FastMSECT Algorithm: A Many-Core Fast Scalable Method for Massive String Exhaustive Comparison Technology such as GMSECT

*Abhishek Narain Singh**

BIRAC E-Yuva Centre, Bhopal, Madhya Pradesh, India

Abstract. FastMSECT is a divide-and-conquer based algorithm to deal with large strings such as DNA sequences of Genome size in many-core processors or co-processors. FastMSECT uses the power of many core processors to adequately match the L3 cache sizes which is uniformly available to all the CPU cores as shared memory and in doing so it expedites the process of string comparison. Sequence comparison algorithms such as Smith Waterman usually have quadratic or exponential time complexity, which FastMSECT drops the time complexity to linear growth with longer sequence lengths. FastMSECT algorithm is implemented in the form of a tool called GMSECT for Genome-wide Massive Sequence Exhaustive Comparison Tool, which is made available via web-service of GenomeKlinik. GenomeKlinik is a versatile internet-based application where biomedical scientists and clinical practitioners can feed in their genomic sequence of interest and retrieve genomic variation as compared to a reference genome such as HuRef for human genome, and also get downstream annotation for the variation such as disease phenotype and features. It internally makes use of GMSECT, SQL and other commercially freely available software tools. The tool can also be used for large scale population data study and novel variation detection, and can be downstream channeled into biomarker discovery. GenomeKlinik although focuses on genomic sequences such as the Next Generation Sequencing (NGS), it is also useful for other kinds of data such as epigenome and other omics. GenomeKlinik serves as a one stop bioinformatics platform where the customer can create recommended pipeline for data processing and get meaningful information such as for the patient whose Exome or Genome or Epigenome data has been obtained. The web application server is a Freemium mode of offering limited services for free.

1 Introduction

The FastMSECT algorithm was initially developed when working on Sharcnet supercomputing facility [23] in Ontario, Canada in 2007, although the initial poster and abstract was published in the year 2011 as a variant tool called GenomeBreak. An accepted method for obtaining the first indication of functional similarity is sequence homology. A relevant paper on GMSECT was published as a bioRxiv preprint in early 2021, and the abstract was included in the BIOCOMP 2020 abstract book [1]. The likelihood of finding closer matches increases as the DNA sequence database expands upon. Nonetheless, as the size of the DNA repository grows, so does the need for managing the sequences quickly and effectively [2]. Although the majority of pairwise alignment tools currently in use are extremely quick when compared to the typical dynamic programming approach, they are neither quick or efficient when handling large sequences, which leads to memory issues address violation together with additional computing difficulties. DMWAS was one such application developed that made use of AI (artificial intelligence) to extract

---

*Corresponding author: abi@abitoeq.net

biomarkers and the pipeline before applying any machine learning algorithms would first involve mapping and aligning sequences to find the variants, where FastMSECT algorithm finds application. Tools such as GenomeBreak which uses concepts similar to GMSECT were used for family of four analysis to make new discoveries such as mitochondrial inheritance from father as discovered in 2012 and also published later in 2018 as a journal paper with more detailed explanations and mathematics , and discovery of Structural Variants having stronger impact on phenotype and for customized biomedical informatics blasand relate the genomic variants to end clinical phenotype.

2 Time complexity for aligners

The time computational complexity of pairwise alignment is approximately, aW+bN2+c N2W/20w [30] Where, W is the number of words generated, N2 is the number of residues in the subject database and a, b and c are constants. The above formula, would have the number 20 replaced by 4 if the sequence comparison was for nucleotides, and then N2 would be the number of bases in the subject sequence. Also, let N1 be the number of bases in the query. Although the number of words generated, W, increases exponentially with decreasing T, it increases only linearly with the length of the query, so that doubling the query length doubles the number of words [30]. That is to say for a given data quality, keeping w and T to be constant, W=d N1, where d is proportionality constant.

$$t(0) = (ab)N1 + (b)N2 + \left(\frac{cd}{20^w}\right)N1N2$$

$$= a'N1 + b'N2 + c'N1N2$$

$$thus\, t(O) = O(N1N2)$$

i.e., query and subject sequences product, which is quadratic in nature.

