RedOak: a reference-free and alignment-free structure for indexing a collection of similar genomes

Recueil de comptes

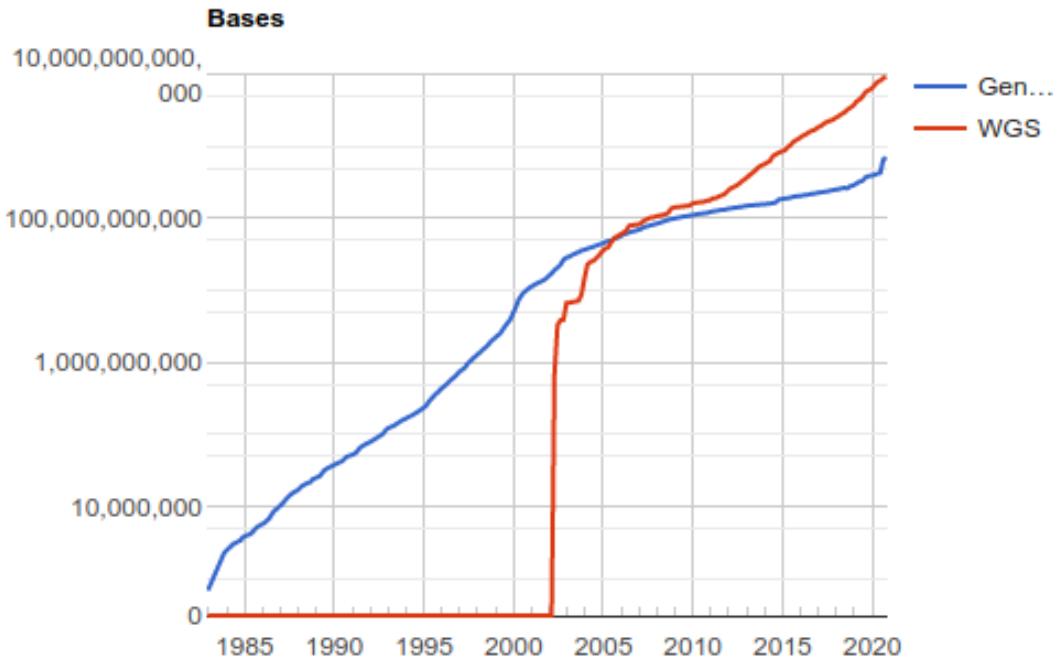
CLÉMENT AGRET, ANNIE CHATEAU, GAËTAN DROC, GAUTIER SARAH, ALBAN MANCHERON, MANUEL RUIZ

SeqBIM, 24 novembre 2020

Sommaire

Chapitre I : Contexte

Le Déluge

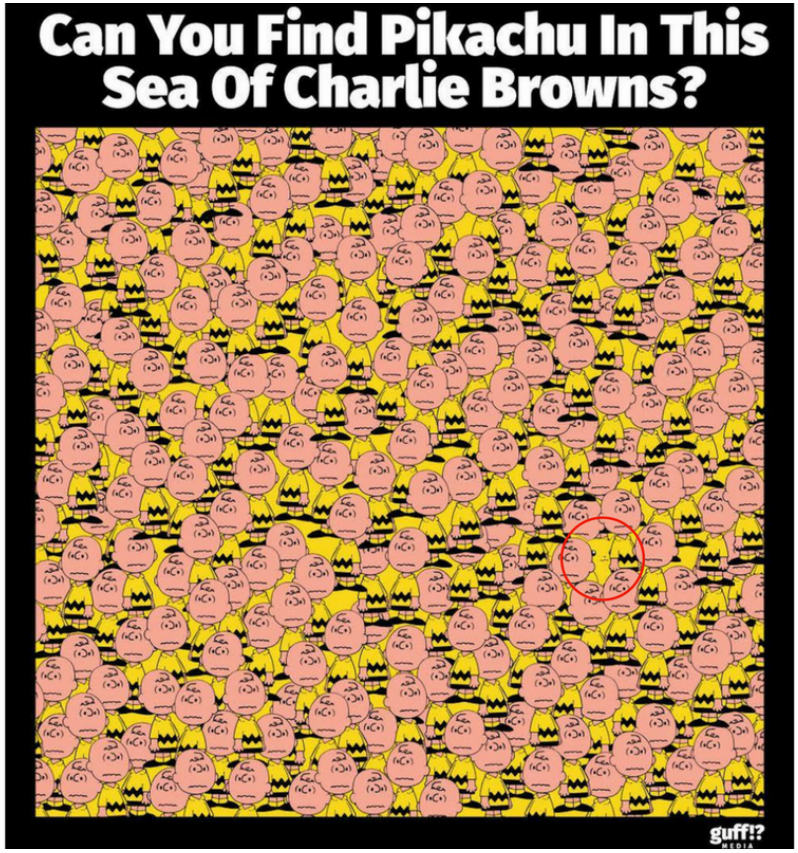


From 1982 to the present, the number of bases in GenBank has doubled approximately every 18 months.

https://www.ncbi.nlm.nih.gov/genbank/statistics/

Chapitre I : Contexte

Le Déluge



Problématique

Comment **structurer** un ensemble de données génomiques pour pouvoir extraire *rapidement* et *facilement* des informations biologiques?

Chapitre I : Contexte

Comptes des mille et un génomes

GenomeHarvest
diversity, organization and dynamics

Agropolis Fondation

2016-2020

Responsables

Angélique D'HONT et Manuel RUIZ

Pluridisciplinaire

Agronomie, mathématique, écologie, évolution et informatique...

(source: www.agropolis-fondation.fr/GenomeHarvest)

Chapitre I : Contexte

Comptes des mille et un génomes

GenomeHarvest
diversity, organization and dynamics

UMR AGAP - Cirad

Équipe Mab - Lirmm

- AGAP : Amélioration Génétique et Adaptation des Plantes méditerranéennes et tropicales
- MAB : Méthodes et Algorithmes pour la Bioinformatique

Chapitre I : Contexte

Comptes des mille et un génomes

GenomeHarvest
diversity, organization and dynamics

Objectif scientifique

Comprendre l'organisation et la dynamique du génome pour l'amélioration des cultures d'intérêt agronomique.

Espèces d'intérêt

Banane, Agrumes, Café, **Riz** et Canne à Sucre

(source: www.agropolis-fondation.fr/GenomeHarvest)

Céréale de la famille des Poacées

Sous espèces principales

- *Indica*
- *Japonica*

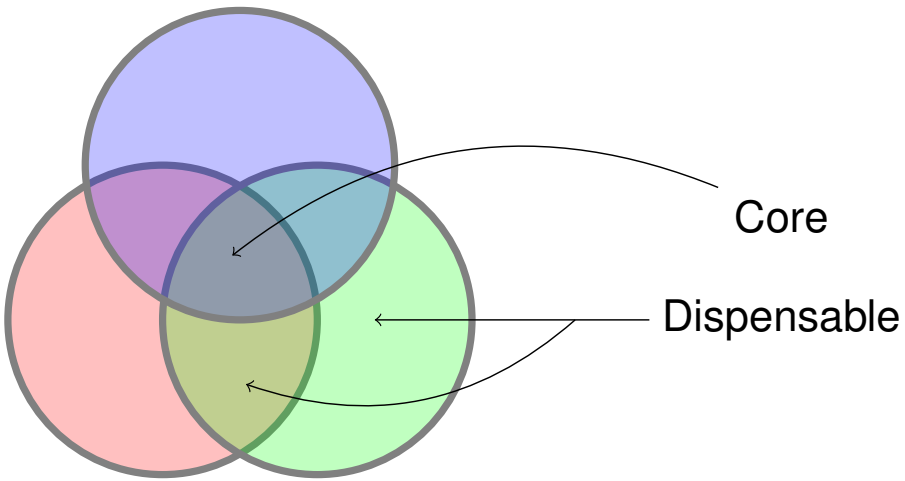
400 et 500 millions de paires de bases d'ADN

38,000 gènes

12 chromosomes

- *The 3,000 rice genomes project* *[Li et al., 2014]*

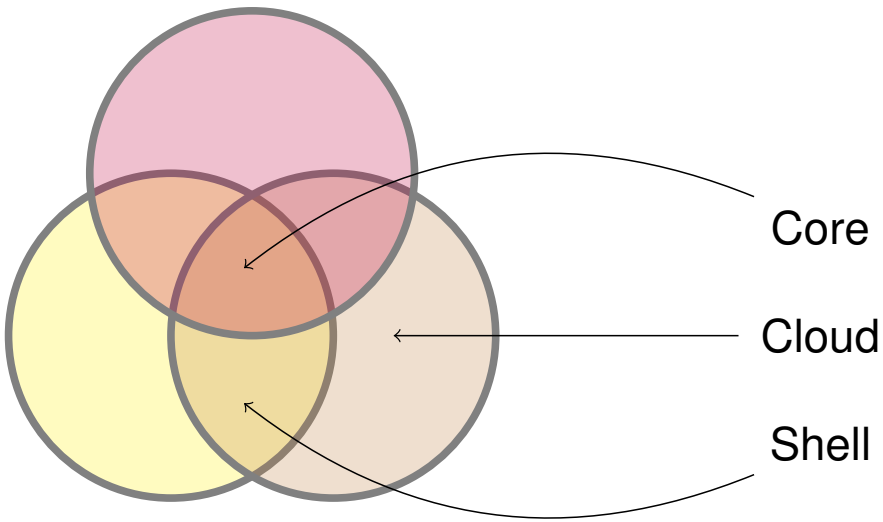
→ 3,000 génomes de riz rendus publics



Définition (Pan-génome)

Ensemble des gènes présents dans toutes les souches d'une même espèce (*core*-genome) et des gènes absents d'une ou de plusieurs souches et des gènes uniques à chaque souche (*dispensable*-genome).

[Tettelin et al., 2005]



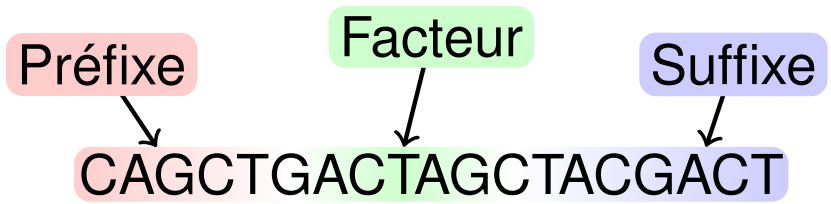
Définition (*core*, *shell* et *cloud*)

Le *core* (noyau : gènes présents dans tous les génomes du pan-génome), le *shell* (la coque : gènes fréquents) et le *cloud* (nuage : gènes rares).

[Koonin and Wolf, 2008, Snipen and Ussery, 2010]

Vocabulaire

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

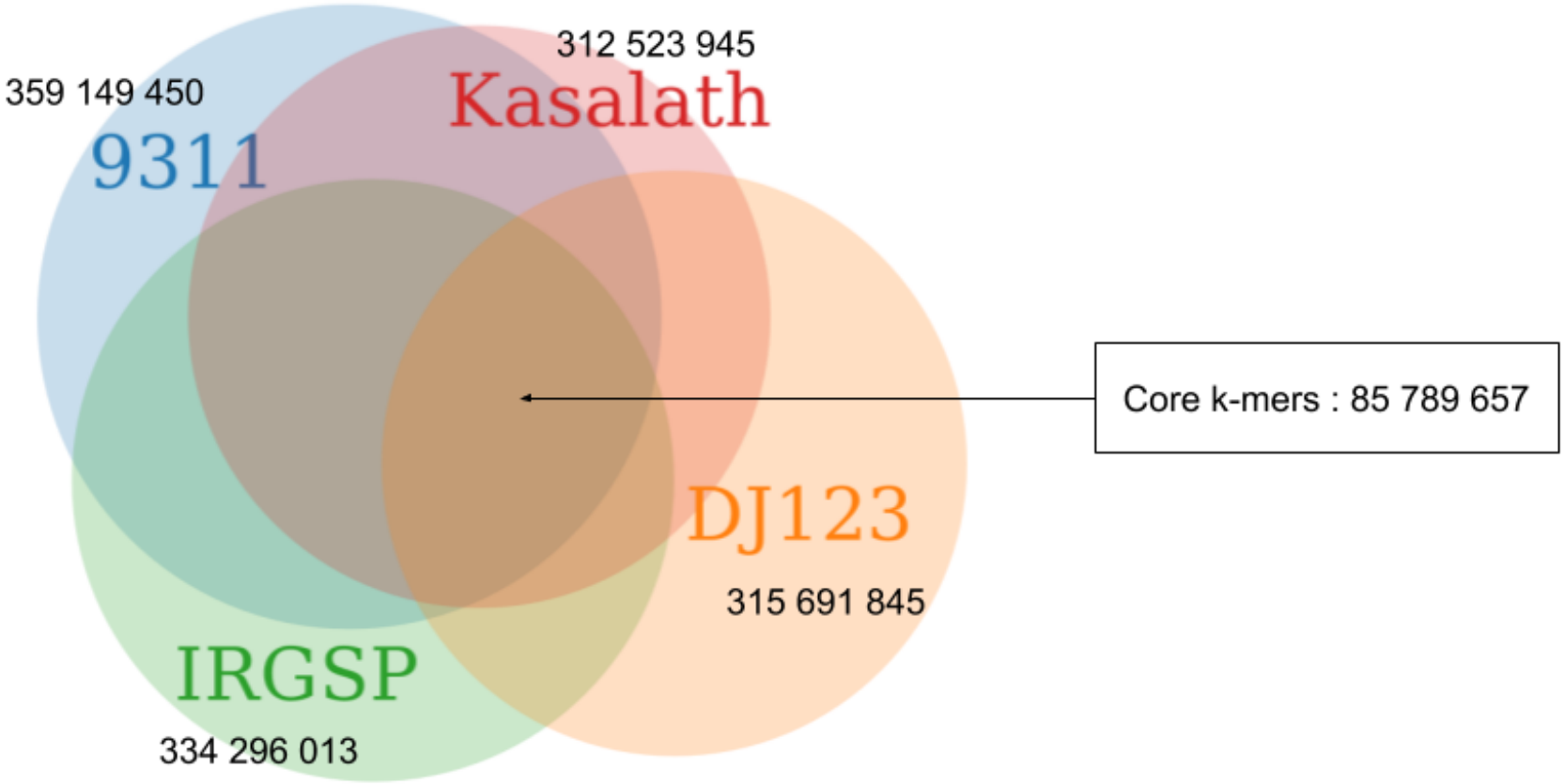


k-mer

Un fragment de k nucléotides consécutifs d'un mot (le cas échéant une séquence provenant d'un génome).
$\rightarrow$ Un k-mer est un facteur de taille k d'un mot.

