Alignement SIMD pour le read-mapping

Quelques slides sur l'alignement SIMD pour le read-mapping ...

Laurent Noé (travail en cours avec Marta Gîrdea)

LIFL, Université Lille 1 - INRIA

Journées au vert

4-5 mai 2009 - Le Cap Hornu

Motivation : alignement rapide pour *read mapping* :

Motivation : alignement rapide pour *read mapping* :

Graines : filtrage des *reads*/du génome à mapper

Motivation : alignement rapide pour *read mapping* :

Graines : filtrage des *reads*/du génome à mapper

Alignement : méthode clef de l'algorithme (la plus gourmande)

Motivation : alignement rapide pour *read mapping* :

Graines : filtrage des *reads*/du génome à mapper

Alignement : méthode clef de l'algorithme (la plus gourmande)

La recherche exacte est très rapide ... mais ...

je préfère la recherche approchée ...

Motivation : alignement rapide pour *read mapping* :

Graines : filtrage des *reads*/du génome à mapper

Alignement : méthode clef de l'algorithme (la plus gourmande)

La recherche exacte est très rapide ... mais ...

je préfère la recherche approchée ...

Motivation : alignement rapide pour *read mapping* :

Graines : filtrage des *reads*/du génome à mapper

Alignement : méthode clef de l'algorithme (la plus gourmande)

La recherche exacte est très rapide ... mais ...

je préfère la recherche approchée ...

Raisons :

- la taille des *reads* augmente, le taux d'erreur aussi.

Motivation : alignement rapide pour *read mapping* :

Graines : filtrage des *reads*/du génome à mapper

Alignement : méthode clef de l'algorithme (la plus gourmande)

La recherche exacte est très rapide ... mais ...

je préfère la recherche approchée ...

Raisons :

- la taille des *reads* augmente, le taux d'erreur aussi.
- *reads* plus longs et significativité.

Motivation : alignement rapide pour *read mapping* :

Graines : filtrage des *reads*/du génome à mapper

Alignement : méthode clef de l'algorithme (la plus gourmande)

La recherche exacte est très rapide ... mais ...

je préfère la recherche approchée ...

Raisons :

- la taille des *reads* augmente, le taux d'erreur aussi.
- *reads* plus longs et significativité.

Approche Mixte : algorithme de mapping étendu :

- 1^{ere} passe *exacte (ou presque)* suivie d'une ...
- 2^{eme} passe *approchée* sur les *reads* non mappés.

Alignement

dans le contexte du *read mapping*

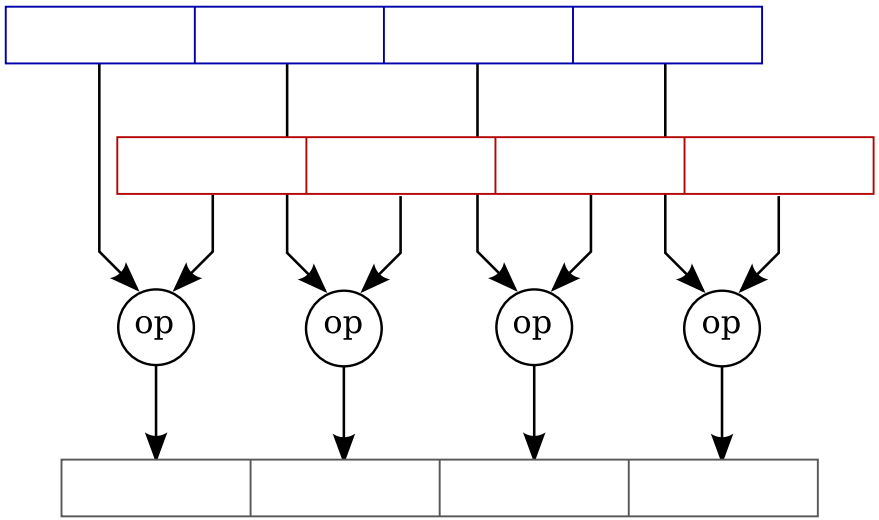
Smith-Waterman : Schrimp (utilise la version SIMD de Farrar)

