

Correction de données de séquençage de troisième génération

Pierre MORISSE

Directeur de thèse : Thierry LECROQ

Co-encadrant : Arnaud LEFEBVRE

Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes

21 mars 2019



- 1 Contexte
- 2 Correction de *reads* longs
- 3 Évaluation des méthodes existantes : ELECTOR
- 4 Correction hybride : HG-CoLoR
- 5 Auto-correction : CONSENT
- 6 Conclusion

Correction de *reads* longs

- Intérêt des *reads* longs face aux *reads* courts
- De nombreuses méthodes de correction existent pour les *reads* courts, mais ne peuvent être appliquées aux *reads* longs
- État de l'art, deux principales approches :
 - Correction hybride, *reads* longs + *reads* courts
 - Auto-correction, *reads* longs uniquement

Correction hybride

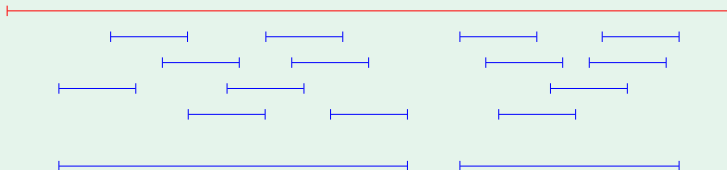
1. L'alignement des *reads* courts sur les *reads* longs

- Première approche de correction hybride décrite dans la littérature
- Utilisation des *reads* courts alignés sur chaque *read* long pour définir un alignement multiple
- Utilisation de l'alignement multiple pour calculer une séquence consensus

Correction hybride

1. L'alignement des *reads* courts sur les *reads* longs

Exemple



Correction hybride

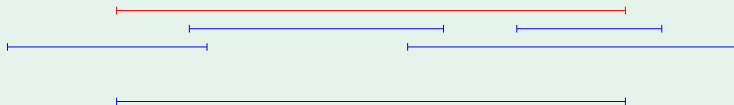
2. L'alignement de contigs de *reads* courts sur les *reads* longs

- Les contigs sont bien plus longs que les *reads* courts
- Plus facile de couvrir les régions très bruitées des *reads* longs lors de l'alignement
- Définition d'un alignement multiple et calcul d'une séquence consensus

Correction hybride

2. L'alignement de contigs de *reads* courts sur les *reads* longs

Exemple



Correction hybride

3. L'utilisation de graphes de de Bruijn

- Un graphe de de Bruijn est un graphe orienté
- Noeuds : k -mers (facteurs de taille k) des *reads*
- Arcs : Représentent les chevauchements préfixe / suffixe de taille $k - 1$ entre les k -mers

Correction hybride

Exemple

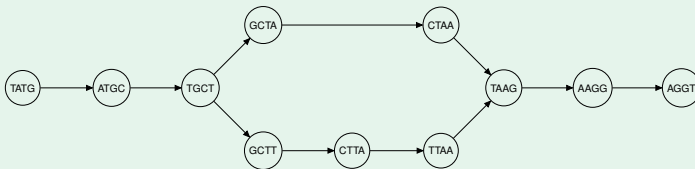
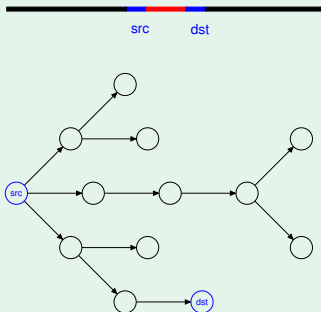


FIGURE – Graphe de de Bruijn d'ordre 4 pour l'ensemble de séquences $S = \{TATGCT, TGCTAA, GCTTAA, TAAGGT\}$.

Correction hybride

3. L'utilisation de graphes de de Bruijn

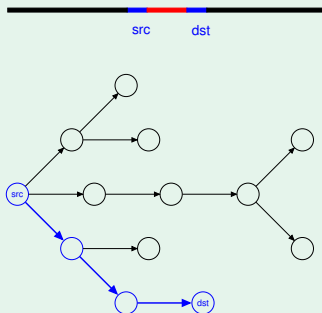
Exemple



Correction hybride

3. L'utilisation de graphes de de Bruijn

Exemple



Méthodes disponibles

Méthode	Approche	Année
PBcR	Correction hybride	2012
LSC	Correction hybride	2012
ECTools	Correction hybride	2014
LoRDEC	Correction hybride	2014
Proovread	Correction hybride	2014
Nanocorr	Correction hybride	2015
NaS	Correction hybride	2015
CoLoRMap	Correction hybride	2016
Jabba	Correction hybride	2016
LSCplus	Correction hybride	2016
HALC	Correction hybride	2017
HECIL	Correction hybride	2017
Hercules	Correction hybride	2017
FMLRC	Correction hybride	2018
HG-CoLoR	Correction hybride	2018
MiRCA	Correction hybride	2018

