

Introduction

Depuis le milieu des années 2000 et le développement des séquenceurs à très haut débit (*Next Generation Sequencing*), la biologie doit faire face au traitement d'énormes quantités de données, formées par des millions de courtes séquences appelées *reads*. Ces *reads* sont utilisés pour le traitement de problèmes de *mapping* ou d'assemblage, et doivent souvent subir une procédure de correction avant utilisation, afin de diminuer le taux d'erreurs de séquençage qu'ils présentent. Philippe et al. ont souligné, dans [1], l'importance de l'indexation des *reads* afin de résoudre ces problèmes, et ont développé un index supportant les 7 requêtes suivantes, pour une séquence f de longueur k donnée :

- Dans quels *reads* f apparaît ?
- Dans combien de *reads* f apparaît ?
- Quelles sont les occurrences de f ?
- Quel est le nombre d'occurrences de f ?
- Dans quels *reads* f n'apparaît qu'une fois ?
- Dans combien de *reads* f n'apparaît qu'une fois ?
- Quelles sont les occurrences de f dans les *reads* où f n'apparaît qu'une fois ?

Nous présentons ici l'état de l'art concernant les technologies de séquençage et les problèmes susmentionnés, ainsi que le problème sur lequel nous nous sommes principalement penchés.

État de l'art

Les tableaux présentés ci-dessous résument brièvement l'état de l'art concernant les technologies de séquençage et les outils existants, utilisant une structure d'index sur les *reads*, permettant de résoudre des problèmes de correction, de *mapping*, ou de traitement des 7 requêtes précédentes.

Technologie	Technique de séquençage	Plateforme	Nombre de <i>reads</i>	Longueur	Précision (en %)	Temps	Débit (en Gb)	Coût (en \$)	Erreurs
Illumina	Synthèse, basé sur ADN polymérisés	HiSeq 2500/1500	3 milliards	36 - 100	99	2 - 11 jours	600	740 000	Substitutions
Roche	Pyroséquençage	454 GS FLX+	17 millions	25 - 250	99	4 - 27 heures	8,5	125 000	Indels
ABI Life Technologies	Ligature	5500xl SOLiD	2,6 millions	75	99,997	23 heures	0,7	450 000	Indels
Pacific Biosciences	Détection de protons	Ion Proton Chip IRT	60 - 80 millions	100 - 200	99,99	10 heures	0,4	108 000	Indels
Oxford Nanopore	Exonuclease par Nanopore	GridION	4 - 10 millions	4 - 10 millions	96	2 heures	10 - 100	243 000	Indels
		MinION	70 000	1000 en moyenne	95	quelques dizaines	variable	750 000	Indels

TABLE 1 : Récapitulatif des différentes technologies de séquençage. Attention, Gb signifie ici Gigabases, et non Gigabits.

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

TABLE 2 : Récapitulatif des différentes méthodes de correction des *reads*. La valeur indiquée pour le nombre de *reads* corrigés est une moyenne obtenue à partir de différents ensembles de données.

Structure de données	Erreurs prises en compte	Nombre de <i>reads</i> (longueur)	Espace mémoire (en Mo)	Temps (en min)	<i>reads</i> mappés (en %)
Table de hachage	subs. + indels	1 000 000 (44)	1 200	331	92,53
Table de hachage	subs.	1 000 000 (100)	20 000	169	90,70

TABLE 3 : Récapitulatif des différentes méthodes de *mapping* de *reads*. La valeur indiquée pour le nombre de *reads* mappés est une moyenne obtenue à partir de différents ensembles de données.

Peu de méthodes sont présentés ici, mais de nombreuses alternatives, n'utilisant pas de structure d'index sur les *reads*, existent et produisent de très bons résultats, aussi bien en espace et en temps, qu'en qualité de *mapping*.

