Méthodes de *mapping* de *reads* avec indexation des *reads*

Pierre Morisse

26 juin 2016

Plan de la présentation

Contexte

- Milieu des années 2000 $\Rightarrow$ Développement des séquenceurs à très haut débit (NGS)

- Production de millions de très courtes séquences appelées *reads*, utilisés pour résoudre des problèmes :
  - De *mapping*
  - D'assemblage
  - De traitement des 7 requêtes suivantes, pour f de longueur k fixé :
    1. Dans quels *reads* f apparaît ?
    2. Dans combien de *reads* f apparaît ?
    3. Quelles sont les occurrences de f ?
    4. Quel est le nombre d'occurrences de f ?
    5. Dans quels *reads* f n'apparaît qu'une fois ?
    6. Dans combien de *reads* f n'apparaît qu'une fois ?
    7. Quelles sont les occurrences de f dans les *reads* où f n'apparaît qu'une fois ?

Contexte

- *Reads* produits bruités $\Rightarrow$ Nécessité d'une procédure de correction avant utilisation
- Nécessité d'index ces *reads* pour traiter les différents problèmes identifiée dans [1], où les 7 requêtes précédentes et un index les supportant ont été développé
- De nombreuses méthodes d'indexation permettant de traiter ces problèmes existent

Définitions et notations

Définitions et notations

Alphabet : $\Sigma = \{A, C, G, T\}$

Séquence : Mot sur l'alphabet Σ

k-mer : Facteur de longueur k d'une séquence

Contig : Séquence générée par l'assemblage de plus courtes séquences se chevauchant

Gb : Gigabases

Description

- Ont pour but de produire des séquences à partir de ...
- Différentes technologies et plateformes $\Rightarrow$ Possibilité de traiter divers problèmes de génomiques
- Prix désormais abordable $\Rightarrow$ Séquençage accessible à tous
- Depuis peu, séquençage de *reads* de plus en plus longs $\Rightarrow$ Très utiles dans les problèmes d'assemblage
- Mais ces *reads* sont très bruités

Principaux séquenceurs

Technologie	Technique de séquençage	Plateforme	Nombre de *reads*	Longueur	Précision	Temps	Débit	Coût	Erreurs
Illumina	Synthèse, basé sur ADN polymérases	HiSeq 2500/1500 MiSeq	3 milliards 17 millions	36 - 100 25 - 250	99 >99	2 - 11 jours 4 - 27 heures	600 8,5	740 000 125 000	Substitutions
Roche	Pyroséquençage	454 GS FLX+ 454 GS Junior	1 million 1 million	700 400	99,997 >99	23 heures 10 heures	0,7 0,4	450 000 108 000	Indels.
ABI Life Technologies	Ligature Détection de protons	5500xl SOLiD Ion Proton Chip I/II	2,8 millions 60 - 80 millions	75 jusqu'à 200	99,99 >99	7 jours 2 heures	180 10 - 100	595 000 243 000	Indels.
Pacific Biosciences	Simple molécule en temps réel	PacBio RS	50 000	3 000 en moyenne	85	2 heures	13	750 000	Indels.
Oxford Nanopore	Exonucléase par Nanopore	GridION MinION	4 - 10 millions 70 000	dizaines de milliers dizaines de milliers	96 70	variable 48 heures	quelques dizaines 0,132	variable 1 000	Indels.

Méthodes de correction

Motivations

- *Reads* bruités
- Difficiles à utiliser
- Nécessité d'améliorer leur précision

Méthodes de correction

Principaux outils :

