Cross-Environment Comparison of a Bioinformatics Pipeline: Perspectives for Hybrid Computations

Nico Curti[1](✉), Enrico Giampieri[1], Andrea Ferraro[2], Cristina Vistoli[2],
Elisabetta Ronchieri[2], Daniele Cesini[2], Barbara Martelli[2],
Cristina Duma Doina[2], and Gastone Castellani[1]

[1] Department of Physics and Astronomy, University of Bologna, Bologna, Italy
nico.curti2@unibo.it
[2] INFN-CNAF, Bologna, Italy

Abstract. In this work a previously published bioinformatics pipeline was reimplemented across various computational platforms, and the performances of its steps evaluated. The tested environments were: (I) dedicated bioinformatics-specific server (II) low-power single node (III) HPC single node (IV) virtual machine. The pipeline was tested on a use case of the analysis of a single patient to assess single-use performances, using the same configuration of the pipeline to be able to perform meaningful comparison and search the optimal environment/hybrid system configuration for biomedical analysis. Performances were evaluated in terms of execution wall time, memory usage and energy consumption per patient. Our results show that, albeit slower, low power single nodes are comparable with other environments for most of the steps, but with an energy consumption two to four times lower. These results indicate that these environments are viable candidates for bioinformatics clusters where long term efficiency is a factor.

Keywords: Whole genome sequencing · Bioinformatic pipeline
Low-power · GATK-LODn pipeline

1 Introduction

Biomedical data are growing both in size and breath of possible uses. Of special importance are the so called biomedical big data, blanket term describing data generated from several machines and used to describe the health state of a person:

1. **Next generation sequencing NGS.** NGS technology. RNA-seq: experimental procedure, challenges and opportunities in statistical data analysis. ChIP-Seq: experimental procedure and statistical data analysis.

N. Curti and E. Giampieri contributed equally to this work.

G. Mencagli et al. (Eds.): Euro-Par 2018 Workshops, LNCS 11339, pp. 638–649, 2019.
https://doi.org/10.1007/978-3-030-10549-5_50

2. **Proteomics and Metabolomics.** LC/MS technology, challenges in data processing. Biological pathways.
3. **Biomedical imaging.** Imaging techniques, acquisition methods and data structures/characteristics for different imaging modalities.
4. **Statistical Analysis of Imaging Data.** Data processing techniques, study designs, analysis strategies, research questions and goals. Radiomics.
5. **Brain Networks and Imaging Genetics.** The importance of brain networks in differentiating between healthy and mentally ill subjects, methods on how to estimate the brain network which may or may not rely on additional clinical, demographic and genetic information.
6. **Molecular genetics and population genetics.** Biological backgrounds for statistical genetics, concepts from population genetics that are most relevant to association analysis.
7. **Genetic association studies.** Tests for association, challenges especially in the context of genome-wide association studies (GWAS), including how to correct for population stratification and multiple testing.

These datasets are known to contain vast amount of information, especially when connected together to enhance the power of the biological modeling [2,7].

Genetic information is important in studying cancer, as frequently the process is kickstarted from a small subset of mutations in the genetic code of the cell [5]. These mutations can, via genomic instability, generate a wide variety of mutations in the cancerous cells, often different not only from case to case, but even inside a single case. To address this problem and to find interesting treatment target, identifying the original mutations is necessary, and this requires an in depth analysis of the genome of both healthy and tumoral tissues, possibly across several subjects.

With the increasing demand of resources from ever-growing datasets, it is not favorable to focus on single server execution, and is better to distribute the computation over cluster of less powerful nodes. The computational pipeline also has to manage a high number of subjects, and several steps of the analyses are not trivial to be done in a highly parallel way. Thus, the importance of system statistics management as the efficiency usage of available resources are crucial to reach a compromise between computational execution time and energy cost. For these reasons our main focus is on the performance evaluation of a single subject without using all the available resources, as these could be more efficiently allocated to concurrently execute several subjects at the same time. Due to the nature of the employed algorithms, not all steps can exploit the available cores in a highly efficient way: some scales sublinearly with the number of cores, some have resource access bottleneck. Other tools are simply not implemented with parallelism in mind, often because they are the result of the effort of small teams that prefer to focus their attention on the scientific development side rather than the computational one.