3 FastMSECT Algorithm implementation in GMSECT

Now, the FastMSECT uses divide-and-conquer method and combines the knowledge of L3 cache in many cores computing architecture to make the computing increase only about linearly with increasing sequence length. This property makes this algorithm versatile for genome comparison and structural variation extraction purposes as presented in the paper on GMSECT. Using a parallel computing technique and L3 cache knowledge the FastMSECT algorithm implemented as GMSECT tool efficiently performs the alignment comparison in the shortest amount of time by accounting for memory and processor resources. Figures 1 & 2 taken from GMSECT paper shows the linear time relation with data and speedup with increasing computing CPU cores in many-core architecture of the FastMSECT algorithm. Figure 3 taken from GMSECT paper, for which the lead author is the same as in this paper, shows the importance of L3 cache hits, as the time to do computation actually first reduces with increase in data (sequence) size until it gradually starts to increase again with increasing data size.

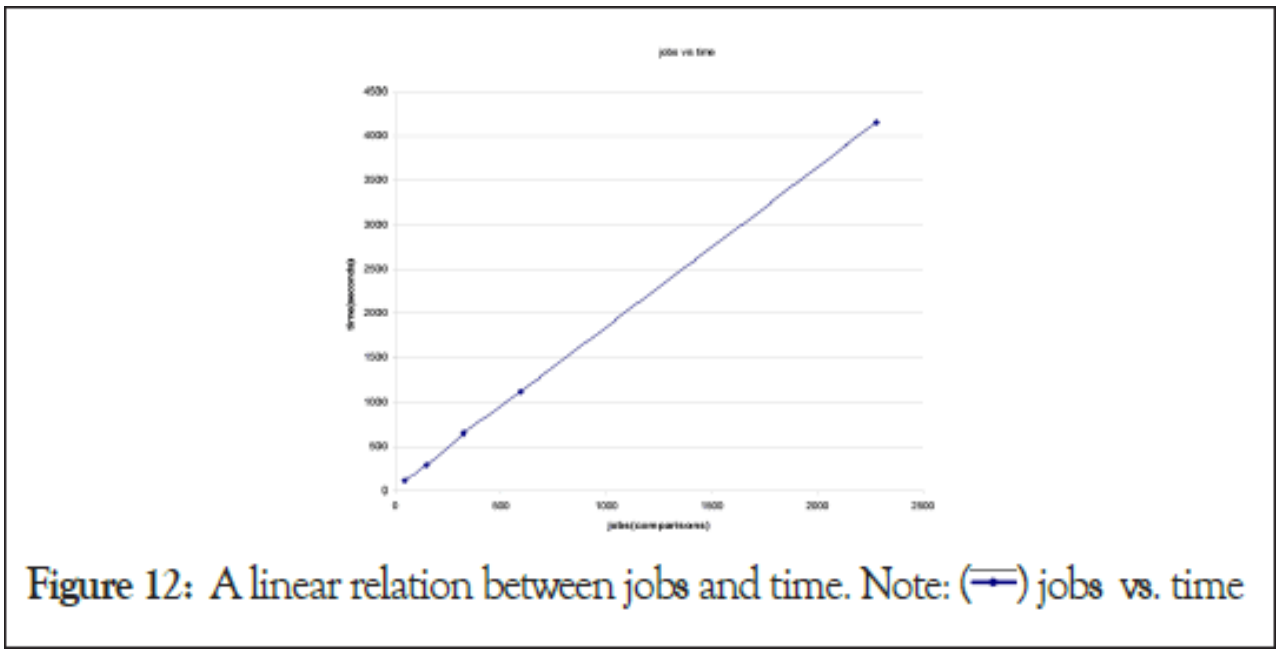

Figure 12: A linear relation between jobs and time. Note: (—•—) jobs vs. time

Figure 1: GMSECT paper shows the linear performance with sequence length as time complexity of FastMSECT algorithm