Méthode	Approche	Année
PBcR-BLASR	Auto-correction	2013
PBDAGCon	Auto-correction	2013
Sprai	Auto-correction	2014
PBcR-MHAP	Auto-correction	2015
FalconSense	Auto-correction	2016
Sparc	Auto-correction	2016
Canu	Auto-correction	2017
Daccord	Auto-correction	2017
LoRMA	Auto-correction	2017
MECAT	Auto-correction	2017
FLAS	Auto-correction	2018
CONSENT	Auto-correction	2019

Contexte

- Principe : Comparer trois versions différentes d'un même *read* :
 - 1 La version de référence (sans erreurs, provenant du génome de référence)
 - 2 La version originale (contenant des erreurs)
 - 3 La version corrigée (après application d'un outil de correction)
- Dédire des métriques décrivant la correction à partir des observations

LRCstats

reference R ATTGCTGAT ...

uncorrected U ATGTCTT ...

corrected C ATTGCAGAT ...

ATTG-CTGAT ...

AT-GTCT--T ...

ATTGCTGAT ...

ATTGCAGAT ...

align
to reference

Contribution : ELECTOR

- Facile d'utilisation
- Rapide
- Passage à l'échelle
- Calcul de métriques additionnelles

Métriques

	ELECTOR	LRCstats
Taux d'erreur avant/après correction	✓	✓
Rappel	✓	✗
Précision	✓	✗
Délétions	✓	✓
Insertions	✓	✓
Substitutions	✓	✓
<i>Reads</i> fragmentés	✓	✗
Taille manquante	✓	✗
%GC avant/après correction	✓	✗
Statistiques d'assemblage	✓	✗

Principe

reference R

ATTGCTGAT . . .

uncorrected U

ATGTCTT . . .

corrected C

ATTGCAGAT . . .

ATTG-CTGAT . . .

AT-GTCT--T . . .

ATTG-CAGAT . . .

multiple
sequence
alignment

Alignement multiple

- Avantages : Précis, permet de calculer des métriques additionnelles
- Inconvénient : Coûteux (NP complet)
- Passage à l'échelle difficile
- Solution : Stratégie de segmentation

Stratégie de segmentation

Calcul des k -mers communs entre les trois séquences



UNIVERSITÉ
DE ROUEN
NORMANDIE

Stratégie de segmentation

Recherche de la plus longue chaîne de k -mers communs



Stratégie de segmentation

Recherche de la plus longue chaîne de k -mers communs



Stratégie de segmentation

Recherche de la plus longue chaîne de k -mers communs



Stratégie de segmentation

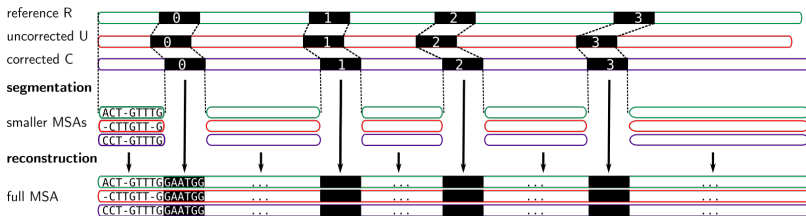
Recherche de la plus longue chaîne de k -mers communs



Alignement multiple segmenté

set of **common, colinear** seeds (k-mers)

0 1 2 3



Validation de la stratégie de segmentation

- Jeux de données de la bactérie *E. coli*
- Expériences : *Reads* de longueurs 1 000, 10 000, 100 000, taux d'erreurs de 10%

Expérience	Rappel (%)	Précision (%)	Bases correctes (%)	Temps
"1k" sans segmentation	99,712	98,996	98.980	2h 05 min
"1k" avec segmentation	99,769	98,992	98.979	28 min
"10k" sans segmentation	99.921	99.781	99.794	20 h 50 min
"10k" avec segmentation	99.921	99.795	99.793	29 min
"100k" sans segmentation	99.913	99.925	99.956	8 jours 18 h 38 min
"100k" avec segmentation	99.924	99.903	99.902	1 h 11 min