Moreover in order to obtain an optimal execution of bioinformatics pipelines, each analysis step might need very different resources. This means that any suboptimal component of a server could act as a bottleneck, requiring bleeding edge technology if all the steps are to be performed on a single machine. Hybrid systems could be a possible solution to these issues, but designing them requires detailed information about how to partition the different steps of the pipeline. This work explores the different behavior of a recent pipeline on different computing environments as a starting point for this partition.

1.1 GATK-LODn Pipeline

This pipeline has been developed in 2016 by Valle et al. [9], and codifies a new approach aimed to Single Nucleotype Polimorphism (SNP) identification in tumors from Whole Exome Sequencing data (WES). WES is a type of "next generation sequencing" data [1,8,11], focused on the part of the genome that actually codifies proteins (the exome). Albeit known that non-transcriptional parts of the genome can affect the dynamic of gene expression, the majority of cancers inducing mutations are known to be on the exome, thus WES data allow to focus the computational effort on the most interesting part of the genome. Being the exome in human approximately 1% of the total genome, this approach helps significantly in reducing the number of false positives detected by the pipeline. The different sizes of next generation sequencing dataset are shown in Table 1.

The GATK-LODn pipeline is designed to combine results of two different SNP-calling softwares, GATK [6] and MuTect [4]. These two softwares employ different statistical approaches for the SNP calling: GATK examines the healthy tissue and the cancerous tissue independently, and identifies the suspect SNPs by comparing them; Mutect compares healthy and cancerous tissues at the same time and has a more strict threshold of selection. In identifying more SNPs, GATK has a higher true positive calling than Mutect, but also an higher number of false positives. On the other end Mutect has few false positives, but often does not recognize known SNPs. The two programs also call different set of SNPs, even when the set size is similar. The pipeline therefore uses a combination of the two sets of chosen SNPs to select a single one, averaging the strictness of Mutect with the recognition of known variants of GATK.

The pipeline workflow includes a series of common steps in bioinformatics analysis and in the common bioinformatics pipelines. It includes also a sufficient representative sample of tools for the performances statistical analysis. In this way the results extracted from the single steps analysis could be easily generalized to other standard bioinformatics pipelines.

1.2 System Resources Management

As mentioned earlier, a bioinformatics pipeline consists of various steps that could be independent or sequential from each other. Each step could need more or less resources (e.g. memory and threads). So the optimal pipeline execution is closely related to the amount of available resources. The number of samples (patients) to process can penalize performances. There are two main optimization strategies: the first is to improve the efficiency of a single run on a single

Table 1. Typical dataset size for a single patient of different types of next generation sequencing. BAM file size refers to the size of the binary file containing the reads from the machine.

	Coverage	No. of reads	Read length	BAM file size	NGS size
Whole genome	37.7x	975,000,000	115	82 GB	104 GB
Whole genome	38.4x	3,200,000,000	36	138 GB	193 GB
Exome	40x	110,000,000	75	5.7 GB	7.1 GB

patient and the second is to employ massive parallelization on various samples. In both cases we have to know the necessary resources of the pipeline (and in a fine grain the resources of each step) and the optimal concurrency strategy to be applied to our workflow (see Fig. 1). In the analyses we want to highlight limits and efficiencies of the most common computational environments used in big data analytics, without any optimization strategy of the codes or systems.

We also focused on a single patient analysis, the base case study to design a possible parallelization strategy. This is especially relevant for the multi-sample parallelization, that is the most promising of the two optimization strategies, as it does not rely on specific implementations of the softwares employed in the pipeline.

