Méthodes de mapping de reads avec indexation des reads

Pierre Morisse[1], *Thierry Lecroq*[2], *Arnaud Lefebvre*[3]

[1] LITIS
Université de Rouen
76000 Rouen, France
e-mail : pierre.morisse1@etu.univ-rouen.fr

[2] LITIS
Université de Rouen
76000 Rouen, France
e-mail : thierry.lecroq@univ-rouen.fr

[3] LITIS
Université de Rouen
76000 Rouen, France
e-mail : arnaud.lefebvre@univ-rouen.fr

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 très courtes séquences appelées $reads$. Dans un papier relativement récent, Philippe et al. ont souligné l'importance de l'indexation de ces $reads$ afin de résoudre des problèmes de correction ou de $mapping$, et ont développé un index supportant les 7 requêtes suivantes :

— 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 ?

De nombreux outils utilisant une structure d'index sur les $reads$ existent, et permettent aussi bien de résoudre des problèmes de $mapping$, de correction, que de traiter les 7 requêtes présentées ci-dessus. Un bref résumé de l'état-de-l'art est donné ci-dessous.

Séquenceurs à très haut débit

De nombreuses technologies et plateformes ont été développées pour permettre le séquençage de $reads$. Elle se différencient principalement par la longueur des $reads$ qu'elles produisent, par leur débit, ainsi que par le taux et le type d'erreurs qu'elles introduisent le plus fréquemment. Le tableau ci-dessous dresse un bref récapitulatif des technologies disponibles.

Technologie	Technique de séquençage	Plateforme	Nombre de $reads$	Longueur	Précision (%)	Temps	Débit	Coût	Erreurs
Illumina	Synthèse basé sur polymères	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	Polyséquençage	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	Ligatures 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 70 000	plusieurs milliers 1 665	96 70	variable 48 heures	quelques dizaines 0,132	variable 1 000	Indels.

Correction de $reads$

Le tableau ci-dessous dresse un récapitulatif des différents outils de correction de $reads$ utilisant une structure d'index sur ces $reads$, des structures de données sur lesquelles ils reposent, et de leur efficacité.

Outil	Structure de données	Erreurs corrigées	Nombre de $reads$ (longueur)	Espace mémoire (en Mo)	Temps (en min)	$reads$ corrigés (en %)
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 échantillonnée	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)	1 437	23	76,65
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

$Mapping$ de $reads$

Le tableau ci-dessous dresse un récapitulatif des différentes outils de $mapping$ de $reads$ utilisant une structure d'index sur ces $reads$, des structures de données sur lesquelles ils reposent, et de leur efficacité.

Outil	Structure de données	Erreurs prises en compte	Nombre de $reads$ (longueur)	Espace mémoire (en Mo)	Temps (en min)	$reads$ mappés (en %)
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

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

Traitement des 7 requêtes

La tableau ci-dessous dresse un récapitulatif des différents outils de traitement des 7 requêtes utilisant une structure d'index sur les $reads$, des structures de données sur lesquelles ils reposent, et et leur efficacité. Les requêtes 5-7 sont exclues du comparatif, car non implémentées dans GkA et CGkA lors des tests réalisés dans l'article introduisant PgSA.

Outil	Structure de données	Nombre de $reads$ (longueur)	Espace mémoire (en Go)	Temps R1 (en ms)	Temps R2 (en ms)	Temps R3 (en ms)	Temps R4 (en ms)
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

Correction de $reads$ longs

Comme il peut-être remarqué dans le tableau présentant les différentes technologies et plateformes de séquençage, de nouvelles technologies, permettant de séquencer des $reads$ de plus en plus longs, se développent. Les $reads$ séquencés par ces nouvelles technologies présentent cependant un taux d'erreur bien plus important que les $reads$ plus courts. Ainsi, au vu de la faible efficacité des méthodes de correction existantes sur ces $reads$ longs, notre travail s'est principalement porté sur ce problème. Nous présentons donc ici une méthode déjà existante, et la méthode que nous avons développé.

NaS (Nanopore $Synthetic$-$long$ $reads$)

Du blabla sur les NaS, il faut partir de seeds et recruter des reads courts Illumina, approche hybride, tout ça tout ça.

Notre méthode

Du blabla sur notre méthode. On garde la même idée que NaS, mais on essaye d'améliorer le temps d'exécution (car recrutement très long) en utilisant une autre technique pour recruter les reads courts.