

## Evaluating the use of smartphones for processing computational tasks: "A case study for the annotation of prokaryotic genomes"

<sup>a</sup> Jamir Nascimento Cardoso Junior, <sup>b</sup> Mônica Silva De Oliveira, Adonney Allan De Oliveira Veras<sup>a</sup> Faculty of Computing, Federal University of Pará campus Castanhal (FACOMP-CCAST), Pará, Brazil,<sup>b</sup> Federal Institute of Pará campus Tucuruí, Pará, Brazil,<sup>c</sup> Biological Engineering Laboratory, Guamá Science and Technology Park, Belém, PA, Brazil.**Authors' Contribution** Junior, J. N. Cardoso & M. S. De Oliveira developed software test tool and wrote user's manual; A. A. O. Veras developed methodology, review & edited, conceptualization the project.**Corresponding Author's Email Address:** allanveras@ufpa.br**ABSTRACT****Review Process:** Peer review

Currently, there is a growing dependence on smartphones in the daily lives of thousands of people, noting that the number of users continues to rise, surpassing the number of computers in use globally. In addition, the performance of smartphone hardware has become an essential factor when choosing these devices, along with price, brand and operating system. Constant innovations in smartphones have made these devices increasingly powerful, even rivaling the performance of many computers. The example of the Samsung S23 5G, which has features comparable to a low-performance notebook, illustrates the advancement of the hardware in these devices. Despite the growing demand for advanced hardware in smartphones, many users may not fully understand the technical specifications of the processors. The article proposes an evaluation of the use of this hardware in the execution of specific tasks, with a focus on the annotation of prokaryotic genomes. The methodology of the study involved selecting popular smartphones from Samsung and Motorola, choosing the Android operating system due to its efficiency in resource management, using the Prokka tool for genomic annotation due to its minimal resource requirements and using Termux, a terminal emulator that allows users to run a full Linux distribution on Android devices. To evaluate performance, the article uses the Linux 'time' command to measure execution time, memory consumption, CPU utilization and other relevant metrics.

**Keywords:** Genomic annotation, software, android, smartphone

**INTRODUCTION:** Today, thousands of people have incorporated smartphones into their daily activities, and this number of users continues to grow every year. According to research, the number of smartphones in operation in the world has already surpassed the number of computers, and 60% of Google searches are carried out using mobile devices. According to IBGE data, Brazil has 242 million smartphones in use, slightly exceeding the mark of 214 million inhabitants. With the increase in the number of users, expectations regarding the performance of smartphones are also growing, and hardware has become one of the main factors considered when making a purchase. In addition to price, brand and operating system, CPU processing speed and processor type are important criteria for decision-making (Tanenbaum and Bos, 2016). The constant innovations in smartphones available on the market are raising the intelligence of these devices to levels that, in some cases, exceed the processing capabilities of many computers. The hardware of these devices has experienced significant advances, as evidenced by the Samsung S23 5G. This device has features similar to those of a mid-range notebook, with 8 GB of RAM, a solid-state storage capacity of up to 128 GB and the option of expansion via an additional card. Equipped with an Octa Core processor operating at 13.36 GHz, 2.8 GHz and 2 GHz, it provides processing performance comparable to many mobile PC processors (Samsung, 2023). Nowadays, many professionals are taking advantage of their smartphones to carry out tasks that previously required specific devices. One example of this is online store professionals, who use their mobile devices to photograph products and even edit the images directly on their cell phones. In addition, the use of smartphones to watch movies in high quality, including those in HD (High-Definition), and for other activities, such as viewing video lessons during studies, which used to be done exclusively on computers, has become increasingly common on mobile devices (Foresti, 2016). From a computing point of view, when evaluating processor speed, several factors are taken into account. These include the ability to run applications efficiently, the ability to handle the simultaneous execution of multiple applications in the background. In the context of mobile devices, the performance of the data connection (3G, 4G, 5G or Wi-Fi) is also analyzed, as well as the power consumption of the processor and other resources during execution (Stallman, 2007).