Structure de données	Nombre de <i>reads</i> (longueur)	Espace mémoire (en Go)	Temps R1 (en ms)	Temps R2 (en ms)	Temps R3 (en ms)	Temps R4 (en ms)
Table des suffixes modifiée						
Table des suffixes modifiée inverse	42 400 000 (75)	20	16	25	25	0,1
Table associant k -mer - nombre d'occurrences						
Table de suffixes échantillonnée	42 400 000 (75)	3 - 7	1203	28	1278	28
2 vecteurs de bits						
Table des suffixes échantillonnée	42 400 000 (75)	1 - 4	70	58	70	58
Table auxiliaire d'information sur les <i>reads</i> et k -mers						

TABLE 4 : Récapitulatif des différentes méthodes permettant de traiter les 7 requêtes. Les requêtes 5-7 sont exclues du comparatif, car non implémentées dans toutes les méthodes de traitement.

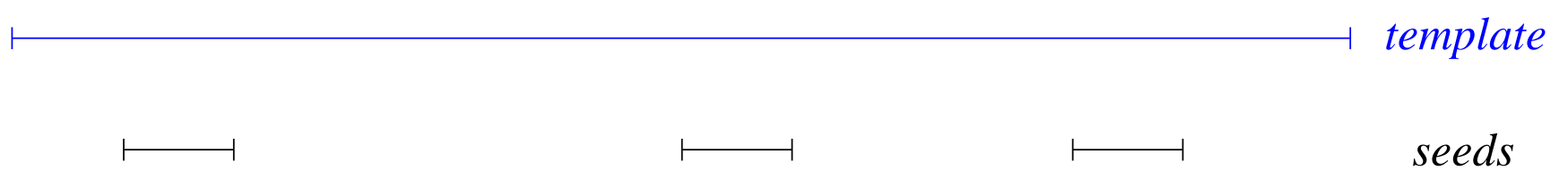
Correction de *reads* longs : Les *reads* NaS

Les nouvelles technologies de séquençage permettent de séquencer des *reads* de plus en plus longs, très utiles pour résoudre des problèmes d'assemblage. Ils disposent cependant d'un important taux d'erreur, avoisinant notamment les 30% pour les *reads* séquencés par la plateforme MinION. Comme le montre la Table 2, les méthodes de correction classiques ne sont pas adaptées à de tels *reads*, et sont donc très peu efficaces. Une solution alternative pour résoudre ce problème est la génération de *reads* dits synthétiques. Ces derniers sont générés via une approche hybride, utilisant des *reads* longs comme *templates* et des *reads* courts disposant d'un plus faible taux d'erreur. Les *reads* ainsi synthétisés, appelés *reads* NaS (Nanopore *Synthetic-long*), peuvent atteindre une longueur de 60 000, et s'aligner intégralement et sans erreurs. Nous présentons ici une première méthode permettant de synthétiser de tels *reads*, et la méthode que nous avons développée.