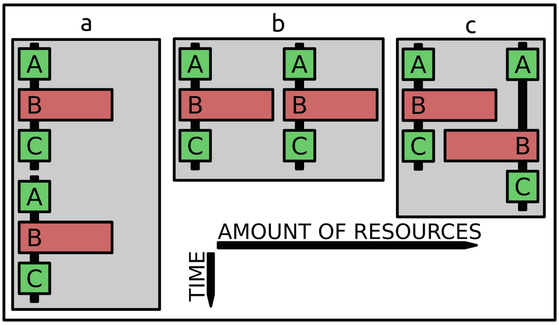

Fig. 1. Examples of concurrency workflow of two processes. The first case (*a*) represents a simple (naive) sequential workflow; the second (*b*) highlights a brute force parallelization; the third (*c*) is the case of a perfect match between the available resources and the requested resources. Often brute force parallelization of pipelines done as in the image *b* ends up overlapping the most computationally intensive steps. Measuring the minimum viable requirements for the execution allow to better allocate resources as seen in the image *c*.

2 Materials and Methods

The pipeline was implemented on 5 computational environments: 1 server grade machine (Xeon E52640), 1 HPC node (Xeon E52683), 2 low power machines

(Xeon D and Pentium J) and one virtual machine built on an AMD Opteron hypervisor. The characteristics of each node are presented in Table 2.

The server-grade node is a typical node used for bioinformatics computation, and as such features hundreds of GB of memory with multiple cores per motherboard: for these reasons we chose it as reference machine and the following results are expressed in relation to it.

The two low-power machines are designed to have a good cost-to-performance ratio, especially for the running cost[1]. These machines have been proven to be a viable solution for high performance computations [3]. Their low starting and running cost mean that a cluster of these machines would be more accessible for research groups looking forward to increase their computational power.

The last node is a virtual machine, designed to be operated in a cloud environment.

The monitoring tool used is Telegraf, which is an agent written in Go for collecting, processing, aggregating, and writing metrics. Each section of the pipeline sends messages to the Telegraf daemon independently.

Regardless of the number of cores of each machine we restrict the number of cores used to only two to compare the statistics: this restriction certainly penalize the environment with multiple cores but with a view of maximizing

Table 2. Characteristics of the tested computational environments. Electrical costs are estimated as 0.25 €/kWh; CPU frequencies are reported in GHz; TDP: Thermal Design Power, an estimation indicator of maximum amount of heat generated by a computer chip when a "real application" runs.

Class	Server grade machines		Low power machines		Virtual machine
CPU	Intel Xeon	Intel Xeon	Intel Pentium	Intel Xeon	AMD Opteron
Version	E5-2683v3	E5-2640v2	J4205	D-1540	6386 SE
Microarchitecture	Haswell	Ivy Bridge EP	Apollo Lake	Broadwell	Piledriver
Launch date	Q3'14	Q3'13	Q4'16	Q1'15	Q3'12
Lithography	22 nm	22 nm	14 nm	14 nm	32 nm
Cores/threads	14/28	8/16	4/4	8/16	16
Base/Max Freq	2.00/3.00	2.00/2.50	1.50/2.60	2.00/2.60	2.80/3.50
L2 Cache	35 MB	20 MB	2 MB	12 MB	16 MB
TDP	120 W	95 W	10 W	45 W	115 W
Total CPUs	2	2	1	1	1
Total cores/threads	28/56	16/32	4/4	8/16	16
Total Memory	256 GB	252 GB	8 GB	32 GB	60 GB
System power	240 + 60 W	190 + 60 W	10 + 2 W	45 + 10 W	115 + 10 W
Electrical costs	650 €/year	550 €/year	26 €/year	120 €/year	273€ /year
System price	4000–6000 €	3000–5000 €	100–130 €	900–1200 €	2000-3000€

[1] Running cost is evaluated as the energy consumption that the node requires per subject, assuming that the consumption scales linearly with the number of cores used in the individual step.

the parallelizations and minimize the energy cost it is the playground to compare all the available environments. Another restriction is applied to the chosen architectures: since available low-power machines provides only x86-architectures also the other environments are forced to work in x86 to allow the statistics comparison.

2.1 Dedicated Bioinformatics Server

The reference node for the tests is one of the servers employed for bioinformatics analyses by the authors. This is a single node with 252 GB memory, 125 TB storage and 2 CPU E5-2640v2, with 16 cores each.