Comparaison : métriques

Sur le jeu de données *S. cerevisiae*

Métriques	Reads non corrigés		Reads corrigés avec HALC	
	ELECTOR	LRCstats	ELECTOR	LRCstats
Débit	238,309,333	237,655,341	212,266,193	214,152,119
Taux d'erreurs	0.1403	0.1751	0.0042	0.0023
Insertions	28,772,841	32,589,970	100,874	215,507
Délétions	5,235,890	8,991,984	1,035,978	120,743
Substitutions	4,058,953	1,633,123	198,853	221,646
Rappel	-	-	0.9997	✖
Précision	-	-	0.9959	✖
Reads fragmentés	-	-	12,043	✖
Taille manquante	-	-	577.5	✖
Reads étendus	-	-	71	✖
Taille extension	-	-	53.2	✖
%GC	38.2	✖	38.2	✖
Métriques	Reads non corrigés		Reads corrigés avec Canu	
	ELECTOR	LRCstats	ELECTOR	LRCstats
Débit	244,560,743	244,633,066	229,555,492	229,825,812
Taux d'erreurs	0.1425	0.1781	0.0506	0.0694
Insertions	30,090,583	34,105,075	12,252,413	12,942,568
Délétions	5,483,119	9,489,618	2,574,320	3,134,365
Substitutions	4,375,017	1,748,302	2,197,172	1,591,650
Rappel	-	-	0.9515	✖
Précision	-	-	0.9495	✖
Reads fragmentés	-	-	2,216	✖
Taille manquante	-	-	35.1	✖
Reads étendus	-	-	178	✖
Taille extension	-	-	30.7	✖
%GC	38.2	✖	38.7	✖

Contexte

- Les premières expériences de séquençage de *reads* longs produisaient des *reads* très bruités ($> 30\%$) d'erreurs
- Les méthodes d'auto-correction ne sont pas efficaces sur de tels taux d'erreurs
- Les méthodes hybrides existantes peuvent être appliquées mais...
 - Induisent de mauvais résultats d'assemblage
 - Sont très coûteuses

Contribution : HG-CoLoR

Nouvelle méthode de correction hybride qui :

- Combine différentes approches de l'état de l'art
 - L'alignement de *reads* courts sur les *reads* longs
 - L'utilisation d'un graphe de de Bruijn
- Adopte une approche *seed and extend*
 - 1 Les *reads* courts sont alignés sur les *reads* $\Rightarrow$ graines
 - 2 Les graines sont reliées entre elles à l'aide d'un graphe de de Bruijn

Workflow

Première étape

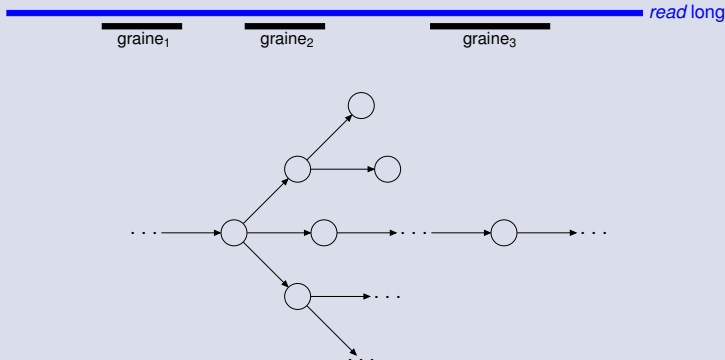
Alignement des *reads* courts sur les *reads* longs



Workflow

Deuxième étape

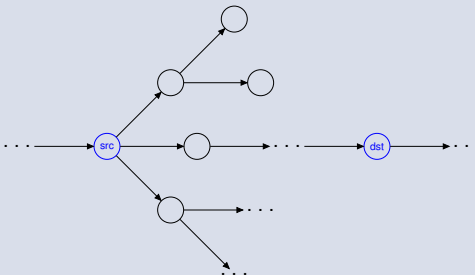
Construction du graphe à partir des k -mers des *reads* courts



Workflow

Deuxième étape

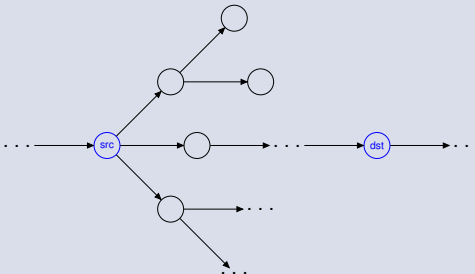
Utilisation des graines comme points d'ancrage sur le graphe



Workflow

Troisième étape

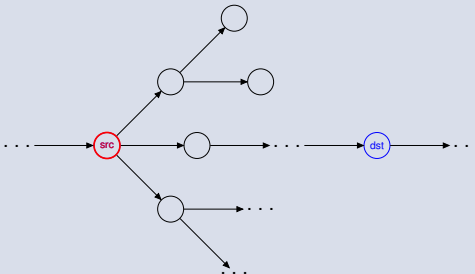
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

