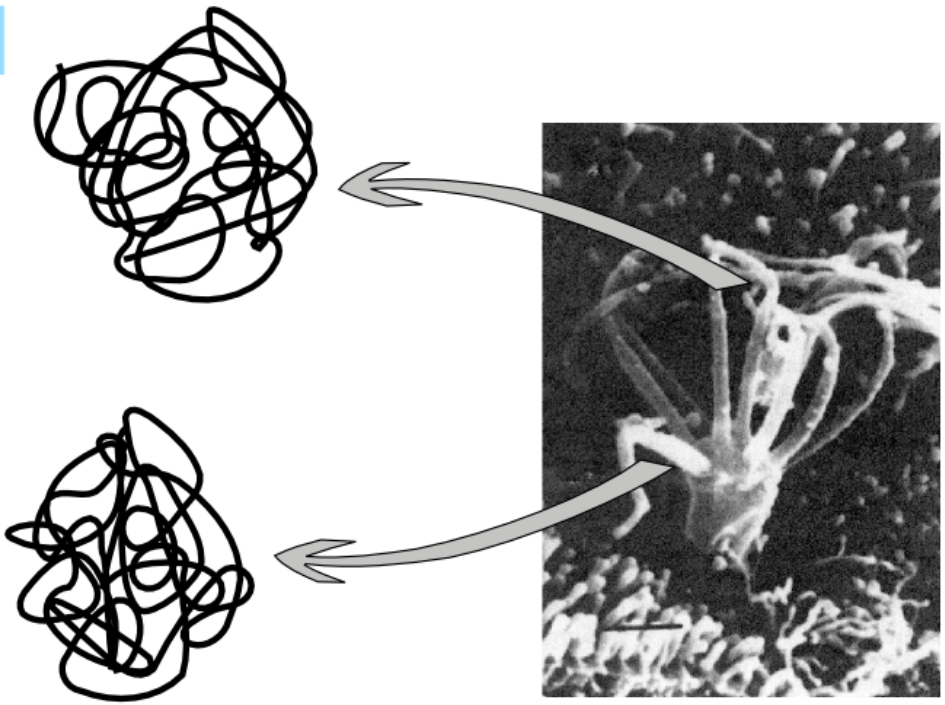


- recherches par annotations

Recherches dans les séquences génomiques

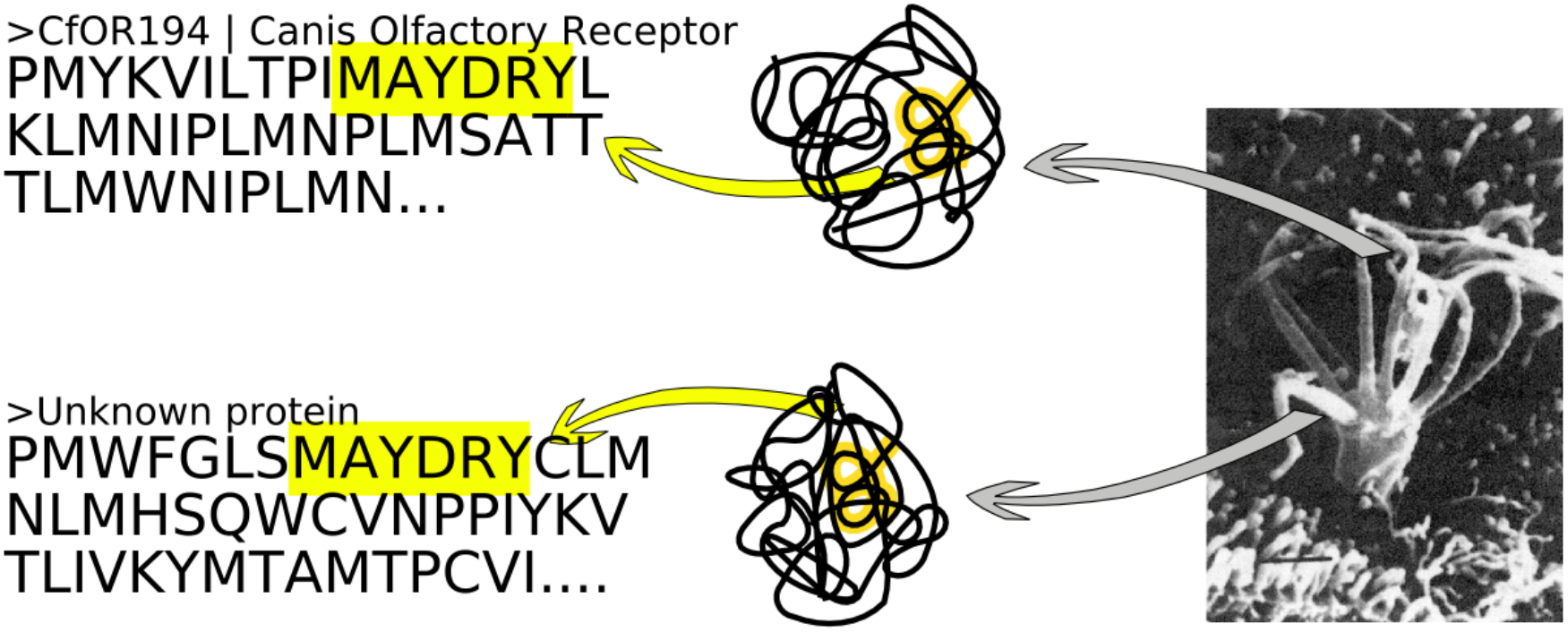
>CfOR194 | Canis Olfactory Receptor
PMYKVILTPIMAYDRYL
KLMNIPLMNPLMSATT
TLMWNIPLMN...

>Unknown protein
PMWFGLSMAYDRYCLM
NLMHSQWCVNPPIYKV
TLIVKYMTAMTPCVI....



- recherches par annotations

Recherches dans les séquences génomiques

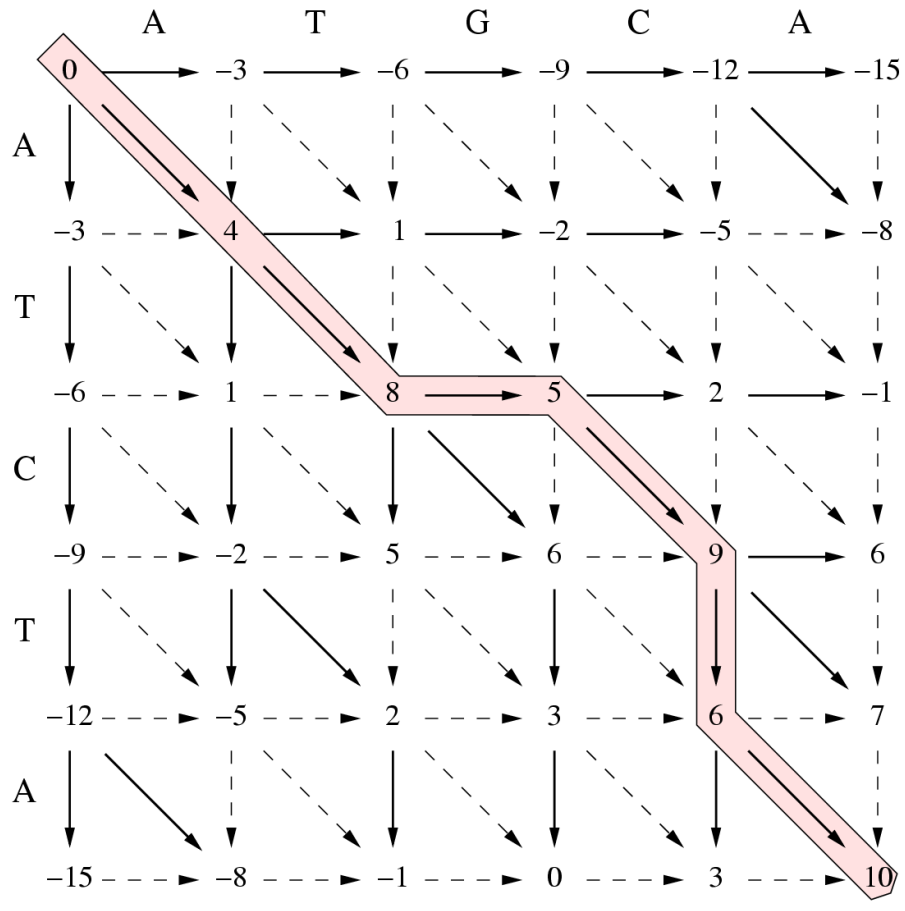


- recherches par annotations / recherches par le contenu
- mutation des séquences : recherche par une partie du contenu
  $\longrightarrow$ recherches de motif, exacts ou approchés

Alignement global : Needleman-Wunsch

Aligner ATGCA et ATCTA

$$\text{match} \longrightarrow +4$$
$$\text{substitution} \longrightarrow -2$$
$$\text{gap} \longrightarrow -3$$


A T G C A
A
T
C
T
A
0 −3 −6 −9 −12 −15
−3 4 1 −2 −5 −8
−6 1 8 5 2 −1
−9 −2 5 6 9 6
−12 −5 2 3 6 7
−15 −8 −1 0 3 10

```
A T G C - A
| |   | |
A T - C T A
```