This machine was designed to be able to sustain most commonly performed bioinformatics pipelines, using high volume memory and storage.

2.2 HPC Cluster Hardware Configuration

The HPC cluster is composed by 27 Infiniband interconnected worker nodes, which provide 640 core (Hyperthreaded, E5-2640 cpu), 48 HT cores X5650, 48 HT cores E5-2620, 168 HT cores E5-2683v3, 15 GPUs (8 Tesla K40, 7 Tesla K20, 2 x (4GRID K1)), 2 MICs (2 x Xeon Phi 5100).

A dedicated storage has been setup for the cluster. Storage is accessible by all the nodes through the GPFS file system. In particular the setup includes 2 disks servers, 60 TB of shared disk space, 4 TB for shared home directories. Disks servers are equipped with dual 10 Gb/s Ethernet.

Worker nodes are connected each other via Infiniband (QDR) and are equipped with 1 Gb Ethernet interfaces for storage and network traffic. Home, data and softwares directories are located on a dedicated GPFS file system and shared between all the cluster nodes. The LSF batch system (version 9.1.3) is used to manage job submission to the cluster nodes. The execution environment is shared with a number of other users, therefore in order to measure resource usage, it has been necessary to monitor our jobs from within.

2.3 The Low-power Cluster

The nodes of the cluster are located in a I.N.F.N. facility located in Bologna (Italy) and are based on the current state-of-the-art low-power processors technology. Low power processors are gaining interest in many scientific applicative fields. Designed for the embedded, mobile or consumer market, they are progressively reducing the performance gap with server grade environments, with the added values of keeping a competitive edge on the bill of material and electrical energy costs.

In particular, low power Systems-on-Chip (SoCs) are designed to meet the best computing performance with the lowest power consumption. The SoCs superior performance/consumption ratio is driven by the growing demands for energy-saving boards in mobile and embedded industries. Indeed, the primary design goal for SoCs has been low power consumption because of their use in battery-powered devices or rugged industrial embedded devices. On the contrary, the current server grade CPUs were designed to meet high performance demand required by data center power-hungry clients. Moving away from their embedded and consumer worlds, SoCs are becoming a valid alternative environment for scientific applications without sacrificing too much the performances of server grade CPUs.

The low-power cluster is equipped with nodes based on ARMv7, ARMv8 and x86 low-power environments and is currently used for scientific benchmarks and real-time application tests. Nevertheless, in this work we have only considered x86-based low-power environments because they do not require porting compiling issues and because on the basis of our experience other low-power architectures (i.e. ARM based) are equivalent to x86 low-power platform in term of CPU performance. GPU-enhanced applications can result in a different scenario between ARM and x86 platforms, however, the software pipeline in this work were developed for CPU only.

We chose the following two x86 low-power architectures because they are deployed in different fields of applications: the extremely low-power Intel Pentium J Series (Apollo Lake code name) and the high-performance low-power Intel Xeon D Family (Broadwell code name). We would stress the fact that the Intel Xeon D Family is on the edge of the low-power boundary definition, as shown in the last two rows at the bottom of the Table 2 with the thermal design power (TDP) and median Bill Of Material (BOM) of each platform, but we chose it because it is a natural glue between the low-power platforms and the server-grade platforms.

2.4 Virtual Machine

The virtual machine used in our tests is made available by the project Cloud@CNAF with 16 VCPUs, 60 GB RAM and an attached persistent storage volume of 1 TB. A small list of the benefits from an end-user point of view is: lower computer costs; flexibility and scalability; virtually unlimited storage capacity; increased data reliability; easier group collaboration; device independent.