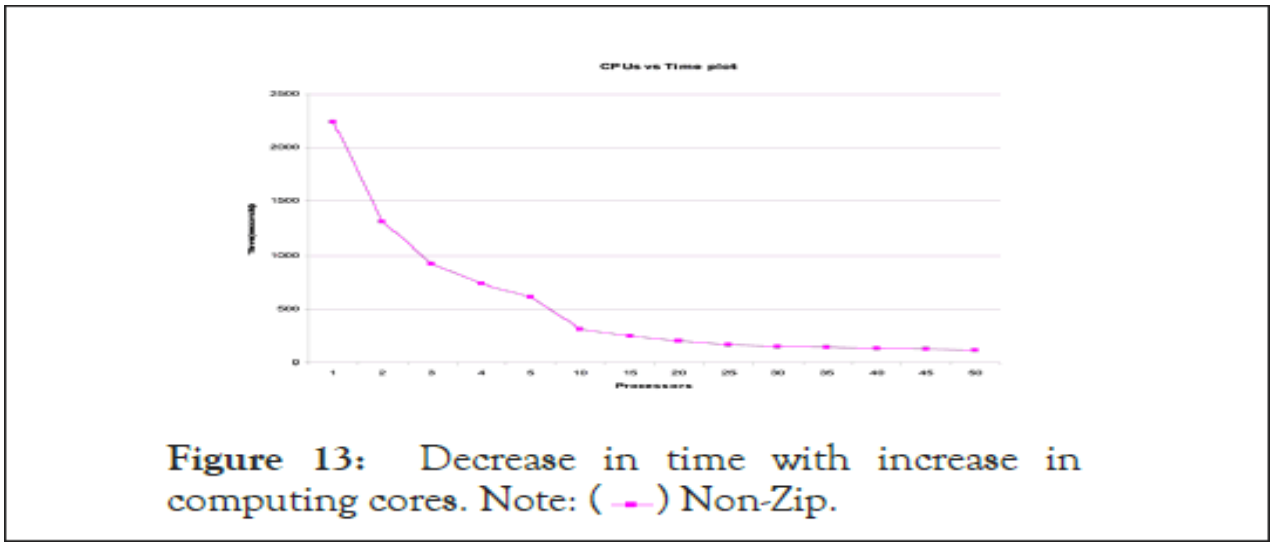

Figure 13: Decrease in time with increase in computing cores. Note: (—•—) Non-Zip.

Figure 2. GMSECT paper shows the value of many-core computing in FastMSECT algorithm