Chapitre II : structuration des données

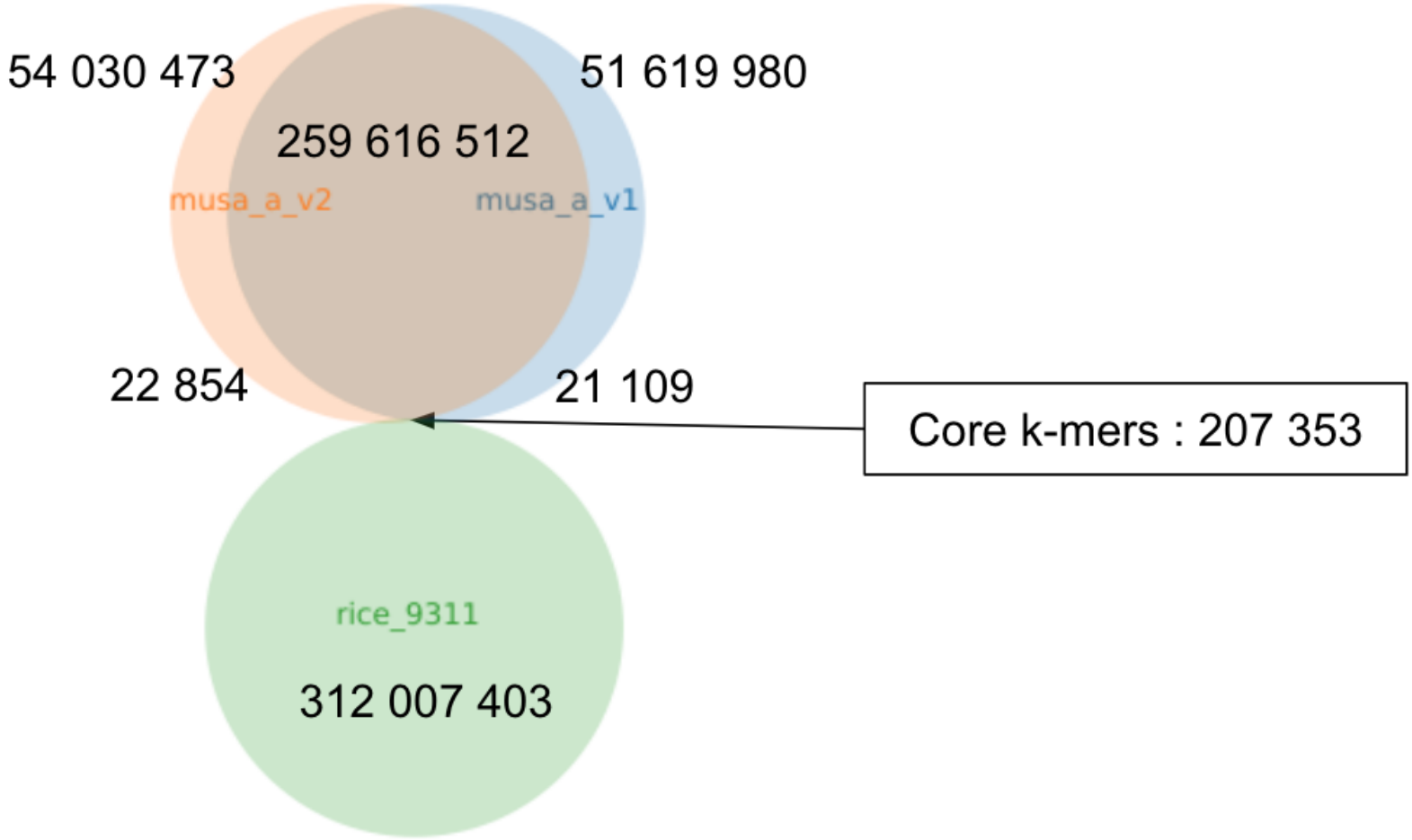
Peau de Banane



Étude des 21-mers

Chapitre II : structuration des données

Peau de Banane



Étude des 21-mers entre deux espèces (banane et riz)

$\rightarrow$ Approche *k*-mers validée

Chapitre II : structuration des données

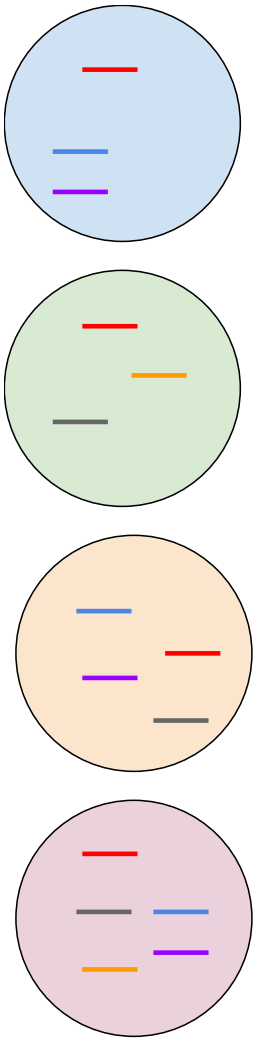
Nipponbare et les sept génomes

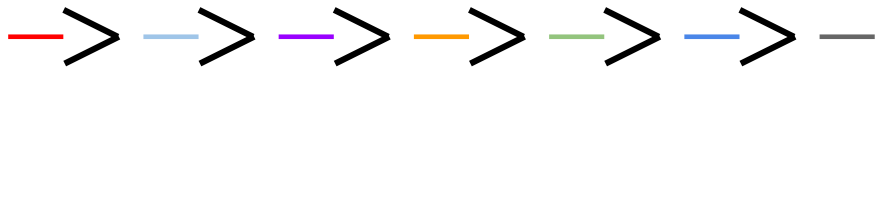
Génome			Séquences		Taille du
N°	cultivar	*ssp.*	nombre	taille totale	fichier
1.	93-11	*indica*	12 chromosomes + 12,718 scaffolds	423,026,874pb	412MiB
2.	BABO01	*japonica*	12 chromosomes + 654,543 contigs	306,177,972pb	390MiB
3.	DJ 123	*aus*	2,819 scaffolds	345,981,746pb	335MiB
4.	IR 64	*indica*	2,919 scaffolds	345,209,449pb	334MiB
5.	Nipponbare (IRGSP-1.0)	*japonica*	12 chromosomes	373,245,519pb	362MiB
6.	Kasalath	*aus*	12 chromosomes + 1 scaffold (14,822 contigs concaténés et séparés par 1,000 N)	401,141,708pb	421MiB
7.	LNNJ01.1	*indica*	237 scaffolds	346,854,256pb	336MiB
8.	LNNK01	*japonica*	181 scaffolds	359,918,891pb	349MiB

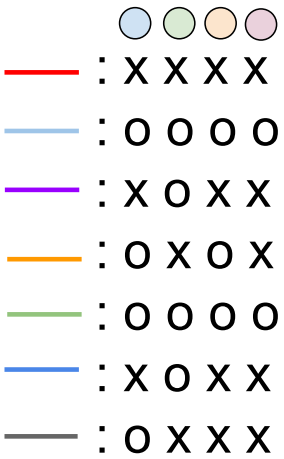
Chapitre II : structuration des données

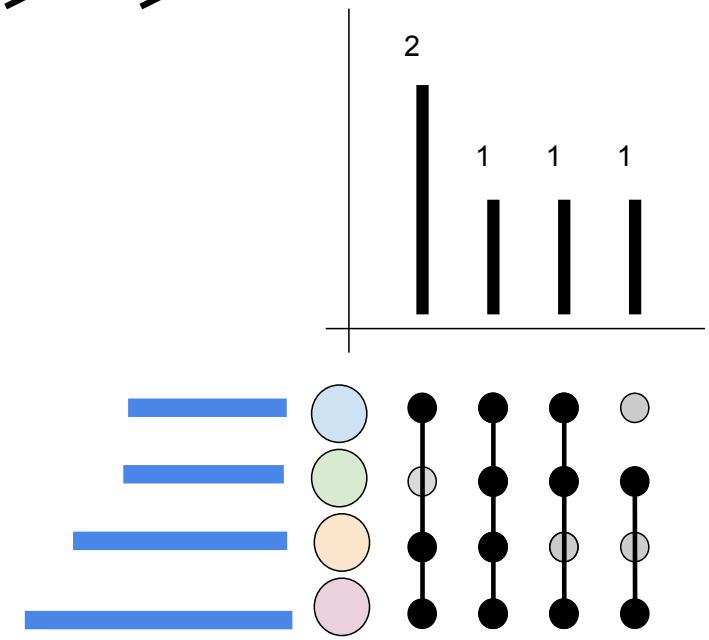
PICTURES : *Matrice de présence / absence et l'image géante*

UpSetR : Un package R pour la visualisation des ensembles.









PICTURES

UpSetR

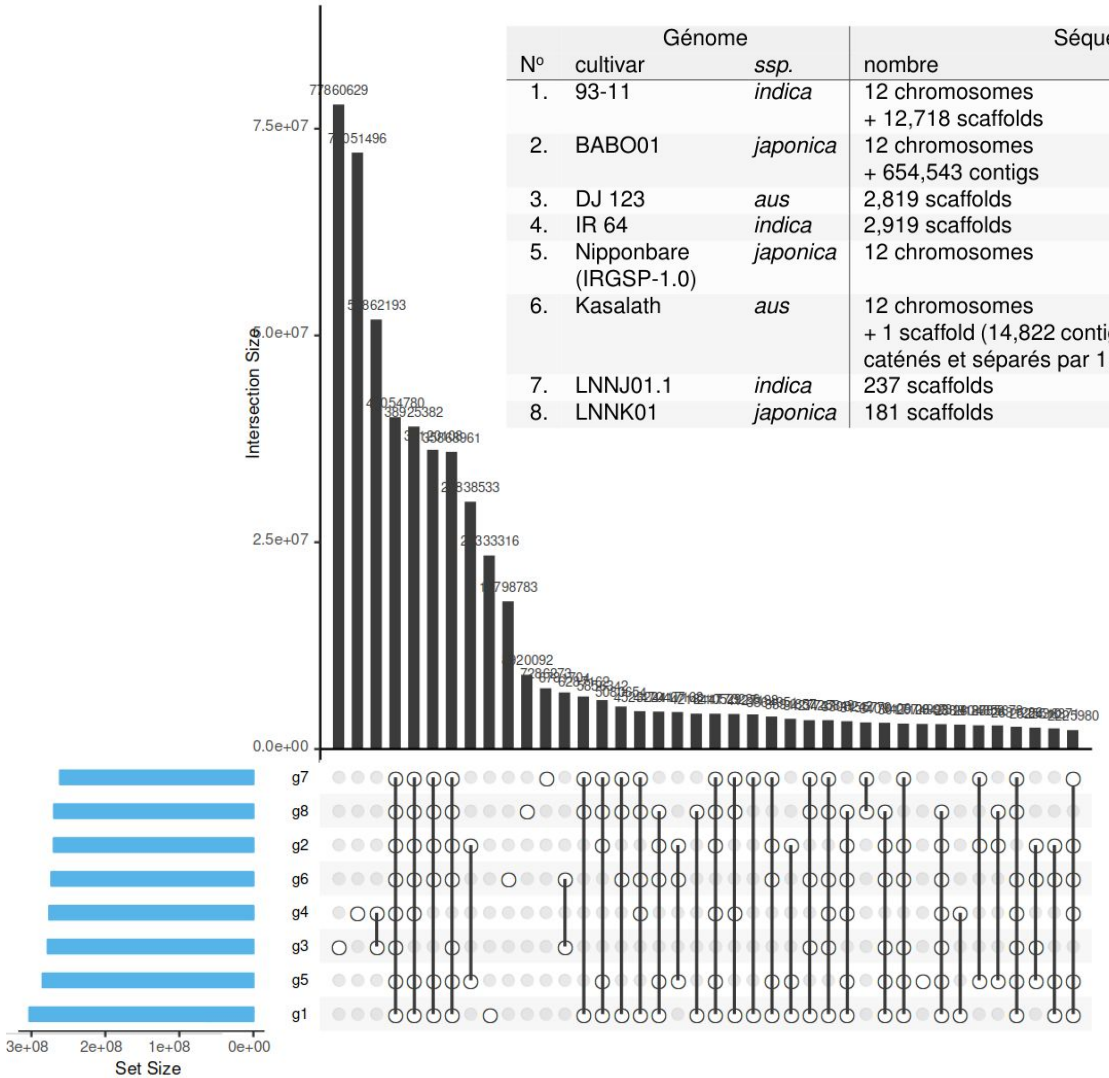
PICTURES: https://gitlab.southgreen.fr/ALICIA/PICTURES

UpSetR : Jake Conway, Nils Gehlenborg

Chapitre II : structuration des données

PICTURES : *Matrice de présence / absence et l'image géante*

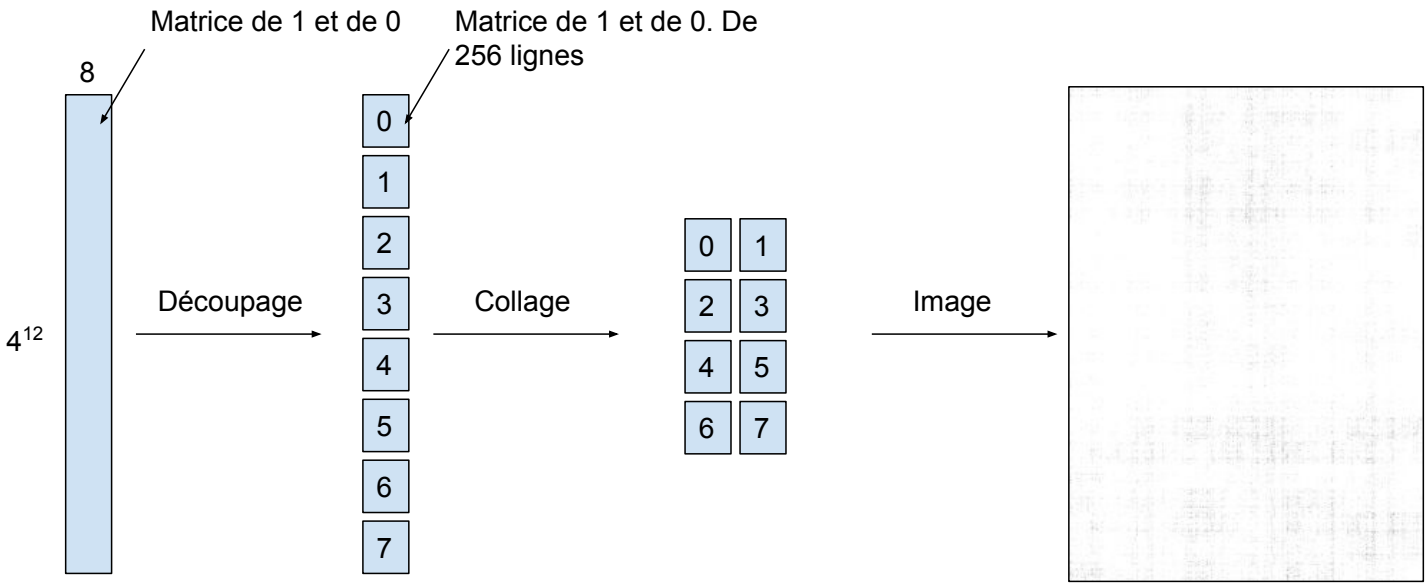
Nº	Génome cultivar	Génome *ssp.*	Séquences nombre	Séquences taille totale	Taille du fichier
1.	93-11	*indica*	12 chromosomes + 12,718 scaffolds	423,026,874pb	412MiB
2.	BABO01	*japonica*	12 chromosomes + 654,543 contigs	306,177,972pb	390MiB
3.	DJ 123	*aus*	2,819 scaffolds	345,981,746pb	335MiB
4.	IR 64	*indica*	2,919 scaffolds	345,209,449pb	334MiB
5.	Nipponbare (IRGSP-1.0)	*japonica*	12 chromosomes	373,245,519pb	362MiB
6.	Kasalath	*aus*	12 chromosomes + 1 scaffold (14,822 contigs concaténés et séparés par 1,000 N)	401,141,708pb	421MiB
7.	LNNJ01.1	*indica*	237 scaffolds	346,854,256pb	336MiB
8.	LNNK01	*japonica*	181 scaffolds	359,918,891pb	349MiB