Although the demand for more advanced hardware in smartphones is growing, many users lack the necessary knowledge to identify the characteristics of a high-performance processor and how to evaluate it effectively. Many simply want an efficient processor in their devices, without a deep understanding of the technical specifications involved. This makes room for an evaluation of the performance of this hardware when processing specific tasks. The main objective of this study is to conduct an analysis centered on the annotation of prokaryotic genomes, using mobile devices as a scientific tool. Genome annotation refers to the process of inferring

biological information from DNA sequences, an essential and frequent task for researchers in the field of genetics, as it reveals the genetic content of a target organism (Da Silva Moia *et al.*, 2022).

**OBJECTIVES:** The objectives of this study were as follows: (1) Evaluate the use of smartphones to perform specific tasks, with an emphasis on annotating prokaryotic genomes. (2) Use the Prokka tool for genomic annotation, making use of the features of Docker and Termux, a terminal emulator that allows users to run a complete Linux distribution on Android devices. (3) Evaluate performance, execution time, memory consumption, CPU utilization, and other relevant considerations.

**MATERIAL AND METHODS:** The methodology used to process and analyze the datasets in this study is briefly described in figure 1.

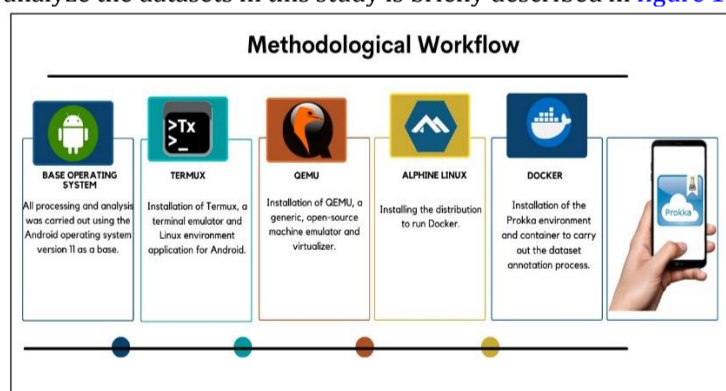


Figure 1: Methodological workflow.

The base operating system used was Android version 11. To emulate the Linux terminal, we used Termux, making permissions adjustments for access to the device's storage. Installation on the Alpine Linux distribution required prior configuration of the QEMU virtualization environment. Finally, we implemented the Docker environment and then the Prokka container to conduct the annotation on the datasets selected for this research.

**Hardware:** The hardware devices chosen for this study were smartphones produced by Motorola and Samsung, selected due to their widespread popularity and the fact that they operate natively on the Android operating system. The initial analyses were conducted on one specific model, the Moto e7, which is equipped with a 64-bit AArch64 Processor rev 4 (aarch64) processor, operating at a frequency of 4 x 400MHz at 1500MHz and 4 x 900MHz at 2001MHz, running the Android 10 system, with 4GB of RAM and 64GB of storage. The other device used was the Samsung Galaxy A03s, equipped with a 4x 2.3GHz + 4x 1.8 GHz Cortex-A53 AArch64 processor, Android 11 operating system with Samsung One UI 3.0, 4GB of RAM and 64GB of storage.

**Android:** The choice of the Android system is not based solely on its popularity, but rather on the presence of a central management system for processes, memory, network protocols and device and

peripheral drivers that are run through the Linux kernel. Android is responsible for controlling these devices and therefore manages energy at a more fundamental level, determining the CPU's operating frequency in order to optimize battery performance in smartphones (Pereira and Da Silva, 2009).

**Annotation software:** Prokka is a computational tool developed to automate the genomic annotation of bacteria and archaea. The choice of this tool for this study is justified by the fact that its annotation process requires minimal computing resources, such as memory and processing capacity, and can be run on a conventional desktop computer. The only requirement of the user is to provide the genomic sequence file in FASTA format (Seemann, 2014).