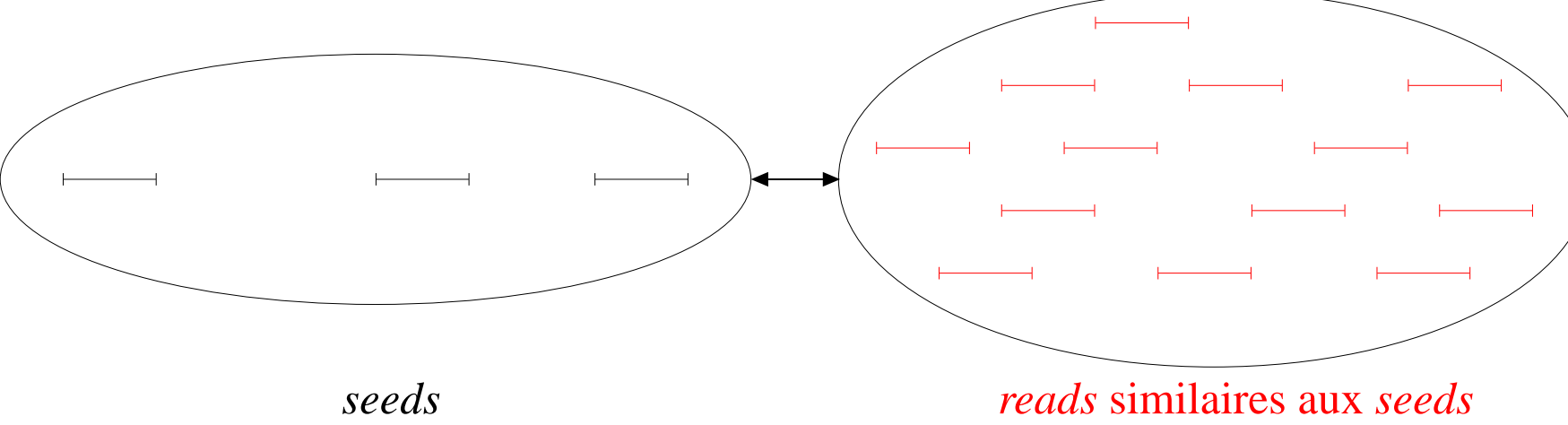
Première méthode

La première méthode de synthèse des NaS [2] aligne les *reads* courts entre eux, et sur les *reads* longs *templates*. Elle repose sur les étapes suivantes :

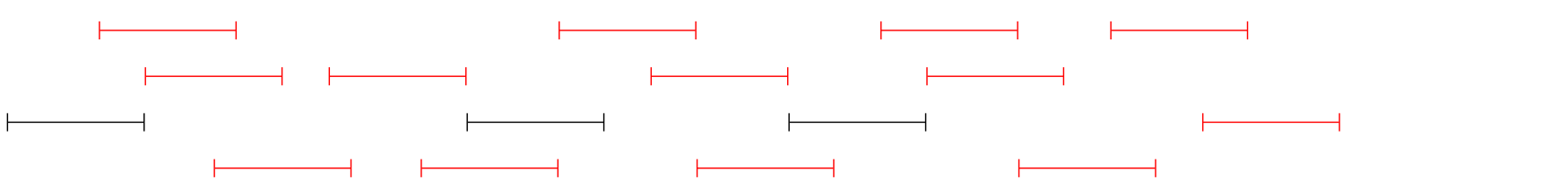
1. Alignement des *reads* courts sur le *template*, pour trouver les *seeds*



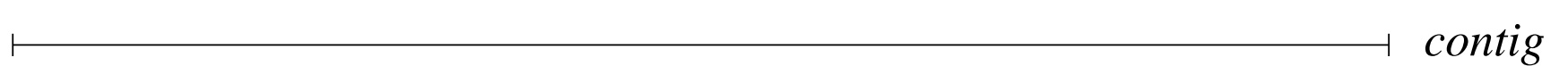
2. Recrutement de nouveaux *reads* similaires aux *seeds*, en alignant les *reads* courts entre eux. Un *read* court est similaire à un *seed* s'ils partagent au moins t k -mers ne se chevauchant pas.



