UCLouvain

ICTEAM - ELEN

GLORIA

Global Rapid and Innovative DNA Analysis

Benchmarking de GSC

Contacts :
Pauline HERMANS

Emails :
pauline.hermans@uclouvain.be

Décembre 2024

Table des matières

1 Introduction

L'objectif de ce rapport est d'étudier l'algorithme de compression de fichiers VCF (Variant Call Format) [2] qui sera probablement utilisé dans le cadre du projet GLORIA. Il existe différents algorithmes qui peuvent convenir. Une étude comparative plus générale a déjà été réalisée et détaillée dans le rapport semestriel du deuxième semestre du projet (octobre 2024). Il en est ressorti que l'algorithme optimal est GSC (Genotype Sparse Compression) [3]. Pour une présentation détaillé de l'algorithme, l'article scientifique de celui-ci est disponible à `https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giae046/7716932` et le code (en C++) de l'algorithme est disponible sur la page GitHub `https://github.com/luo-xiaolong/GSC/tree/master`. Seuls les éléments nécessaires à la compréhension de ce rapport seront expliqués ici.

Plusieurs hyperparamètres peuvent être définis lors de l'utilisation de GSC. Ce rapport a donc pour but de trouver quels sont les hyperparamètres optimaux (nombre de sujets par fichiers et nombre de threads utilisés lors de la compression) en réalisant un benchmarking.

À titre comparatif, des tests similaires à ceux présentés ici pour GSC ont également été réalisés sur SVC. L'ensemble des graphes et quelques commentaires rapides sont présentés en annexe.

2 Méthodes

Dans cette section, les méthodes utilisées pour réaliser le benchmarking seront détaillées. L'ensemble des scripts utilisés pour lancer les tests, les données obtenues ainsi que les codes pour réaliser les graphes et les graphes obtenus sont disponibles sur le GitHub `https://github.com/PaulineHrms/VCF_compression_benchmarking.git` .

2.1 Hyperparamètres

Il y a 2 hyperparamètres principaux sur lequel nous pouvons jouer :

- **Le nombre de sujets dans un fichier VCF** : La question est de savoir quel nombre de sujets permet d'avoir des performances optimales. Le but de ce benchmarking est d'évaluer les performances pour des fichiers VCF de différentes taille afin de choisir le meilleur compromis entre vitesse, RAM et taux de compression. Il est également important de tenir compte de la sécurité et naturellement le nombre de sujets par fichier est fortement lié à la sécurité. Nous ne discuterons cependant pas de cet aspect dans ce rapport.

- **Nombre de threads** : Lors de la compression, le nombre de threads utilisés peut être choisi. Les performances peuvent également variées en fonction de la machine utilisée, ce benchmarking a donc pour but de donner une idée globale de l'influence du nombre de threads sur la vitesse et la RAM utilisée.

2.2 Métriques de performances

Le projet GLORIA a certaines attentes pour l'algorithme de compression qui sera utilisé pour le stockage à froid. Ci-dessous sont listés les **tests (en gras)** qui ont été réalisés et les *métriques (en italique)* qui ont été mesurées au cours des ces tests :

- **Compression** : *taux de compression* entre le fichier initial et la version compressée, *vitesse de compression* (en secondes par MB du fichier VCF à compresser), *RAM maximale* utilisée (en MB de RAM par MB du fichier VCF à compresser).

- **Décompression complète (de tous les sujets du fichier VCF)** : *vitesse de décompression* (en secondes par MB du fichier VCF compressé), *RAM maximale* utilisée (en MB de RAM par MB du fichier VCF compressé).

- **Décompression d'un seul sujet** : *vitesse de décompression* (en secondes), *RAM maximale* utilisée (en MB).

- **Random access complet (de tous les sujets du fichier VCF) pour 1 seule position** : *vitesse de random access* (en secondes par MB du fichier VCF compressé), *RAM maximale* utilisée (en MB de RAM par MB du fichier VCF compressé).

- **Random access d'un seul sujet pour 1 seule position** : *vitesse de random access* (en secondes), *RAM maximale* utilisée (en MB).

Pour le benchmarking lié au nombre de sujets dans les fichiers VCF, les fichiers ont été sous-échantillonnés à 2500, 2250, 2000, 1500, 1000, 900, 800, 750, 700, 600, 500, 250, 200, 150, 100, 50, 25, 15, 10, 5, 2 et 1 sujet(s).

Pour ce qui est du nombre de threads, cela ne concerne que la compression, les mêmes tests que présentés au premier point de la liste ci-dessus ont été réalisés pour des fichiers contenants aussi différents nombres de sujets et en utilisant 1,2,3,4 et 8 threads.

2.3 Dataset

Le dataset utilisé est celui de la phase 3 du 1000 Genomes Project [1] téléchargeable à partir du lien `http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/?C=S;O=A`. Selon les chercheurs de l'UMons partenaires du projet GLORIA, ce type de fichier devraient ressembler à ceux qui seront utilisés dans le projet. Le projet 1000 Genomes Project rassemble 2504 sujets dont les génomes sont regroupés dans 24 fichiers VCF : un par chromosome. Les tests ont été réalisés sur les chromosomes 1 à 22. Comme mentionné précédemment, pour tester les performances en fonction du nombres de sujets dans les fichiers VCF, ces fichiers seront sous-échantillonnés pour ne garder que 2500, 2250, 2000, 1500, 1000, 900, 800, 750, 700, 600, 500, 250, 200, 150, 100, 50, 25, 15, 10, 5, 2 et 1 sujet(s) choisis aléatoirement.

3 Résultats et analyses

Les résultats sous formes de graphes seront directement commentés et discutés. Ils sont présentés dans l'ordre chronologique dans lequel ils ont été réalisés.

3.1 Nombre de sujets

Premièrement, l'influence du nombre de sujets à été étudié.

3.1.1 Premiers tests

Un premier groupe de tests ont été réalisés sur l'ensemble des chromosomes 1 à 22 pour des sous-échantillonnages de 2500, 2000, 1500, 1000, 750, 500, 250, 200, 150, 100, 50, 25, 15, 10, 5, 2 et 1 sujet(s). Sur les graphes, les résultats moyens sur tous les chromosomes sont représentés par la courbe noire.

Taux de compression

Sur les Figure 1 et 2, nous voyons que le taux de compression varie aussi bien avec le nombre de sujets qu'avec la taille du fichier d'input. Les deux sont bien sûr liés mais pas totalement car les chromosomes sont numérotés (principalement) selon leur longueur (chrom 1 le plus long, chrom 22 le plus court avec tout de même des variations entre les individus). Le chromosome 1 avec 1 sujet est donc plus lourd que le chromosome 22 avec 1 sujet.

Dans GSC, les données génotypiques du fichier sont découpées en bloc et compressées par bloc, chaque bloc contenant v variants consécutives. Si le nombre de sujets s est inférieur à 2^{13} (ce qui est toujours notre cas ici), v est fixé à s, (sinon, v est fixé à 2^{13}). Plus il y a de sujets, plus les blocs sont grands et la compression sera efficace, ce qui explique pourquoi le taux de compression augmente avec la taille de

sous-échantillonnage dans la Figure 1 (presque indépendamment du chromosome).

Nous voyons tout de même que, pour des nombres de sujets élevés, il y a une différence non négligeable de taux de compression entre les chromosomes. Si le taux de compression de chaque bloc était uniquement dépendant de sa taille, on devrait avoir une taux de compression identique pour un nombre de sujets donné peu importe la taille du fichier (donc peu importe le chromosome). Une des différences entre les chromosomes est la taille des headers. Mais, même s'ils sont plus grands pour certains chromosomes, ils n'influent, selon moi, pas pour autant si fort le taux de compression (ils occupent une très petite proportion de tout le fichier VCF). On observe que pour les petits nombres de sujets, notre hypothèse est respectée : sur la Figure 1, les lignes se superposent bien si on se rapprochent de la gauche et sur le Figure 2, pour les petites tailles de fichier, on voit que les points sur chaque courbe forme une droite de taux de compression constant qui corresponde en fait à chaque fois à un même nombre de sujets. C'est seulement lorsque le nombre de sujets augmentent fort que des écarts apparaissent et s'accentuent. Cela pourrait venir du fait que les derniers blocs de chaque fichier ne sont jamais parfaitement remplis (on remplit le dernier bloc avec les variants restant mais il y en a quasiment toujours moins que s). Ce dernier bloc sera donc compressé de manière moins optimale que les autres blocs et influera le taux de compression du VCF au global. Plus le fichier VCF est petit, plus le taux de compression de ce dernier bloc aura une influence importante sur le taux de compression global. Alors que dans des très grand VCF, le dernier bloc n'aura que peu d'influence par rapport au grand nombre de blocs complets compressés optimalement. Cela pourrait expliquer que les grands chromosomes présentent des meilleurs taux de compression.