**Termux:** Termux stands out as an efficient solution for Android users who want to integrate Linux functionality into their mobile devices. With it, it is possible to operate a complete Linux distribution, expanding the capabilities of smartphones and tablets beyond their conventional functions. Through its command line interface (CLI), users have access to a vast array of Linux commands, optimizing the potential of their devices (Okah, 2023).

In addition, Termux's package management system offers a diverse range of applications and tools, including renowned programming languages such as Python, Node.js and Git, promoting an environment conducive to experimentation on mobile platforms.

**Docker:** Docker, an innovative container virtualization platform, has revolutionized the development, distribution and execution of applications. Focusing on virtualized containers, Docker makes it possible to encapsulate applications and their dependencies in complete images. These images, which encompass both the application and its execution environment, are stored in the Docker Registry, simplifying the integration and continuous customization of software (Docker, 2020).

In addition to their versatility, Docker containers stand out for their lightness and efficiency, operating independently of the operating system. This ensures consistent performance, both in a developer's local environment and on large production servers. In addition, Docker simplifies the deployment process by managing all the dependencies needed to ensure that the application works correctly.

**Alpine linux:** Alpine Linux stands out as an interesting option for using containers in the Android environment due to its remarkably low resource consumption. With a minimalist approach and a design focused on simplicity, Alpine Linux presents itself as a robust alternative to traditional Android operating systems. Its lightweight structure not only optimizes the use of device resources, but also boosts performance, providing a fluid experience even on older devices or those with more basic configurations. This system is renowned for its efficiency: a container consumes less than 8 MB, while a basic installation requires approximately 130 MB of storage. So, as well as offering a complete Linux environment, Alpine makes a wide range of packages available through its repository (Alpine, 2023).

**The QEMU:** QEMU is a virtualization tool that makes it possible to create virtual machine environments, allowing the execution of tools with architectures other than Android. It also acts as a hypervisor that uses dynamic translation to emulate CPU instructions in a foreign architecture. Its usefulness covers a range of activities, including testing and development, cloud computing and system administration. When used in conjunction with other virtualization technologies, such as KVM or Xen, QEMU enables direct execution of the guest system on the host CPU, providing a speed close to native (Galgani, 2023).

**Benchmark:** To acquire the essential statistics during the evaluation process, the 'time' command was used in the Linux environment (<http://man7.org/linux/man-pages/man1/time.1.html>). These statistics include the program's execution time in user mode, execution time in kernel mode and the program's average memory consumption and CPU usage. The data from this evaluation includes the size of the file used as input, the number of contigs (genomic sequences) and any errors that occurred during processing.

**Data source:** To conduct this research, thirty organisms were chosen from the National Center for Biotechnology Information (NCBI) database, accessible at <https://www.ncbi.nlm.nih.gov>. The datasets selected for this analysis include draft (incomplete) genomes, complete genomes and information from metagenomic analysis, in order to provide comprehensive diversification in the analysis in question. Table 01 lists the names of all the organisms that were analyzed.

Table 1: List of all the organisms used in the study.