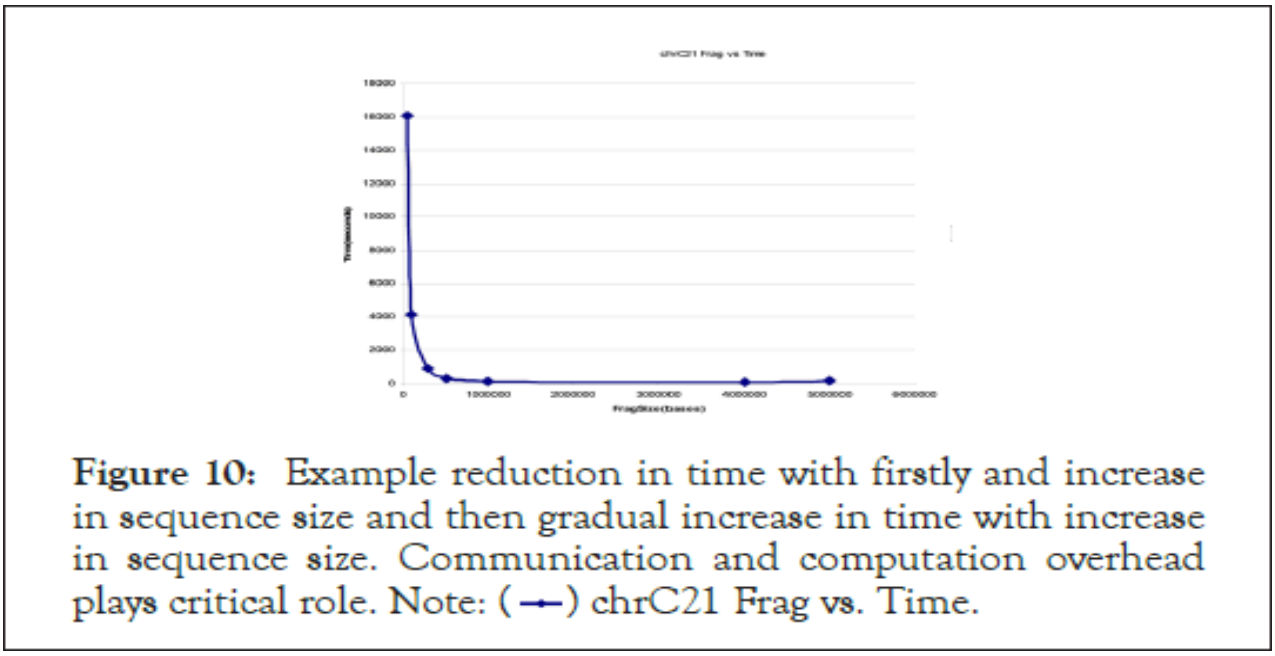

Figure 3: The role of L3 Cache as depicted in time that it takes with various sequence lengths in GMSECT paper figure 10.

4 DISCUSSIONS

Any pairwise alignment tool, including BLAST[22], BLAT, FASTA, and other potential alignment tools, can be utilized with GMSECT, an application interface. The computational complexity that arises with sequence size also determines how massive a sequence is, in addition to genome size. Stated differently, a large sequence is one that requires a significant amount of time to process with conventional computer resources and frequently does not process satisfactorily. Sequences that are characteristic of higher eukaryotes, like plants and animals, whose genome sizes range into the hundreds mega-bases of giga-bases, are known as massive sequences. The human genome and the model plant Arabidopsis thaliana are two common examples. For example, the human Build 35 reference sequence's chromosomes 1, 2, and 3 have corresponding sizes of 241 MB, 237 MB, and 195 MB in megabytes. Massive, however, is a relative adjective that should be used in relation to the algorithm that has been put into place. The previous statement can be attributed, in part, to variations in the word counts produced by various algorithms, which lead to hits that result in 'High Scoring Pairs' (HSPs) and their continuance until the score falls below the predetermined level. As a result, the computational limit for the memory resource exists. The compared sequences shouldn't be so large that they result in a "memory address violation," which creates "core" files, or that cause other errors like "segmentation fault," "mpid:Broken pipe!," or "machine hang." For example, a 2 GB, 2.2 GHz processor may have a massiveness limit of about 50,000 bases for the tool BLASTn, but only 20,000 bases for BLAT. BWA, Burrow- Wheeler-Aligner [10] is the new aligner but typically is used for very short sequences such as Next Generation Sequences [11]. BWA-SW[14] has come as an alternative for long read sequences and can be useful for a lot of coming research and medical informatics work in coming years.