Outil	Structure de données	Erreurs corrigées	Nombre de *reads* (longueur)	Espace mémoire	Temps	*reads* corrigés
SHREC	Arbre des suffixes	subs.	1 090 946 (70)	1 500	183	88,56
HybridSHREC	Arbre des suffixes	subs. + indels	977 971 (178)	15 000	28	98,39
HiTEC	Table des suffixes	subs.	1 090 946 (70) 4 639 675 (70)	757 3 210	28 125	94,43
Fiona	Table des suffixes partielle	subs. + indels	977 971 (178) 2 464 690 (142)	2 000 3 000	15 32	66,76
Coral	Table de hachage	subs. + indels	977 971 (178)	8 000	5	92,88
RACER	Table de hachage	subs.	2 119 404 (75) 101 548 652 (457 595)	1 437 41 700	23 104	76,65 42,95
BLESS	Filtres de Bloom	subs. + indels	1 096 140 (101)	11	6	84,38
LoRDEC	Graphe de De Bruijn	subs. + indels	33 360 *reads* longs (2 938) et 2 313 613 *reads* courts (100)	960	10	85,78

Méthodes de *mapping*

Motivations

- Comparer ADN d'un individu à un génome de référence
- Détection de mutations dans l'ADN séquencé
- => TODO : un gène en particulier, à rechercher

Méthodes de *mapping*

Principaux outils :

Outil	Structure de données	Erreurs prises en compte	Nombre de *reads* (longueur)	Espace mémoire	Temps	*reads* mappés
MAQ	Table de hachage	subs. + indels	1 000 000 (44)	1 200	331	92,53
MrsFAST	Table de hachage	subs.	1 000 000 (100)	20 000	169	90,70
MrsFAST-Ultra	Table de hachage	subs.	2 000 000 (100)	2 000	57	91,41

Remarques

- Peu d'outils présentés ici
- De nombreux outils, n'utilisant pas de structure d'index sur les *reads*, existent et produisent de bons résultats

Méthodes de traitement des 7 requêtes

Motivations

- TODO : Revoir motivations requêtes

Méthodes de traitement des 7 requêtes

Principaux outils :

Outil	Structure de données	Nombre de *reads* (longueur)	Espace mémoire	Temps R1	Temps R2	Temps R3	Temps R4
GkA	Table des suffixes modifiée + Table des suffixes modifiée inverse + Table associant *k*-mer - nombre d'occurrences	42 400 000 (75)	20	16	25	25	0,1
CGkA	Table de suffixes échantillonnée + 3 vecteurs de bits	42 400 000 (75)	3 - 7	1203	28	1278	28
PgSA	Table des suffixes échantillonnée + Table auxiliaire d'information sur les *reads* et *k*-mers	42 400 000 (75)	1 - 4	70	58	70	58

Remarque

Les requêtes 5-7 sont exclues du comparatifs, car non implémentées dans GkA et CGkA au moment des tests réalisés.

Problématique

- *Reads* longs très utiles, notamment pour résoudre des problèmes d'assemblage longs et complexes
- Séquencer de tels *reads* est devenu rapide, peu coûteux et facile, notamment à l'aide de MinION
- Ces *reads* présentent un fort taux d'erreur
- La correction de ces *reads* longs par des méthodes classiques n'est pas aussi efficace que la correction de *reads* courts
- $\Rightarrow$ Nécessité de proposer une méthode alternative

Solution : les *reads* NaS

- Création de *reads* longs synthétiques via une approche hybride
- Peuvent atteindre une longueur de 60 000, disposent d'une précision de 99,99%, et peuvent donc s'aligner intégralement et sans erreurs
- $\Rightarrow$ Première solution efficace permettant d'appliquer un traitement correctif aux *reads* longs

Solution : les *reads* NaS

Nous présentons ici deux méthodes de synthèse des *reads* NaS :

- La première [2] nécessite d'aligner les *reads* courts sur les *reads* longs, mais également entre eux
- La deuxième, que nous avons mis en place, vise à ne déduire des informations qu'à partir de l'alignement des *reads* courts sur les *reads* longs

Jeu de données utilisé

- 66 492 *reads* longs MinION répartis en 5 ensembles comme suit :

Ensemble	Nombre de *reads*	% *reads* 2D	% taille totale
1	9 241	6,5	14,6
2	3 990	13,6	27,1
3	6 052	43,3	57,1
4	11 957	11,6	42,7
5	35 252	9,7	44,6