Organism Name	Amount Contigs	GenBank
Sulfolobus sp. SCGC	86	GCA_003086375.1
MAG: Aquificaceae bacterium isolate SKW97	67	JAOAIS000000000.1
MAG: Chlorobiota bacterium isolate Skiv_18-Q3-R9-52_MAXAC.146	2	JADKIV000000000.1
Fusobacteriaceae bacterium genome ERS537392_50	30	GCA_900757965.1
Nostoc sp. HK-01 DNA	1	AP018318.1
Aquificaceae bacterium CF Pisc bin19	48	GCA_026627465.1
Fusobacteriaceae bacterium HGM14322	41	GCA_900758045.1
MAG: Chlorobiota bacterium isolate CHB3	19	RHKQ000000000.1
Aquificaceae bacterium isolate SKYB2	66	JANXBA000000000.1
MAG: Nostocales cyanobacterium LE14-WE12	226	JASJEY000000000.1
Sulfolobus sp. E5-1-F chromosome	1	CP045687.1
MAG: Dolichospermum sp. UKL201 chromosome	1	CP050891.1
Microcystis aeruginosa FD4 chromosome	1	CP046973.1
Planktothrix agardhii NIES 204 DNA	1	AP017991.1
Anabaena catenula FACHB-362	109	GCA_014698735.1
Cyanobacterium sp. HL-69 chromosome	1	CP024912.1
aff. Roholtiella sp. LEGE 12411 genome assembly ASM1520699v1	338	GCA_015206995.1
Scytonema sp. RU44	77	JAAUSC000000000.1
MAG TPA_asm: Richelia sp. isolate UBA11571	463	DLZW000000000.1
MAG: Rivularia sp. ALOHA_DT_140	2574	JALJPP000000000.1
MAG: Rivularia sp. T60_A2020_040	139	JACYMD000000000.1
Nostoc Azollae 0708	3	CP002059.1
MAG: Nostocales cyanobacterium isolate PH2015_08D_37_15	603	RDYK000000000.1
MAG: Nostocales cyanobacterium LacPavin_0920_SED1_MAG_38_18	198	JAPLHY000000000.1
MAG: Nostocales cyanobacterium isolate PH2015_08D_39_10	271	RDYJ000000000.1
Anabaena sp. CCAP 1446/1C	254	JAAMUX000000000.1
Sulfolobus sp. E11-6 chromosome	1	CP045706.1
Sulfolobus sp. A20	1	CP017006.1
Sphaerospermopsis aphanizomenoides LEGE 00250	399	GCA_015207835.1
MAG: Scytonema sp. CRU_2_7	628	JAAUOR000000000.1

**RESULTS:** To begin presenting the results in a comprehensive manner, it is essential to examine not only the final results, but also the number of contigs (DNA fragments resulting from the genomic assembly process) present in each file used during processing. Figure 2 was designed to provide a clear visual representation of this information. Looking at the distribution of contigs in the different datasets analyzed. It is important to note that some of these files contain complete genomic sequences, indicating that the file includes a single sequence that represents the entire genome of the organism in question. However, in order to reflect the reality faced by researchers, it was decided to use several organisms with multiple contigs, resulting from so-called draft assemblies, in this study. Figure 3 presents an analysis of the total processing time required to annotate each organism. It can be inferred that the total number of contigs in the input file does not correlate directly with the total processing time required for its annotation. An illustrative example is the organism MAG: Rivularia sp. ALOHA\_DT\_140, which despite having a high number of contigs as shown in figure 2, showed a similar average processing time compared to the other datasets. Based on the analysis shown in figure 4, it can be seen that file size is not always directly correlated with the amount of memory needed to perform the task. In detail, seven data sets had sizes close to two megabytes, of which four required significantly more memory to complete their analysis.

Analyzing the memory required for different organisms, it stands out that "Anabaena catenula FACHB-362" and "aff. Roholtiella sp. LEGE 12411 genome assembly ASM1520699v1" have a memory close to 229.92 Mbytes, while "Fusobacteriaceae bacterium HGM14322" has a lower memory, around 223.02 Mbytes. These observations highlight the complexity of the relationship between file size and the amount of memory needed for processing. They indicate that factors other than file size influence memory requirements, suggesting the importance of considering specific characteristics of datasets when allocating computing resources.

Figure 5 showd the distribution of the total CPU allocation required to process all the datasets used in this study. Notably, it can be seen that 40% of the datasets processed were allocated 5%, while 36.7% received 6% of the total processing capacity dedicated to them.