[Schatz et al., 2014]

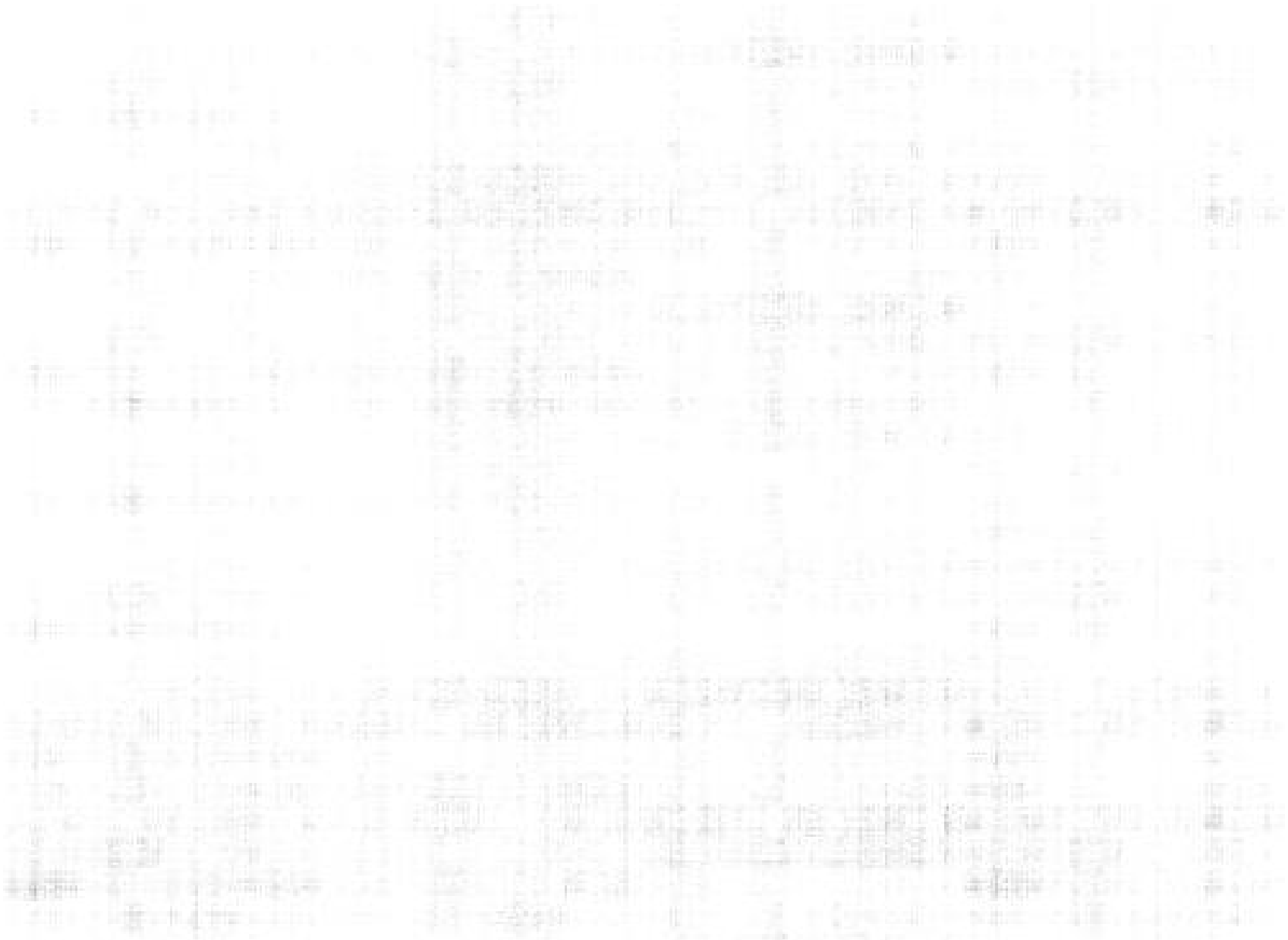
Chapitre II : structuration des données

PICTURES : *Matrice de présence / absence et l'image géante*



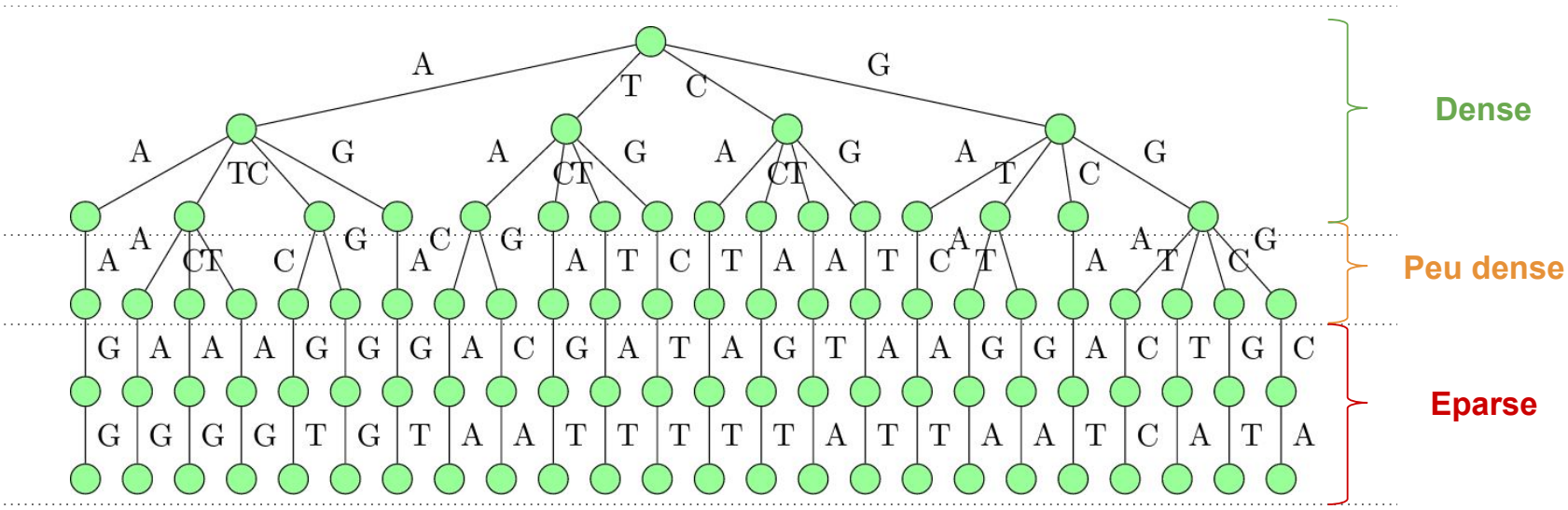
Chapitre II : structuration des données

PICTURES : *Matrice de présence / absence et l'image géante*



Chapitre II : structuration des données

Les comptes de la mère Loi



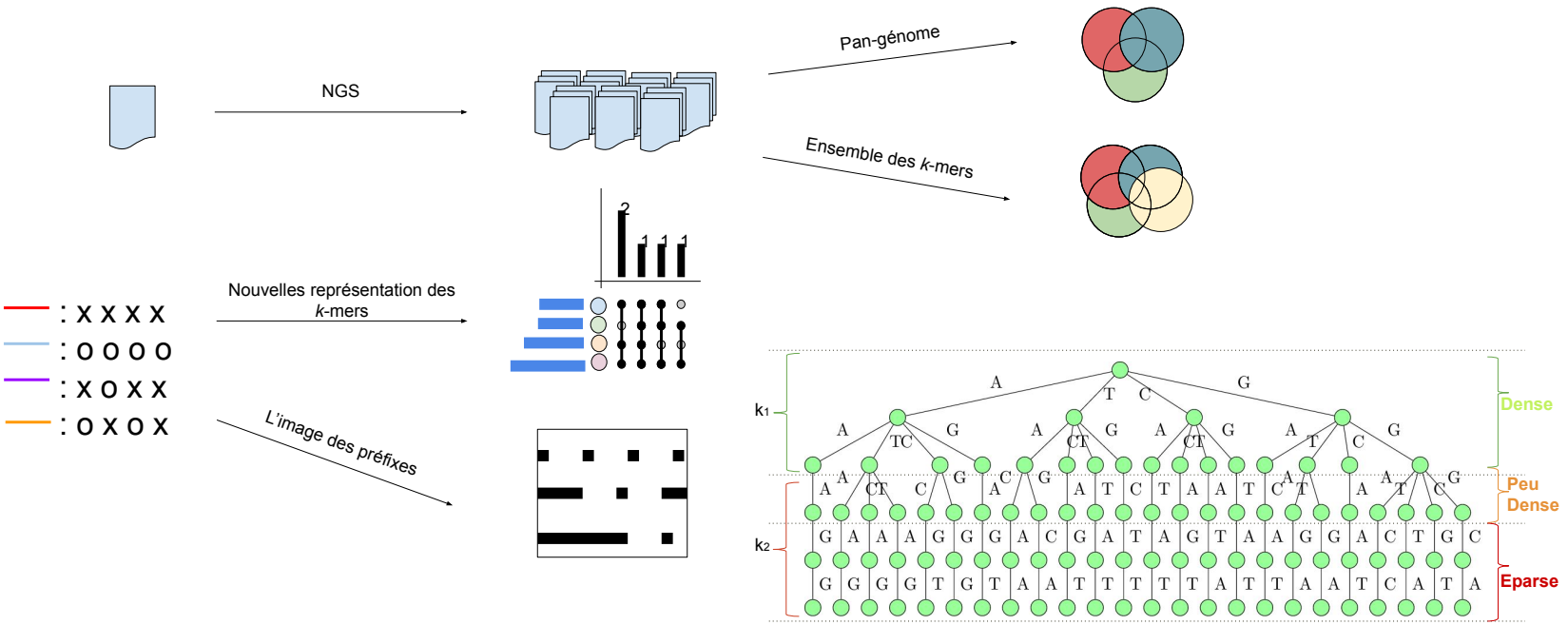
$\mathcal{N}$: Nombre total de k-mers

k_1 : Hauteur de la zone dense

$$2k_1 \leq \log(\mathcal{N}) - \log\log(\mathcal{N}) + O\left(\frac{1}{\log(\mathcal{N})}\right)$$

[Park et al., 2009]

Chapitre II : structuration des données



Chapitre III : Méthode d'indexation des *k*-mers

gkampi: *Le vaillant petit tailleur de k-mers*

JellyFish *[Marcais and Kingsford, 2011]*
khmer *[Zhang et al., 2014]*
KMC *[Deorowicz et al., 2013]*
KMC2 *[Deorowicz et al., 2014]*
KMC3 *[Kokot et al., 2017]*
. . .
gkampi

Chapitre III : Méthode d'indexation des *k*-mers

gkampi: *Le vaillant petit tailleur de k-mers*

Tools	Paradigm	Strategy	Data Structure	Storage	Input	Output	k
Jellyfish	C++/Multithread	-	Bloom filter/Hash table	HD	Fasta/Fastq	binary/tsv/fasta/hist.	2^{32}
DSK	C++/Multithread	-	Hash table	HD	Fasta/Fastq	binary/tsv	128
KMC	C++/Multithread	MP/MC	Splitted tables	HD	Fasta/Fastq (+ gzip)	binary/tsv	256
KHMER	python	-	Hash tables	HD	Fasta/Fastq	binary	int size
KCMBT	C++/Multithread	-	Burst tries	RAM	Fasta/Fastq (+ gzip)	binarys/tsv	32
Gerbil	C++/Multithread/GPU	MP/MC	Hash tables	HD	Fasta/Fastq (+gzip)	binary/fasta/hist.	136
Turtle	C++/Multithread	SP/MC	Bloom filter/Hash table	RAM	Fasta/Fastq	fasta/tsv	64
kmerstream	C++/Multithread	-	Arrays list	RAM	Fasta/Fastq	binary/tsv	int size
GKAMPI	C++/Multithread	MP/MC	Gk-arrays	RAM	Fasta/Fastq (+gzip)	binary/tsv/fasta/hist.	no limit