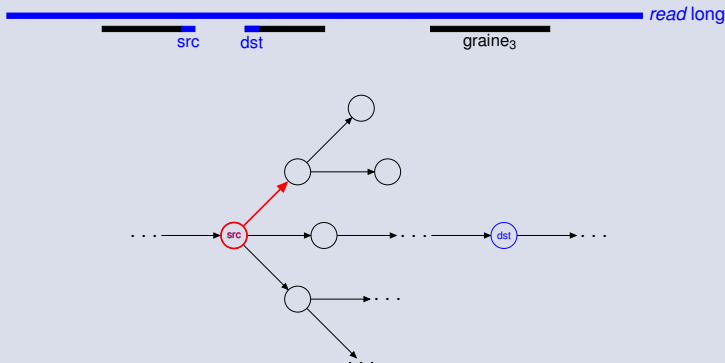
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

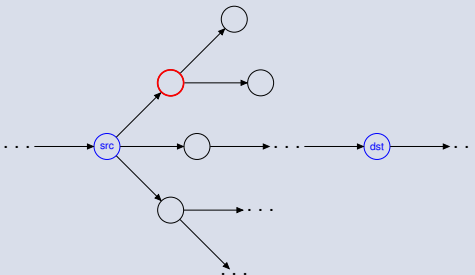
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

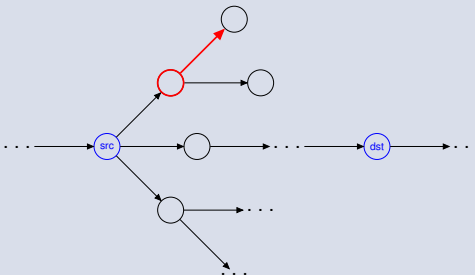
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

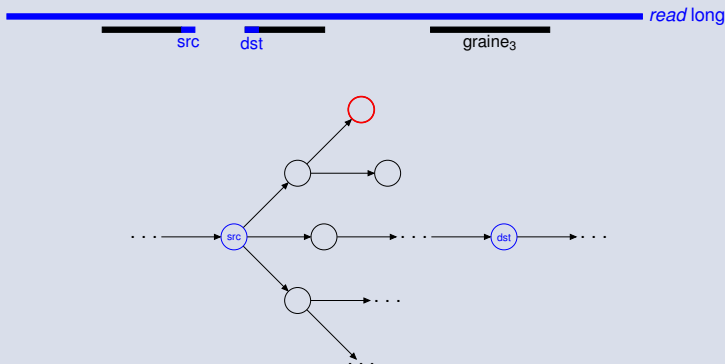
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

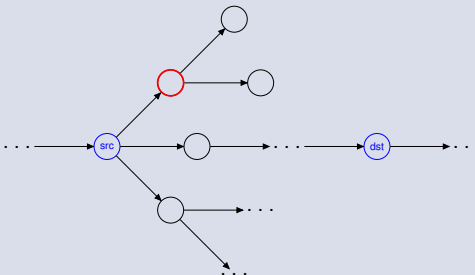
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

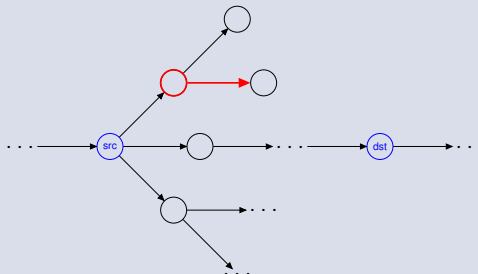
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

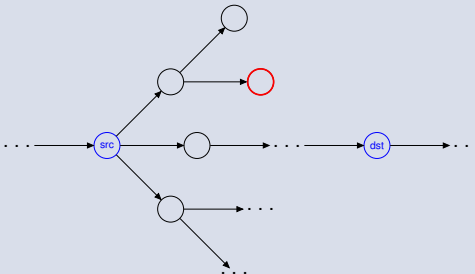
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

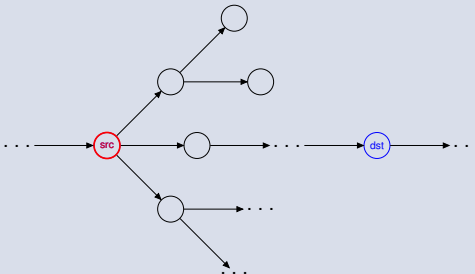
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