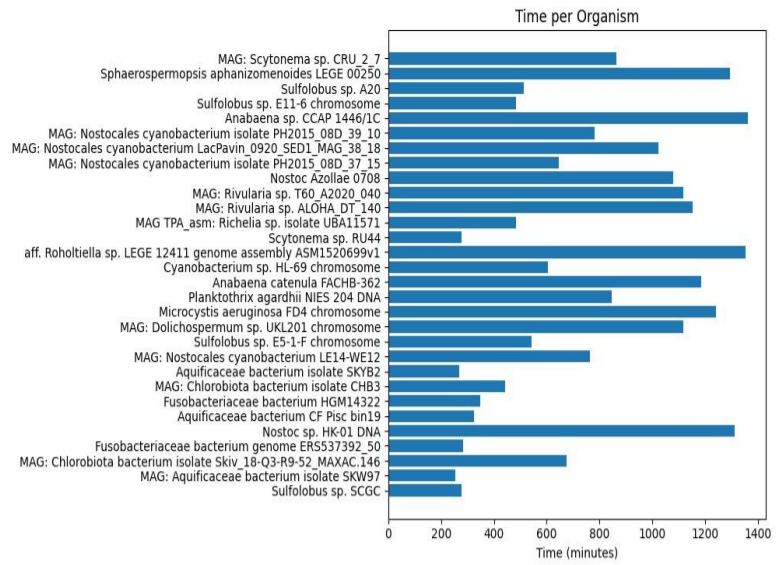
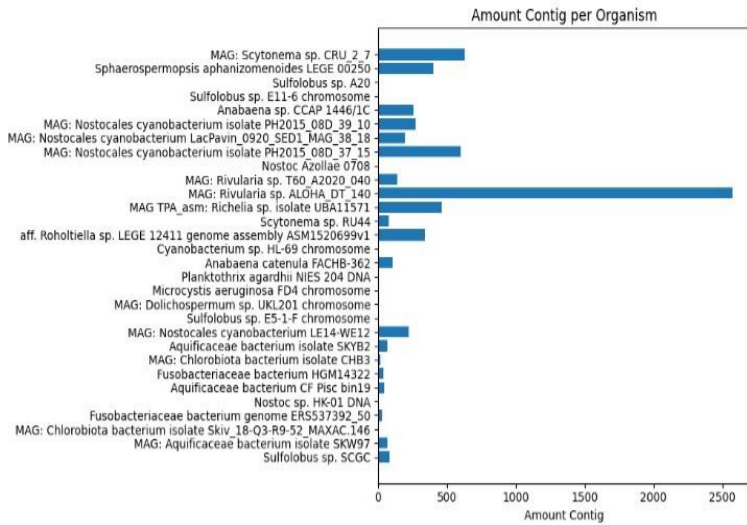


Figure 2: Total number of contigs per file.