Chapitre III : Méthode d'indexation des *k*-mers

gkampi: *Le vaillant petit tailleur de k-mers*

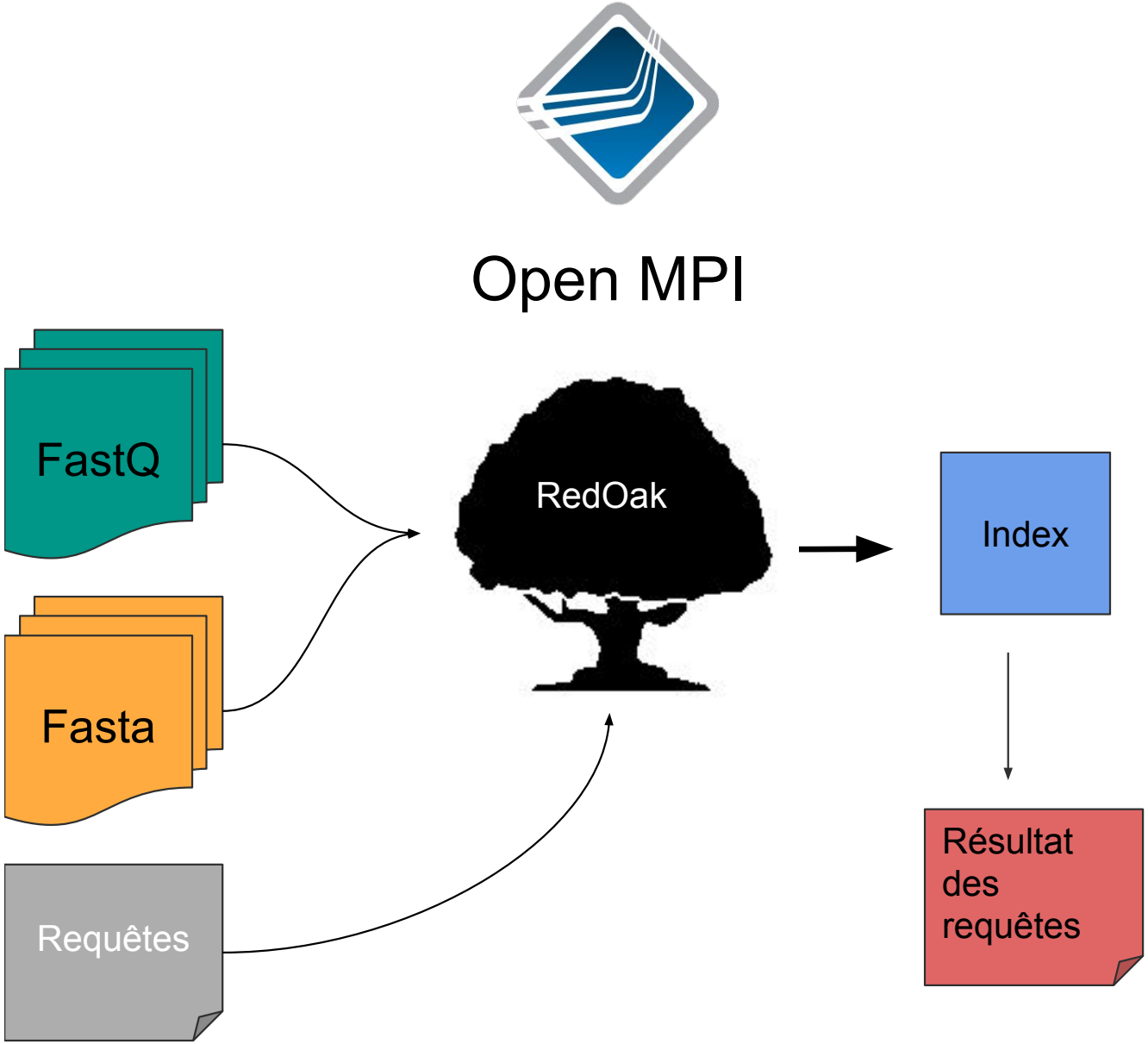
L'outil *gkampi*

- *Gk-Arrays* et *PgSA* *[Philippe et al., 2011, Kowalski et al., 2015]*
- Parallélisation
  - Massive : grille de calcul (Open MPI)
  - Légère : multicœur (OpenMP)
- Découpage des *k*-mers en préfixes/suffixes
- Indexation des *k*-mers

(git: https://gitlab.info-ufr.univ-montp2.fr/doccy/libGkArrays-MPI.git)

Chapitre III : Méthode d'indexation des *k*-mers

RedOak: *Le joueur de flûte de Montpellier*



Chapitre III : Méthode d'indexation des *k*-mers

RedOak: *Le joueur de flûte de Montpellier*

```
1  Input :
2    K* % The core k−mers of G = {G_1, ··· , G_N} %
3    K+ % The shell k−mers of G = {G_1, ··· , G_N} %
4    K− = ⊎_{i=1}^{N} K^i % The cloud k−mers of G = {G_1, ··· , G_N} %
5    g % A new genome to add %
6  Output :
7    ⟨K*, K+, K− = ⊎_{i=1}^{N+1} K^i⟩ % The updated index of G ∪ {g} = {G_1, ··· , G_{N+1}} %
8  Begin
9    K ← {w|w is a k-mer of g}
10   If N = 0 Then
11     K* ← K % All k−mers are in core %
12     K+ ← ∅ % There is no shell k−mers %
13     K^1 ← ∅ % There is no cloud k−mers %
14   Else
15     K' ← K* \ K % Those k−mers are not in core anymore %
16     K* ← K* \ K' % Only core k−mers that are in g remains in core %
17     K ← K \ K* % Removing core k−mers from K %
18     If N = 1 Then
19       K^1 ← K' % Move old core k−mers to cloud k−mers of G_1 %
20     Else
21       K ← K \ K+ % The shell k−mers that are in g remains shell %
22       K+ ← K+ ⊎ K' % Moving old core k−mers to shell k−mers %
23       For i in {1, ··· , n}
24         K' ← K^i ∩ K % Those k−mers are both in G_i and g %
25         K^i ← K^i \ K' % So they are removed from the cloud of G_i %
26         K ← K \ K' % and from the cloud of g %
27         K+ ← K+ ⊎ K' % Finally they are added to the shell k−mers %
28       End For
29     End If
30     K^{N+1} ← K % Add remaining k−mers from g to its cloud k−mers %
31   End If
32   Return ⟨K*, K+, K− = ⊎_{i=1}^{N+1} K^i⟩
33 End
```

Chapitre III : Méthode d'indexation des k-mers

RedOak: *Le joueur de flûte de Montpellier*

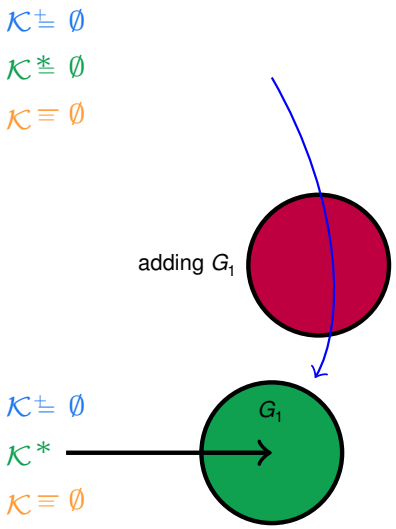
$\mathcal{K}^{\pm} \leftarrow \emptyset$

$\mathcal{K}^{*} \leftarrow \emptyset$

$\mathcal{K}^{=} \leftarrow \emptyset$

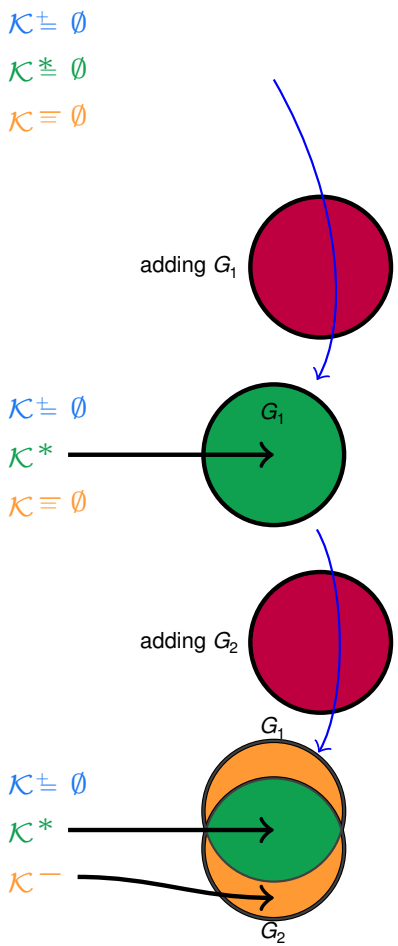
Chapitre III : Méthode d'indexation des *k*-mers

RedOak: *Le joueur de flûte de Montpellier*



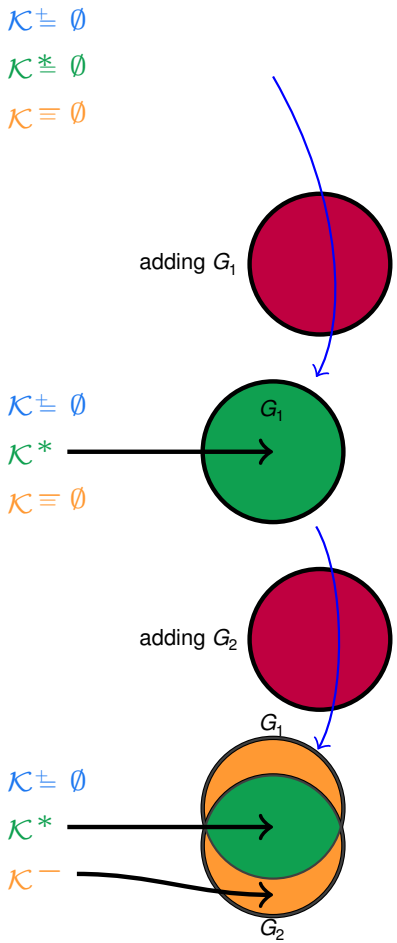
Chapitre III : Méthode d'indexation des *k*-mers

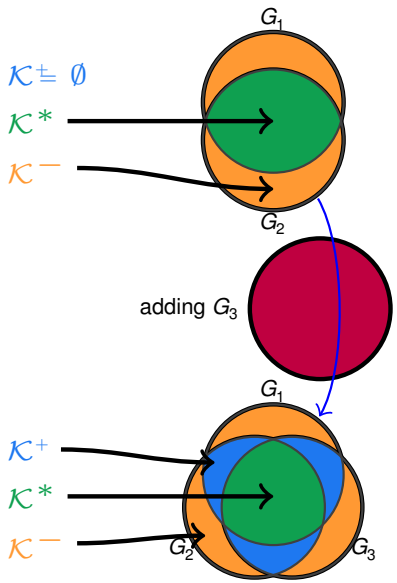
RedOak: *Le joueur de flûte de Montpellier*



Chapitre III : Méthode d'indexation des *k*-mers

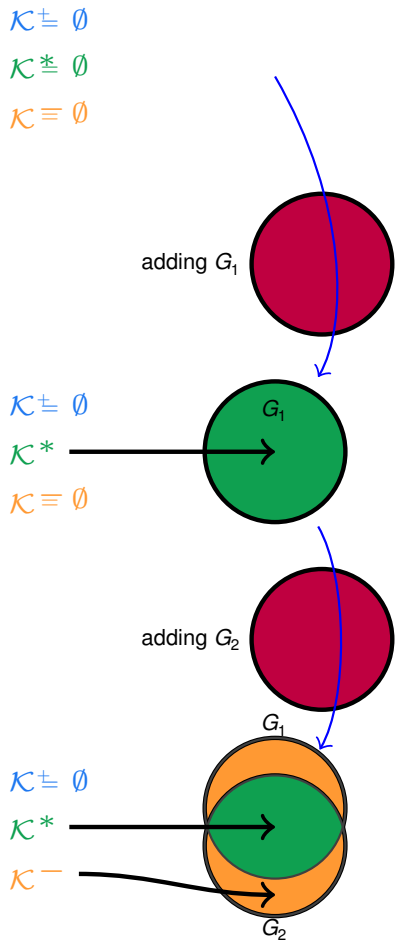
RedOak: *Le joueur de flûte de Montpellier*

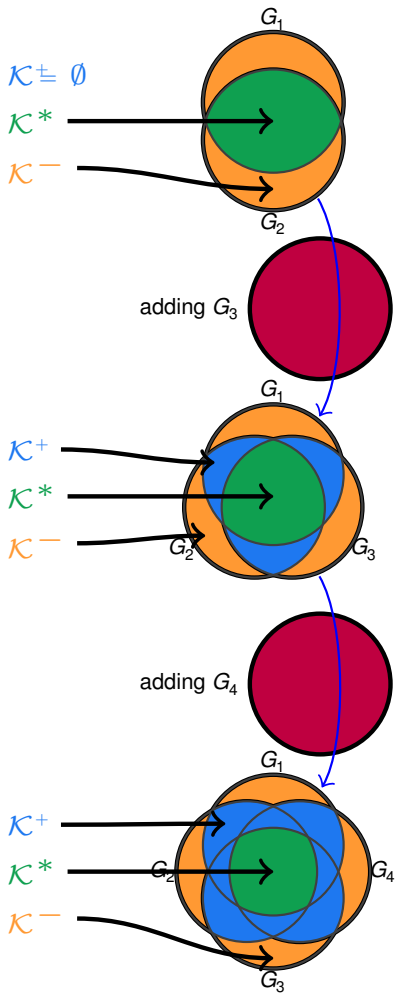




Chapitre III : Méthode d'indexation des *k*-mers

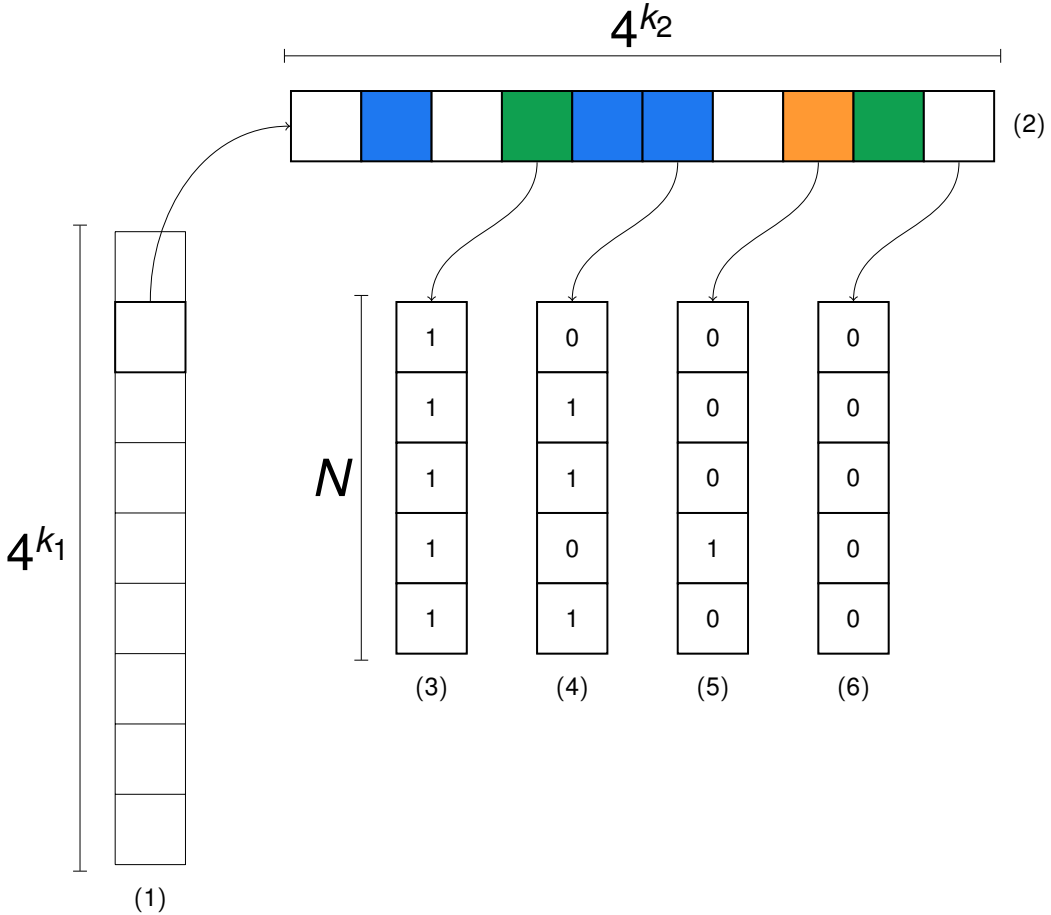
RedOak: *Le joueur de flûte de Montpellier*