Alignement global : Needleman-Wunsch

- $X = (x_1, x_2 \dots x_m)$ et $Y = (y_1, y_2 \dots y_n)$
- $H(i,j)$ similarité entre $x_1 \dots x_i$ et $y_1 \dots y_j$

$\forall i : \quad H(i,0) = g_{\text{penalty}} \times i \quad \forall j : \quad H(0,j) = g_{\text{penalty}} \times j$
$\forall i,j,ij \neq 0 :$

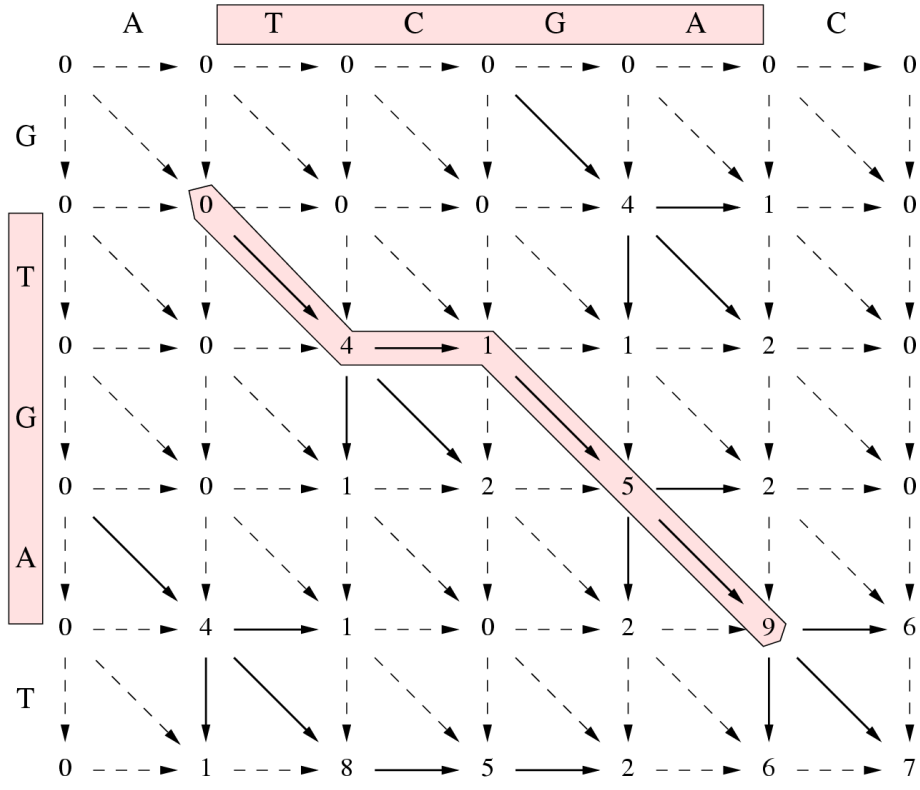
$$H(i,j) = \max \begin{cases} H(i-1,j-1) + d(x_i, y_j) & \text{(match ou substitution)} \\ H(i-1,j) - g_{\text{penalty}} & \text{(insertion)} \\ H(i,j-1) - g_{\text{penalty}} & \text{(délétion)} \end{cases}$$

$\longrightarrow$ $H(n,m)$ similarité globale entre X et Y

Alignement local : Smith-Waterman

Aligner localement ATGCAC et GTCTAT

$$\text{match} \longrightarrow +4$$
$$\text{substitution} \longrightarrow -2$$
$$\text{gap} \longrightarrow -3$$


A
T C G A
C
G
T G A
T
0 0 0 0 0 0 0
0 0 0 0 4 1 0
0 0 4 1 1 2 0
0 0 1 2 5 2 0
0 4 1 0 2 9 6
0 1 8 5 2 6 7

T C G A
| | |
T – G A

Alignement local : Smith-Waterman

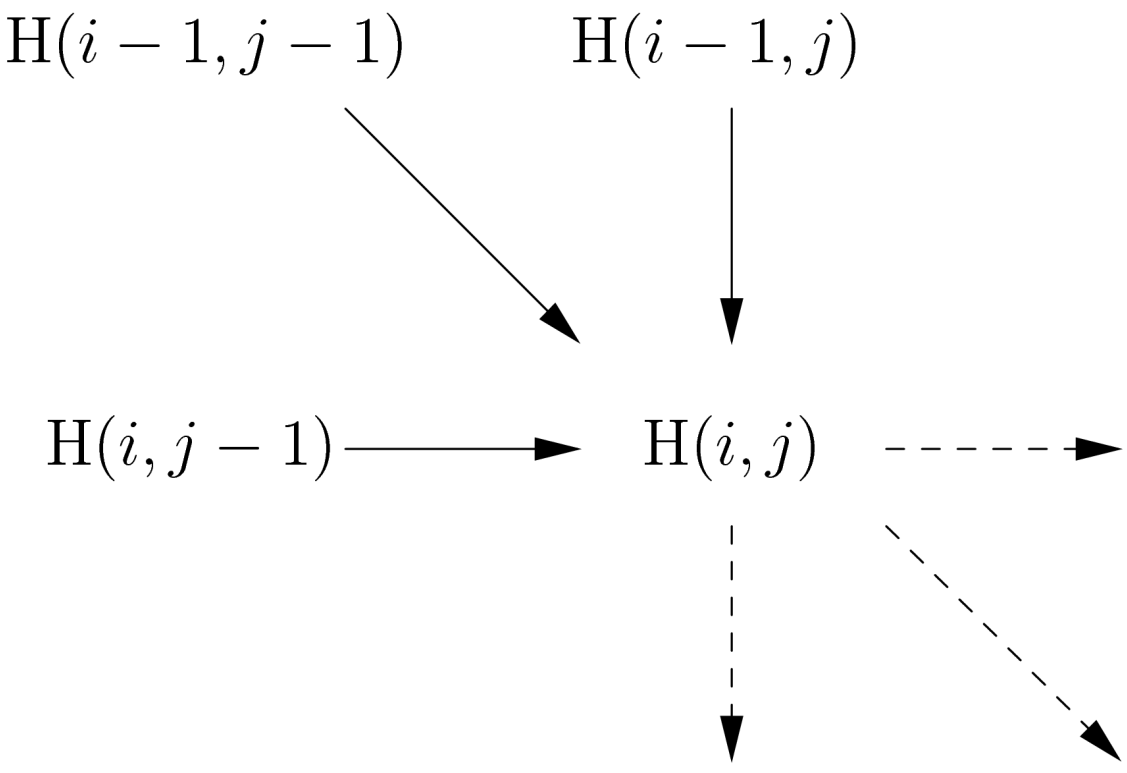
- $H(i,j)$ score maximum entre toutes les sous-séquences $x_a \dots x_i$ et $y_b \dots y_j$

$$\forall i,j : \quad H(i,0) = H(0,j) = 0 \quad \forall i,j, ij \neq 0 :$$

$$H(i,j) = \max \begin{cases} 0 & \text{(début d'un nouvel alig.)} \\ H(i-1,j-1) + d(x_i, y_j) & \text{(match ou substitution)} \\ H(i-1,j) - g_{\text{penalty}} & \text{(insertion)} \\ H(i,j-1) - g_{\text{penalty}} & \text{(délétion)} \end{cases}$$

$\longrightarrow$ $\max_{i,j} H(i,j)$ meilleure similarité entre X et Y

Localité du calcul

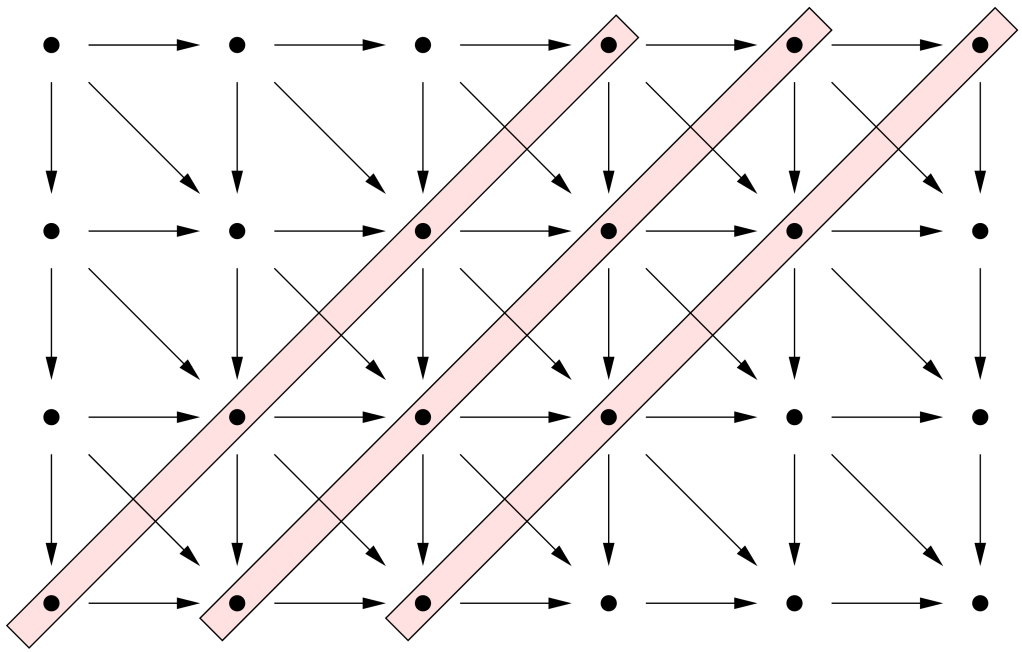


- Dépendance de trois cellules précédentes
- Seule information extérieure : $d(x_i, y_j)$
- Transmission des données vers les trois cellules suivantes