Figure 3: Total processing time per organism

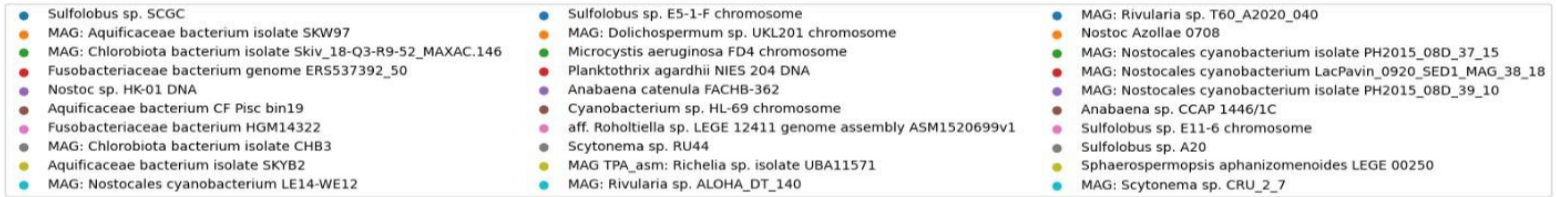
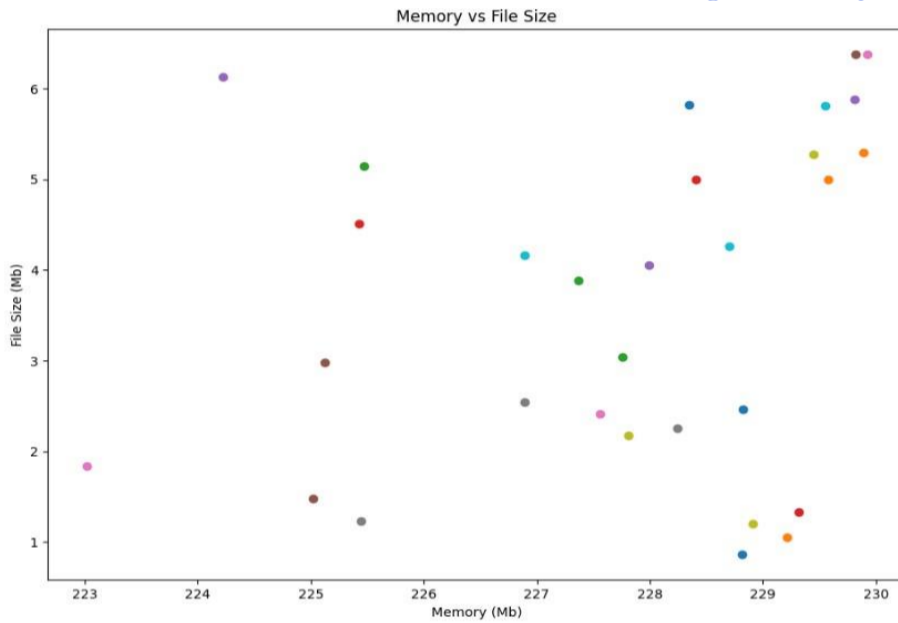


Figure 4: Distribution of dataset size versus amount of memory.