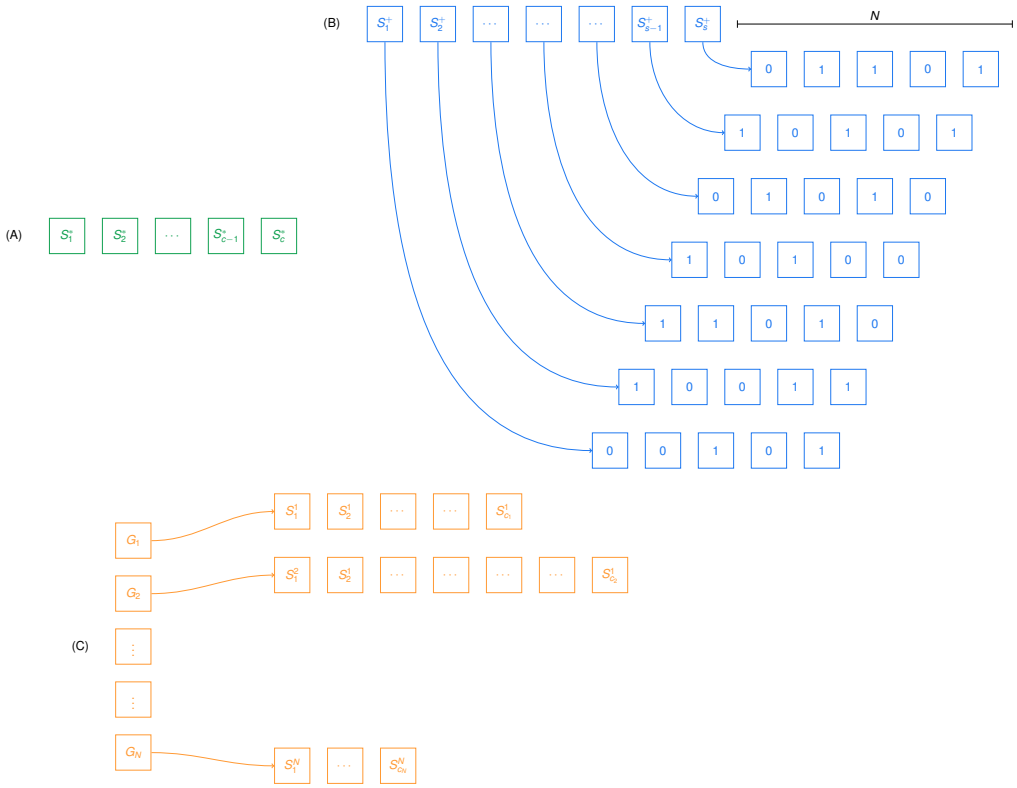
Chapitre III : Méthode d'indexation des *k*-mers

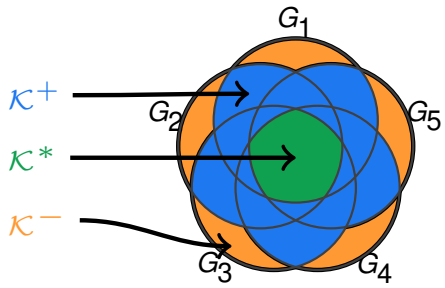
RedOak: *Le joueur de flûte de Montpellier*



Chapitre III : Méthode d'indexation des *k*-mers

RedOak: *Le joueur de flûte de Montpellier*





Requête : recherche les *k*-mers d'une séquence dans l'index et affiche les résultats.

--query 'ATAACGAGGGATGCTGGGTAAAATGCAAAGCTAG'
--query 'Reverse complement:CTAGCTTTGCATTTTACCCAGCATCCCTCGTTAT'

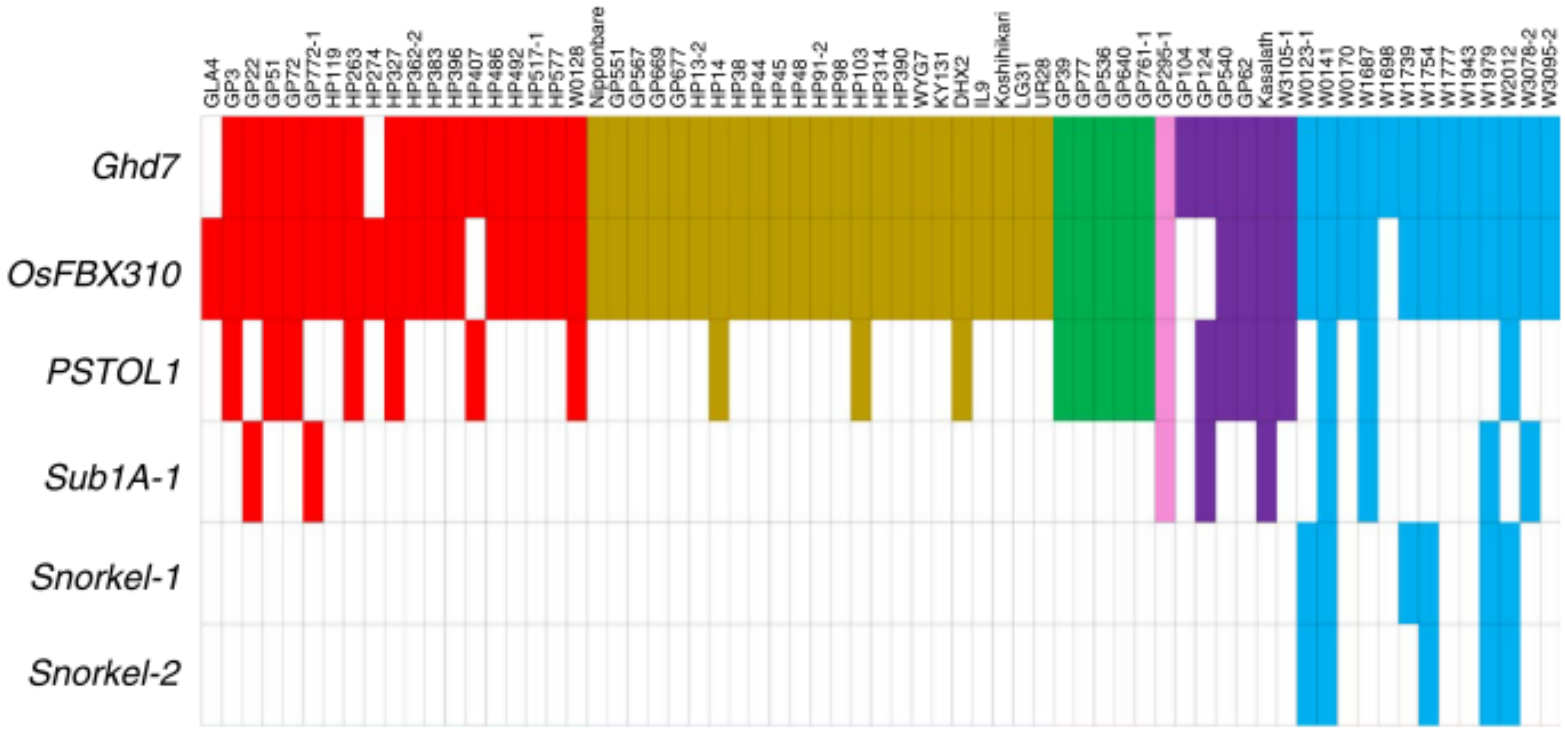
Les différentes valeurs disponibles pour les positions de la requête sont :

- " " Le *k*-mer et son complément inverse sont absents
- "." Il y a un *k*-mer précédent qui chevauche cette position
- "+" Le *k*-mer à cette position est présent
- "−" Le complément inverse du *k*-mer à cette position est présent
- "*" Le *k*-mer et son complément inverse à cette position sont présents

Chapitre III : Méthode d'indexation des *k*-mers

Les requêtes avec *RedOak*: *Aladin et la lampe merveilleuse*

Exemple d'application pour valider les requêtes :

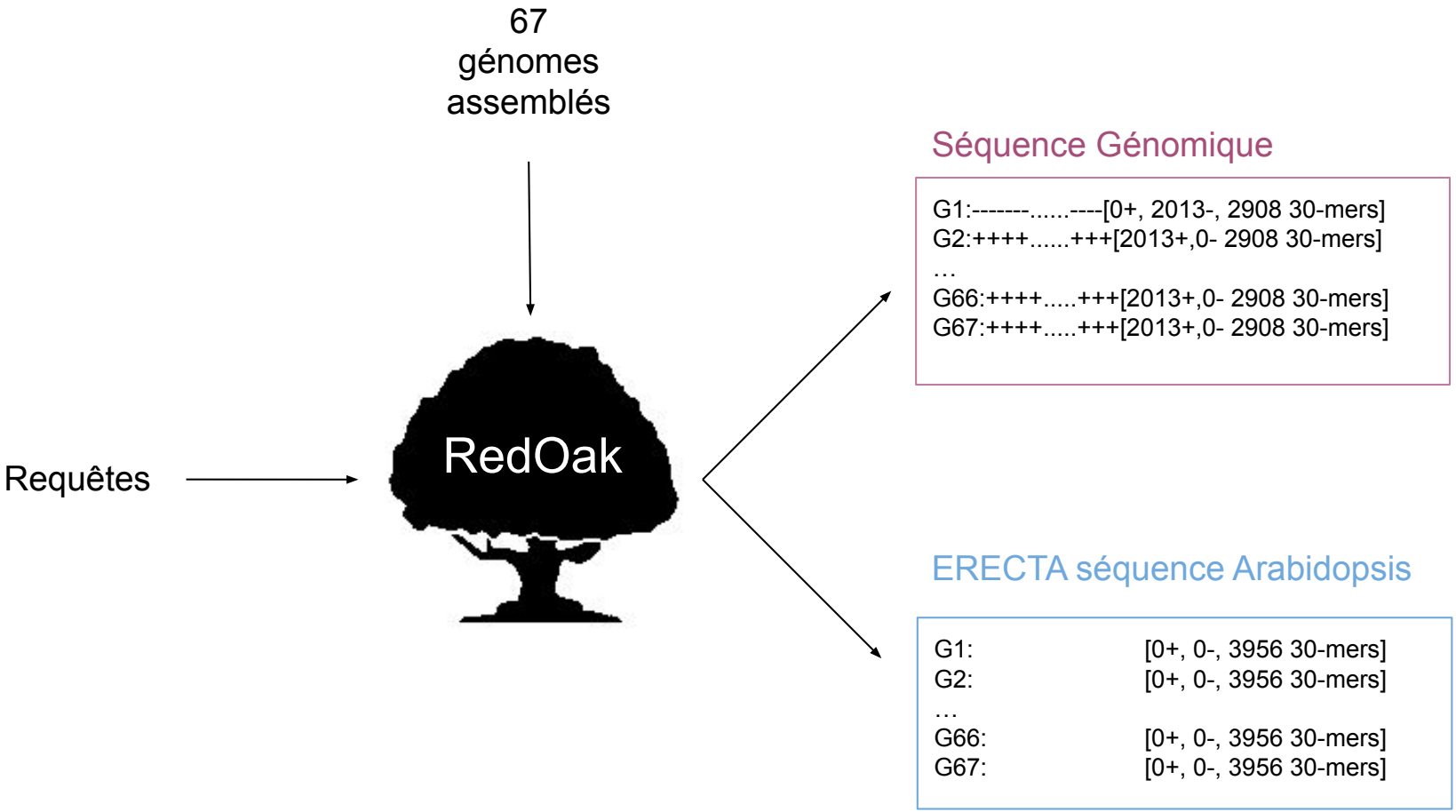


[Zhao et al., 2018].

Chapitre III : Méthode d'indexation des *k*-mers

Les requêtes avec *RedOak*: *Aladin et la lampe merveilleuse*

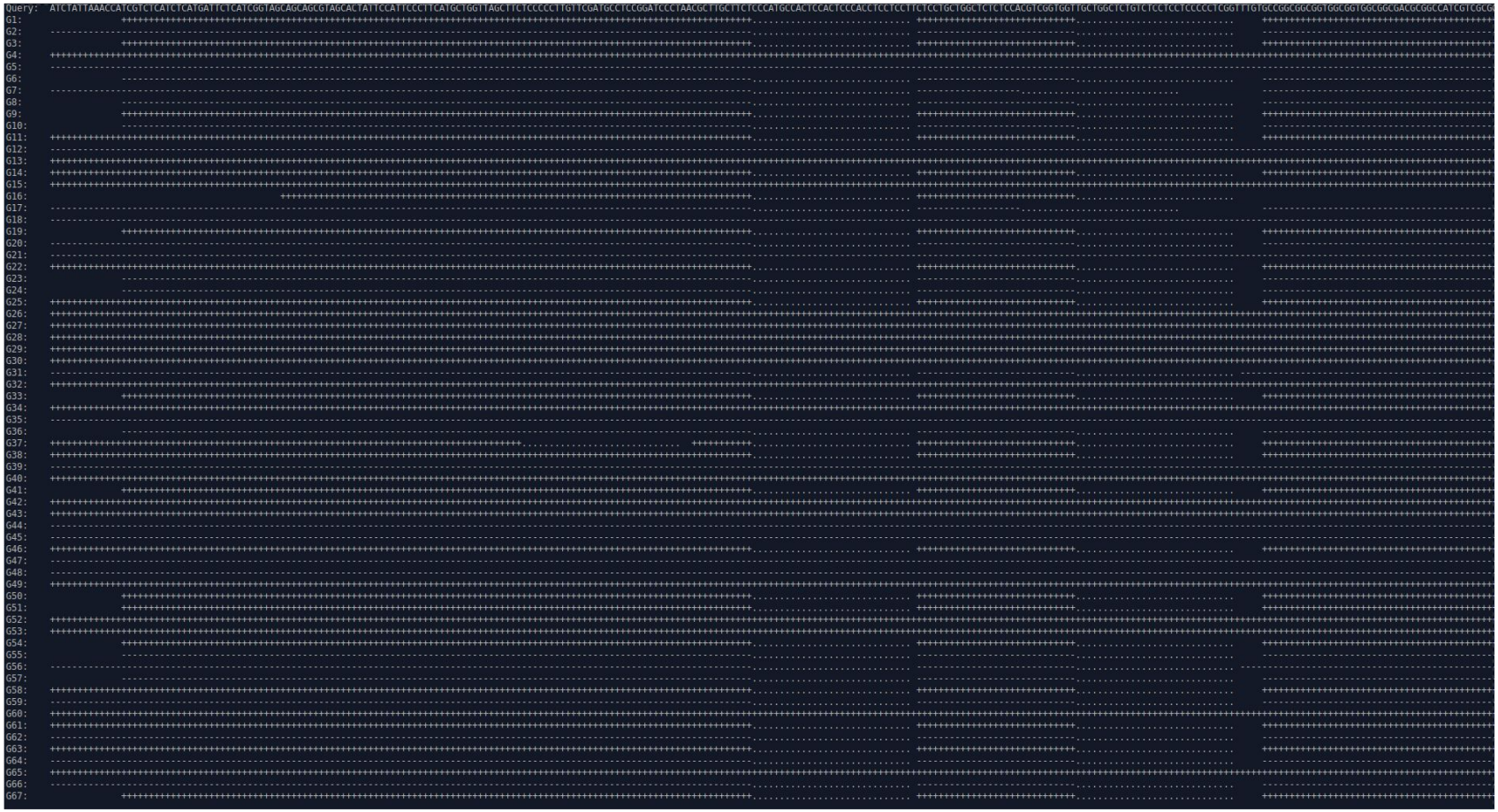
Les requêtes avec un découpage en 30-mers