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



4. Obtention d'un contig

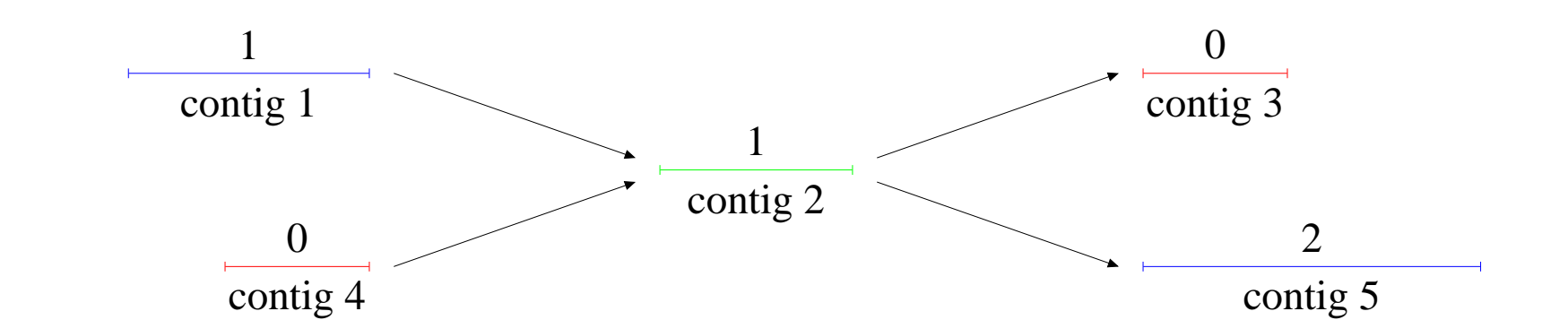


En général, un seul contig est produit, mais il est possible que de mauvais *reads* soient recrutés, et que des contigs erronés, ne devant pas être associés au *template*, soient produits. Pour résoudre ce problème et ne produire qu'un seul contig, et donc un NaS, en sortie, il suffit d'employer la démarche suivante :

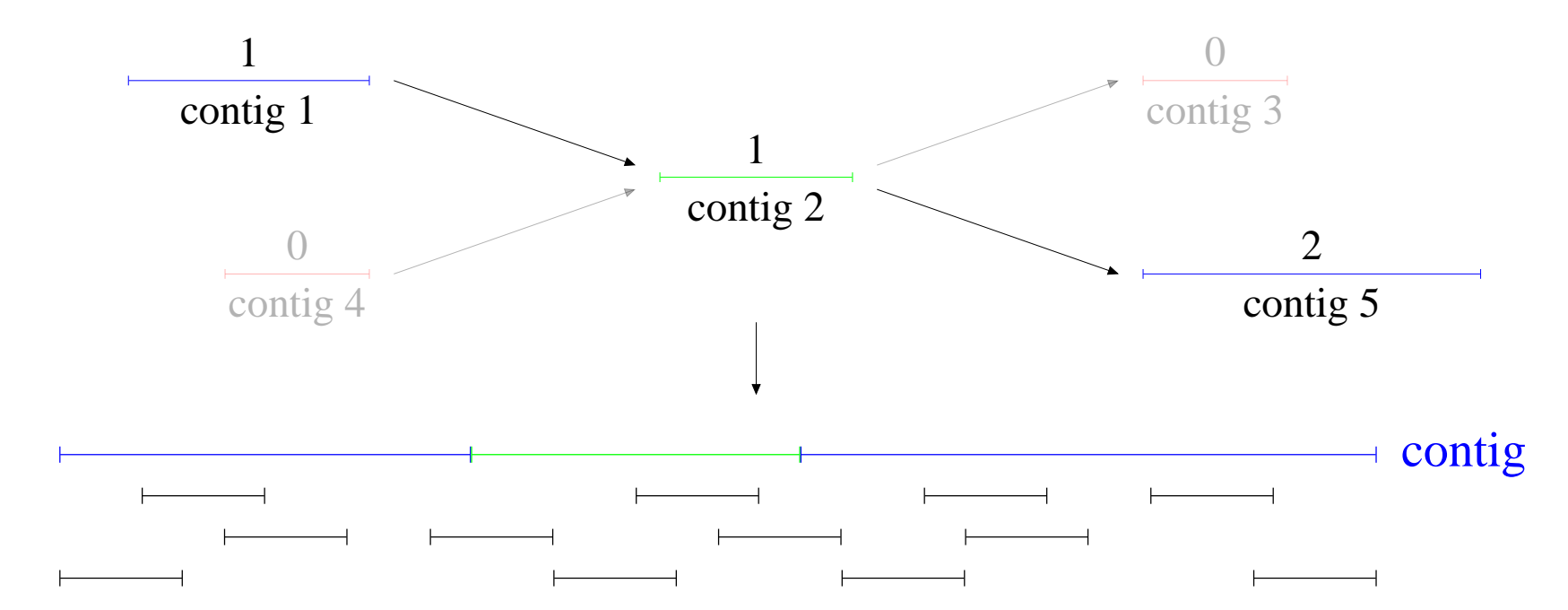
1. Obtention de plusieurs contigs



2. Construction du graphe des contigs, pondéré par le degré de couverture des contigs par les *seeds*



3. Sélection du chemin optimal, passant par les contigs ayant le plus haut degré de couverture par les *seeds*, puis vérification du contig produit par alignement des *reads* courts, et acceptation si la couverture est suffisante