- 83,2% des *reads* 2D et 16,6% des *reads* 1D alignés
- Identité moyenne de 74,5% et 56,5%, respectivement
- Deux ensembles de 5 984 858 *reads* courts Illumina

Première méthode

Nous présentons ici la méthode pour le traitement d'un *read* long :

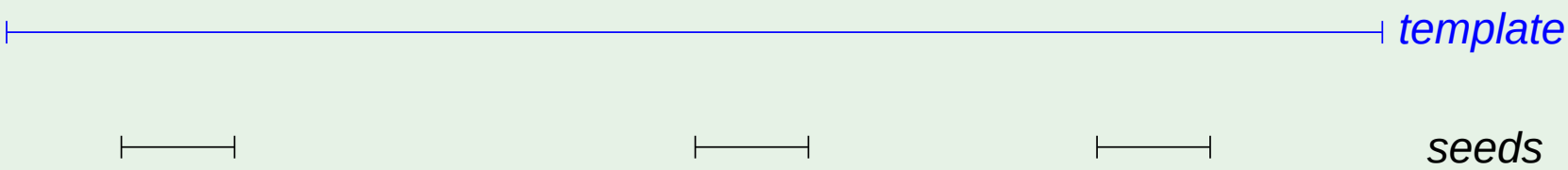
4 étapes

1. Alignement des *reads* courts sur le *read* long *template*
2. Recrutement de nouveaux *reads*, en alignant les *reads* courts entre eux
3. Micro-assemblage de l'ensemble de *reads* obtenu
4. Obtention d'un contig

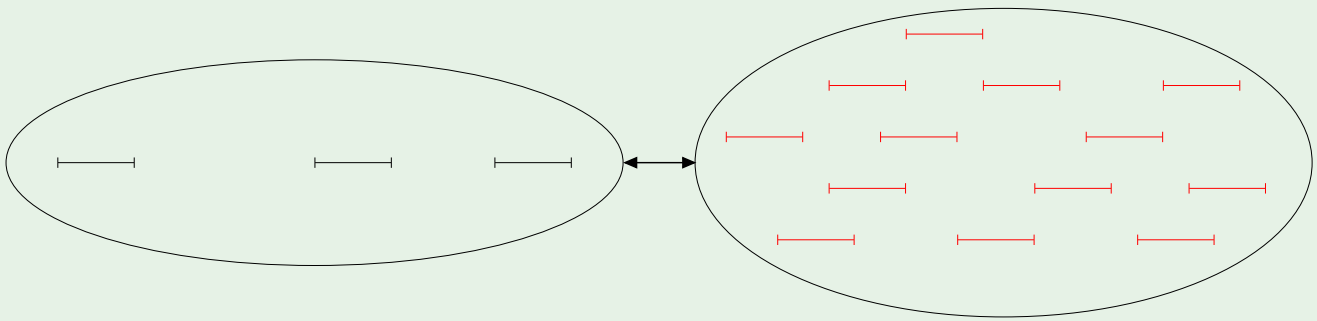
Première méthode

Illustration

1. Alignement des *reads* courts sur le *read* long *template*



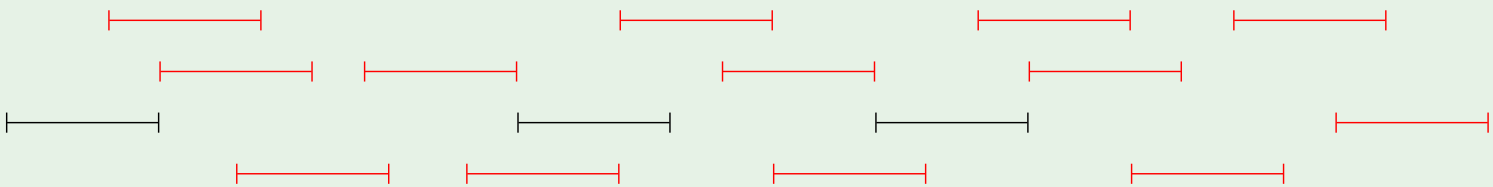
2. Recrutement de nouveaux *reads* courts