Complexités

Comparaison exhaustive, séquence de taille n contre séquence de taille m :

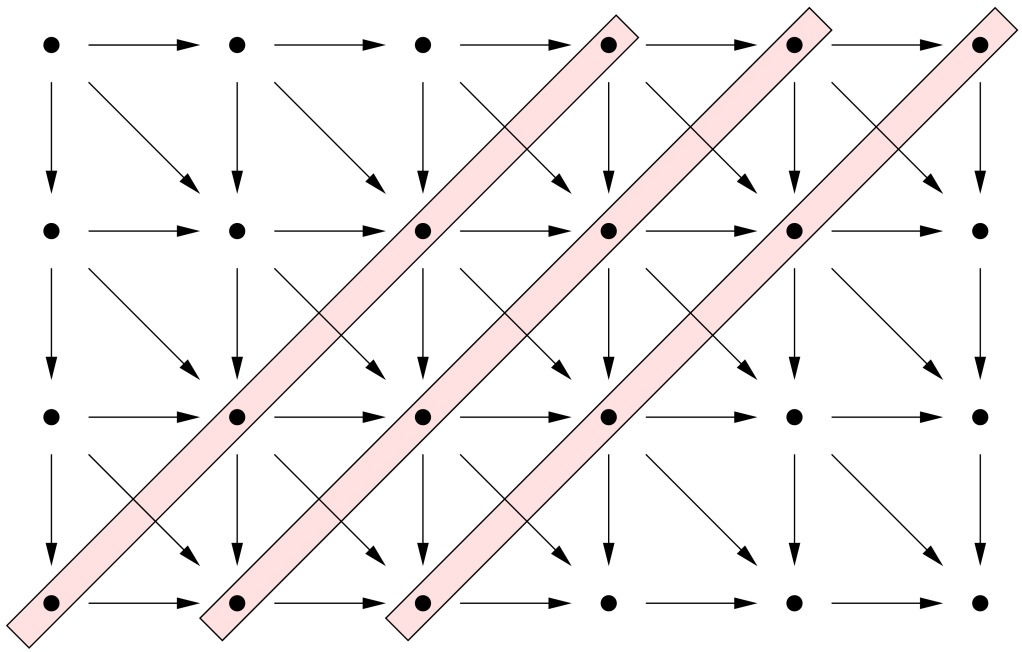
- $\mathcal{O}(mn)$ cellules
- calcul simultané de m cellules
- espace $\mathcal{O}(m)$



Complexités

Comparaison exhaustive, séquence de taille n contre séquence de taille m :

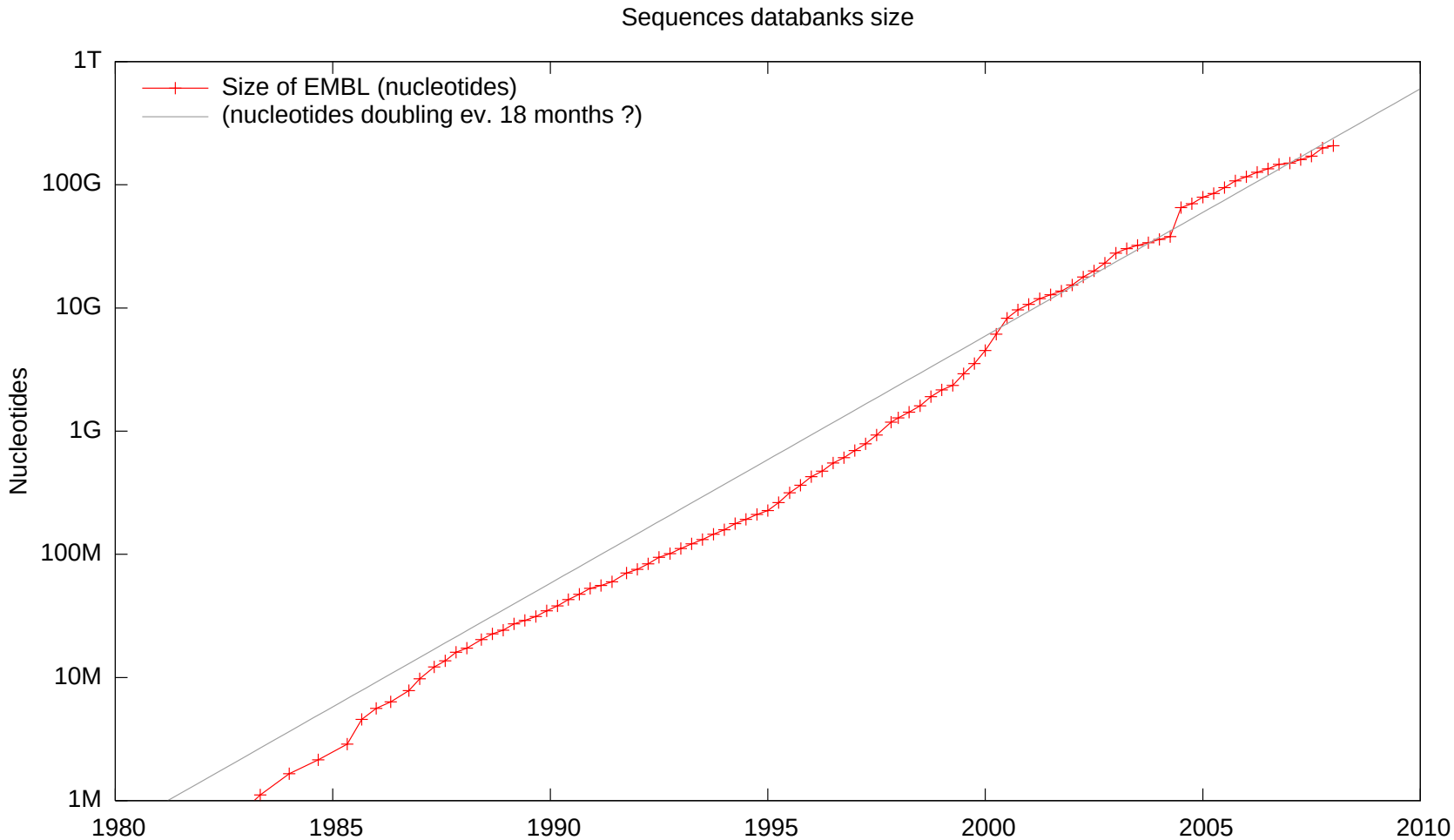
- $\mathcal{O}(mn)$ cellules
- calcul simultané de m cellules
- espace $\mathcal{O}(m)$



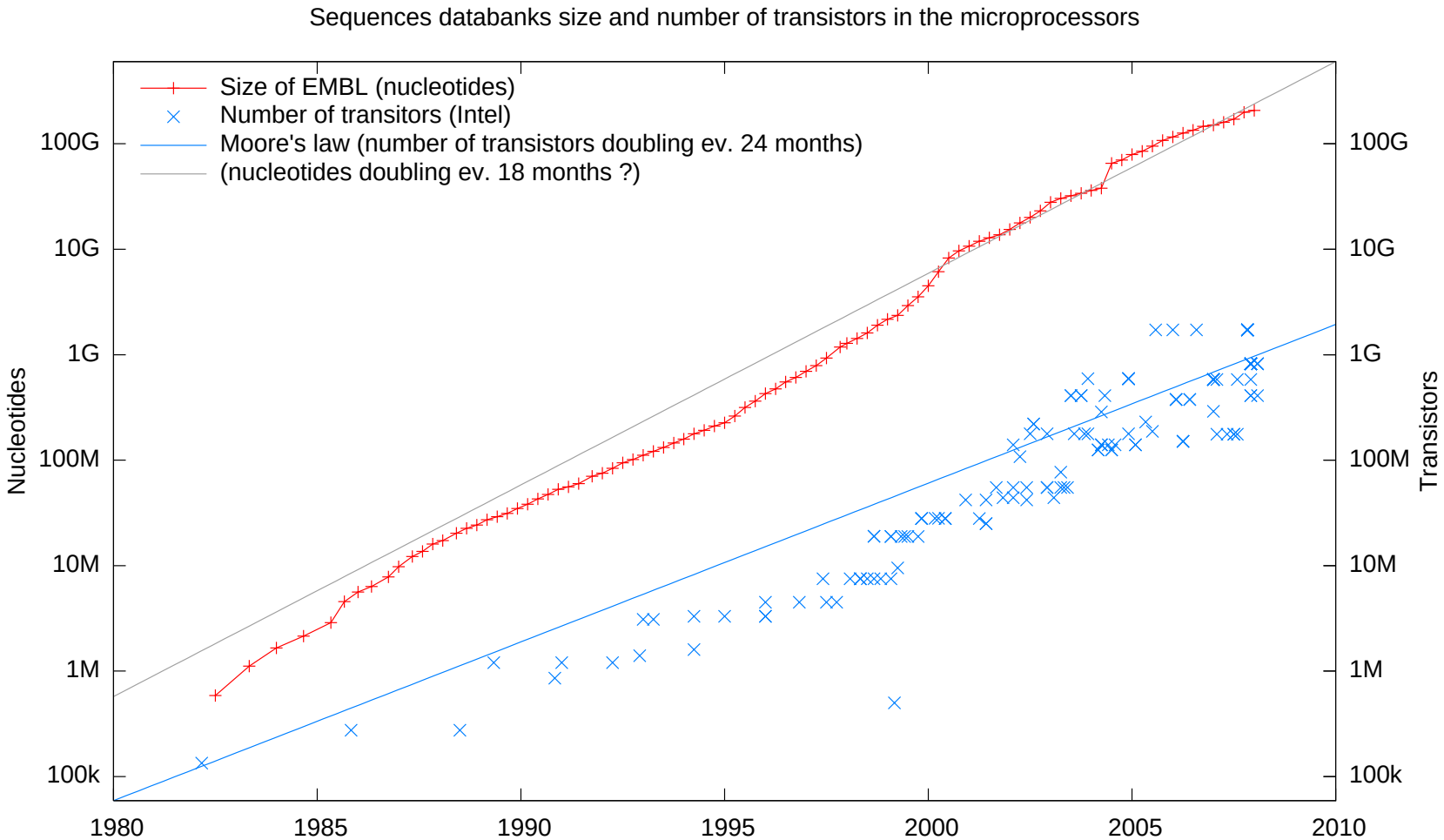
Algorithmes sous-quadratiques ?