Samtools [12] and GATK [13] are the annotators that are the natural choice of tools for bioinformatics professional, although the legacy also is of having used MAQ by many bioinformatics scientists. An alternative to GATK is the FreeBayes which uses Bayesian statistics for the purpose of variant calling and is freely available for commercial use unlike GATK. Naturally, each of these heuristics-based tools and aligning extracting tools has advantages and disadvantages based on the diverse conditions that determine their applicability. "The number of bases could be considered roughly equal to the number of bytes" under fair approximation. Message Passing Interface [MPI] [21] is commonly used for creating parallel jobs such as described in parallel pattern search papers for gene sequences. Many massive data for human genome projects have been launched after the successful technology development of human genome sequencing, that includes the 1000 genome project [15], The Genome of the Netherlands [16], the Indian genome project or the UK Biobank project [18] or even in the GTEx Genome Tissue Expression project [19], in each one the GMSECT tool can be deployed for meaningful information extraction. Due to the enormous length of the sequences and the processing demands,

various pairwise alignment strategies are impossible to do without a supercomputer. The current approach functions as an interface for parallel computing for the heuristic tools that are currently available and can be used on a cluster of computers. The Needleman Wunsch Algorithm for global alignment and the Smith-Waterman algorithm for local alignment are the generalizations of the dynamic programming technique. In order to find matches of all conceivable words of size w that match score threshold T or greater, a well-known local alignment tool like BLAST [22] searches for matches. It then uses dynamic programming to lengthen the matches until the score falls below the threshold value T.

With four bases—T, C, A, and G—there are four words that can be created for nucleotides, or 4w. A large value of w would produce more words but fewer HSPs (large Scoring Pairs), whereas a little value of w would produce fewer words but more HSPs. Thus, in the earlier instance High sensitivity is present, whereas greater specificity is provided in the latter scenario. Issues about high noise and repetitive information arise with increased sensitivity, whereas issues about losing out on pertinent matches arise with increased specificity. As an extreme case scenario, let's say that w=1. In such situation, we get 41, or just 4 words: T, C, A, and G. But each of these four words would produce a large computational time quantity of duplicated data by creating HSPs to match the full sequence. Consider the opposite extreme scenario, where we compare a sequence of size 'N' with itself while maintaining w=N for the word size. Due to this, 4N would produce a high number of words, but only one of those words would be able to create an HSP. As a result, all potential intra-sequence matches, including copy numbers, inversions, LINES, SINES, mini-satellites, microsatellites, and SNPs, would be lost. The story is the same for protein sequence matches, with the small change that since there are 20 naturally occurring amino acids, there would now be 20 words instead of only 16. Because different alignment techniques have varied values for w and T, they are therefore appropriate for varying levels of data quality. Notably, the elongation of the HSPs takes up the majority of the time. An algorithm's sensitivity, specificity, and speed on a given dataset depend on the quality of the data, w and T. The number of word hits increases exponentially with decreasing T, despite the fact that the quantity of words generated and execution time have a linear relationship [3]. The sequencing of DNA is not random. For instance, the existence of CpG islands and the locally biased A and T rich region are known to exist. It is well known that higher GC percentages are linked to higher DNA stability, higher thermal stability, and species evolution with codon biasing as a result of stability criteria or other factors. Large segments of DNA or single nucleotides are subject to a survival selection process that determines their existence. The metabolic route that is now in place in a cell determines the selection check, which is why distant species react differently to changes in the genetic sequence. These events raise questions about why the data quality of a distant species differs from that of humans, such as in the case of Arabidopsis thaliana. Because of the distinct metabolic processes that higher organisms have compared to lower organisms, their survival criteria and complexity differ. As a result, whereas the ORF of a prokaryote lacks introns, the eukaryotic genomic sequence is divided into exons and introns by splice sites. Furthermore, eukaryotic genes can have lengths of up to 15 Kbp, whereas prokaryotic genes typically have lengths of around 1 Kbp. Since the complexity of the genome is yet unknown, some researchers think that before drawing conclusions, they should dust and cover the "uninformative" sections of the genome, like tandem repeats and fingerprints. Regardless of how the genomic data is handled, it is certain that the genome sequences of various organisms will differ in terms of data quality, which will impact the quantity of words created and, consequently, the extension as well, leading to variance in execution time.