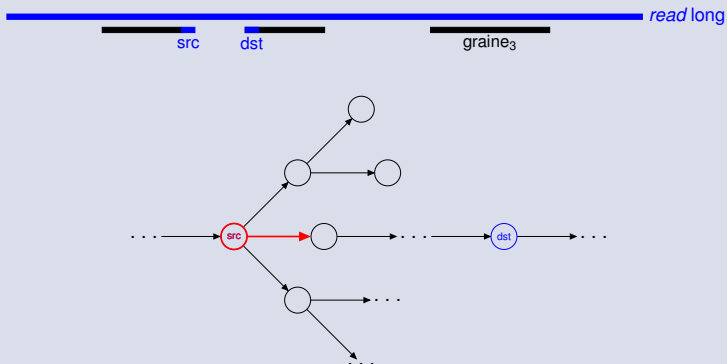
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

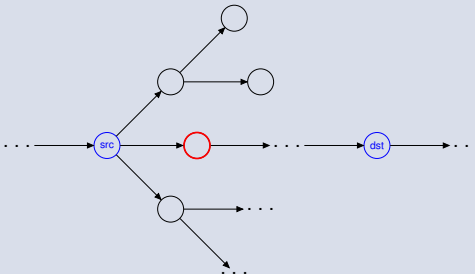
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

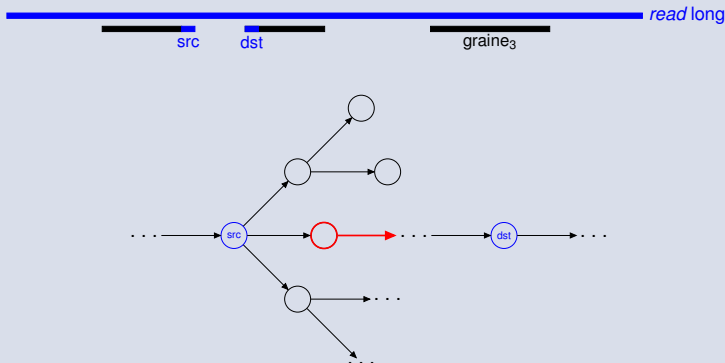
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

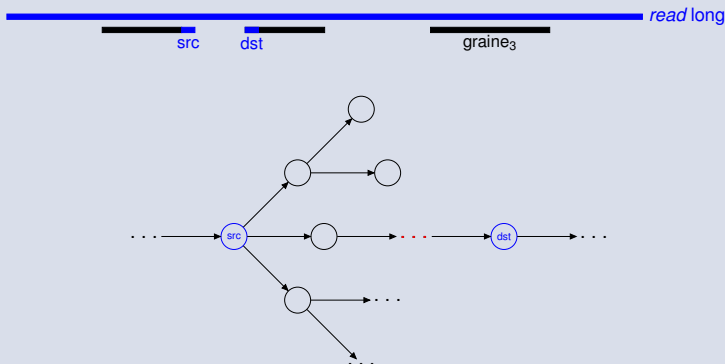
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

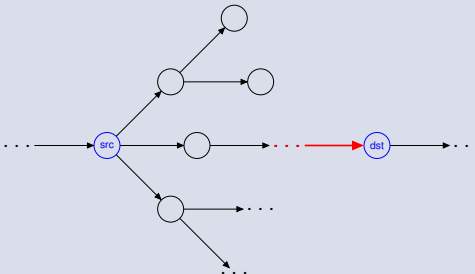
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

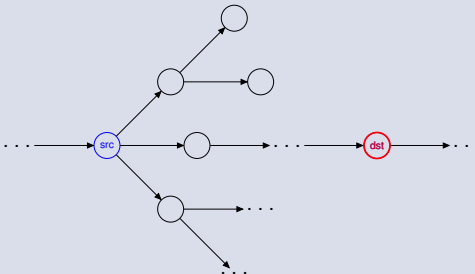
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

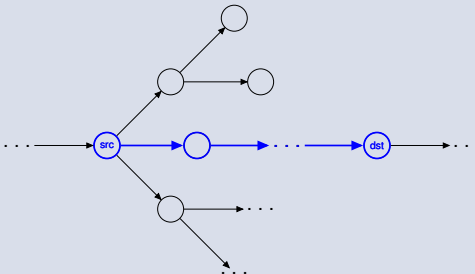
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