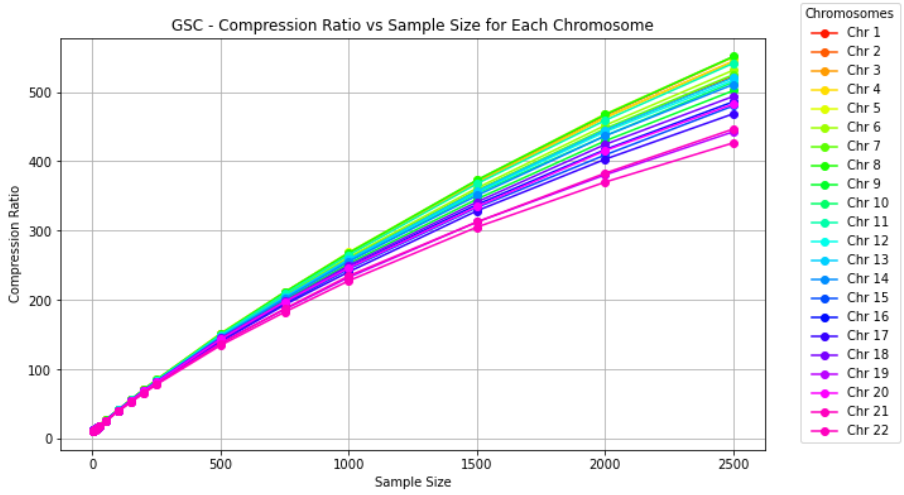

Figure 1 – Taux de compression en fonction de la taille de sous-échantillonnage

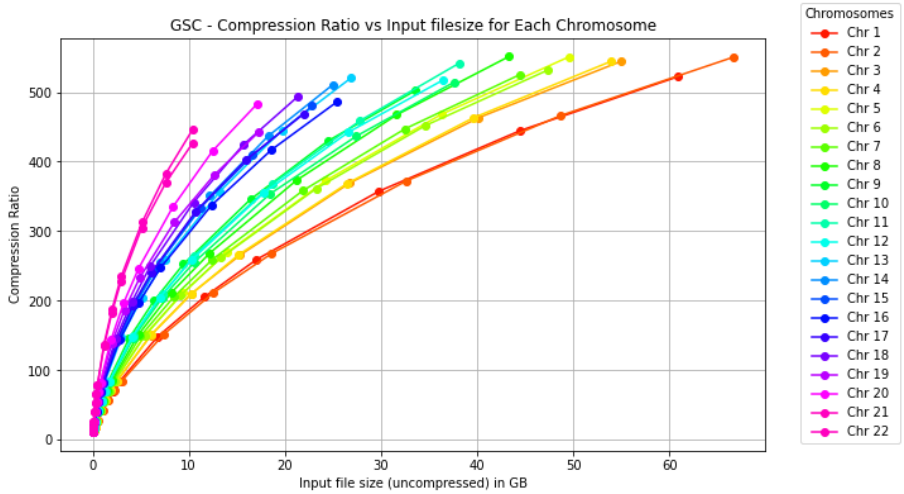

Figure 2 – Taux de compression en fonction de la taille du fichier d'input

Vitesse et RAM en compression

Les métriques, temps et RAM, ont été divisées par la taille du fichier d'input permettant de normaliser les valeurs obtenues et de plus facilement comparer les tests réalisés sur les différents chromosomes. Dans le cadre du projet GLORIA, on aura un certain nombre de sujets à traiter, la question est ici de savoir si c'est mieux d'avoir x fichiers de 1 sujet ou 1 fichier de x sujet. Compresser un fichier de 1 seul sujet sera toujours plus rapide que de compresser un fichier de x sujets, mais il faut tenir compte du fait que si on utilise des fichiers de 1 sujet, il faudra en compresser x au lieu de tout compresser en une fois à partir du fichier de x sujets.

Plus les blocs sont grands, plus leur vitesse de compression (en sec/MB) est élevée. C'est effectivement ce que nous observons sur la Figure 3, où le temps de compression présente une diminution très rapide entre 1 et 100 sujets, qui est toujours présente mais moins forte jusqu'à 500-1000 sujets pour finir par être quasi constant après 1000-1500 sujets.

Les observations sont relativement semblables pour la RAM avec un diminution qui devient rapidement moins forte au fur et à mesure que le nombre de sujets augmente (voir Figure 4).

Note : Je pense avoir un problème d'unité que je ne comprends pas très bien. Pour la RAM, cela me semble totalement faux que l'algorithme de compression ait un ratio RAM/MB de fichier si élevé pour des faibles nombres de sujets. Cela me semble énorme de voir que l'algorithme à besoin d'environ 4000 MB de RAM pour compresser 1 MB de VCF par exemple.

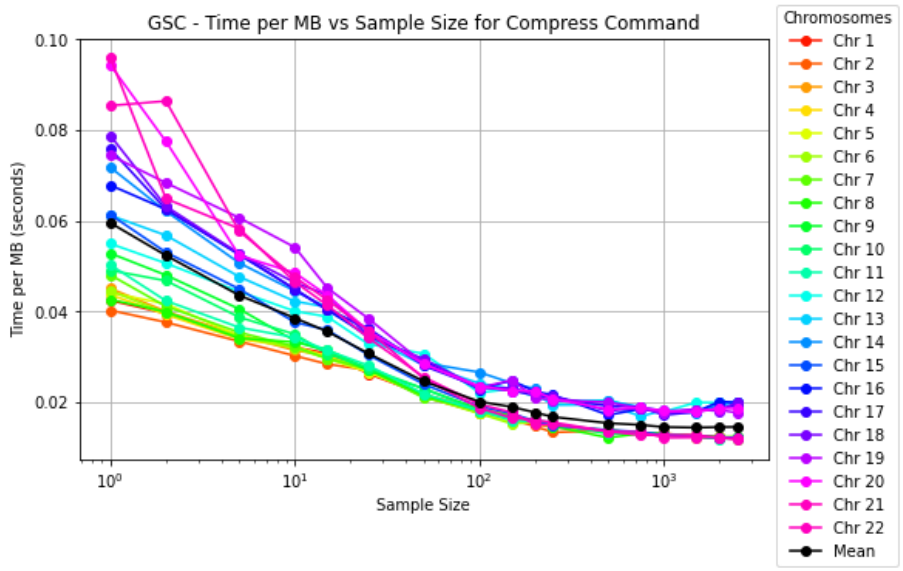

Figure 3 – Temps normalisé [sec/MB] de compression en fonction de la taille de sous-échantillonnage

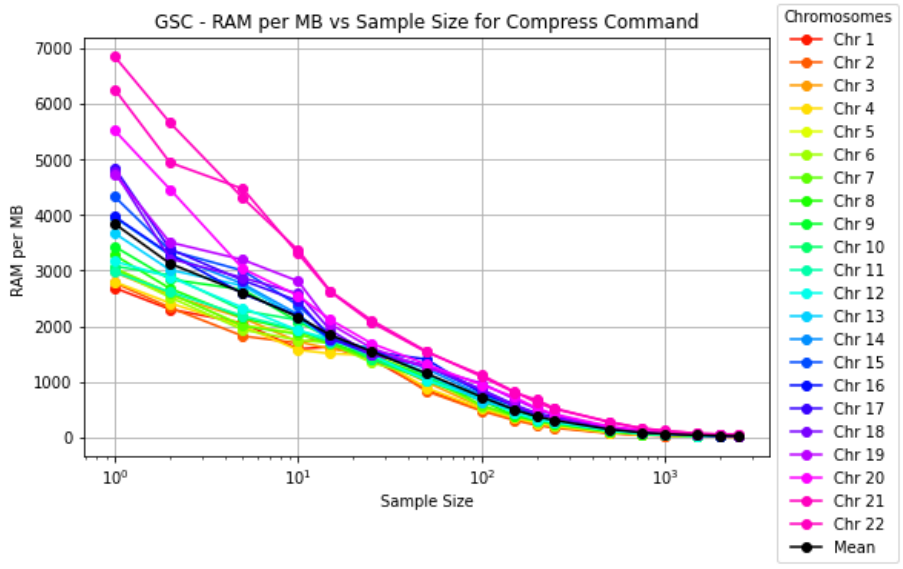

Figure 4 – RAM normalisée [MB/MB] en compression en fonction de la taille de sous-échantillonnage

Vitesse et RAM en décompression complète

Pour ces tests de décompression, nous considérons la décompression complète de tous les sujets du fichier pour retrouver le fichier VCF initial. L'analyse de la Figure 5 permet de montrer que la vitesse est constante (voir légèrement descendante) de 1 à 10 sujets, et augmentent ensuite linéairement (graphe en échelle logarithmique) avec le nombre de sujets.

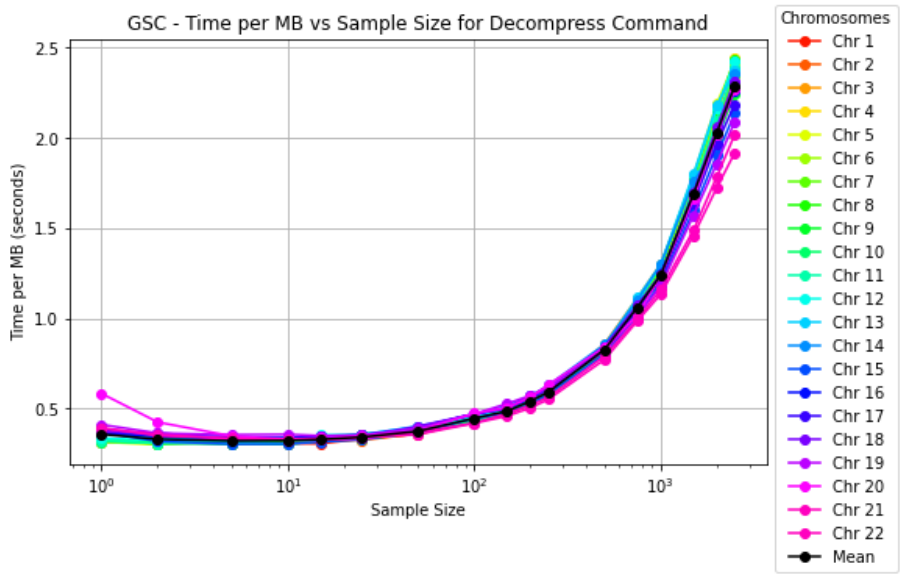

Figure 5 – Temps normalisé [sec/MB] de décompression en fonction de la taille de sous-échantillonnage

Pour ce qui est de la RAM, voir Figure 6, les résultats sont plus éparpillés selon les chromosomes, mais nous observons une diminution de la RAM avec le nombre de sujets, très rapide pour les nombres de sujets faibles et qui tend vers une constante lorsque le nombre de sujets augmente. Nous pouvons observer une diminution assez marquée entre 500 et 750 sujets et entre 2000 et 2500 sujets. Je ne fais que le mentionner ici mais nous en reparlerons dans le suite de ce rapport.

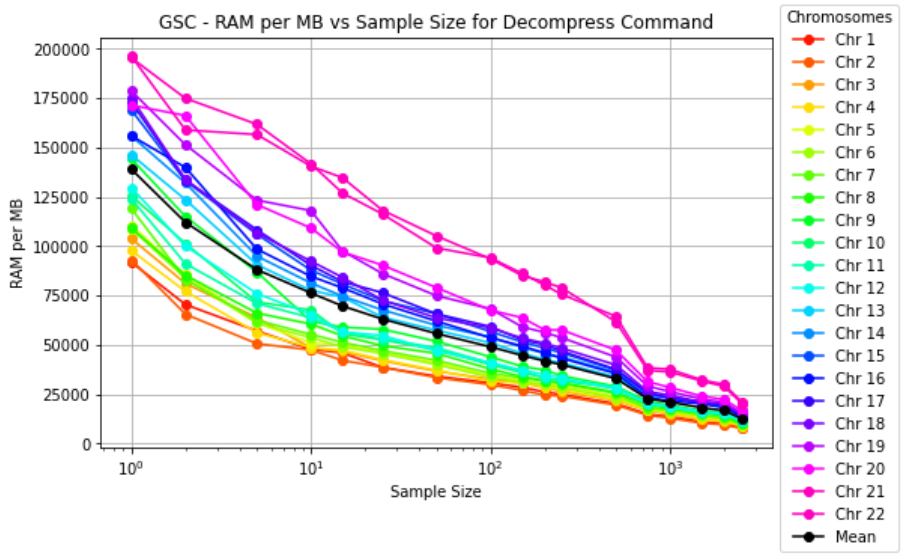

Figure 6 – RAM normalisée [MB/MB] en décompression en fonction de la taille de sous-échantillonnage

Vitesse et RAM de random access complet

Pour rappel, le random access ici se fait sur l'ensemble des sujets. Que ce soit pour la vitesse (Figure 7) ou pour le RAM (Figure 8), nous observons une diminution progressive de la métrique avec un nombre de sujets croissant et nous remarquons toujours cette brusque diminution entre 500 et 700 sujets et entre 2000 et 2500 sujets.