- Masek et Paterson (1980) :
  $\mathcal{O}\left(n^2/\log n\right)$ pour scores rationnels
- Crochemore, Landau et Ziv-Ukelson (2002) :
  $\mathcal{O}\left(hn^2/\log n\right)$ (h entropie de la séquence)

Genomic bank sizes and Moore's Law



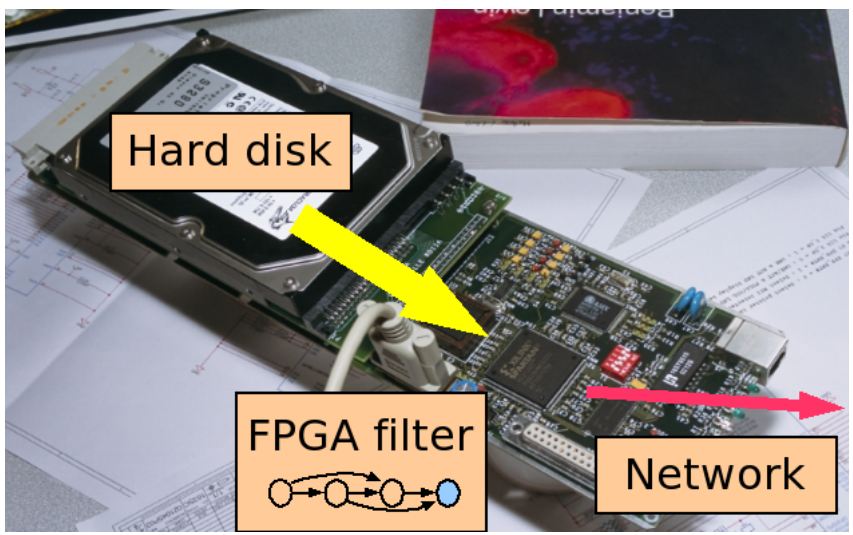
Genomic bank sizes and Moore's Law

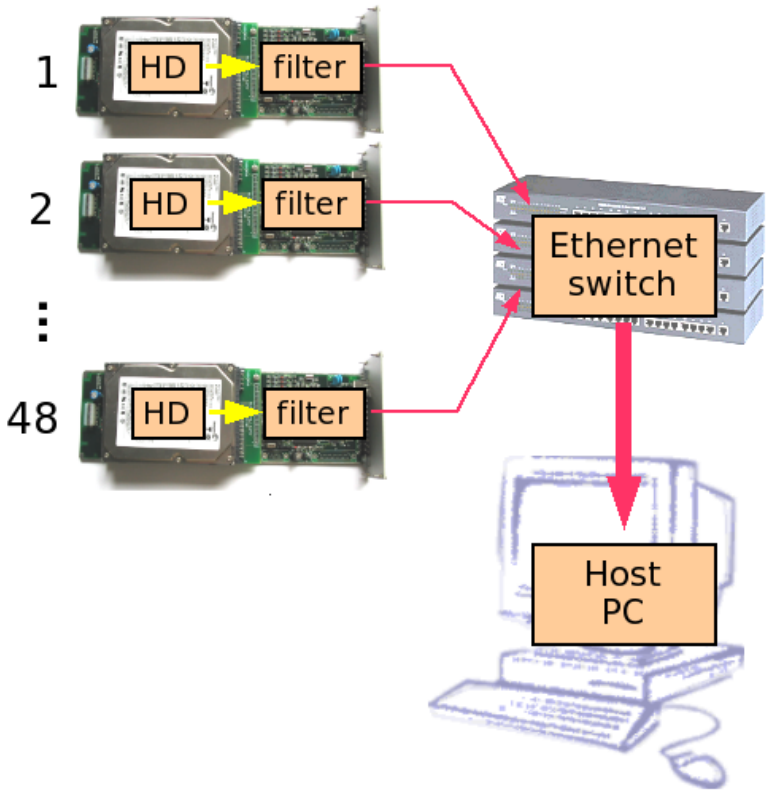


Architectures spécialisées FPGA

Rdisk

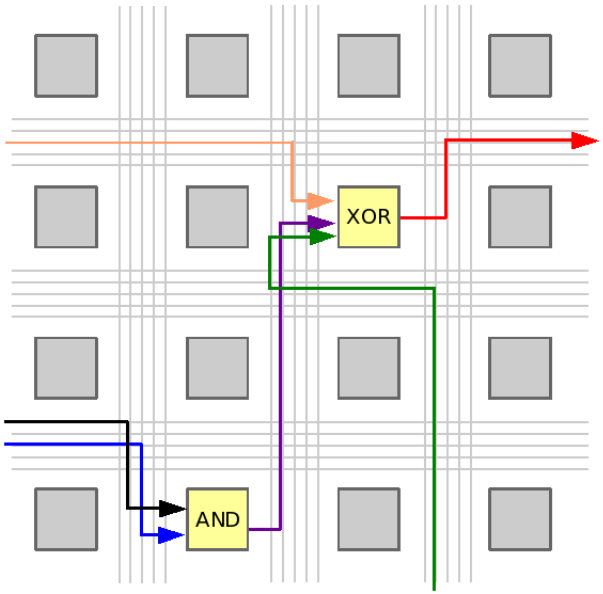
Filtrage de données directement à la sortie des disques durs





Système “économique” : $<$ 200 euros de composants

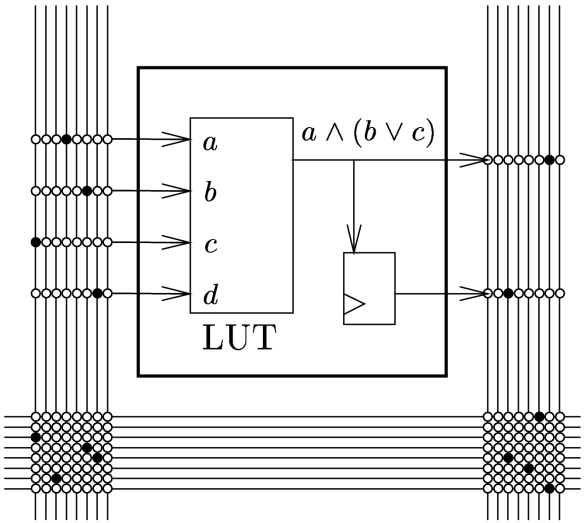
Les FPGAs, une puissance de calcul reconfigurable



- Grille de cellules logiques
- Interconnexion (routage)
- Reconfigurable
- Prototypage
- Circuits économiques

2007 : 100×10^6 portes logiques

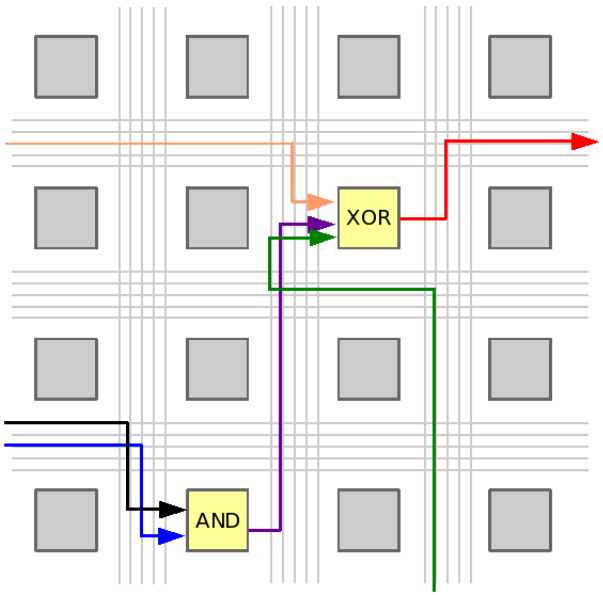
Les FPGAs, une puissance de calcul reconfigurable



- Grille de cellules logiques
- Interconnexion (routage)
- Reconfigurable
- Prototypage
- Circuits économiques