Chapitre III : Méthode d'indexation des *k*-mers

Les requêtes avec *RedOak*: *Aladin et la lampe merveilleuse*

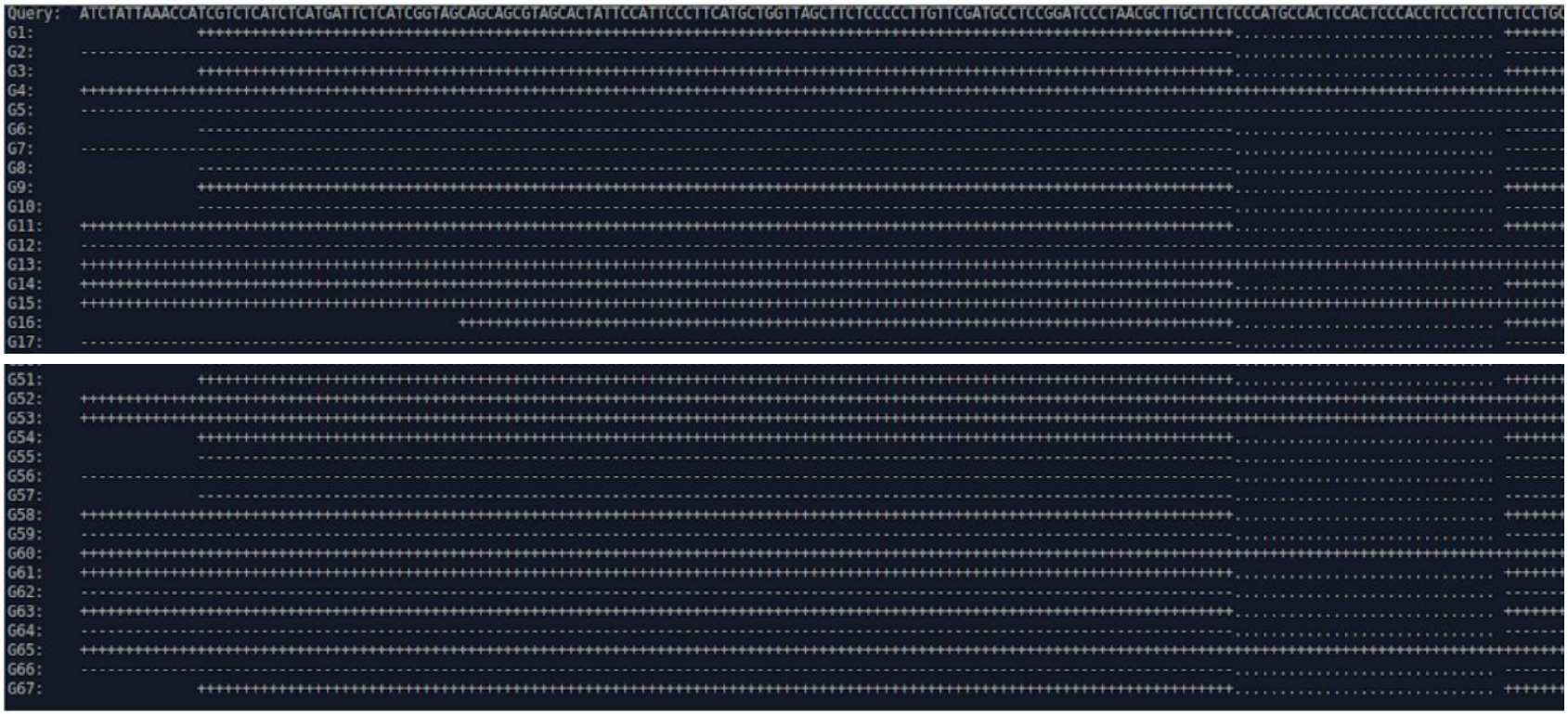
Recherche du gène PSTOL dans l'index de *RedOak*



Chapitre III : Méthode d'indexation des *k*-mers

Les requêtes avec *RedOak*: *Aladin et la lampe merveilleuse*

Recherche du gène PSTOL dans l'index de *RedOak*



Chapitre III : Méthode d'indexation des *k*-mers

Les requêtes avec *RedOak*: *Aladin et la lampe merveilleuse*

Recherche du gène ERECTA dans l'index de *RedOak*

Chapitre III : Méthode d'indexation des *k*-mers

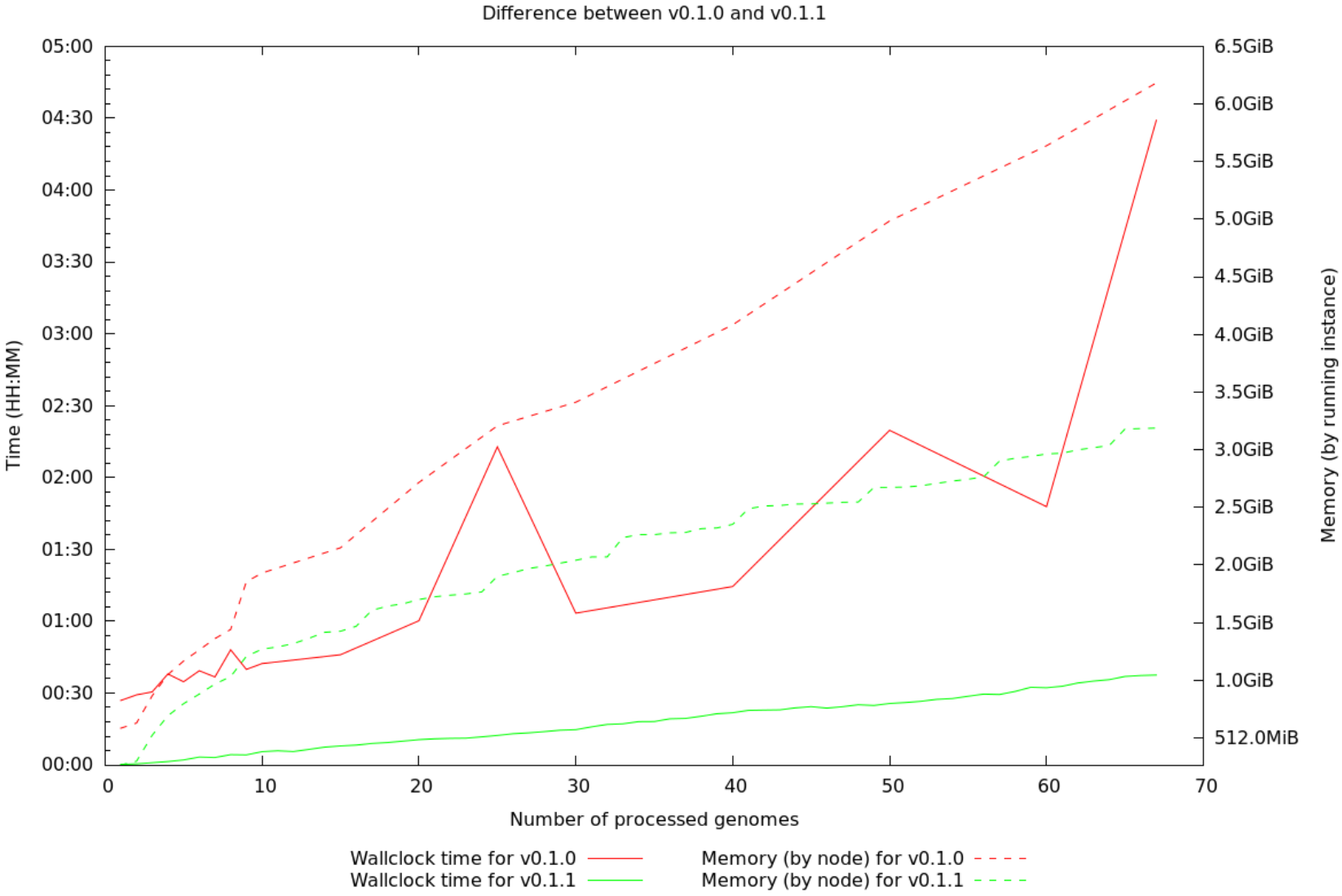
One Index to rule them all, One Index to find them, One Index to bring them all, and in the light show them

- Massivement parallélisé
- Conçu pour des clusters de calcul
- Permet d'indexer des génomes assemblés compressés ou non
- Permet de retrouver des gènes d'intérêt

$\rightarrow$ Est-ce que *RedOak* est adapté à une collection plus large de génomes ?