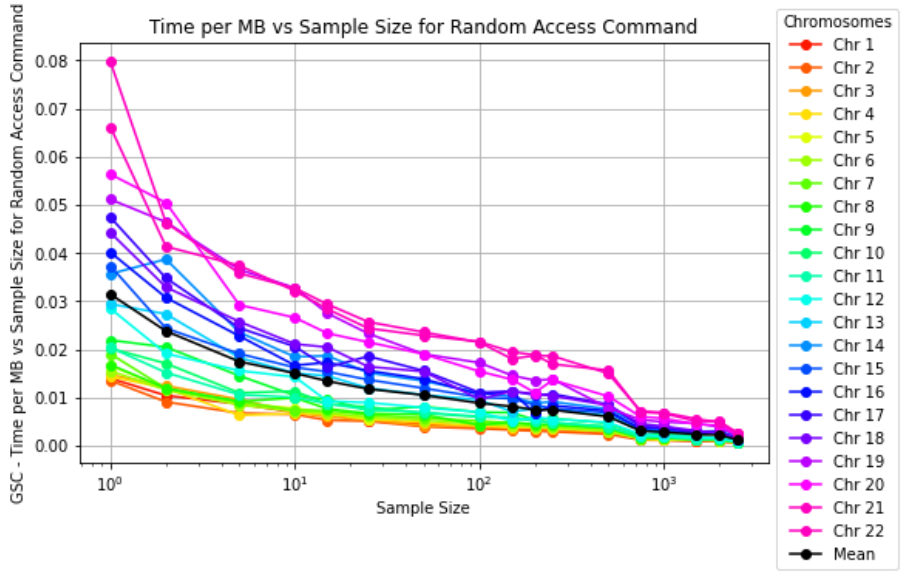

Figure 7 – Temps normalisé [sec/MB] de random access en fonction de la taille de sous-échantillonnage

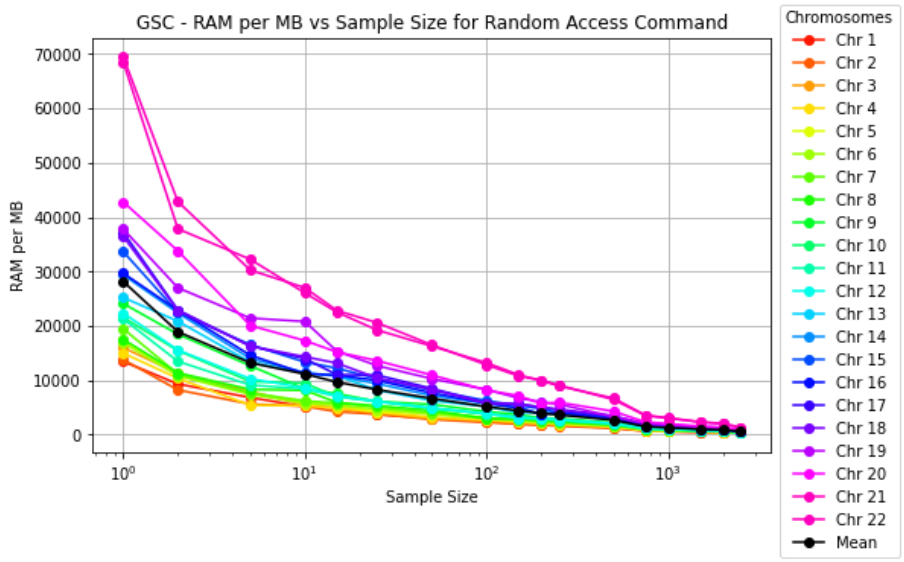

Figure 8 – RAM normalisée [MB/MB] en random access en fonction de la taille de sous-échantillonnage

$\Longrightarrow$ Pourquoi ces variations brusques ?

À la fin de ces tests, une grosse question reste : pourquoi observons nous des brusques variations de performances entre 500 et 750 et entre 2000 et 2500 sujets ? Et pouvons-nous réduire ces intervalles pour savoir avec plus de précisions le nombre de sujets à partir duquel se produisent ces brusques variations ? À la lecture de l'article de GSC, nous lisons que :

1. Les blocs ont une taille de v, définie comme suit : si le nombre de sujets s est inférieur à 2^{13} (ce qui est toujours notre cas ici), v est fixé à s, (sinon, v est fixé à 2^{13}).
2. « To improve the compression ratio while also maintain query speed, the genotype blocks are further merged into chunks. We adopt a chunk size of $l = 65536$ variants. The chunks are compressed with the general-purpose compressor BSC.»

Avec ces infos je n'ai pas réussi à expliquer les brusques changements et je n'ai pas relevé d'autres éléments dans l'article qui me permettraient de comprendre. Dans les seconds tests, les pas de sous-échantillonnage ont été modifiés pour définir de manière plus précise l'intervalle de nombre de sujets dans lequel se produisent ces brusques variations.

3.1.2 Seconds tests

Une deuxième groupe de tests a été conduit pour 2 raisons :

1. Mieux définir l'intervalle de nombre de sujets dans lequel se produisent les brusques variations de performances. Les sous-échantillonnages suivant ont

été ajoutés à ceux déjà testés pendant les premiers tests : 600, 700, 800, 900 et 2250.

2. Ajouter les tests de décompression d'un seul sujet et de random access d'un seul sujet.

Étant donné que cela augmente le nombre de tests à réaliser, seuls les chromosomes 1, 10, 15 et 22 ont été testés. Les graphes sur le taux de compression n'ont pas fondamentalement changés par rapport aux premiers tests donc ils ne sont pas refaits.

Vitesse et RAM en compression

Pour la vitesse de compression, la Figure 9 est identique à la Figure 3, les conclusions sont donc les mêmes. Pour la RAM de compression, la Figure 10 est aussi identique à la Figure 4, les conclusions sont donc les mêmes également.

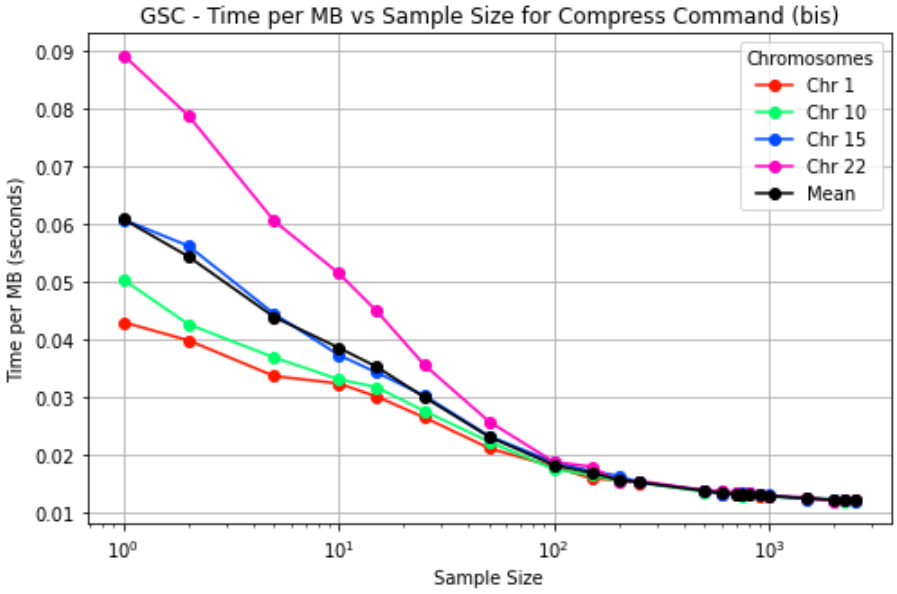

Figure 9 – Temps normalisé [sec/MB] de compression en fonction de la taille de sous-échantillonnage

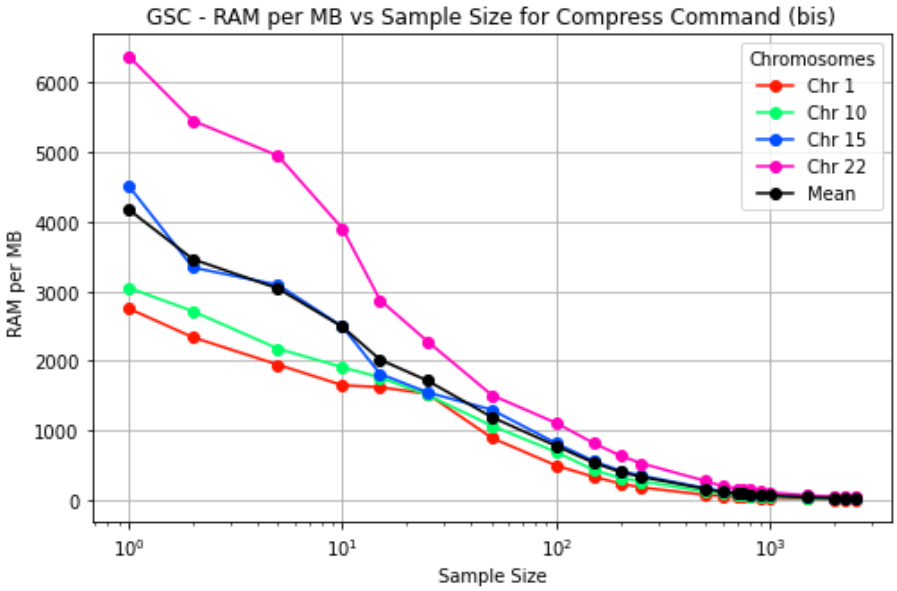

Figure 10 – RAM normalisée [MB/MB] en compression en fonction de la taille de sous-échantillonnage

Vitesse et RAM en décompression complète

Pour le décompression complète, nous re-obtenons aussi des résultats semblables en vitesse (Figure 11 et 5) et en RAM (Figure 12 et 6). Dans la Figure 12, nous pouvons cependant mieux évaluer l'endroit de la diminution brusque. Nous voyons qu'elle se produit entre 500 et 600 sujets et ensuite entre 2000 et 2250 sujets.