2007 : 100×10^6 portes logiques

Les FPGAs, une puissance de calcul reconfigurable

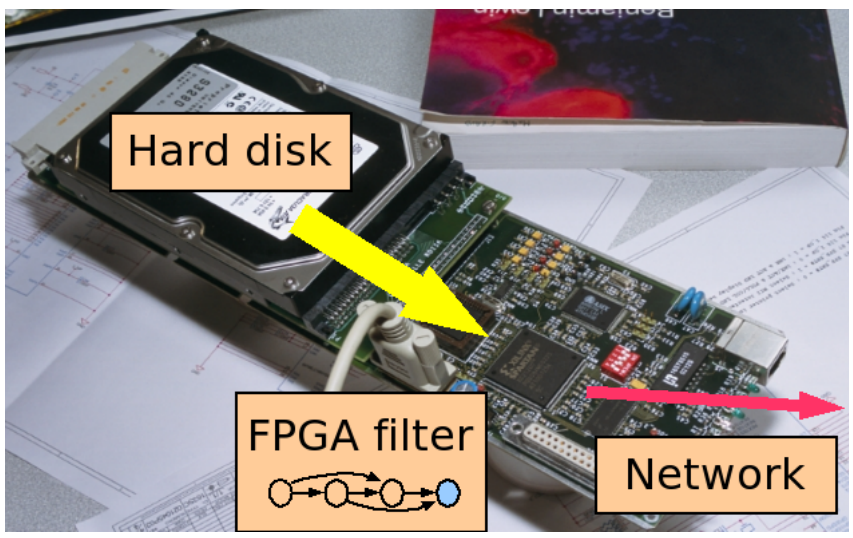


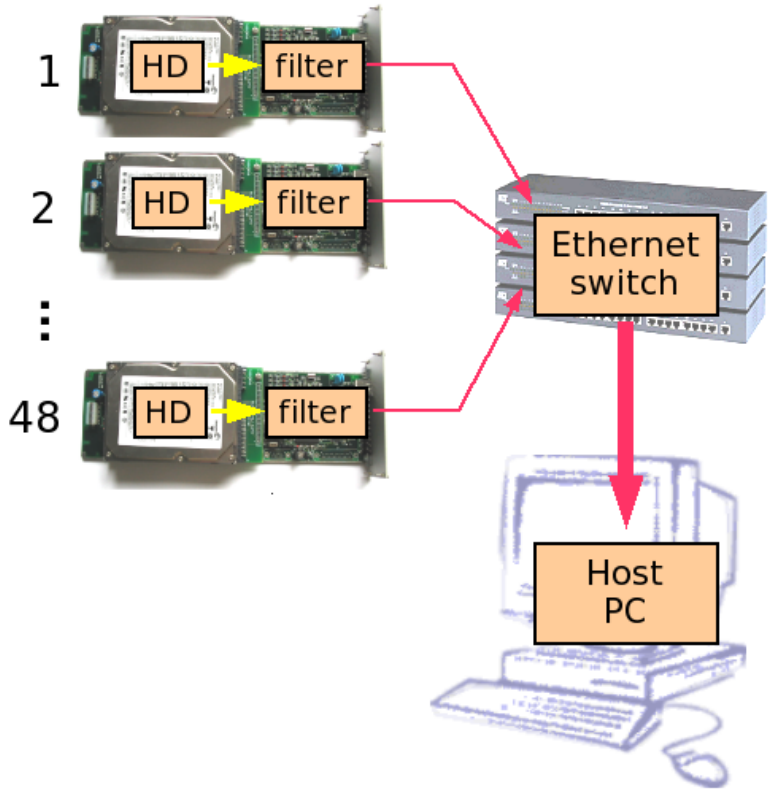
- Grille de cellules logiques
- Interconnexion (routage)
- Reconfigurable
- Prototypage
- Circuits économiques

2007 : 100×10^6 portes logiques

Rdisk

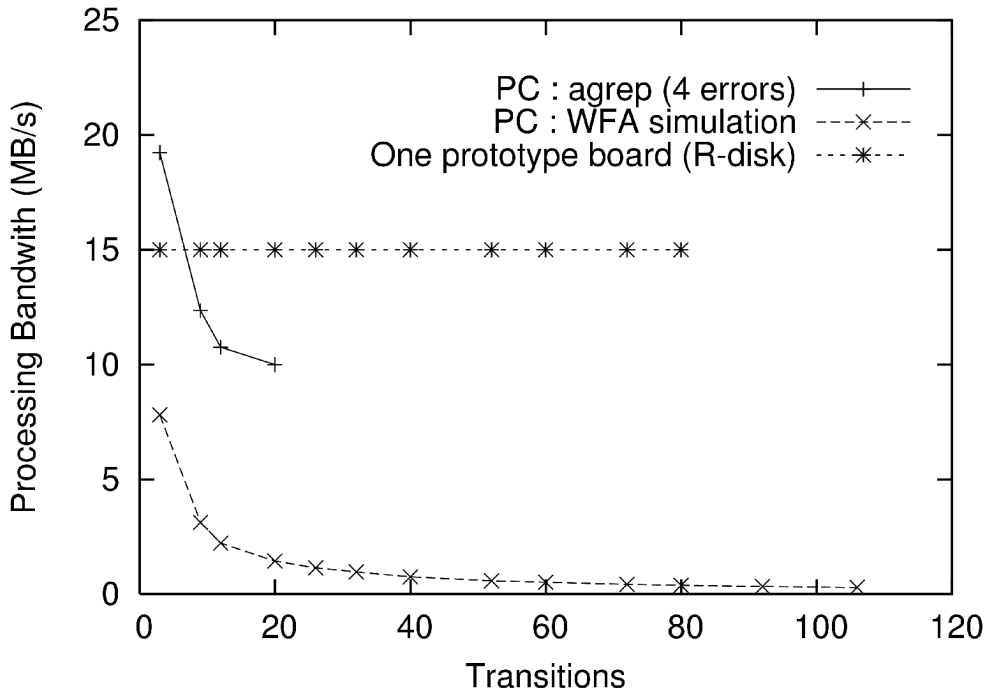
Filtrage de données directement à la sortie des disques durs





Système “économique” : $<$ 200 euros de composants

wapam/Rdisk : vitesse

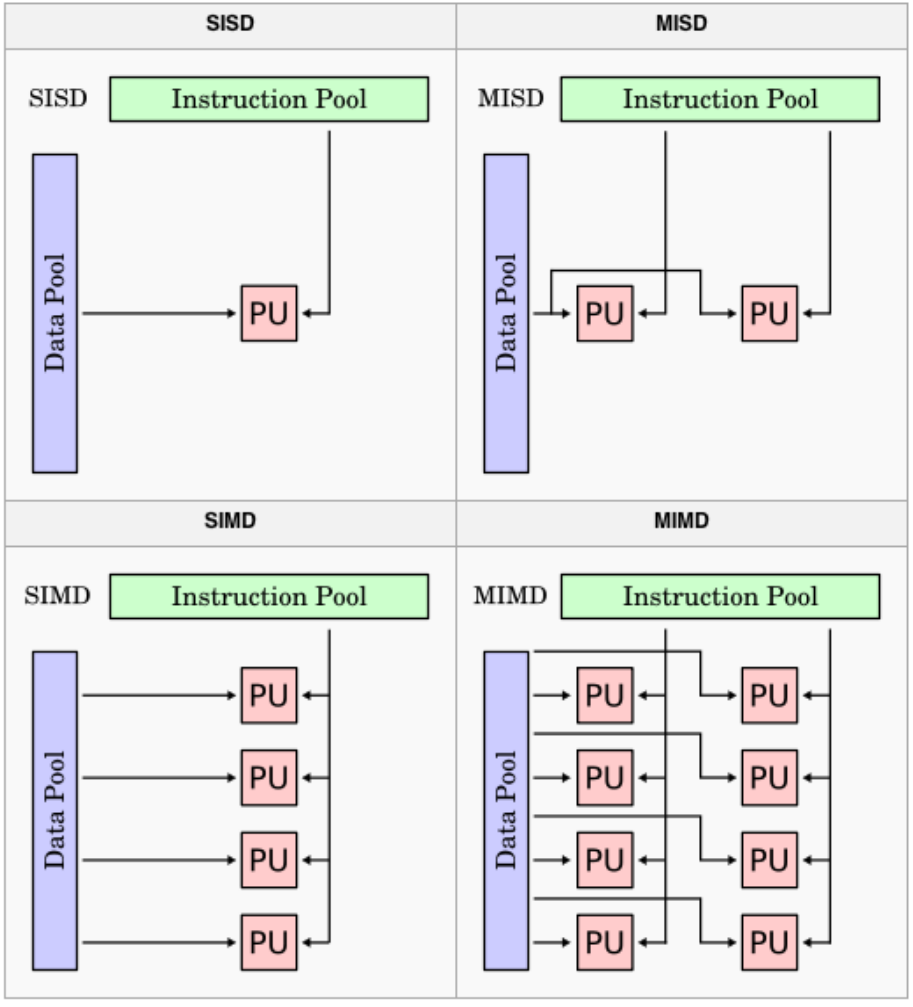


- Parallélisme
  - grain fin : 6 Gop/s
    (Spartan II, 40 MHz)
  - grain fort : R-disk 48