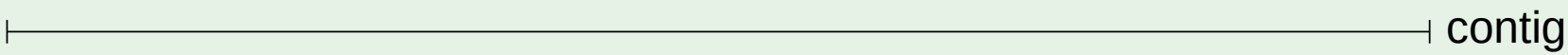
Première méthode

Illustration

1. Micro-assemblage de l'ensemble de *reads* obtenu



2. Obtention d'un contig



Première méthode

En général un unique contig est produit, mais de mauvais *reads* peuvent être recrutés et produire des contigs erronés

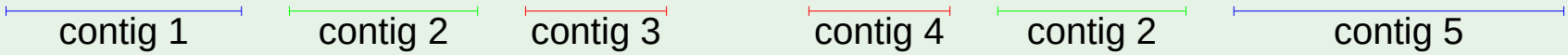
4 étapes supplémentaires

1. Obtention de plusieurs contigs
2. Construction du graphe des contigs
3. Sélection du chemin optimal
4. Vérification du contig obtenu, par alignement des *reads* courts

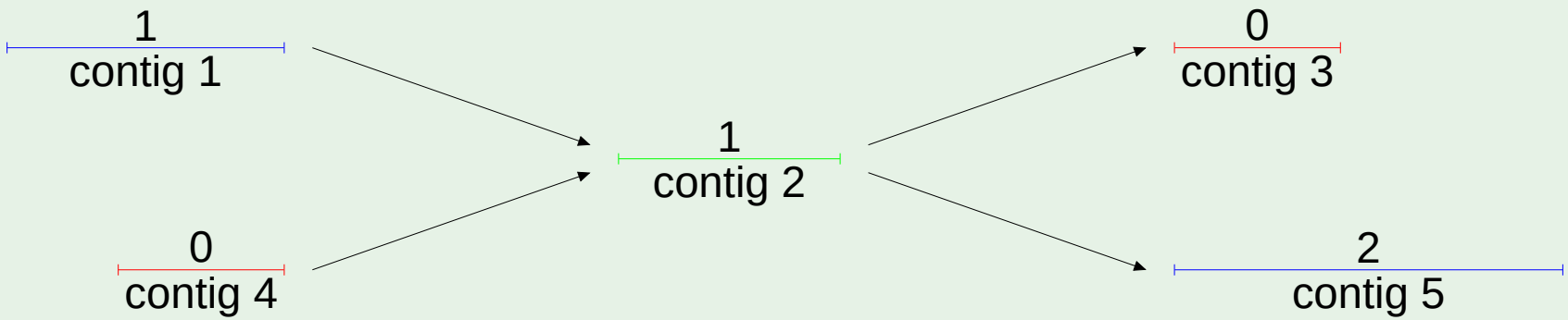
Première méthode

Illustration

1. Obtention de plusieurs contigs



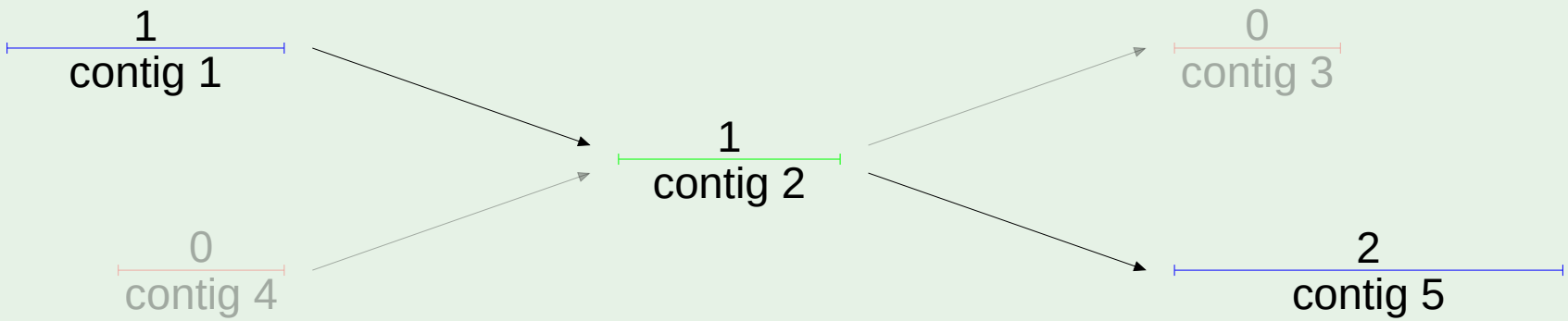