5 Usages of FastMSECT in Personalized Genomics Era

As the era of personalized genome sequencing approaches, we will need a more convenient and potent tool to draw relevant conclusions from the sequence in order to link the individual with genetic causes of diseases, like autism, or to determine the genetic basis of a particular trait. To do this, a reference genome would need to be established as the benchmark against which the genome of the individual might be evaluated. A single comparison between the reference genome and itself would be necessary [4]. It would also be necessary for each individual genome to compare both with the reference genome and with itself. But the selection of the standard reference genome is debatable in and of itself because no person's genome can be biased to be designated as a reference genome, only for the rationale that no single individual can be entirely indicative of every potential variation during the course of human evolution. The goal should be to assemble a representative reference genome from a population of different racial backgrounds by obtaining as much statistically significant data as possible from as many genomes as possible. The various structural variants found in the genome would be included in this statistically significant information, ensuring that every significant structural variant is included in the fictitious genome. Naturally, as more and more individual genomes from diverse populations were available, the reference genome's structure would also need to be revised. More-and-more people's genomes should become available if sequencing technology advances quickly enough. The reference genome should ideally be updated dynamically; however, this would obviously require a lot of processing power. By making the update periodic, it is possible to reduce processing demand and make the dynamic update discrete. We select mice as our prospective animal for conducting tests on them for a variety of

therapeutic goals. The identification of a person's genome in relation to structural variants of the conventional mouse genome would be of interest to scientists. The structural differences between the chimpanzee and human genomes are of interest to scientists as well. Similarly, to comprehend genetic composition, evolution, and disease susceptibility, it is necessary to compare two nearby and two distant species. The previously mentioned genome segmentation and alignment approach does not require intensive inter-processor communication. The tasks could alternatively be turned in in a serial basis [5].

When GMSECT was used under the same conditions on a distant species, such chromosome 2 of Arabidopsis thaliana, a comparable hyperbolic resemblance profile was produced using the BLAST tool selection. About 4 million bases later, the Minima was discovered once more. The decrease in data quality may be the cause of the minor shift in minima [6]. We can reasonably assume the Minima to be a "general" ideal fragmentation size for HPF cluster nodes with processors of the previously specified kind, since even for distant species, it is approximately 4 million bases [7]. In addition to increasing calculation time, reducing the fragmentation size to a smaller size would also raise the approximate stitching back of the matches at the junction [8]. Using higher units of chromosome 21 units of Celera's Human Genome compilation, a number of comparisons against time graph was created using the BLAST tool selection and standard output format [9].

6 CONCLUSIONS

We talked in a crisp summary of what the FastMSECT algorithm is that was used int the GMSECT [1] papers such as by highlighting the importance of divide-and-conquer algorithm becoming beneficial for a quadratic time complexity algorithms such as that used in many sequence comparison methods, and we also talked about the importance of L3 cache which is shared among the various processors. GMSECT uses the FastMSECT algorithm internally that exploits the value of many-core computing.