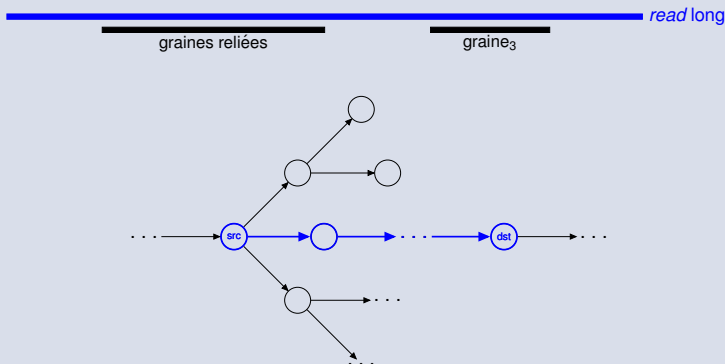
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

Parcours du graphe afin de relier les graines



Workflow

Troisième étape

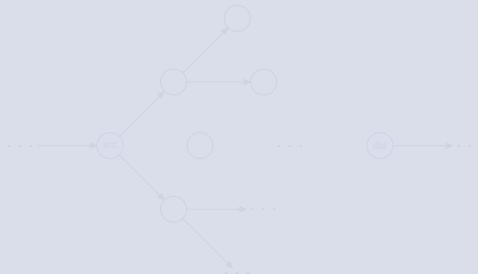
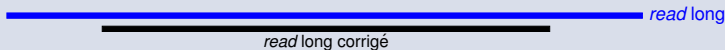
Parcours du graphe afin de relier les graines



Workflow

Troisième étape

Parcours du graphe afin de relier les graines



Comparaison aux autres outils

- Outils comparés :

- CoLoRMap
- LoRDEC
- NaS
- Canu
- Daccord

- Jeu de données : *S. cerevisiae*, 44% d'erreurs

- Comparaison de la qualité des *reads* et de la qualité de l'assemblage

Comparaison de la qualité des *reads*

Méthode	-	HG-CoLoR	CoLoRMap	LoRDEC	NaS	Canu	Daccord
Nombre de <i>reads</i>	205,923	72,228	39,088	59,075	71,793	-	-
Longueur moyenne	5,698	6,724	2,294	1,126	5,938	-	-
Nombre de bases (M)	1,173	512	165	221	426	-	-
Identité moyenne (%)	55.49	99.10	99.45	98.45	99.59	-	-
Temps	N/A	8h36min	10h44min	1h09min	> 16 days	-	-

Contexte

- La longueur des *reads* continue également à augmenter
- La technologie Oxford Nanopore parvient à séquences des *reads* d'un million de bases (*ultra-long reads*)
- Corriger de tels *reads* est un challenge
- Les méthodes de l'état de l'art ne passent pas à l'échelle

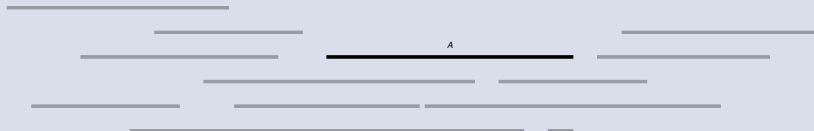
Pré-traitement

Aligner les *reads* longs



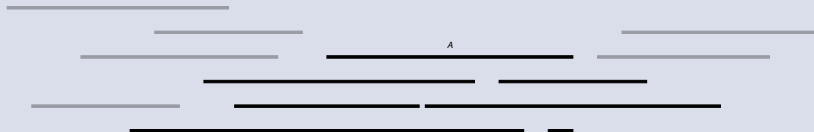
Première étape : définir une pile d'alignement

Sélectionner un *read* long à corriger



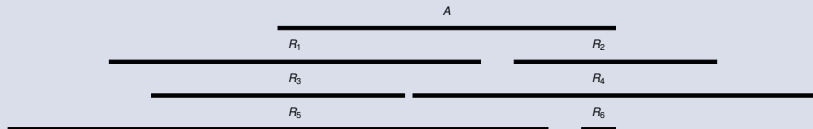
Première étape : définir une pile d'alignement

Retrouver les autres *reads* alignés



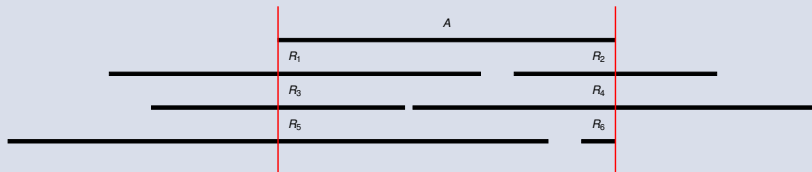
Première étape : définir une pile d'alignement

Obtenir la pile d'alignement



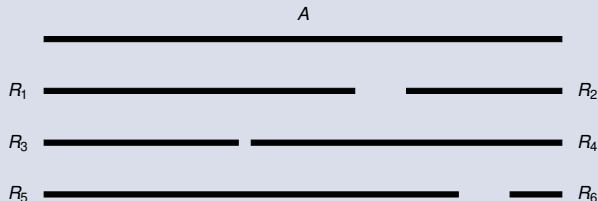
Première étape : définir une pile d'alignement