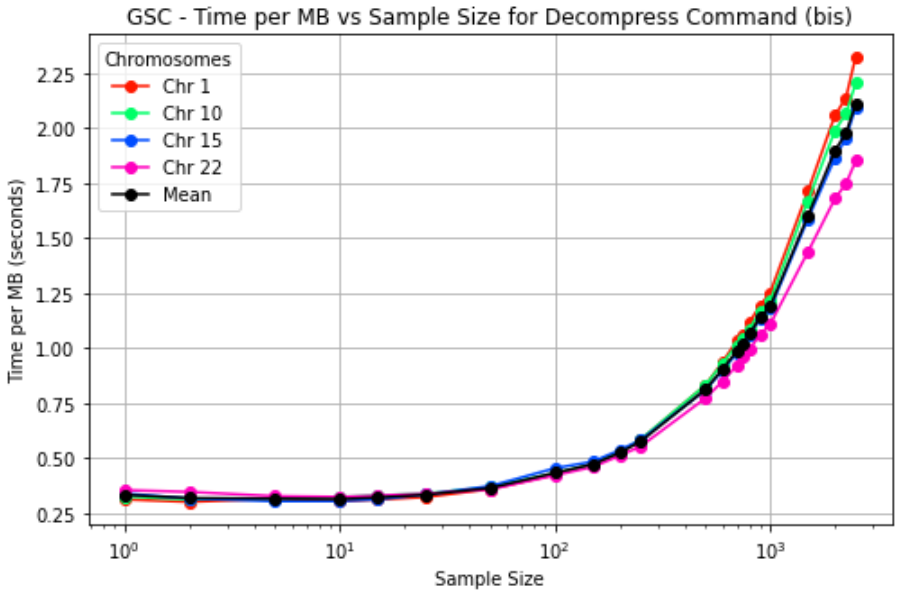

Figure 11 – Temps normalisé [sec/MB] de décompression en fonction de la taille de sous-échantillonnage

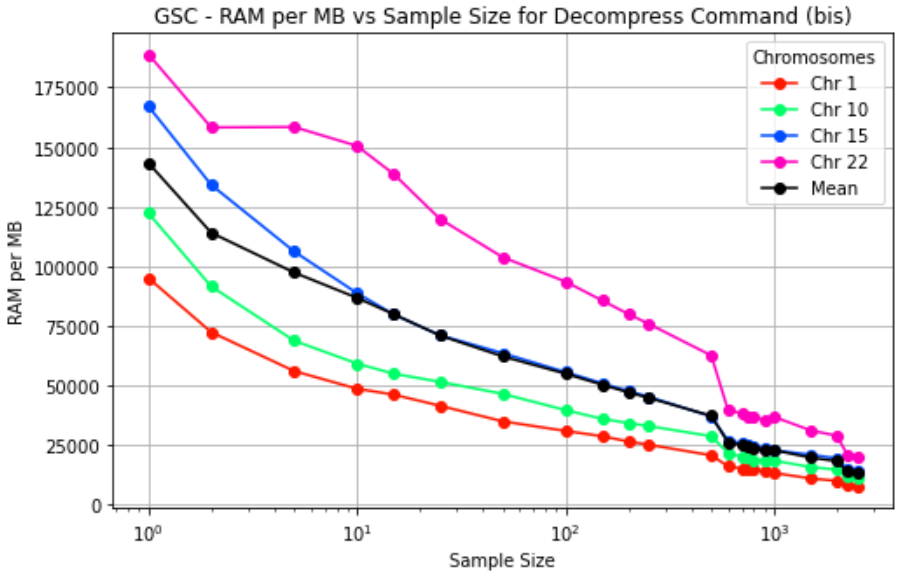

Figure 12 – RAM normalisée [MB/MB] en décompression en fonction de la taille de sous-échantillonnage

Vitesse et RAM de random access complet

Pour le random access complet, les observations sont également identiques. Nous re-obtenons des résultats semblables en vitesse (Figure 13 et 7) et en RAM (Figure 14 et 8). Dans la Figure 14, nous pouvons également mieux évaluer l'endroit de la diminution brusque (entre 500 et 600 sujets et ensuite entre 2000 et 2250 sujets).

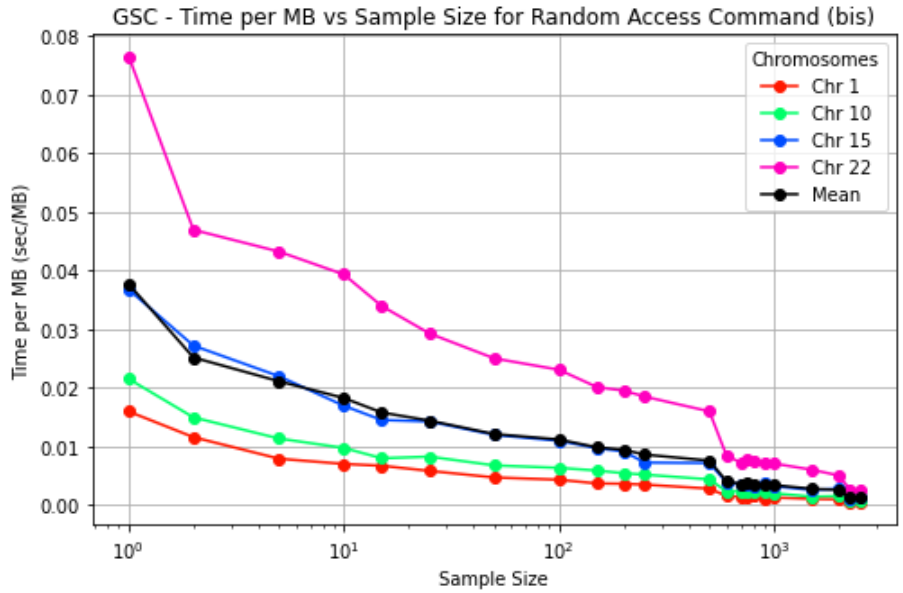

Figure 13 – Temps normalisé [sec/MB] de random access en fonction de la taille de sous-échantillonnage

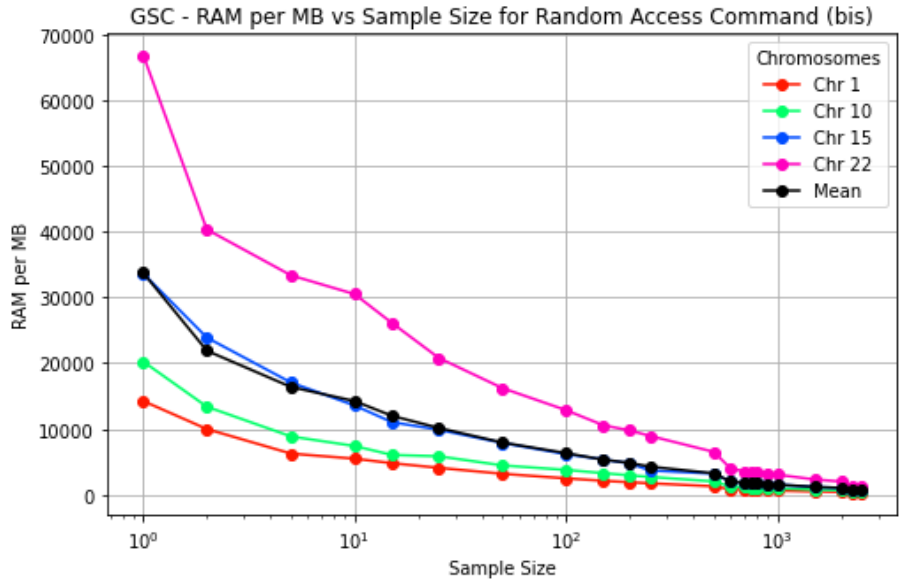

Figure 14 – RAM normalisée [MB/MB] en random access en fonction de la taille de sous-échantillonnage

Vitesse et RAM de décompression d'un seul sujet

Pour la décompression et le random access sur un seul sujet, nous ne normalisons pas les résultats en fonction de la taille du fichier d'input. En effet, dans le cadre du projet GLORIA, lorsque nous voudrons accéder à une ou des données pour un seul sujet, nous ne devrons le faire que pour ce sujet-là peu importe le nombre d'autres

sujets dans le fichier, donc peu importe la taille du fichier d'input.

Sur la Figure 15, nous voyons que le temps de décompression augmente linéairement avec le nombre de sujets dans le fichier compressé, avec un maximum d'environ 18 secondes pour le chromosome 1 (le plus long) avec 2500 sujets, ce qui reste relativement rapide et environ 4 secondes pour le chromosome 22 (le plus petit) avec 2500 sujets également.

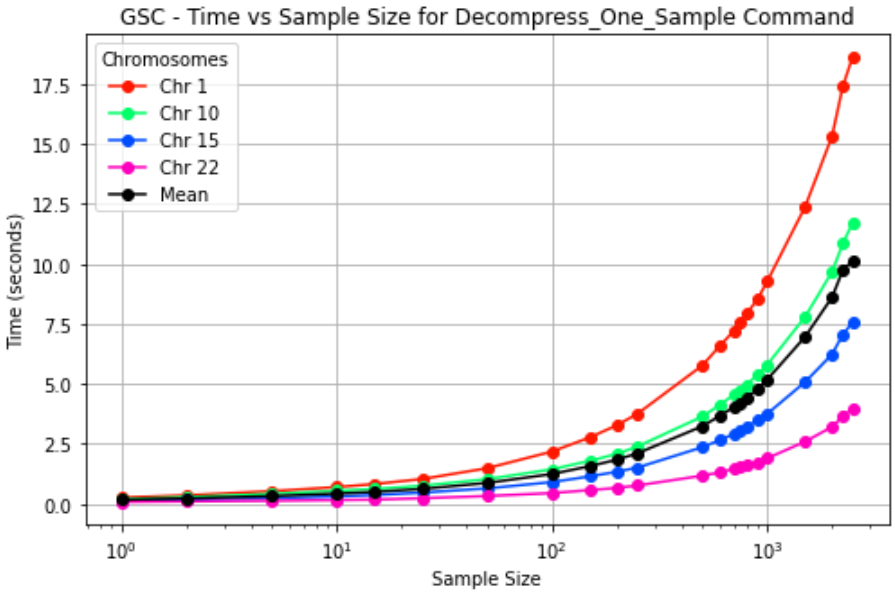

Figure 15 – Temps [sec] de décompression d'un seul sujet en fonction de la taille de sous-échantillonnage

Pour comparer la vitesse de décompression de tous les sujets (complète) par rapport à la décompression d'un seul sujet, nous pouvons observer les vitesses non normalisées sur la Figure 16. Nous voyons ainsi que GSC permet un gain de temps important lorsque nous ne décompressons qu'un seul sujet surtout lorsque le nombre de sujets augmente puisque pour des petits nombres de sujets les différences sont pas très marquées.