2. Construction du graphe des contigs



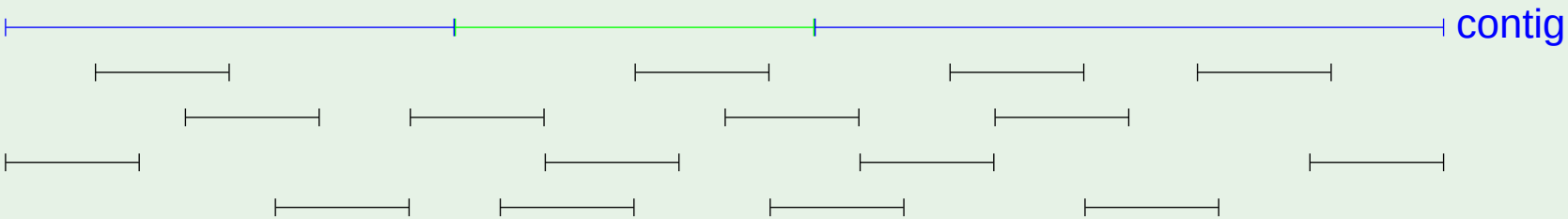
Première méthode

Illustration

1. Sélection du chemin optimal



2. Vérification du contig, par alignement des *reads* courts



Première méthode

Résultats

- 11 275 *reads* NaS produits
- Longueur maximale de 59 863
- Seulement 17% des *reads* longs ont produit un *read* NaS (76,4% 2D, 8,1% 1D)
- Certains *reads* NaS sont plus longs que leur *template* de référence
- Temps de traitement d'un *read* long : Moins d'une minute en moyenne

Première méthode

Résultats

- Les *reads* NaS produits couvrent 99,96% du génome de référence
- Identité moyenne de 99,99%
- 97% s'alignent sans erreur
- 99,2% s'alignent avec au plus une erreur