Ignorer les extrémités



Première étape : définir une pile d'alignement

Ignorer les extrémités

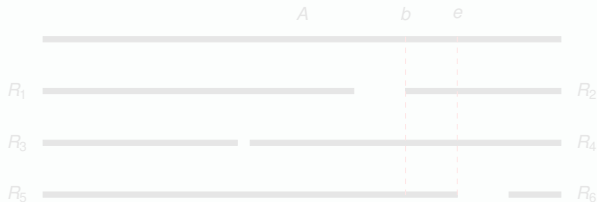


Deuxième étape : diviser la pile en fenêtres

Definition

Une fenêtre $w = (b, e)$ est un "facteur" d'une pile d'alignement

Exemple

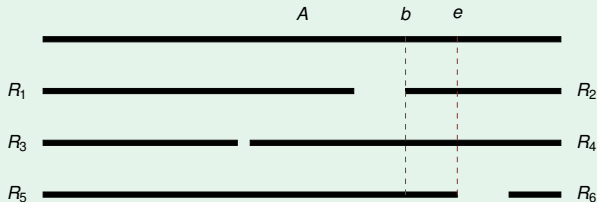


Deuxième étape : diviser la pile en fenêtres

Definition

Une fenêtre $w = (b, e)$ est un "facteur" d'une pile d'alignement

Exemple



Deuxième étape : diviser la pile en fenêtres

Pour la correction, divisons de la pile en fenêtres de taille fixée :

Exemple

Sur la pile précédente :

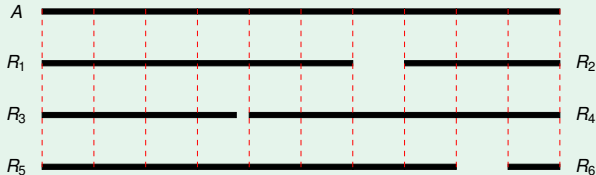


Deuxième étape : diviser la pile en fenêtres

Pour la correction, divisons de la pile en fenêtres de taille fixée :

Exemple

Sur la pile précédente :



Troisième étape : corriger les fenêtres

- Calculer un alignement multiple avec les séquences de chaque fenêtre
- Calculer une séquence consensus à partir de l'alignement multiple
- En pratique, fenêtres de quelques centaines de bases de long
- L'alignement multiple est coûteux
- Généralisation de la stratégie de segmentation d'ELECTOR
- Calcul d'alignements multiples et de consensus pour de plus petites séquences

Troisième étape : corriger les fenêtres

Stratégie de segmentation

1. Calcul des k -mers communs entre les séquences des fenêtres



Troisième étape : corriger les fenêtres

Stratégie de segmentation

2. Recherche de la plus longue chaîne de k -mers tel que $\forall A_i, A_{i+1}$:

- 1 A_i est suivi par A_{i+1} dans au moins N séquences
- 2 A_{i+1} n'est jamais suivi par A_i

Troisième étape : corriger les fenêtres

Stratégie de segmentation

2. Recherche de la plus longue chaîne de k -mers tel que $\forall A_i, A_{i+1}$:

- 1 A_i est suivi par A_{i+1} dans au moins N séquences
- 2 A_{i+1} n'est jamais suivi par A_i



Troisième étape : corriger les fenêtres

Stratégie de segmentation

2. Recherche de la plus longue chaîne de k -mers tel que $\forall A_i, A_{i+1}$:

- 1 A_i est suivi par A_{i+1} dans au moins N séquences
- 2 A_{i+1} n'est jamais suivi par A_i



Troisième étape : corriger les fenêtres

Stratégie de segmentation

2. Recherche de la plus longue chaîne de k -mers tel que $\forall A_i, A_{i+1}$:

- 1 A_i est suivi par A_{i+1} dans au moins N séquences
- 2 A_{i+1} n'est jamais suivi par A_i



Troisième étape : corriger les fenêtres

Stratégie de segmentation

2. Recherche de la plus longue chaîne de k -mers tel que $\forall A_i, A_{i+1}$:

- 1 A_i est suivi par A_{i+1} dans au moins N séquences
- 2 A_{i+1} n'est jamais suivi par A_i



Troisième étape : corriger les fenêtres

Stratégie de segmentation

2. Recherche de la plus longue chaîne de k -mers tel que $\forall A_i, A_{i+1}$:

- ① A_i est suivi par A_{i+1} dans au moins N séquences
- ② A_{i+1} n'est jamais suivi par A_i