- Vitesse d'entrée : 16 Mo/s sur une carte
  - $4\times - 10\times$ speed-up vs PC (2 GHz, 728 Mo RAM)
  - Temps de calcul pour EMBL
    PC : $>$ 2 heures, 1 carte : 35 min, 48 cartes : 40 s
- Temps de compilation : 60 – 100 secondes

Calculs bioinformatiques sur cartes graphiques

Instruction parallelism



Flynn's Taxonomy [wikipedia]

- No parallelism : one instruction, one data
- SIMD (single instruction, multiple data)
  - vector processors (1970's), MMX, SSE...
  - bit-parallelism
- MIMD (multiple instruction, multiple data)
  - clusters, multi-core

Bioinformatics computations on GPU

- 2005 : RAxML
- 2006 : ClustalW
- 2007 : mumMER
- 2008 : Smith-Waterman, spliced sequences, Cell SW

- en cours : PWM, ADP, Séquenceurs

2005 : RAxML (phylogeny) [BrookGPU]

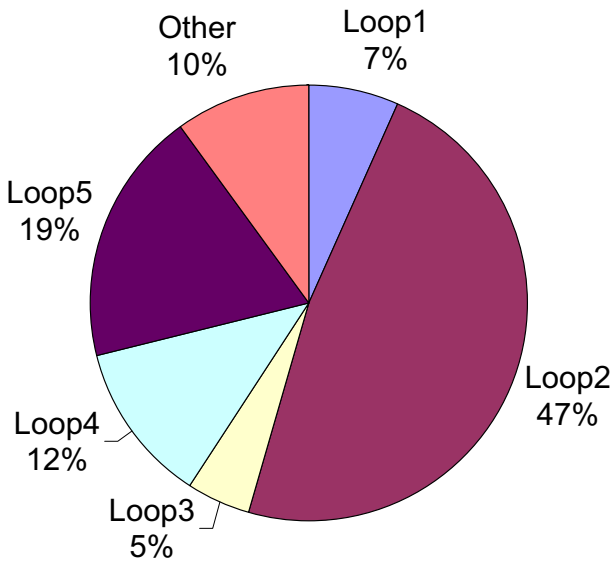
Initial Experiences Porting a Bioinformatics Application to a Graphics Processor

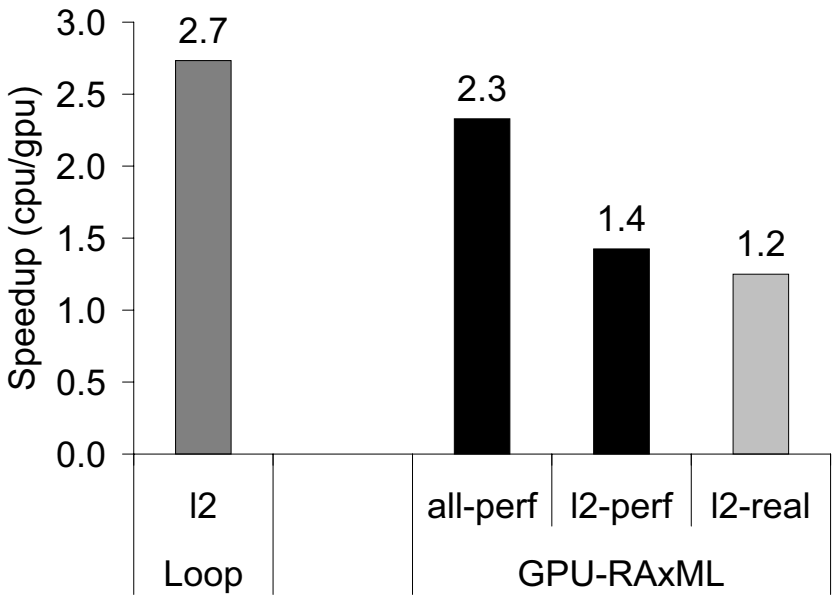
Maria Charalambous[1], Pedro Trancoso[1], and Alexandros Stamatakis[2]

2005 : RAxML (phylogeny) [BrookGPU]

Initial Experiences Porting a Bioinformatics Application to a Graphics Processor

Maria Charalambous[1], Pedro Trancoso[1], and Alexandros Stamatakis[2]





2006 : GPU-ClustalW [OpenGL Shading Lang]

GPU-ClustalW: Using Graphics Hardware to Accelerate Multiple Sequence Alignment

Weiguo Liu, Bertil Schmidt, Gerrit Voss, and Wolfgang Müller-Wittig

2006 : GPU-ClustalW [OpenGL Shading Lang]

GPU-ClustalW: Using Graphics Hardware to Accelerate Multiple Sequence Alignment

Weiguo Liu, Bertil Schmidt, Gerrit Voss, and Wolfgang Müller-Wittig

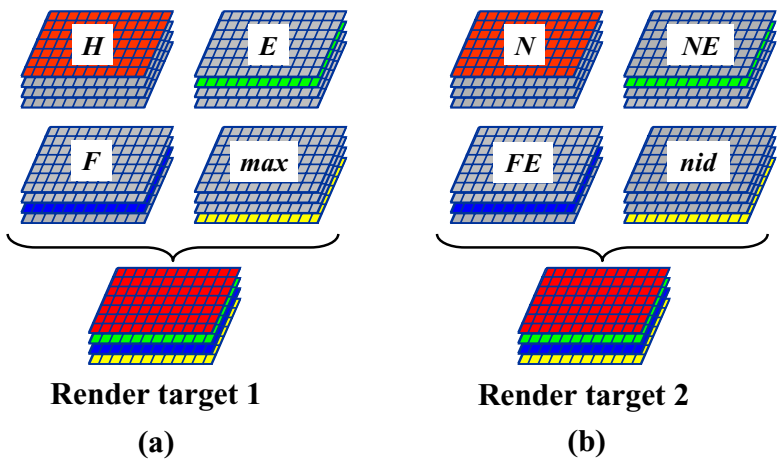


Fig. 6. Using the RGBA channels of two-dimensional texture buffers for the computation of H, E, F, max, N, NE, FE and nid

2006 : ClustalW GPU (Pairwise comparison)

Number of sequences (average length)		200 (412)	400 (408)	600 (462)
ClustalW (P4, 3GHz)	**Overall**	194.9	891.9	1818.1
	Pairalign	183.8 (94.4%)	833.1 (93.4%)	1697 (93.3%)
	Guided Tree	0.07 (0.03%)	0.8 (0.09%)	4.1 (0.2%)
	Malign	11.0 (5.6%)	58.0 (6.5%)	117.0 (6.4%)
GPU-ClustalW (GeForce 7800)	**Overall**	27.2	134.1	272.4
	Pairalign	16.1 (59.2%)	75.3 (56.2%)	151.3 (55.5%)
	Guided Tree	0.07 (0.3%)	0.8 (0.6%)	4.1 (1.5%)
	Malign	11.0 (40.4%)	58.0 (43.3%)	117.0 (43%)
Speedups	**Overall**	7.2	6.7	6.7
	Pairalign	11.4	11.1	11.2

2007 : MUMmerGPU (CUDA)

BMC Bioinformatics

BioMed Central

Software

Open Access

High-throughput sequence alignment using Graphics Processing Units

Michael C Schatz*†[1,2], Cole Trapnell†[1,2], Arthur L Delcher[1,2] and Amitabh Varshney[2]

2007 : MUMmerGPU (CUDA)

BMC Bioinformatics

BioMed Central

Software

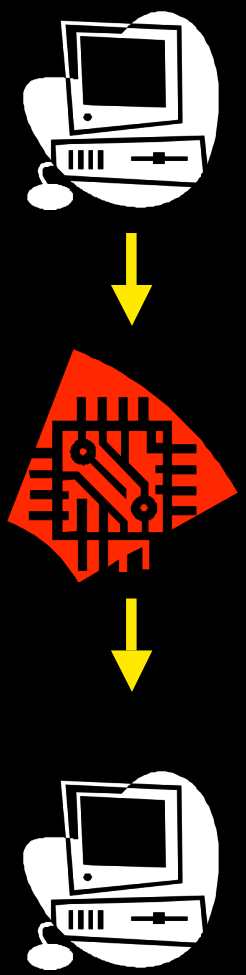
Open Access

High-throughput sequence alignment using Graphics Processing Units

Michael C Schatz*†[1,2], Cole Trapnell†[1,2], Arthur L Delcher[1,2] and Amitabh Varshney[2]