Chapitre IV : Performances

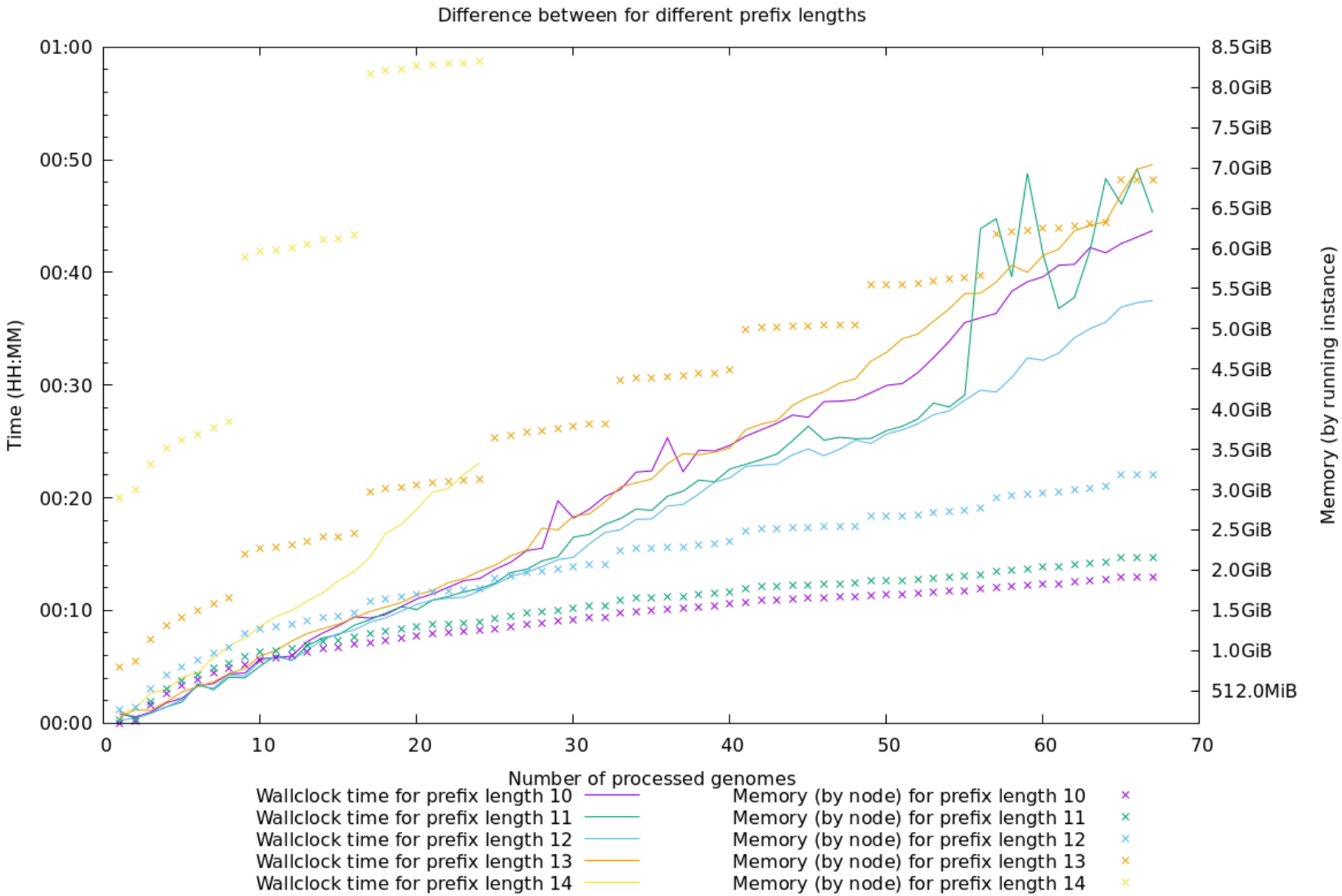
Le lièvre et la tortue



Chapitre IV : Performances

Le lièvre et la tortue

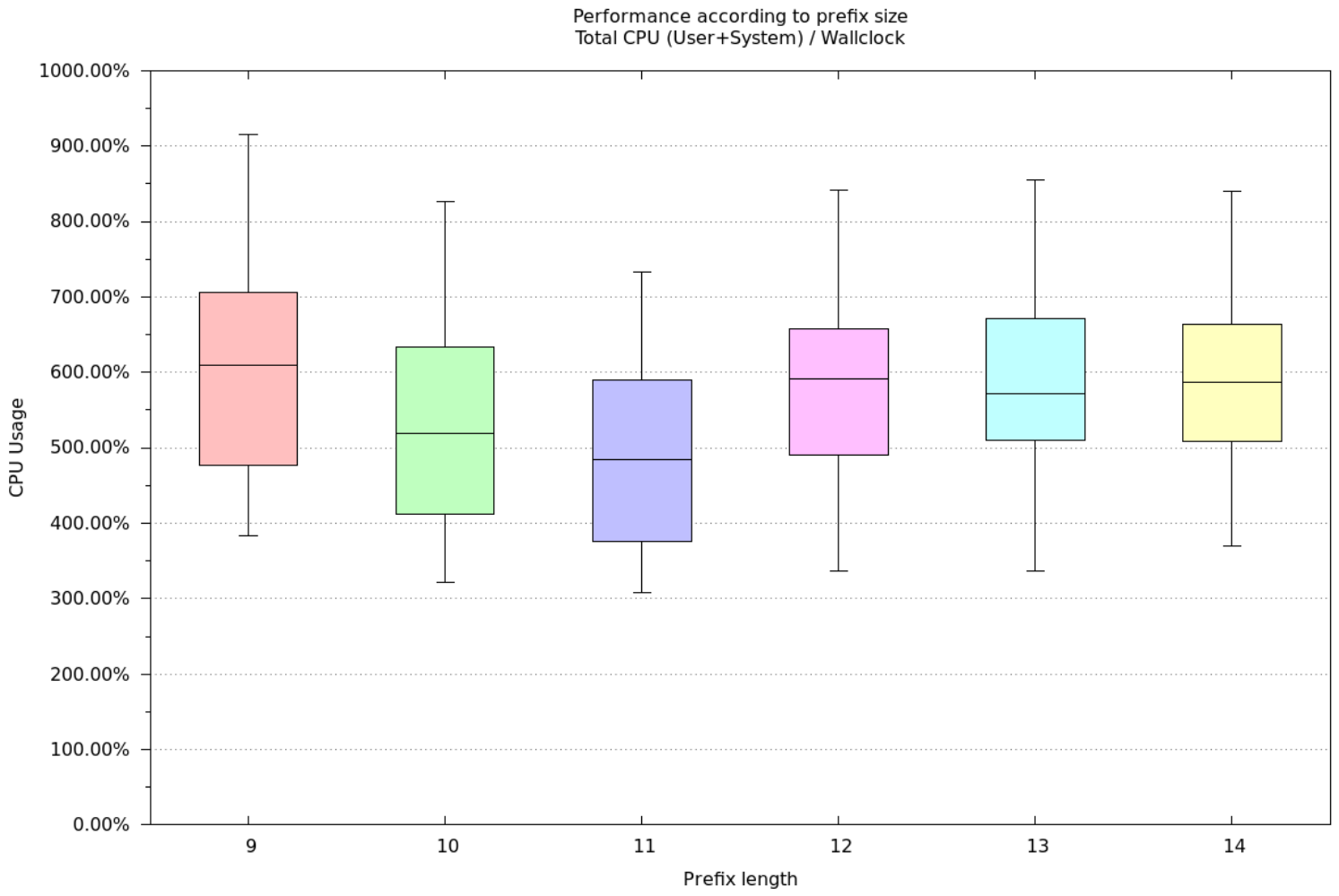
Indexation réalisée avec 2 nœuds et 20 instances par nœud



Chapitre IV : Performances

Ressources processeurs (calcul) : *les six petits préfixes*

Utilisation processeurs en fonction des différentes taille de préfixes



Bilan : taille des préfixes

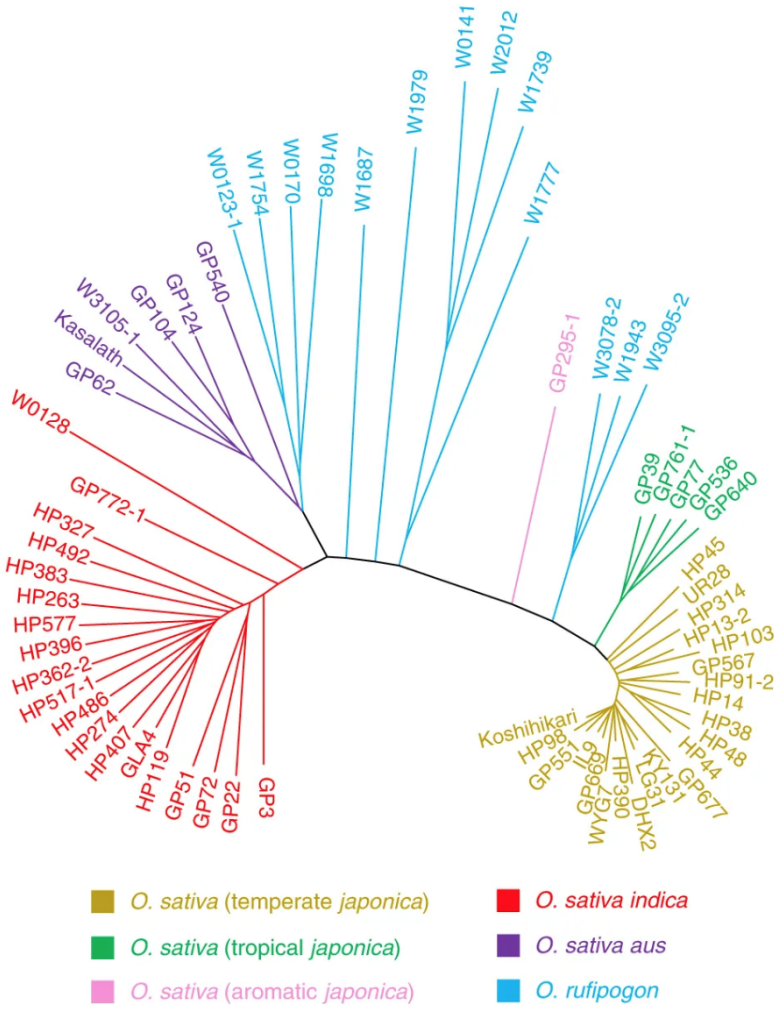
- Optimal en mémoire : $k_1 = 10$
- Optimal en temps : $k_1 = 12$
- Utiliser au mieux les ressources $k_1 = 10$

$\rightarrow$ Pour le riz la taille des préfixes qui semble être optimale est $k_1 = 10$ nucléotides

Chapitre IV : Performances

RedOak et les 67 génomes

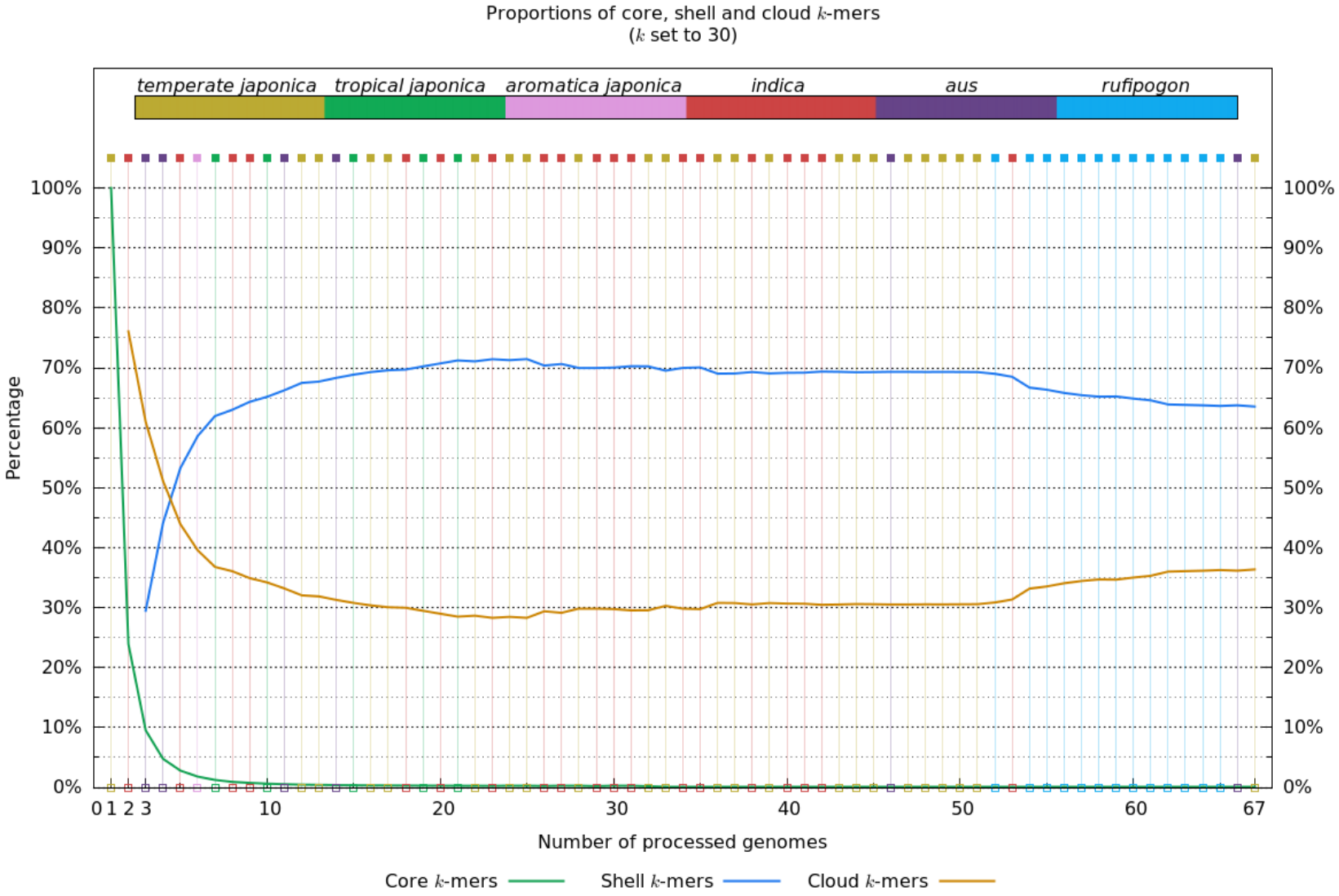
Évolution des *core*, *shell* et *cloud* *k*-mers



[Zhao et al., 2018]