Banded (semi-)global alignment : Maq

SIMD

Single Instruction Multiple Data

- Les processeurs courants proposent chacun leur version
  - Altivec (IBM/Motorola Power),
  - NEON (ARM),
  - MAX-2 (HP PA-RISC),
  - MMX/SSEx/AVX (Intel/AMD ix86{_64}),
  - VIS (Sun UltraSPARC).
- Sur ces processeurs, est implémenté par subdivision d'un mot machine en mots plus petits indépendants.
- Jeu d'instructions verticales (très majoritairement)



SIMD

(1/2) l'utiliser via l'auto-vectorisation

- GCC peut vectoriser automatiquement du code, à condition d'avoir :
  - des données alignées (malloc aligné),
  - des boucles sans dépendances (programmation explicite des diagonales)
  - de la chance ...
- pas de garantie de vectorisation, et assez complexe : même Fortran90 est plus adapté à ces aspects ...

```
a[0:N] = b[0:N] + c[0:N];
```

SIMD

(2/2) l'utiliser via les *Intrisics* d'Intel

- Manipulation explicite des vecteurs

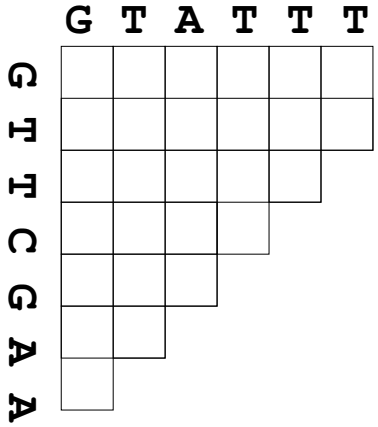
```
v4float va = mm_set (ptr_a);
v4float vb = mm_set (ptr_b);
v4float vr = mm_add (va, vb);
v4float vrmax = mm_max (vr, vrmax);
```

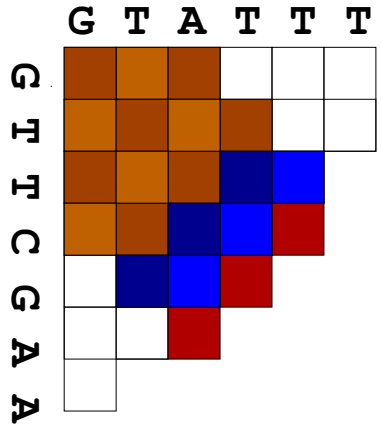
- lié à une architecture (Intel SSE2 par exemple),
- garantie de WYCIWYG, quel que soit le compilateur,
- aucun `__asm__("movl ...")` comme dans l'ancien temps.

SIMD

pour l'alignement *Banded-SW*

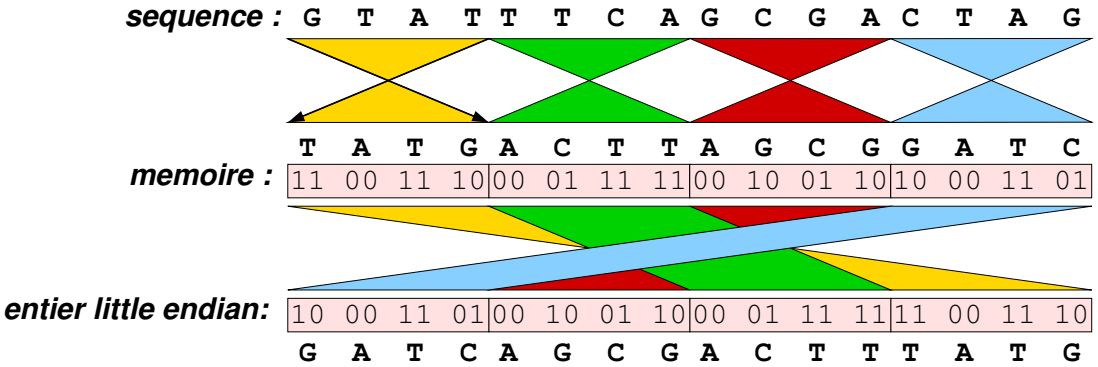
Un évidence : gérer une diagonale dans un vecteur SIMD.