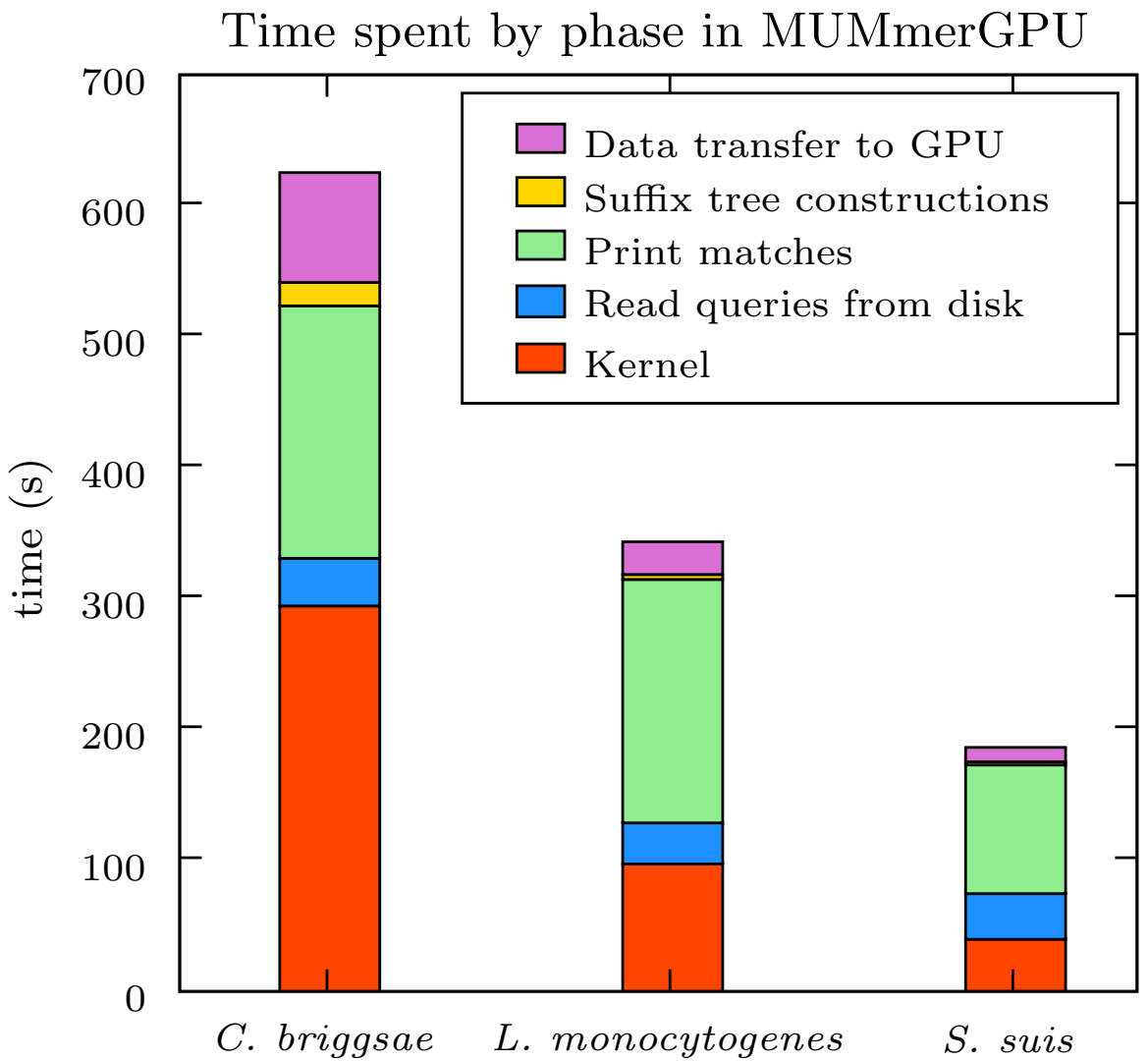
Table 1: Runtime parameters and speedup for MUMmerGPU test workloads. MUMmerGPU is consistently more than 3 times faster than mummer for a variety of sequencing data.

Reference	Reference length (bp)	# of queries	Query length mean ± stdev.	Min alignment length (*l*)	# of suffix trees (*k*)	Speedup
C. briggsae Chr. III (Sanger)	13,163,117	2,357,666	717.84 ± 159.44	100	2	3.71
L. monocytogenes (454)	2,944,528	6,620,471	200.54 ± 60.51	20	1	3.79
S. suis (Illumina/Solexa)	2,007,491	26,592,500	35.96 ± 0.27	20	1	3.47



1. Load Reference String
2. Create Suffix Tree
3. Reorder Tree Layout
4. Load Query Strings
5. Transfer data to GPU
6. Execute Query Kernel
   - Up to 128 simultaneous matches on GPU
7. Fetch Results from GPU
8. Output results

2007 : MUMmerGPU (CUDA)



2008 : Smith-Waterman (CUDA)

BMC Bioinformatics

BioMed Central

Research

Open Access

CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment

Svetlin A Manavski*[1,2] and Giorgio Valle[1]

2008 : Smith-Waterman (CUDA)

BMC Bioinformatics

BioMed Central

Research

Open Access

CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment

Svetlin A Manavski*[1,2] and Giorgio Valle[1]

Table 2: Smith-Waterman in CUDA running on single and double GPU vs. BLAST and SSEARCH

Sequence Name	*Length*	*SW-Cuda** *Time (s)*	*MCUPS*	*SW-Cuda*** *Time (s)*	*MCUPS*	*Ssearch(Fasta) Time (s)*	*MCUPS*	*Blast Time (s)*	*MCUPS*
O29181	63	2.98	1849	1.547	3561	46	119	3.7	1488
P03630	127	5.88	1889	3.075	3612	93	119	5.7	1948
P53765	255	12.31	1811	6.505	3428	184	121	11	2027
Q8ZGB4	361	17.44	1810	9.162	3446	275	114	16.3	1936
P58229	511	24.89	1795	13.326	3353	362	123	16.6	2691

2008 : Alignment of spliced sequences (CUDA)

GEAgpu: Improved alignment of spliced DNA sequences to genomic data using Graphics Processing Units

Svetlin A. Manavski, Alessandro Albiero, Claudio Forcato, Nicola Vitulo, Giorgio Valle

CRIBI, University of Padova, Padova, Italy E-mail: svetlin.manavski@cribi.unipd.it

- alignement genome / EST
- scan (heuristique)
- puis extension

	Runtime (min)	# of alignments	# of queries aligned	Runtime factor
GEAGpu	8	87,596	46,382	1
Blast	2,297	56,799	39,073	277
Blat	388	75,820	43,210	47
Soap	18	69,424	43,513	2

2008 : Sequence Alignment (Cell)

BMC Bioinformatics

BioMed Central

Software

Open Access

CBESW: Sequence Alignment on the Playstation 3

Adrianto Wirawan*, Chee Keong Kwoh, Nim Tri Hieu and Bertil Schmidt

2008 : Sequence Alignment (Cell)

BMC Bioinformatics

BioMed Central

Software

Open Access

CBESW: Sequence Alignment on the Playstation 3

Adrianto Wirawan*, Chee Keong Kwoh, Nim Tri Hieu and Bertil Schmidt

- 6 processeurs
- SIMD 128 bits
- 3.2 GHz

Projets GPU en cours, Sequoia, LIFL / INRIA Lille

- GPU PWM (Position Weight Matrices)
- Compilateur GPGPU pour méthodologie ADP
- GPU et séquenceurs à haut-débit
- Collaborations, sujets de stage : www.lifl.fr/∼giraud
- Soutien de NVIDIA

Projets GPU en cours, Sequoia, LIFL / INRIA Lille

- GPU PWM (Position Weight Matrices)
  - avec J.-S. Varré
  - speed-up de 10× à 20× sur scan et comparaison
- Compilateur GPGPU pour méthodologie ADP
- GPU et séquenceurs à haut-débit
- Collaborations, sujets de stage : www.lifl.fr/∼giraud
- Soutien de NVIDIA

GPU PWM Scan (with J.-S. Varré)

GPU PWM Scan (with J.-S. Varré)

H. Boukhatem, A. Ysmal (MSc students)

CPU	Core2 Duo 6600	2×2.4 GHz	2.0 Gop/s

GPU PWM Scan (with J.-S. Varré)

H. Boukhatem, A. Ysmal (MSc students)

CPU	Core2 Duo 6600	2×2.4 GHz	2.0 Gop/s
GPU 1	GeForce 8800 GTX	$16 \times 8 \times 576$ MHz	21.5 Gop/s
GPU 2	GeForce 8800 GTS	$16 \times 8 \times 650$ MHz	24.2 Gop/s
GPU 3	Quadro FX 570	$4 \times 8 \times 208$ MHz	2.7 Gop/s

GPU PWM Scan (with J.-S. Varré)

H. Boukhatem, A. Ysmal (MSc students)

CPU	Core2 Duo 6600	2×2.4 GHz	2.0 Gop/s
	+ TFM-Scan	2.4 GHz	2 – 8 Gop/s
GPU 1	GeForce 8800 GTX	$16 \times 8 \times 576$ MHz	21.5 Gop/s
GPU 2	GeForce 8800 GTS	$16 \times 8 \times 650$ MHz	24.2 Gop/s
GPU 3	Quadro FX 570	$4 \times 8 \times 208$ MHz	2.7 Gop/s

- $10\times$ speed-up

GPU PWM Scan (with J.-S. Varré)

H. Boukhatem, A. Ysmal (MSc students)

CPU	Core2 Duo 6600	2×2.4 GHz	2.0 Gop/s
	+ TFM-Scan	2.4 GHz	2 – 8 Gop/s
GPU 1	GeForce 8800 GTX	$16 \times 8 \times 576$ MHz	21.5 Gop/s
GPU 2	GeForce 8800 GTS	$16 \times 8 \times 650$ MHz	24.2 Gop/s
GPU 3	Quadro FX 570	$4 \times 8 \times 208$ MHz	2.7 Gop/s

- $10\times$ speed-up
- $2 - 3\times$ speed-up compared to dedicated algorithms
- Good scaling on the GPU to 1 GB genomes (500 seconds)