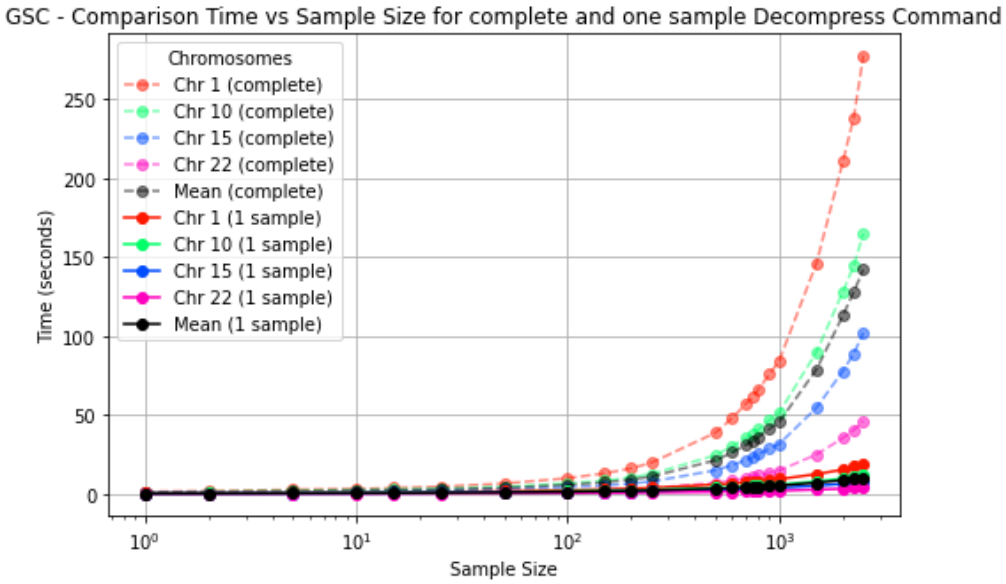

Figure 16 – Comparaison du Temps [sec] de décompression complète et d'un seul sujet en fonction de la taille de sous-échantillonnage

Sur la Figure 17, nous voyons que la RAM évolue de manière très particulière avec le nombre de sujets (et même en fonction du chromosome pour des plus petits nombres de sujets). De 1 à 500 sujets, le RAM augmente avec une pente plus ou moins importante en fonction de la longueur du chromosome. Ensuite, nous observons une diminution très marquée à 600 sujets, au point que la RAM à 600 sujets est inférieure à la RAM pour des fichiers plus petits en nombre de sujets. Entre 600 et 2000 sujets, la RAM augmente linéairement avec le nombre de sujets et puis une diminution entre 2000 et 2250 sujets. La diminution entre 2000 et 2250 est beaucoup moins forte que la diminution entre 500 et 600.

Pour comparer avec une décompression complète, nous pouvons observer la Figure 18. Nous voyons que la RAM nécessaire pour une décompression d'un seul sujet est bien moins grande et que à cette échelle, en dehors des diminutions brusques, la RAM nécessaire est quasiment constante. De manière plus détaillée, le RAM suit environ la même évolution avec une augmentation proportionnelle au nombre de sujets sauf entre 500 et 600 et entre 2000 et 2250 sujets où nous observons des diminutions encore plus forte en décompression complète qu'en décompression d'un seul sujet.

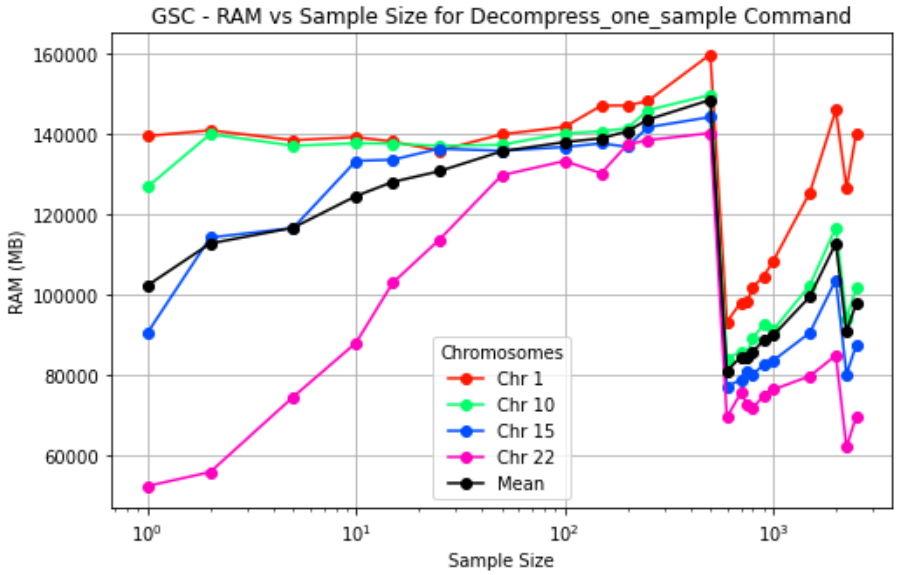

Figure 17 – RAM [MB] en décompression d'un seul sujet en fonction de la taille de sous-échantillonnage

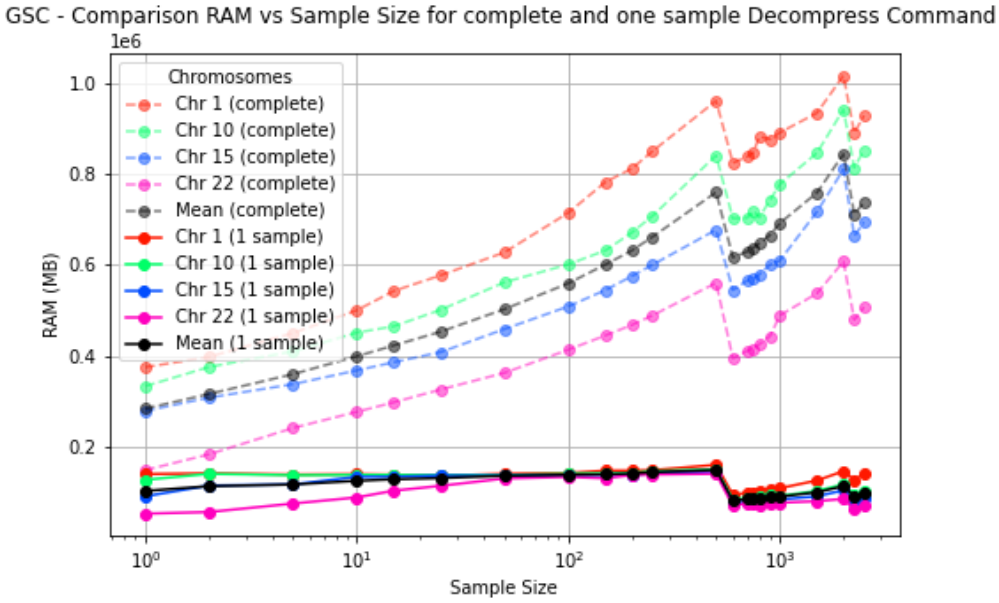

Figure 18 – Comparaison de la RAM [MB] en décompression complète et d'un seul sujet en fonction de la taille de sous-échantillonnage

Vitesse et RAM de random access sur un seul sujet

Sur la Figure 21, nous observons que le temps nécessaire pour un random access sur un seul sujet augmente exponentiellement avec le nombre de sujet dans le fichier compressé jusque 500 sujets. À 600 sujets, il y a une diminution importante. Le temps augmente à nouveau entre 600 et 2000 et il y a une deuxième diminution

brusque à 2250 sujets. Peu importe le nombre de sujets, les vitesses de random access restent tout de même très rapides, les temps varient autour de 1/10 de secondes ce qui est très proche de l'immédiat.

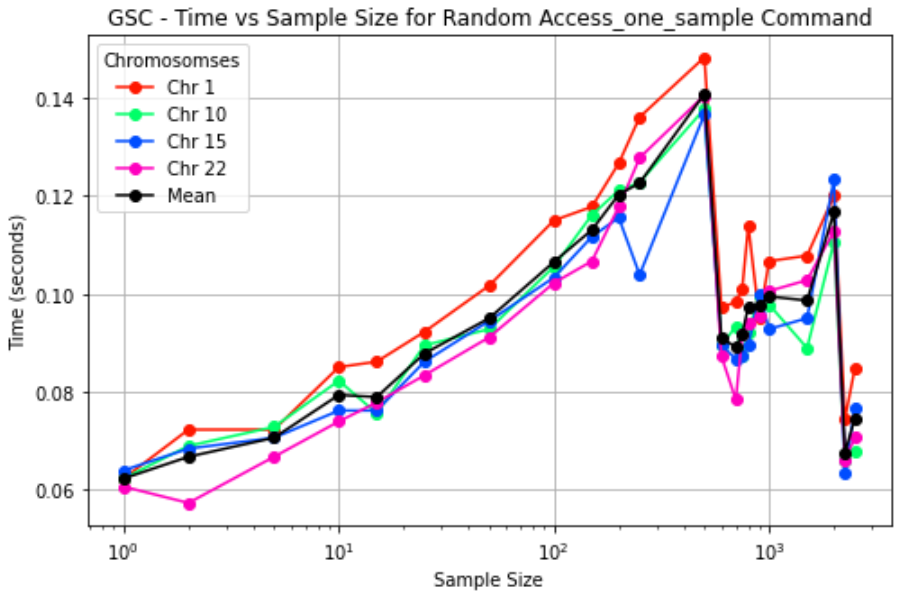

Figure 19 – Temps [sec] de random access sur un seul sujet en fonction de la taille de sous-échantillonnage

La Figure 20 nous permet de constater que les temps de random access pour tous les sujets ou un seul sujet sont quasiment identiques. Le random access complet va même quelques centièmes de secondes plus vite que le random access pour un seul sujet. (Peut être que la commande de random access pour un seul sujet décompresse en réalité la (les) position(s) demandée(s) pour tous les sujets du fichier puis filtre pour ne garder que le(s) sujet(s) demandé(s). Cette hypothèse n'a pas été vérifiée.)