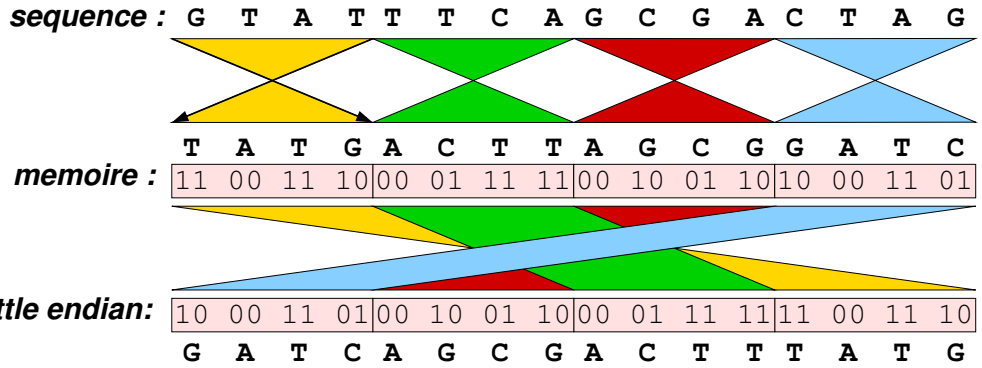


Mieux encore : faire **plusieurs** alignements "indépendants" dans un **même** vecteur SIMD.

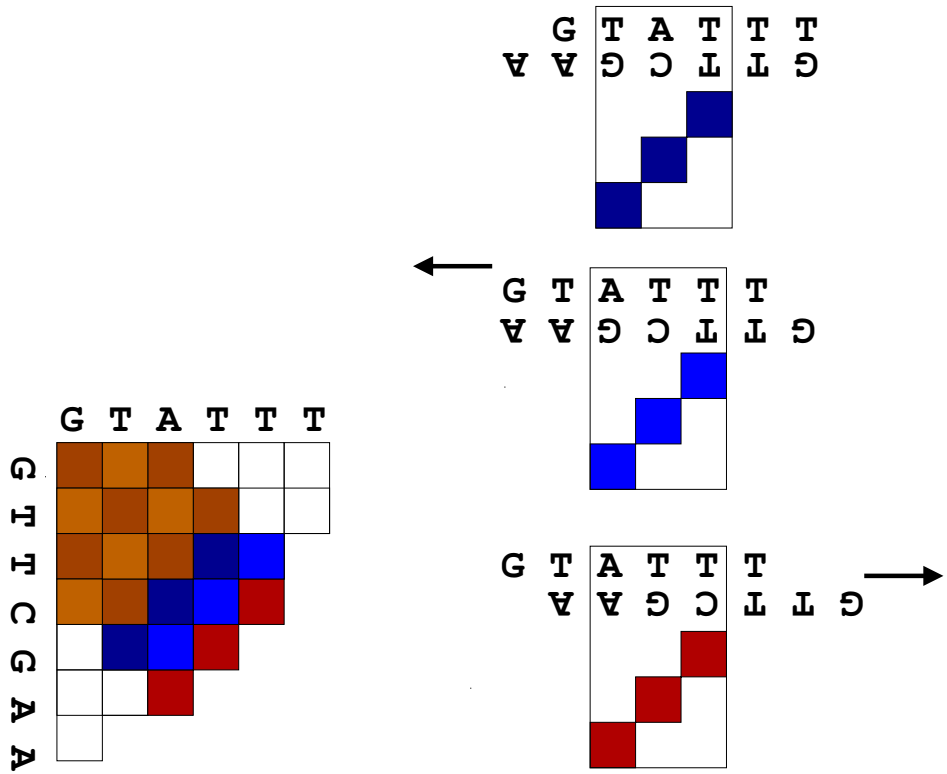
1. Accès aux données :