References

1 Singh A, MSECT, Book of abstract, The 21st International Conference on Bioinformatics and Computational Biology (BIOCOMP 2020) as part of American Council on Science and Education/CSCE 2020, ISBN # 1-60132-512-6.
2 Goodwin S, McPherson JD, McCombie WR. Coming of age: Ten years of next- generation sequencing technologies. Nat Rev Genet. 2016;17(6):333-51.
3 Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990;215(3):403-10.
4 Riley AB, Kim D, Hansen AK. Genome sequence of ―Candidatus Carsonella ruddii‖ strain BC, a nutritional endosymbiont of Bactericera cockerelli. Genome Announc. 2017;5(17):e00236-17.
5 Neale DB, Wegrzyn JL, Stevens KA, Zimin AV, Puiu D, Crepeau MW, et al. Decoding the massive genome of loblolly pine using haploid DNA and novel assembly strategies. Genome Biol.2014;15(3):1-3.
6 Nakato R, Gotoh O. Cgaln: Fast and space-efficient whole-genome alignment. BMC Bioinformatics. 2010;11(1):1-4.
7 Reinert G, Chew D, Sun F, Waterman MS. Alignment-free sequence comparison (I): Statistics and power. J Comput Biol. 2009;16(12):1615-34.
8 Orlov YL, Potapov VN. Complexity: An internet resource for analysis of DNA sequence complexity. Nucleic Acids Res. 2004;32(suppl_2): W628-33.
9 Torreno O, Trelles O. Breaking the computational barriers of pairwise genome comparison. BMC Bioinformatics. 2015;16(1):1-3.Li H. and Durbin R. (2009) Fast and accurate short read alignment with Burrows- Wheeler Transform. Bioinformatics, 25:1754-60. [PMID: 19451168].
10 Shendure J, Ji H (October 2008). "Next-generation DNA sequencing". Nature Biotechnology. 26 (10): 1135–1145. doi:10.1038/nbt1486. PMID 18846087. S2CID 6384349
11 Li H. and Durbin R. (2010) Fast and accurate long-read alignment with Burrows- Wheeler Transform. Bioinformatics, Epub. [PMID: 20080505]
12 Twelve years of SAMtools and BCFtools Petr Danecek, James K Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Martin O Pollard, Andrew Whitwham, Thomas Keane, Shane A McCarthy, Robert M Davies, Heng Li GigaScience, Volume 10, Issue 2, February 2021, giab008, https://doi.org/10.1093/gigascience/giab008
13 Van der Auwera GA & O'Connor BD. (2020). Genomics in the Cloud: Using Docker, GATK, and WDL in Terra (1st Edition). O'Reilly Media.
14 Li H, Ruan J, Durbin R: Mapping short DNA sequencing reads and calling variants using mapping quality scores. Gen Res. 2008, 18 (11): 1851-1858.10.1101/gr.078212.108.

15 The 1000 Genomes Project Consortium. A global reference for human genetic variation. Nature 526, 68–74 (2015). https://doi.org/10.1038/nature15393
16 Boomsma, D., Wijmenga, C., Slagboom, E. et al. The Genome of the Netherlands: design, and project goals. Eur J Hum Genet 22, 221–227 (2014). https://doi.org/10.1038/ejhg.2013.118
17 Rathnasamy, Narmadha; Mullapally, Sujith; Sirohi, Bhawna (April 4, 2022). "Precision oncology in Low and Middle income countries: a word of caution". International Journal of Cancer Care and Delivery. doi:10.53876/001c. 29768. eISSN 2770-3533
18 Conroy MC, Lacey B, Bešević J, Omiyale W, Feng Q, Effingham M, Sellers J, Sheard S, Pancholi M, Gregory G, Busby J, Collins R, Allen NE. (2023). "UK Biobank: a globally important resource for cancer research". British Journal of Cancer. 128: 519–527. doi:10.1038/s41416-022-02053-5. PMC 9938115. PMID 36402876
19 Lonsdale, J., Thomas, J., Salvatore, M. et al. The Genotype-Tissue Expression (GTEx) project. Nat Genet 45, 580–585 (2013). https://doi.org/10.1038/ng.2653
20 Garrison E, Marth G. Haplotype-based variant detection from short-read sequencing. arXiv preprint arXiv:1207.3907 [q-bio.GN] 2012
21 The MPI Forum, CORPORATE (November 15–19, 1993). "MPI: A Message Passing Interface". Proceedings of the 1993 ACM/IEEE conference on Supercomputing. Supercomputing '93. Portland, Oregon, USA: ACM. pp. 878– 883. doi:10.1145/169627 169855. ISBN 0-8186-4340-4.
22 Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990;215(3):403-10.
23 https://www.sharcnet.ca/my/front/