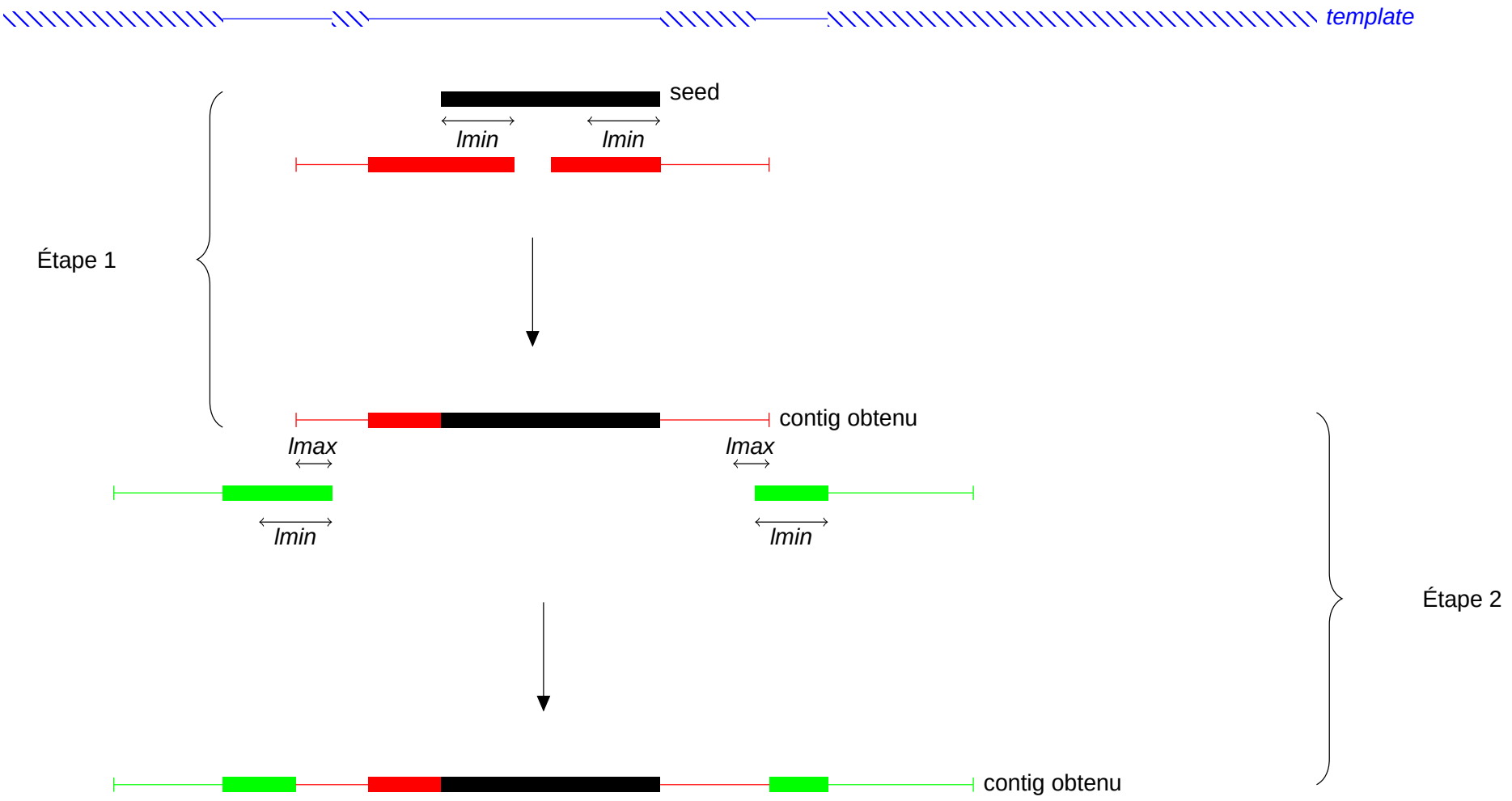
Notre méthode

Nous présentons la méthode pour le traitement d'un *read* long

Principe

- Alignement des *reads* courts sur le *read* long *template*, en se fixant un sueil *lmin*, pour récupérer les *reads* :
  - Totalement alignés, et servant de *seeds*
  - Avec un préfixe de longueur $\geq$ *lmin* aligné
  - Avec un suffixe de longueur $\geq$ *lmin* aligné
- Deux étapes d'extensions :
  1. Recrutement de *reads* partiellement alignés, similaires aux *seeds*
  2. Recrutement de nouveaux *reads* partiellement alignés, sans relation de similarité, en se fixant un nouveau seuil *lmax*

Notre méthode



Notre méthode

Détails sur la première étape ?

Notre méthode

Détails sur la deuxième étape ?

Notre méthode

Présentation des résultats

Conclusion

Conclusion

N. Philippe, M. Salson, T. Lecroq, M. Leonard, T. Commes, and E. Rivals.

Querying large read collections in main memory : a versatile data structure.

BMC bioinformatics, 12(1) :242, 2011.

M.-A. Madoui, S. Engelen, C. Cruaud, C. Belser, L. Bertrand, A. Alberti, A. Lemainque, P. Wincker, and J.-M. Aury.

Genome assembly using Nanopore-guided long and error-free DNA reads.

BMC Genomics, 16 :327, 2015.