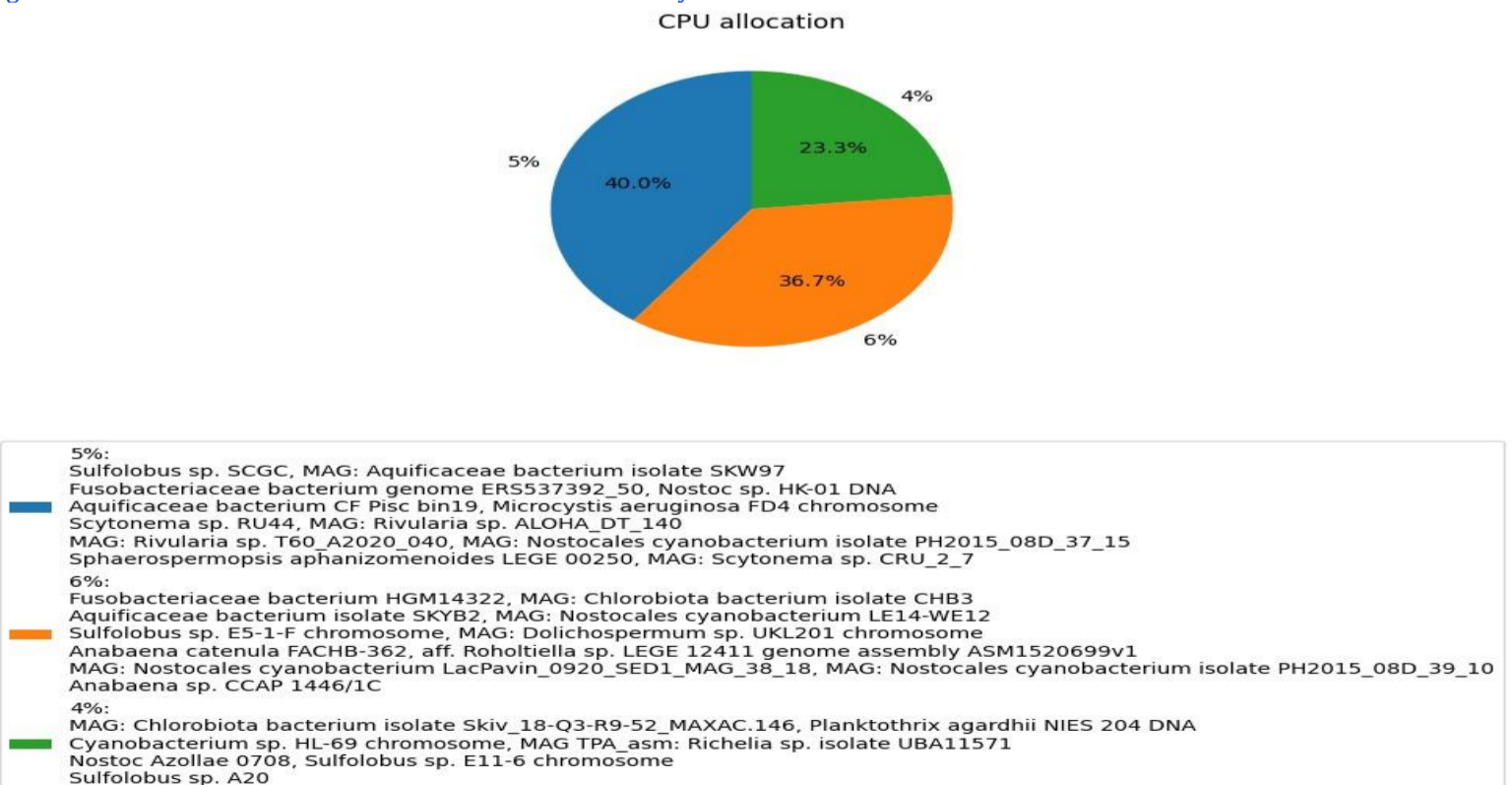


Figure 5: The CPU allocation.

Furthermore, when analyzing the CPU allocation for different organisms, it can be seen that the majority (65%) are between 5% and 6%, suggesting a relatively even distribution of processing resources between the datasets investigated. In this context, it is

interesting to note that some organisms, such as "MAG: Chlorobiota bacterium isolate", "Planktothrix agardhii NIES 204 DNA", "Cyanobacterium sp. HL-69 chromosome", "MAG TPA\_asm: Richelia sp. isolate UBA11571", "Nostoc Azollae 0708", "MAG: Nostocales cyanobacterium isolate PH2015\_08D\_37\_15", "MAG: Rivularia sp. T60\_A2020\_040", "Sulfolobus sp. A20", "Sphaerospermopsis aphanizomenoides LEGE 00250" and "Sulfolobus sp. E11-6 chromosome", show CPU allocations of 4%, standing out as cases with a slightly lower demand for resources. These observations highlight not only the efficiency in distributing the computational load, but also the variation in CPU requirements between different organisms. It is important to note that the entire analysis was carried out exclusively on the SAMSUNG device. This makes it possible to draw conclusions about the minimum capacity required to perform this task, taking into account crucial elements such as processing power, memory capacity and internal space available on the device. This result not only highlights the effectiveness of the device in question, but also provides valuable insights for determining the essential requirements for the successful execution of similar tasks in other hardware environments.

**CONCLUSION:** In the current study, we share the experience of using smartphones with the Android operating system, focusing in particular on the processing of tasks aimed at annotating prokaryote genomes. The emulated environment was configured to operate in a similar way to existing computational tools for this purpose, but this paper highlights both the achievements and limitations of this process. It should be noted that in order to carry out this analysis, it was necessary to overcome a variety of challenges, from defining the necessary software structure, given the limitations of Linux commands in the Android operating system. Although Termux emulates a Linux distribution, it does not have superuser permissions, making it impossible to install various dependencies of the annotation software (Prokka). In addition, Termux presented significant obstacles when installing the Docker environment. Given these limitations, it was essential to proceed with the installation of Termux, and within it, configure the QEMU virtualized environment. In this environment, we chose to install Alpine Linux, where we conducted all the processing for the study in question. This approach made it possible to overcome the initial difficulties, ensuring that the analysis could be carried out effectively in the context of the restrictions imposed by the Android operating system and its peculiarities. In total, thirty organisms were processed, including complete genomes, draft genomes and data resulting from metagenomic assemblies, thus ensuring a wide diversity of datasets within the scope of the study. It was observed that the smartphone hardware used proved to be efficient in processing the tasks, with the main obstacle encountered being related to the operating system. Both mobile devices used in this analysis had the AArch64 architecture in their version of Android, while most software is developed for the x86\_64 architecture. Overcoming the limitations of the operating system, it was clear that the hardware is suitable for carrying out computational tasks. Based on this study, it is possible to infer that, should a future version of Android be made available in the x86\_64 architecture, the