Sur un ensemble de 66 492 *reads* longs MinION, et à l'aide de *reads* courts Illumina, cette méthode a permis de produire 11 275 *reads* NaS, d'une longueur maximale de 59 863. Seulement 17% des *reads* longs ont donc produit un NaS, ce qui est dû au fort taux d'erreurs des *reads* MinION. De plus, 97% des *reads* ainsi synthétisés ont été alignés sur le génome de référence sans aucune erreur, démontrant l'efficacité de cette méthode.

Le temps de traitement d'un *read* long, et donc la synthèse d'un *read* NaS, est de moins d'une minute en moyenne, soit un temps total d'environ 7 jours pour la synthèse des 11 275 *reads* NaS obtenus. La majorité de ce temps est utilisée par la méthode peu efficace de recrutement de *reads* similaires, nécessitant l'alignement des *reads* courts entre eux.

Notre méthode

Notre méthode repose sur le même principe que la méthode précédente, mais ne déduit des informations qu'à partir de l'alignement des *reads* courts sur les *reads* longs. Les *reads* courts sont donc alignés sur les *reads* longs *templates*, en se fixant un seuil $lmin$, pour récupérer les *reads* :

- Totalemment alignés, et servant de *seeds*
- Avec un préfixe de longueur $\geq lmin$ aligné
- Avec un suffixe de longueur $\geq lmin$ aligné

Une fois ces trois ensembles calculés, notre méthode se décompose en deux étapes :

1. Recrutement de *reads* partiellement alignés, similaires aux *seeds*, afin d'étendre ces derniers. Un *read* est considéré comme similaire à un *seed* si son préfixe (respectivement son suffixe) correctement aligné chevauche ce *seed* sur une longueur supérieure ou égale au seuil $lmin$.
2. Recrutement de *reads* partiellement alignés afin d'étendre à nouveau les contigs obtenus à l'étape précédente. Un *read* partiellement aligné est ici recruté si son préfixe (respectivement son suffixe) chevauche le contig sur une longueur inférieure ou égale à un seuil fixé $lmax$.



FIGURE 1 : Illustration des deux étapes de notre méthode. Les parties en gras des *reads* courts correspondent aux parties correctement alignées sur le *template*. Les parties hachurées du *template* correspondent aux zones fortement bruitées de celui-ci.

Nombre de <i>reads</i>	Longueur moyenne	Contigs / <i>read</i>	Longueur moyenne	Précision moyenne	Temps
492	1 873	2,68	665	88,65	1 min 26 s
1 713	1 549	2,89	742	87,94	15 min 35 s
1 616	8 910	3,03	2 139	87,91	1 h 57 min
2 871	11 583	2,41	4 349	88,83	11 h

TABLE 5 : Résultats produits par notre méthode sur différents ensembles de données, avec $lmin = 100$ et $lmax = 10$. La troisième colonne représente le nombre moyen de contigs obtenus par *read* long.

Notre méthode permet donc de produire rapidement (moins de 10 seconde en moyenne pour le traitement d'un *read* long) des contigs relativement longs et plus précis que leurs *templates* d'origine, et semble constituer un prétraitement efficace pour la production de *reads* NaS. Il serait nécessaire d'étudier les contigs produits, afin de voir s'ils se chevauchent et peuvent ainsi être fusionnés, afin de produire, pour chaque *read* long, un unique contig de longueur plus importante, et donc un éventuel *read* NaS.

Références

- [1] 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.
- [2] 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.