Chapitre IV : Performances

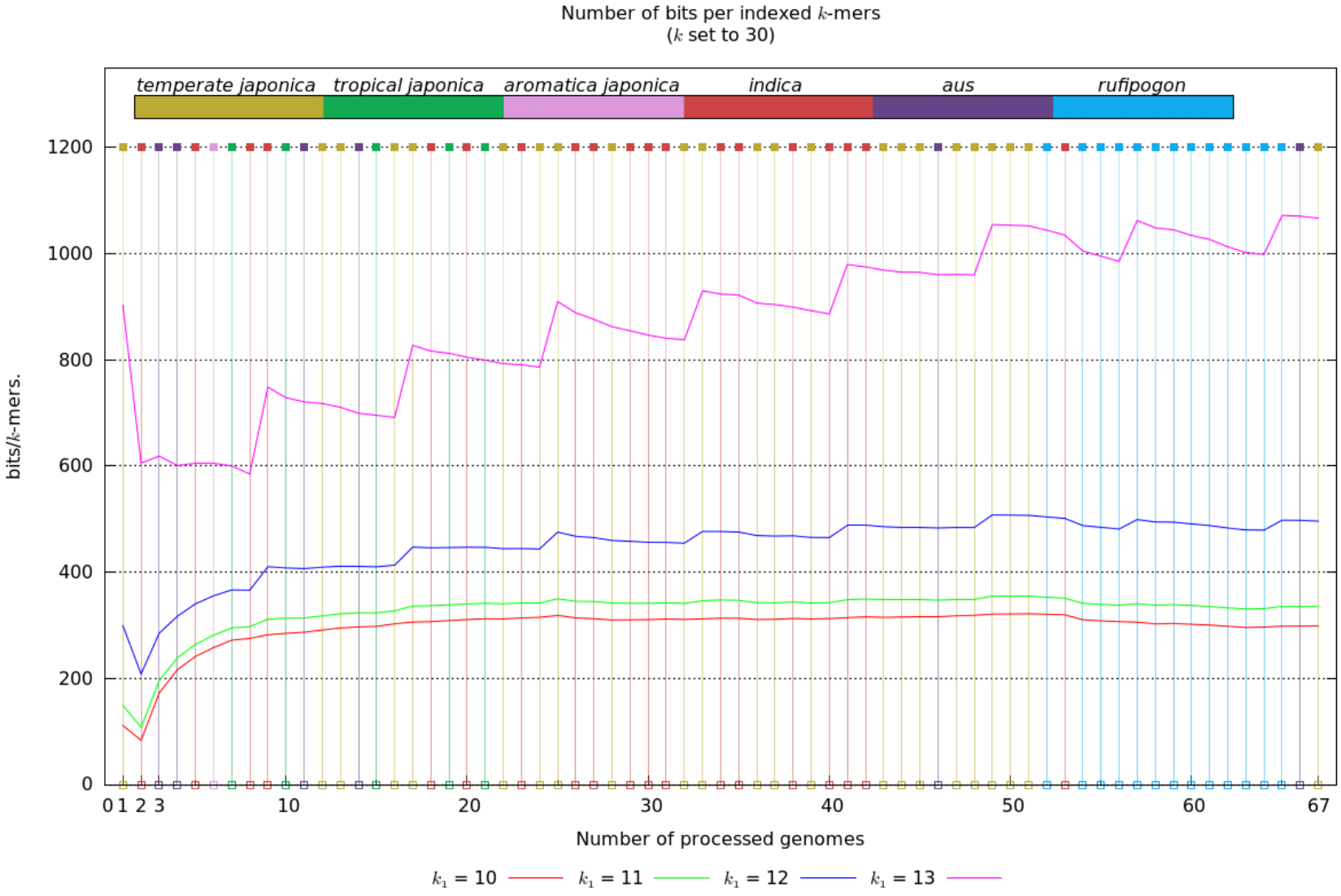
RedOak et les 67 génomes



Chapitre IV : Performances

RedOak et les 67 génomes

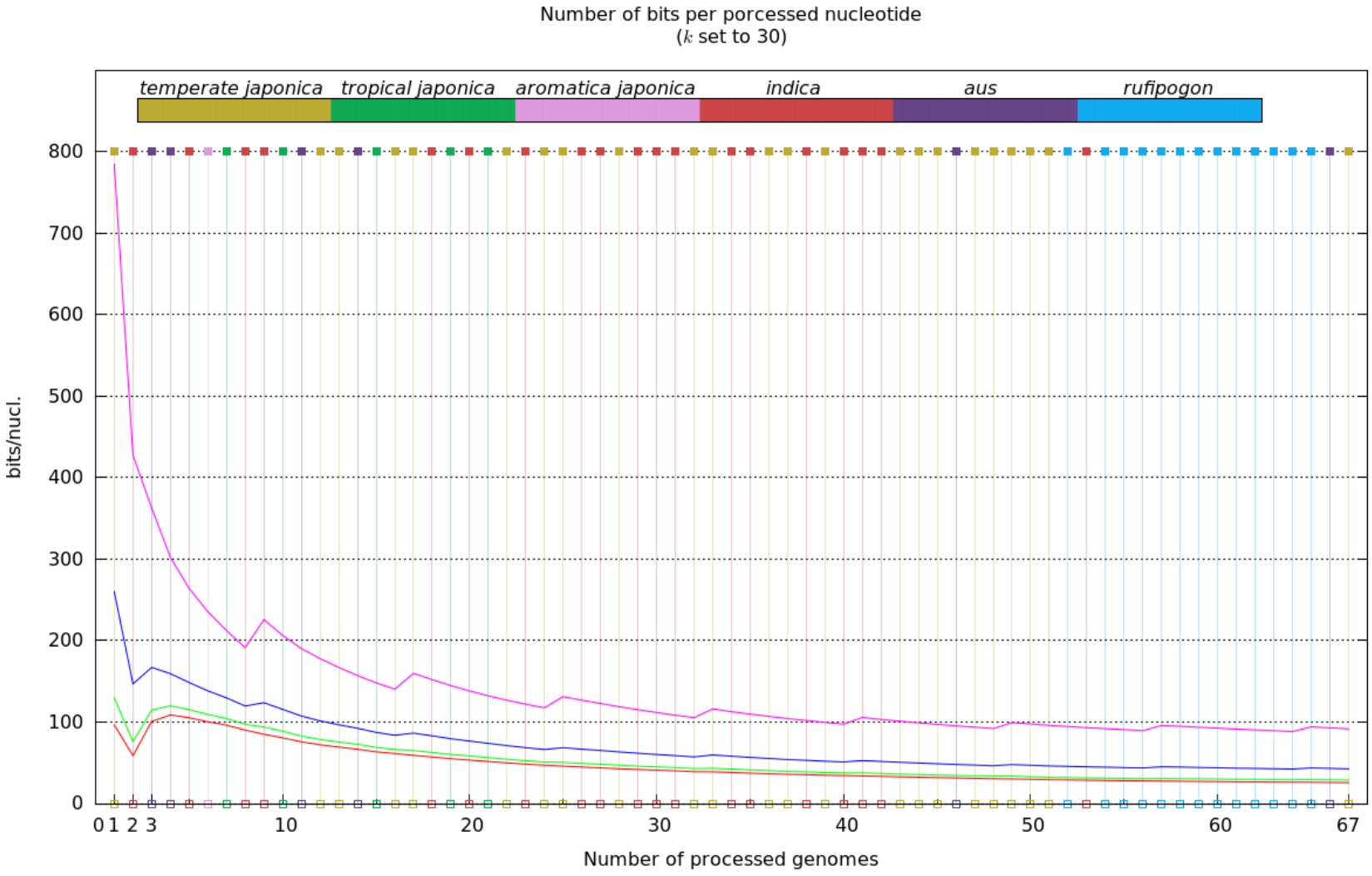
Étude du nombre de bits par *k*-mers pour différentes tailles de préfixe



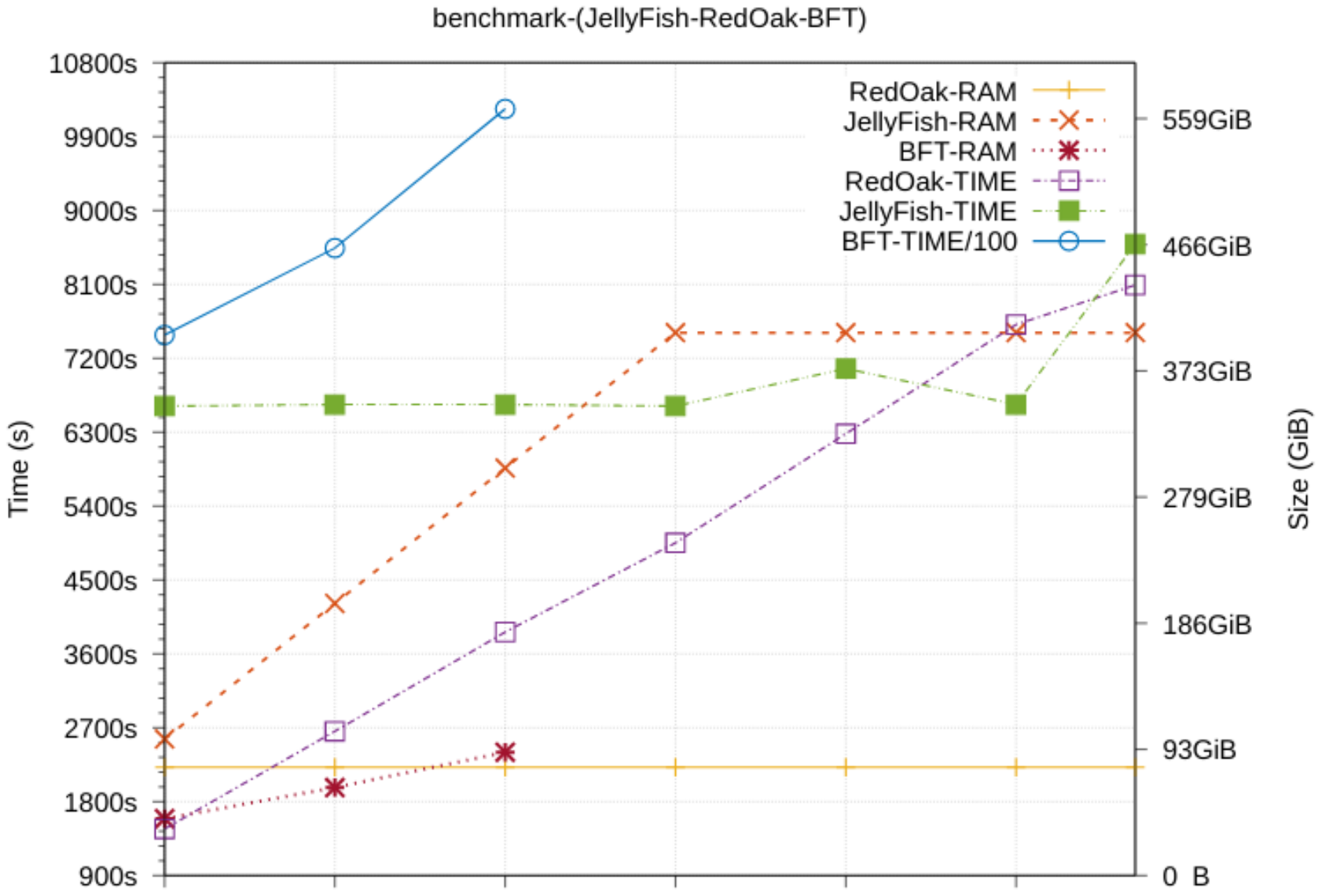
Chapitre IV : Performances

RedOak et les 67 génomes

Étude du nombre de bits par nucléotide pour différentes tailles de préfixe



Comparaison avec d'autres méthodes : construction de l'index



Chapitre IV : Performances

Le petit Poucet

Comparaison avec d'autres méthodes : requêtage

	Memory RAM (GiB) and Query Time (s)				
Query length	RedOak-RAM	RedOak-Query	JellyFish-RAM	JellyFish-Query	JellyFish-Query-Total
270	7	1.179	16	6360	257926
540	7	3.325	16	7919	301113
810	7	2.703	16	11586	374489
1080	7	6.847	16	12996	368945
1350	7	3.839	16	12280	351517
2700	7	9.006	16	12880	391060
5400	7	19.634	16	13397	337673
8100	7	24.976	16	11701	349188
10800	7	34.939	16	10668	262252
13500	7	43.997	16	9779	263889
27000	7	60.389	16	9569	295048

Conclusion

- La structure d'indexation :
  - Massivement parallélisée (grille de calcul)
  - L'indexation de données génomiques
  - La recherche de séquences d'intérêt
  - La comparaison de centaines de génomes
- Implémentée en C/C++, documentée et disponible : (git: https://gitlab.info-ufr.univ-montp2.fr/DoccY/RedOak)

Conclusion

Perspectives

- Élargir à d'autres espèces (*GenomeHarvest*)
- Étudier la loi de distribution des *k*-mers
- La Phylokmérie
  - Faire une classification avec les *k*-mers
  - Arbre des liens entre êtres vivants basé sur les *k*-mers

Merci pour votre attention

https://www.montpellier-omics-days.fr

Montpellier Omics Days

MONTPELLIER OMICS DAYS 2021

CONFÉRENCE ANNUELLE DE BIOINFORMATIQUE & STATISTIQUES

9 & 10 FÉVRIER - FACULTÉ DES SCIENCES DE MONTPELLIER

Accueil | Programme | Inscriptions | Partenaires | Editions Précédentes | Videos | Concours | Contact & Accès

Mon projet filmé en 180 secondes

Cette année, un concours est mis en place pour cette 9ème édition des Montpellier Omics Days (MOD), qui exceptionnellement se fera à distance. Ce concours consiste pour les participants volontaires à réaliser une **vidéo** présentant **leurs projets et leurs travaux de recherche** : une sélection de 6 vidéos sera faite par les organisateurs des MOD sur la base des critères définis ci-dessous. Ces vidéos seront diffusées lors des deux demi-journées de conférences en ligne des MOD 2021 (9 et 10 février 2021). Les participants des MOD pourront voter pour la vidéo qu'ils auront préféré parmi les 6. Un prix qui sera prochainement défini, sera attribué au chercheur ayant réalisé la vidéo préférée de cette édition. Les autres vidéos seront disponibles sur le site des MOD pour être visibles par les personnes intéressées par le sujet abordé.

Public visé : les doctorants, post-doctorants et les ingénieurs d'étude ou de recherche qui développent ou utilisent des approches bioinformatiques et statistiques pour l'omique (génomique, épigénomique, transcriptomique, protéomique, métagénomique, etc.).

Descriptif des vidéos demandé : une vidéo pendant laquelle le chercheur présente son projet et ses travaux de recherche en **180 secondes**. Les travaux doivent être en lien avec la bioinformatique ou les statistiques pour l'omique. Les participants pourront présenter par exemple les résultats de leurs travaux, une méthode qu'ils auraient développée, ou encore un outil bioinformatique qu'ils auraient mis en place.

Critères de sélection des vidéos : les vidéos seront sélectionnées par les organisateurs des MOD en favorisant celles **vulgarisant** vos travaux de recherche, et ayant une approche **pédagogique**. Le dynamisme et l'originalité des vidéos sont également attendus. La forme des vidéos est libre. C'est-à-dire que vous pouvez faire ce que vous voulez, comme par exemple : réaliser un dessin animé ou 3D façon pixar, imiter un documentaire vidéo, un reportage TV, reprendre les codes d'une série TV avec vos collègues ou vos amis. En résumé : **vous êtes libre, soyez créatif !** Mais n'oubliez pas, nous insistons sur la vulgarisation et la pédagogie de vos vidéos. Les vidéos doivent être réalisées en français ou en anglais. Vous pouvez ajouter des sous-titres en anglais, mais ce n'est pas obligatoire.


Deorowicz, S., Debudaj-Grabysz, A., and Grabowski, S. (2013).
Disk-based k-mer counting on a PC.
BMC Bioinformatics, 14(1).


Deorowicz, S., Kokot, M., Grabowski, S., and Debudaj-Grabysz, A. (2014).
KMC 2: Fast and resource-frugal k-mer counting.
Bioinformatics, 31(10):1569–1576.


Kokot, M., Długosz, M., and Deorowicz, S. (2017).
KMC 3: counting and manipulating k-mer statistics.


Koonin, E. V. and Wolf, Y. I. (2008).
Genomics of bacteria and archaea: the emerging dynamic view of the prokaryotic world.
Nucleic Acids Res, 36(21):6688–719.


Kowalski, T., Grabowski, S., and Deorowicz, S. (2015).
Indexing arbitrary-length k-mers in sequencing reads.
PLOS ONE, 10(7):1–16.


Li, J. Y., Wang, J., and Zeigler, R. S. (2014).
The 3,000 rice genomes project: new opportunities and challenges for future rice research.

Gigascience, 3:8.


Marcais, G. and Kingsford, C. (2011).
A fast, lock-free approach for efficient parallel counting of occurrences of k-mers.
Bioinformatics, 27(6):764–770.


Park, G., Hwang, H.-K., Nicodème, P., and Szpankowski, W. (2009).
Profiles of Tries.
SIAM Journal on Computing, 38(5):1821–1880.


Philippe, N., Salson, M., Lecroq, T., Léonard, M., Commes, T., and Rivals, E. (2011).
Querying large read collections in main memory: a versatile data structure.
BMC Bioinformatics, 12(1):242.


Schatz, M. C., Maron, L. G., Stein, J. C., Wences, A. H., Gurtowski, J., Biggers, E., Lee, H., Kramer, M., Antoniou, E., Ghiban, E., Wright, M. H., ming Chia, J., Ware, D., McCouch, S. R., and McCombie, W. R. (2014).
Whole genome de novo assemblies of three divergent strains of rice, oryza sativa, document novel gene space of aus and indica.

Genome biology, 15:506.


Snipen, L. and Ussery, D. W. (2010).
Standard operating procedure for computing pangenome trees.
Standards in Genomic Sciences, 2(1):135–141.


Tettelin, H., Masignani, V., Cieslewicz, M. J., Donati, C., Medini, D., Ward, N. L., Angiuoli, S. V., Crabtree, J., Jones, A. L., Durkin, A. S., DeBoy, R. T., Davidsen, T. M., Mora, M., Scarselli, M., Margarit Y Ros, I., Peterson, J. D., Hauser, C. R., Sundaram, J. P., Nelson, W. C., Madupu, R., Brinkac, L. M., Dodson, R. J., Rosovitz, M. J., Sullivan, S. A., Daugherty, S. C., Haft, D. H., Selengut, J., Gwinn, M. L., Zhou, L., Zafar, N., Khouri, H., Radune, D., Dimitrov, G., Watkins, K., O'Connor, K. J., Smith, S., Utterback, T. R., White, O., Rubens, C. E., Grandi, G., Madoff, L. C., Kasper, D. L., Telford, J. L., Wessels, M. R., Rappuoli, R., and Fraser, C. M. (2005).
Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: Implications for the microbial "pan-genome".
Proceedings of the National Academy of Sciences of the United States of America, 102(39):13950–13955.

Zhang, Q., Pell, J., Canino-Koning, R., Howe, A. C., and Brown, C. T. (2014).
These Are Not the K-mers You Are Looking For: Efficient Online K-mer Counting Using a Probabilistic Data Structure.
PLoS ONE, 9(7):e101271.

Zhao, Q., Feng, Q., Lu, H., Li, Y., Wang, A., Tian, Q., Zhan, Q., Lu, Y., Zhang, L., Huang, T., Wang, Y., Fan, D., Zhao, Y., Wang, Z., Zhou, C., Chen, J., Zhu, C., Li, W., Weng, Q., Xu, Q., Wang, Z. X., Wei, X., Han, B., and Huang, X. (2018).
Pan-genome analysis highlights the extent of genomic variation in cultivated and wild rice.
Nat Genet, 50(2):278–284.