processing of various tasks could be carried out using the hardware in these devices. Another aspect relevant to the study is that most prokaryotic organisms have genomes whose size generally doesn't exceed a few megabytes. This makes processing these genomes feasible in environments that are increasingly popular and accessible to the general population, such as smartphones. In this analysis, we used 8GB of the smartphone's disk space, which leads us to infer that it is entirely possible to take advantage of this hardware, especially in situations where the student may not have a computer or notebook, but does have a smartphone. This approach highlights the adaptability of genome processing on mobile devices, offering an accessible alternative for students and researchers. In addition, it highlights the importance of exploiting the potential of smartphones in academic activities, promoting access to computational resources even in scenarios where traditional equipment may not be available.

**CONFLICT OF INTEREST:** The authors declared no conflict of interest.

**ACKNOWLEDGEMENT:** Thanks to the Brazilian Research Council (CNPq) and the Federal University of Pará, this work received help from PROPEP/UFPA. This work is part of the research developed by BIOD (Bioinformatics, Omics, and Development research group - [www.biod.ufpa.br](http://www.biod.ufpa.br)).

**ETHICAL RESPONSIBILITY:** This manuscript is original research, and it is not submitted in whole or in parts to another journal for publication.

**INFORMED CONSENT:** The author(s) have reviewed the whole manuscript and approved the final version of the manuscript before submission.

**REFERENCES:** Alpine., 2023. Alpine linux development team. C. E. D. O. D. D. E. <https://alpinelinux.org/about>. (Ed.).

Da Silva Moia, G., A. S. C. Gaia, M. S. de Oliveira, V. C. Dos Santos, J. T. C. Alves, P. H. C. G. De Sá and A. A. De Oliveira Veras, 2022. Renoteweb—web platform for the improvement of assembly result and annotation of prokaryotic genomes. *Gene*, 844: 146819.

Docker, O., 2020. Inc. Docker. Linea. Junho De 2017. [Citado em 11 De outubro De 2023]. Disponível em: <https://www.Docker.Com/what-docker>, 2020.

Foresti, F., 2016. O uso de dispositivos móveis entre os estudantes de pós-graduação da universidade federal de santa catarina: Os novos fluxos de informação.

Galgani, F. (Ed.) 2023. How to use qemu from the linux command-line, *baeldung linux*. 04 de setembro de 2023.

Okah, J., 2023. Getting started with termux, *dev community*, 20 de julho de. In: <https://dev.to/jjokah/getting-started-with-termux-307f>.

Pereira, L. C. O. and M. L. Da Silva, 2009. *Android para desenvolvedores*. Brasport.

Seemann, T., 2014. Prokka: Rapid prokaryotic genome annotation. *Bioinformatics*, 30(14): 2068-2069.

Stallman, R. J. A. M., 2007. *Linux and the gnu system*.

Tanenbaum, A. and H. Bos, 2016. *Modern operating systems*. 4th edition. São Paulo: Pearson Education do Brasil.



Except where otherwise noted, this item's licence is described as © The Author(s) 2024. Open Access. This item is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the [Creative Commons license](https://creativecommons.org/licenses/by/4.0/), and indicate if changes were made. The images or other third party material in this it are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.