Troisième étape : corriger les fenêtres

Stratégie de segmentation

2. Recherche de la plus longue chaîne de k -mers tel que $\forall A_i, A_{i+1}$:

- ① A_i est suivi par A_{i+1} dans au moins N séquences
- ② A_{i+1} n'est jamais suivi par A_i



Troisième étape : corriger les fenêtres

Stratégie de segmentation

2. Recherche de la plus longue chaîne de k -mers tel que $\forall A_i, A_{i+1}$:

- ① A_i est suivi par A_{i+1} dans au moins N séquences
- ② A_{i+1} n'est jamais suivi par A_i



Troisième étape : corriger les fenêtres

Stratégie de segmentation

2. Recherche de la plus longue chaîne de k -mers tel que $\forall A_i, A_{i+1}$:

- ① A_i est suivi par A_{i+1} dans au moins N séquences
- ② A_{i+1} n'est jamais suivi par A_i



Troisième étape : corriger les fenêtres

Stratégie de segmentation

2. Recherche de la plus longue chaîne de k -mers tel que $\forall A_i, A_{i+1}$:

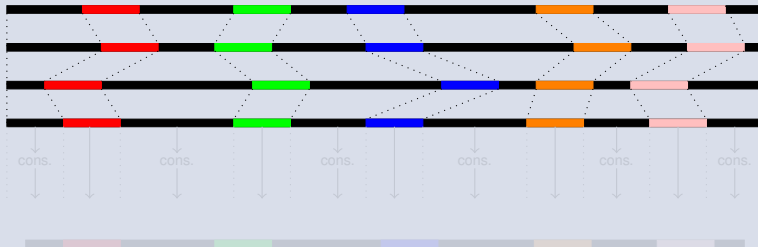
- ① A_i est suivi par A_{i+1} dans au moins N séquences
- ② A_{i+1} n'est jamais suivi par A_i



Troisième étape : corriger les fenêtres

Stratégie de segmentation

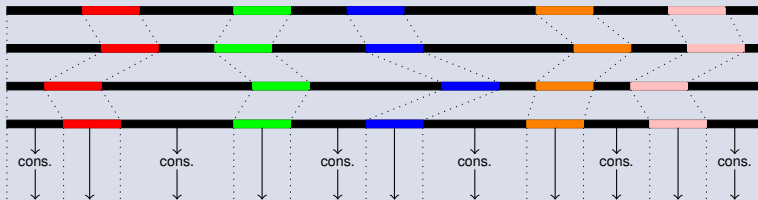
3. Calcul des alignements multiples et des consensus pour les séquences bordées par des ancres



Troisième étape : corriger les fenêtres

Stratégie de segmentation

3. Calcul des alignements multiples et des consensus pour les séquences bordées par des ancres



Validation de la stratégie de segmentation

Expérience sur un jeu de données à 12% d'erreurs de la bactérie *E. coli*

	Sans segmentation	Avec segmentation
Nombre de bases	214,667,382	215,693,736
Taux d'erreurs (%)	0.0757	0.0722
Temps	5h31min	7min
Mémoire (MB)	750	675

Comparaison aux autres outils

- Outils comparés :
 - Canu
 - Daccord
 - FLAS
 - LoRMA
 - MECAT
- Jeu de données humain, 18% d'erreurs, contenant des *ultra-long reads*

Comparaison de la qualité des reads

Méthode	Nombre de reads	Nombre de bases (M)	Longueur moyenne	Identité moyenne (%)	Temps	Mémoire (Mo)
-	1,075,867	7,256	10,568	82.40	N/A	N/A
Canu ¹	-	-	-	-	-	-
daccord ¹	-	-	-	-	-	-
FLAS ¹	670,708	5,695	10,198	91.00	4 h 57 min	14,957
MECAT ¹	667,532	5,479	10,343	91.69	1 h 53 min	11,075
CONSENT	869,462	6,349	10,839	93.00	8 h 30 min	10,021

¹ en filtrant les *ultra-long reads*

Conclusion

- CONSENT est la seule méthode permettant de passer à l'échelle sur des *ultra-long reads*
- Généralise la stratégie d'alignement multiple segmenté d'ELECTOR
- Meilleurs résultats que les méthodes de l'état de l'art
- Temps d'exécution plus important

Conclusion

- Étude approfondie des problématiques liées à la correction de *reads* longs
- Développement d'outils variés
 - Évaluation de la correction
 - Correction hybride
 - Auto-correction
- Pour répondre à des problèmes rencontrés par les outils de l'état de l'art

The end

Merci pour votre attention !