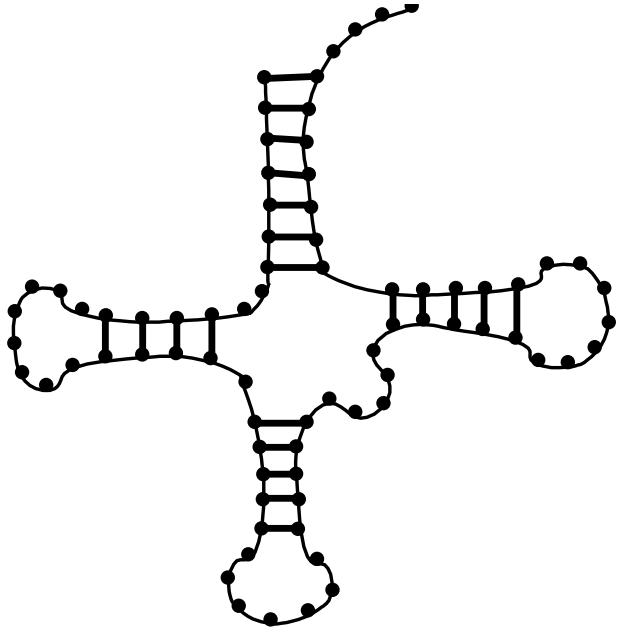
Projets GPU en cours, Sequoia, LIFL / INRIA Lille

- GPU PWM (Position Weight Matrices)
  - avec J.-S. Varré
  - speed-up de 10× à 20× sur scan et comparaison
- Compilateur GPGPU pour méthodologie ADP
  - avec P. Steffen, R. Giegerich (Univ. Bielefeld)
  - programmation dynamique générique
- GPU et séquenceurs à haut-débit
- Collaborations, sujets de stage : www.lifl.fr/∼giraud
- Soutien de NVIDIA

ADP (Algebraic Dynamic Programming)

Generic framework
for dynamic programming

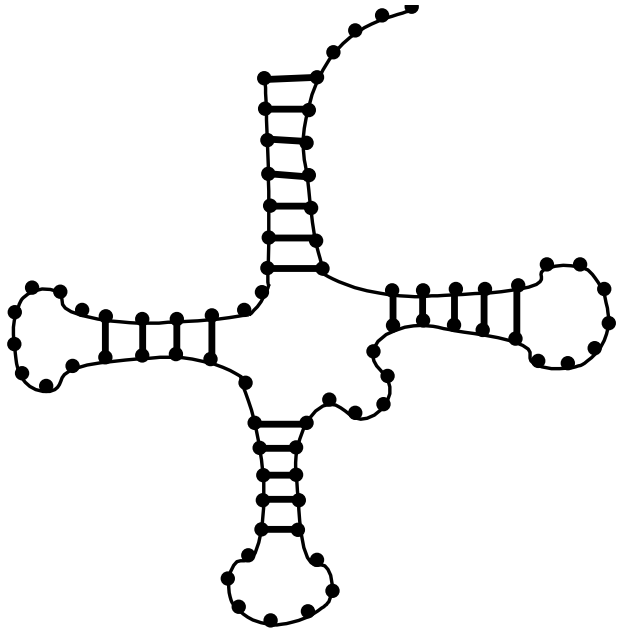
- Sequence alignments
- RNA foldings, co-foldings



ADP (Algebraic Dynamic Programming)

Generic framework
for dynamic programming

- Sequence alignments
- RNA foldings, co-foldings



nussinov78 $\quad Z = \mathrm{s}$

$$\mathrm{s} \rightarrow \underset{\mid}{\mathrm{nil}} \;\Big|\; \underset{\mathrm{s}\ \ \mathrm{s}}{\mathrm{right}} \;\Big|\; \underset{\mathrm{s}}{\underset{\mid}{\mathrm{pair}}} \quad \textbf{with basepairing} \quad \Big|\; \underset{\mathrm{s}\ \ \mathrm{s}}{\mathrm{split}}$$

$$\mathrm{nil} \rightarrow \varepsilon \qquad \mathrm{right} \rightarrow \mathrm{s}\ \mathrm{base} \qquad \mathrm{pair} \rightarrow \mathrm{base}\ \mathrm{s}\ \mathrm{base} \qquad \mathrm{split} \rightarrow \mathrm{s}\ \mathrm{s}$$

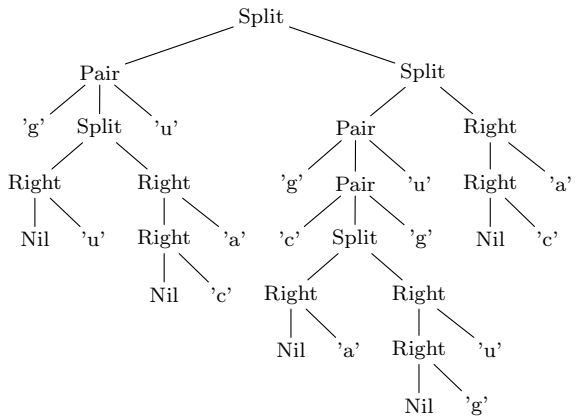
ADP (Algebraic Dynamic Programming)

Generic framework
for dynamic programming

- Sequence alignments
- RNA foldings, co-foldings

```
gucaugcaguguca
(...)((...))..
```

$t_1 =$



$$
\begin{array}{lcl}
\mathtt{Ans}_{bpmax} & = & \mathbb{N} \\
\mathtt{nil}_{bpmax}\,\mathtt{(s)} & = & \mathtt{0} \\
\mathtt{right}_{bpmax}\,\mathtt{(s,b)} & = & \mathtt{s} \\
\mathtt{pair}_{bpmax}\,\mathtt{(a,s,b)} & = & \mathtt{s + 1} \\
\mathtt{split}_{bpmax}\,\mathtt{(s,s')} & = & \mathtt{s + s'}
\end{array}
$$

$$
\begin{array}{lcl}
Ans_{\mathtt{bpmax}} & = & \mathbb{N} \\
\multicolumn{3}{l}{\mathtt{bpmax = (nil, right, pair, split, h) where}} \\
\mathtt{nil(s)} & = & \mathtt{0} \\
\mathtt{right(s,b)} & = & \mathtt{s} \\
\mathtt{pair(a,s,b)} & = & \mathtt{s + 1} \\
\mathtt{split(s,s')} & = & \mathtt{s + s'} \\
\mathtt{h([])} & = & \mathtt{[]} \\
\mathtt{h([}s_1,\ldots,s_r\mathtt{])} & = & \mathtt{[}\max_{1<i<r} s_i\mathtt{]}
\end{array}
$$

Projets GPU en cours, Sequoia, LIFL / INRIA Lille

- GPU PWM (Position Weight Matrices)
  - avec J.-S. Varré
  - speed-up de 10× à 20× sur scan et comparaison
- Compilateur GPGPU pour méthodologie ADP
  - avec P. Steffen, R. Giegerich (Univ. Bielefeld)
  - programmation dynamique générique
- GPU et séquenceurs à haut-débit
  - avec J.-M. Batto, N. Pons, F. Boumezbeur (INRA Jouy)
  - projet MetaHIT : méta-génome intestinal humain
- Collaborations, sujets de stage : www.lifl.fr/∼giraud
- Soutien de NVIDIA

Perspectives

- Calcul haute-performance : une révolution ?
  - non dans les concepts

Perspectives

- Calcul haute-performance : une révolution ?
  - oui dans les concepts, oui économiquement
  - $50\times$ peak speed-up $\longrightarrow$ $10\times$ vraiment possible

Perspectives

- Calcul haute-performance : une révolution ?
  - oui dans les concepts, oui économiquement
  - 50× peak speed-up $\longrightarrow$ 10× vraiment possible
- Côté informatique
  - Efferverscence, beaucoup de publications en 2008-09
  - Intérêt sur réflexion parallèle (et non un simple portage)
- Côté applications biologiques
  - Forte demande de solutions accélérées
  - Maturité des codes et des APIs ?

Perspectives

- Calcul haute-performance : une révolution ?
  - oui dans les concepts, oui économiquement
  - 50× peak speed-up $\longrightarrow$ 10× vraiment possible
- Côté informatique
  - Efferverscence, beaucoup de publications en 2008-09
  - Intérêt sur réflexion parallèle (et non un simple portage)
- Côté applications biologiques
  - Forte demande de solutions accélérées
  - Maturité des codes et des APIs ?

Merci !

Résumé

Les données bioinformatiques issues des séquenceurs sont toujours en croissance exponentielle. Aux génomes de référence s'ajoutent maintenant les variations individuelles tout comme les méta-génomes (séquences d'organismes prélevés dans un même milieu).

Nous présenterons dans cet exposé quelques traitements parallèles sur ces données : certains se contentent d'un parallélisme à gros grain, facile à mettre en oeuvre sur cluster ou sur GPU, d'autres demandent des analyses plus fines pour traiter au mieux les différents accès mémoire. La comparaison intensive de séquences est souvent au coeur de ces algorithmes, mais d'autres défis surgissent des dernières technologies, notamment avec les séquenceurs de dernière génération. Nous parlerons aussi d'une méthode générique pour certains problèmes de programmation dynamique.