Pour le RAM, nous voyons, sur la Figure 21, qu'elle est quasiment constante entre 1 et 500 sujets et entre 600 et 2000 sujets et diminue fortement à 600 et 2250 sujets. Il est étonnant de voir que de manière générale, plus il y a de sujet, moins le RAM nécessaire est importante. Je n'ai pas d'explication à cela. Le graphe de la comparaison de la RAM en random access complet et d'un seul sujet n'est pas repris ici car il se superpose quasi parfaitement, il n'y a aucune différence significative.

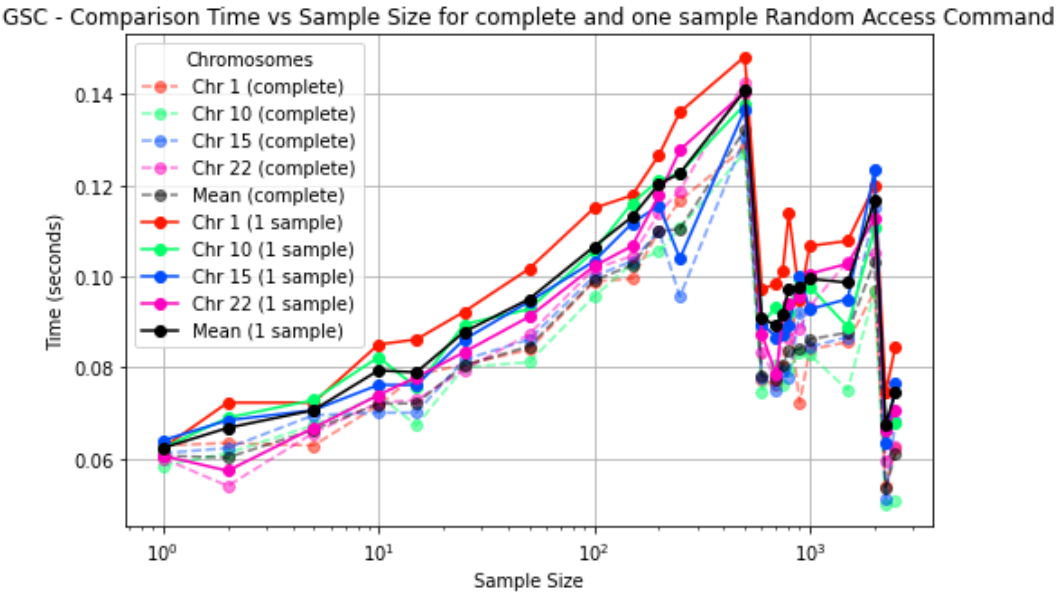

Figure 20 – Comparaison du Temps [sec] de random access complet et sur un seul sujet en fonction de la taille de sous-échantillonnage

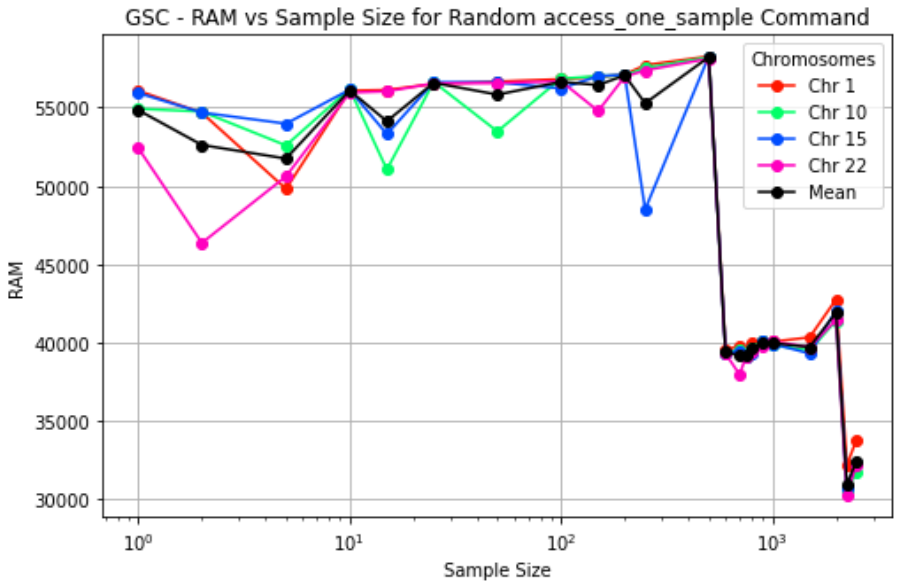

Figure 21 – RAM [MB] en random access sur un seul sujet en fonction de la taille de sous-échantillonnage

$\Longrightarrow$ Variations brusques

Les intervalles dans lesquels se trouvent les variations brusques se sont précisés (entre 500 et 600 et entre 2000 et 2250). Cela ne me permet cependant toujours pas de comprendre pourquoi ces variations brusques surviennent et pourquoi à ces moments-là.

3.2 Nombre de threads

Par la suite, l'influence du nombre de threads à été étudiée. Ce paramètre n'a été étudié que sur la commande de compression car seule la compression peut être multi-threadée. Ce n'est pas le cas de la décompression ou du random access. Les tests ont été réalisé sur les chromosomes 1,10,15 et 22. Les résultats présentés dans les graphes ci-dessous correspondent à des moyennes sur les différents chromosomes.

Sur les Figures 22 et 23, nous pouvons étudier l'effet du nombre de threads sur, respectivement, le temps par MB et la RAM par MB pour des fichiers contenant différents nombres de sujets. Premièrement, sur les deux figures, nous voyons que pour un grand nombre de sujets, le nombre de threads n'influe pas ou très très peu le temps et la RAM. Pour des plus petits nombres de sujets, en terme de vitesse, on observe un minimum pour 4 threads et en terme de RAM, on observe un minimum pour 1 thread. Nous voyons cependant que les différences ne sont pas très grandes selon le nombre de threads et qu'il est plus intéressant de trouver un nombre de sujets par fichier optimal pour améliorer significativement les performances.

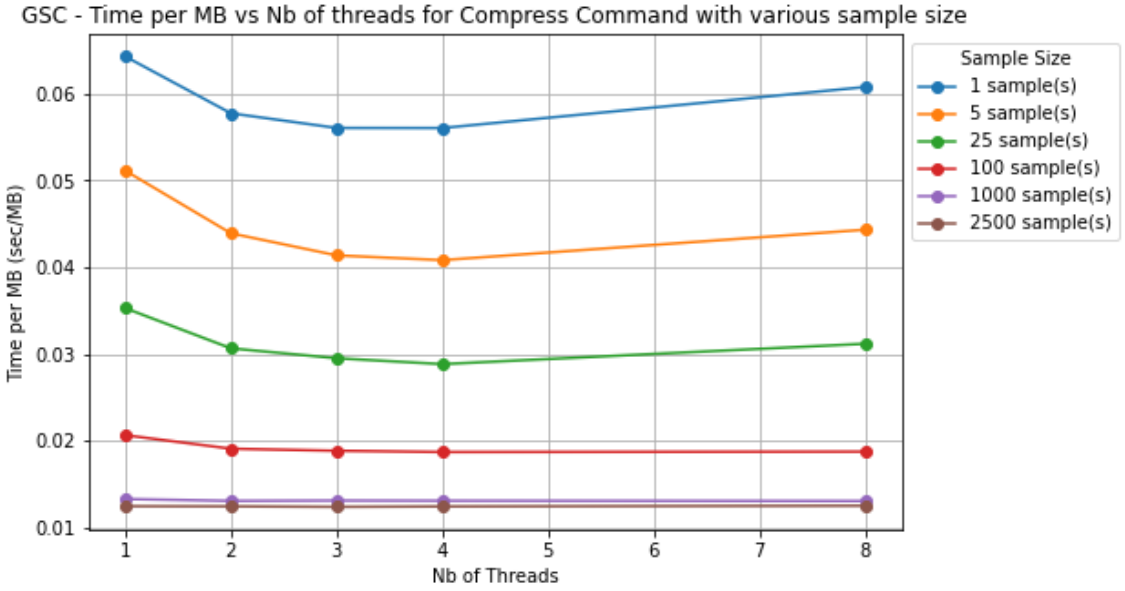

Figure 22 – Temps normalisé [sec/MB] de compression en fonction du nombre de threads

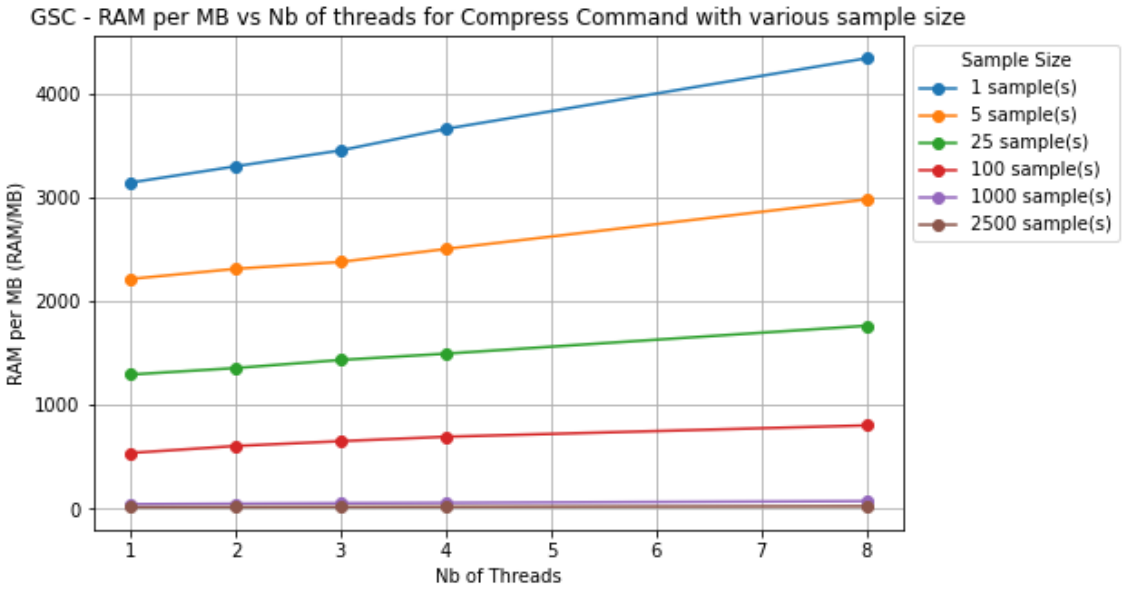

Figure 23 – RAM normalisé [MB/MB] de compression en fonction du nombre de threads

Références

[1] *1000 Genomes | A Deep Catalog of Human Genetic Variation*. URL : `https://www.internationalgenome.org/` (visité le 07/05/2024).