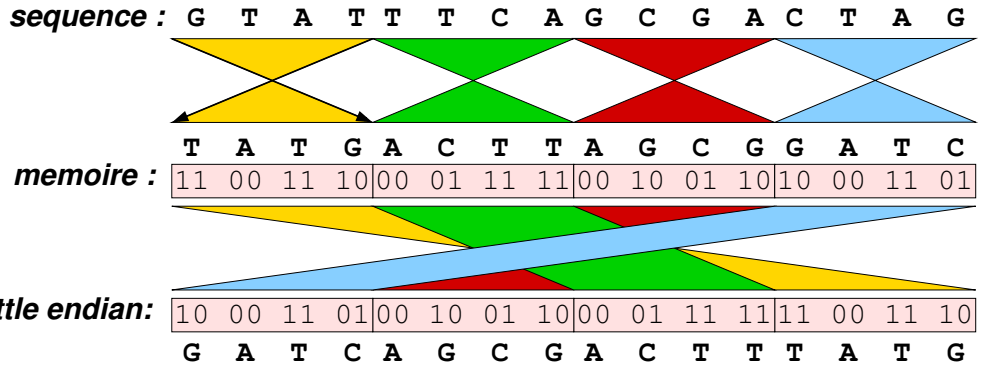
1. Accès aux données :



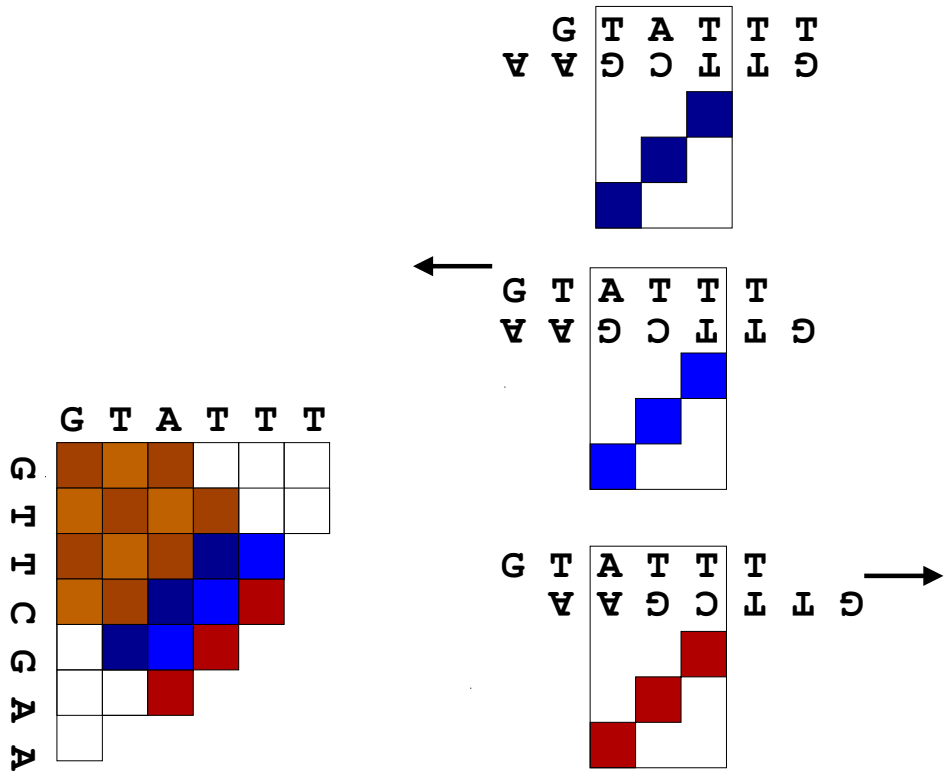
2. Calcul :



1. Accès aux données :



2. Calcul :



3. Résultat : un booleen seulement

SIMD *Banded-SW*

premiers tests

- réalisés avec le logiciel `map` de Marta:

graine : une graine espacée de poids 9,

reads : 50000 reads de taille 35 (sans facteur qualité), extrait de données SOLiD

genome : Ecoli K12.

SIMD *Banded-SW*

premiers tests

- réalisés avec le logiciel `map` de Marta:

graine : une graine espacée de poids 9,
reads : 50000 reads de taille 35 (sans facteur qualité), extrait de données SOLiD
genome : Ecoli K12.

- temps de la partie alignement (ignore *entrées/indexation/sorties*)

nb indels autorisés	sans filtre sse2	avec filtre sse2 (toujours fait sur 16 diagonales)
7	416s	49s
3	203s	30s (possibilité $\times 2$ sur partie filtre)
1	85s	20s (possibilité $\times 4$ sur partie filtre)

(analyse différentielle : facteur d'accélération $\geq$ 11 pour le filtre)