[2] Petr Danecek et al. « The variant call format and VCFtools ». In : *Bioinformatics* 27.15 (1er août 2011), p. 2156-2158. ISSN : 1367-4803. DOI : `10.1093/bioinformatics/btr330`. URL : `https://doi.org/10.1093/bioinformatics/btr330` (visité le 06/05/2024).

[3] Xiaolong Luo et al. « GSC : efficient lossless compression of VCF files with fast query ». In : *GigaScience* 13 (1er jan. 2024), giae046. ISSN : 2047-217X. DOI : `10.1093/gigascience/giae046`. URL : `https://doi.org/10.1093/gigascience/giae046` (visité le 25/10/2024).

A Benchmarking de SVC

Taux de compression

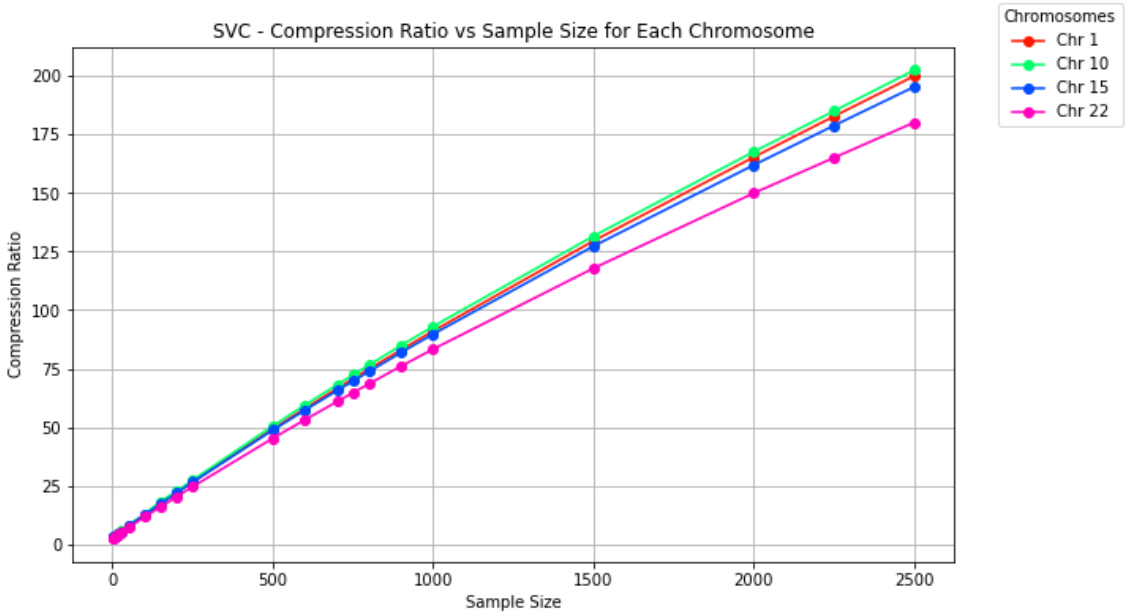

Figure 24 – Taux de compression en fonction de la taille de sous-échantillonnage

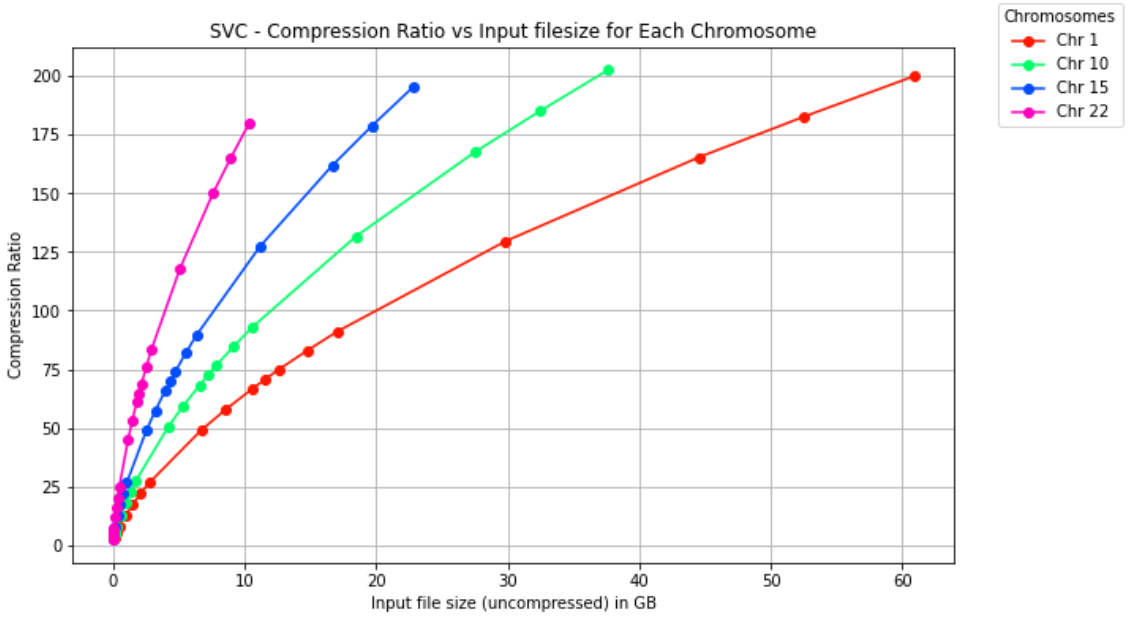

Figure 25 – Taux de compression en fonction de la taille du fichier d'input

Vitesse et RAM en compression

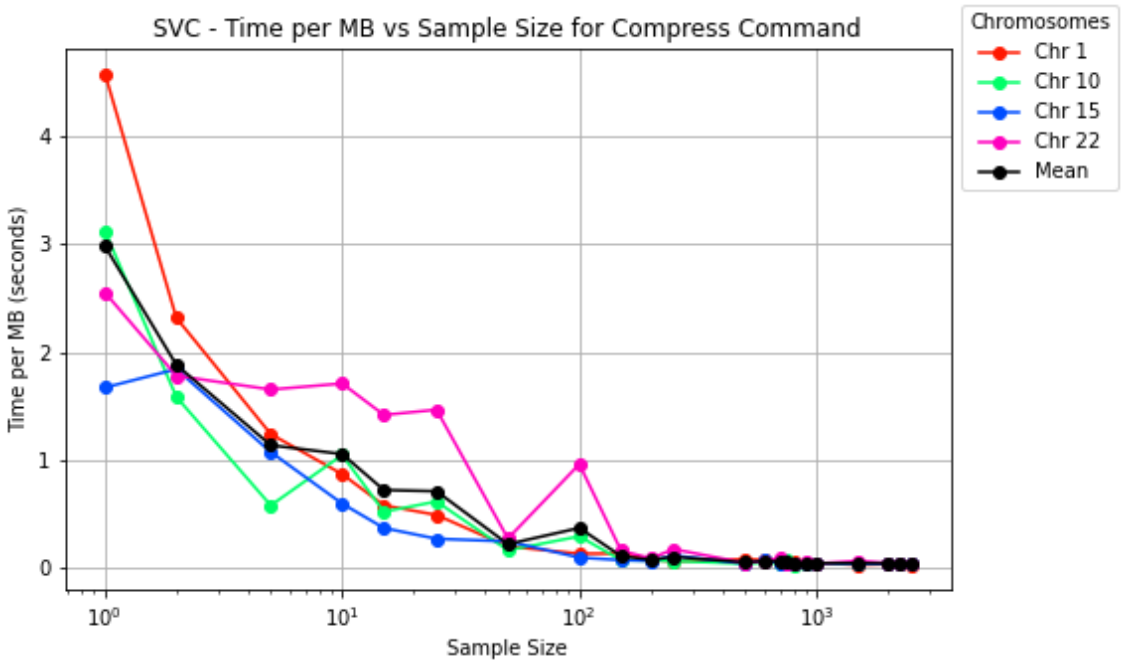

Figure 26 – Temps normalisé [sec/MB] de compression en fonction de la taille de sous-échantillonnage

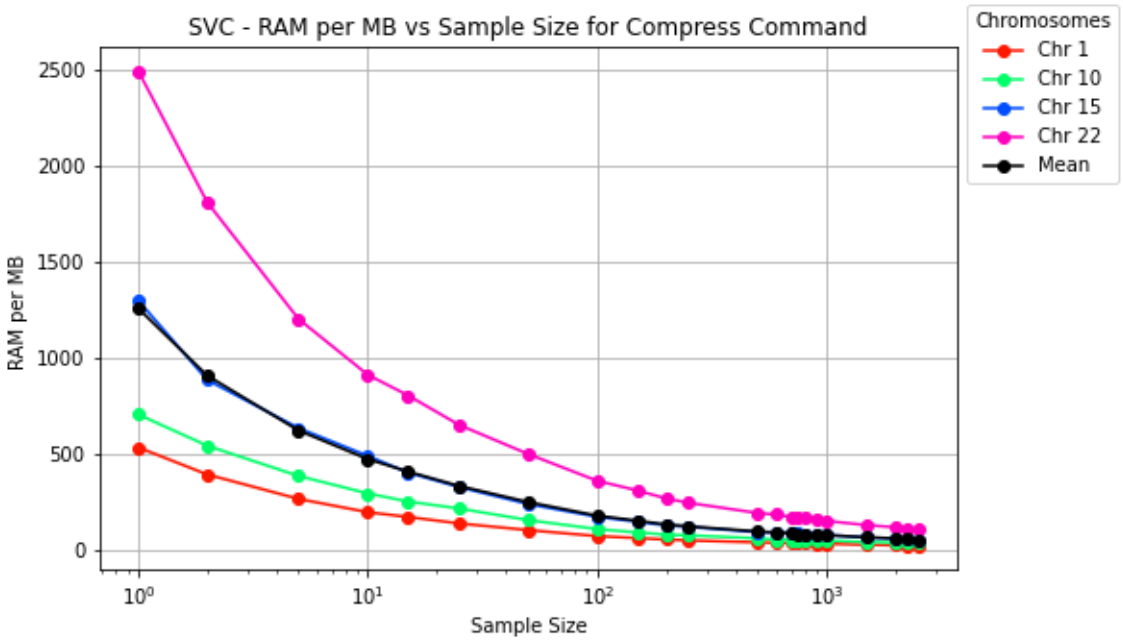

Figure 27 – RAM normalisée [MB/MB] en compression en fonction de la taille de sous-échantillonnage

Vitesse et RAM en décompression complète

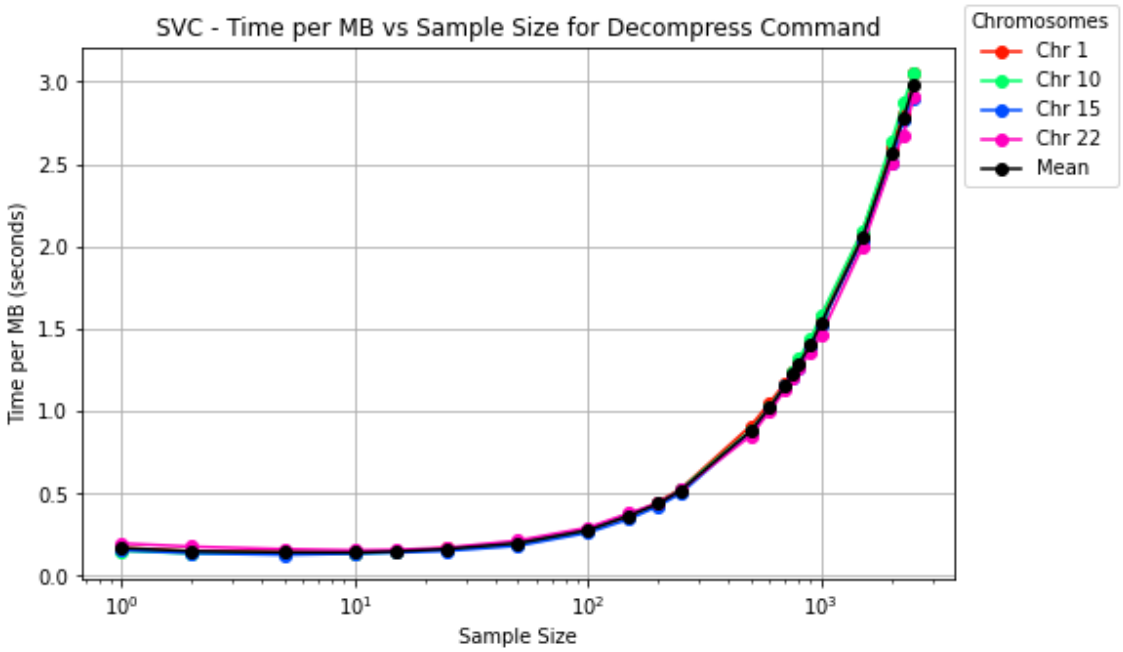

Figure 28 – Temps normalisé [sec/MB] de décompression en fonction de la taille de sous-échantillonnage

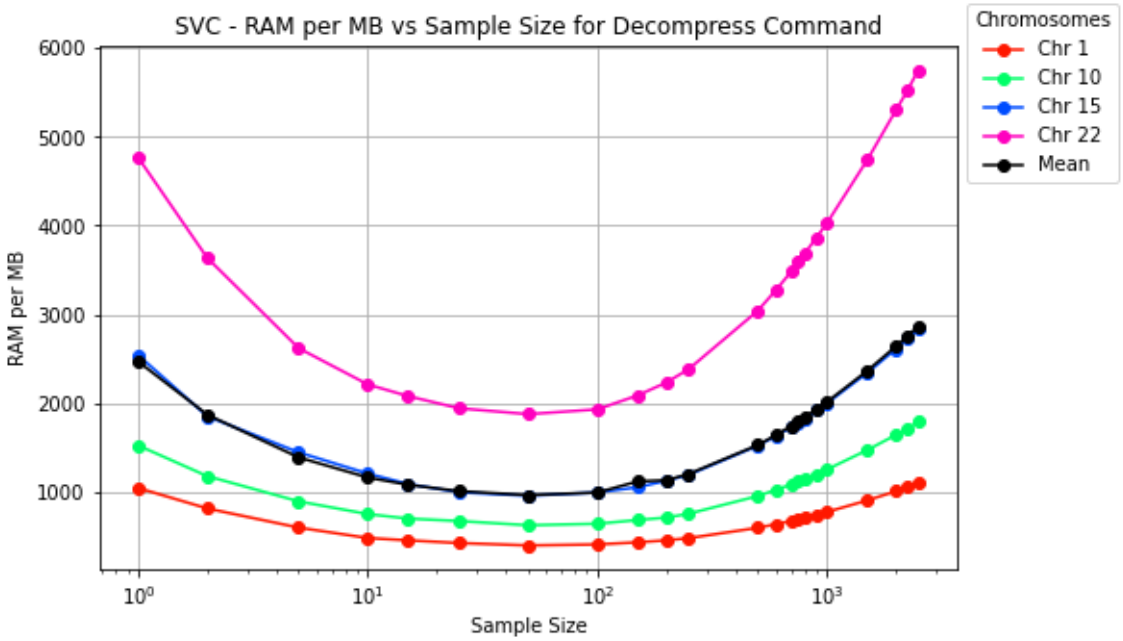

Figure 29 – RAM normalisée [MB/MB] en décompression en fonction de la taille de sous-échantillonnage

Vitesse et RAM en random access complet

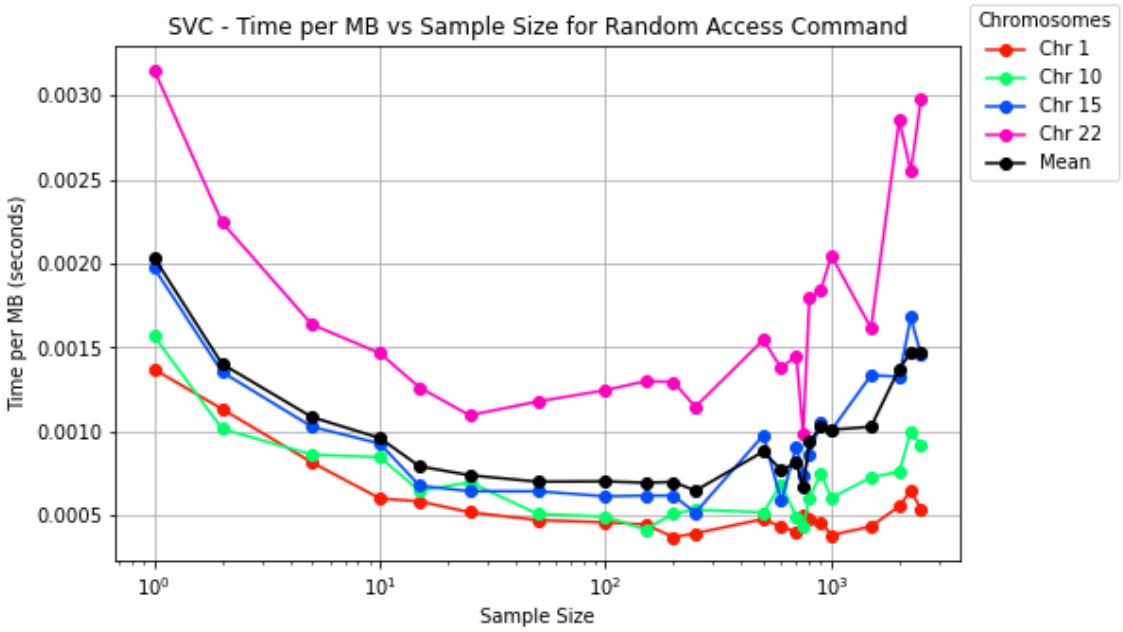

Figure 30 – Temps normalisé [sec/MB] de random access en fonction de la taille de sous-échantillonnage

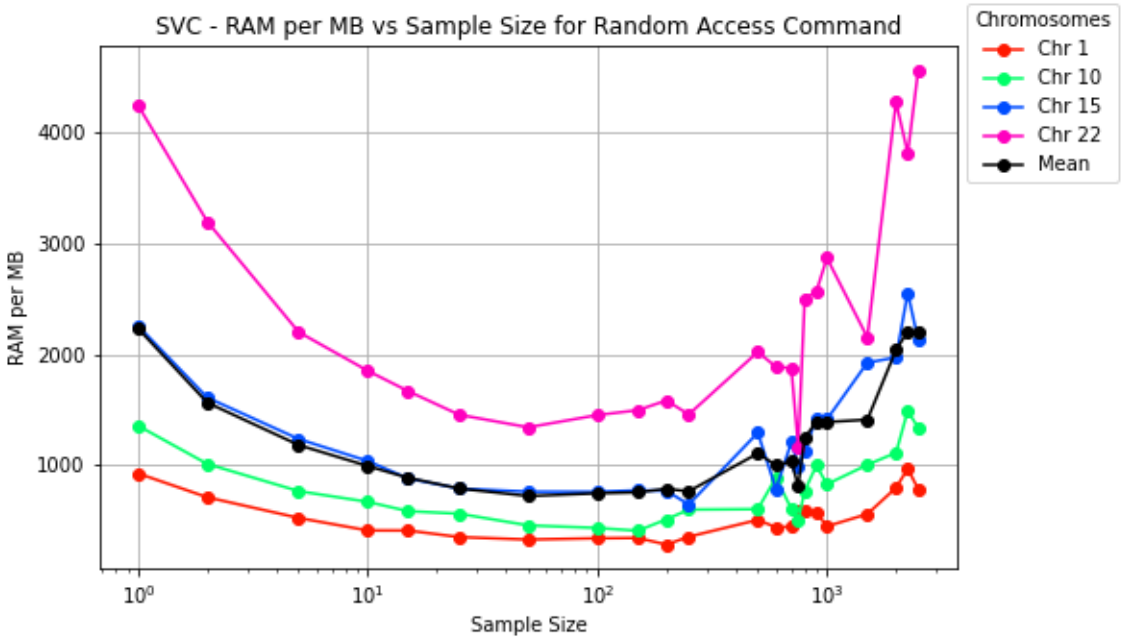

Figure 31 – RAM normalisée [MB/MB] en random access en fonction de la taille de sous-échantillonnage