The Cloud@CNAF IaaS (Infrastructure as a Service) is based on OpenStack, a free and open-source cloud-computing software platform and it has all the services deployed using a High-Availability (HA) setup or in a clustered manner (for ex. using a Percona XtraDB MySQL clustering solution for the deployment of the DBs). It is able to satisfy diversified users needs of compute and storage resources, having available, up to now, 66 hypervisors, with a total of approximately 1400 CPUs, 4 TB of memory and more than 70 TB of storage. The hypervisors range from SuperMicro nodes with 2 × 8 Core AMD Opteron Processor 6320, 64 GB of memory to 2 × 12 AMD Opteron Processor 6238, 80 GB of memory, connected to a PowerVault MD3660i through a GPFS cluster, acting as backend for the cloud VMs ephemeral storage and the persistent, block-storage one.

2.5 Pipeline Steps

The pipeline steps that have been examined are a subset of all the possible steps: we only focus on those whose computational requirements are higher and thus require the most computational power. These steps are:

1. **mapping:** takes all the reads of the subjects and maps them on the reference genome;
2. **sort:** sorts the sequences based on the alignment, to improve the reconstruction steps;
3. **markduplicates:** checks for read duplicates (that could be imperfections in the experimental procedures and would skew the results);
4. **buildbamindex:** indexes the dataset for faster sorting;
5. **indexrealigner:** realigns the created data index to the reference genome;
6. **BQSR:** base quality score recalibration of the reads, to improve SNPs detection;
7. **haplotypecaller:** determines the SNPs of the subject;
8. **hardfilter:** removes the least significant SNPs.

The following statistics were evaluated:

1. **memory per function:** estimate percentage of the total memory available to the node used for each individual step of the pipeline;
2. **energy consumption:** estimated as the time taken by the step, multiplied by the number of cores used in the step and the power consumption per core (TDP divided by the available cores). As mentioned before this normalization unavoidably penalize the multi-core machines but give us a term of comparison between the different environment;
3. **elapsed time:** wall time of each step.

The pipeline was tested on the patient data from the 1000 genome project with access code NA12878, sample SRR1611178. It is referred as a Gold Standard reference dataset [10]. It is generated with an Illumina HiSeq2000 platform, SeqCap EZ Human Exome Lib v3.0 library and have a 80x coverage. As Gold Standard reference it is commonly used as benchmark of new algorithm and for our purpose can be used as valid prototype of genome.

3 Results

Memory occupation is one of the major drawbacks of the bioinformatics pipelines, and one of the greater limits to the possibility of parallel computation of multiple subjects at the same time. As it can be seen in Fig. 2, the memory occupation is comprised between 10% and 30% on all the nodes. This is due to the default behavior of the GATK libraries to reserve a fixed percentage of the total memory of the node. The authors could not find any solution to prevent this behavior from happening. As it can be noticed, in the node with the greatest amount of total memory (both Xeon E5 and the virtual machine) the

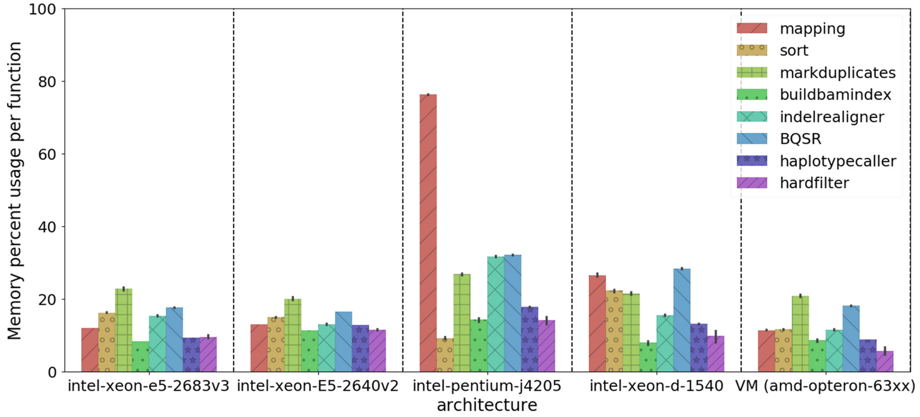

Fig. 2. Memory used for each step of the pipeline. Due to the GATK memory allocation strategy, all steps use a baseline amount of memory proportional to the available memory. Smaller nodes, like the low power ones, require more memory as the baseline allocated memory is not sufficient to perform the calculation.

requested memory is approximately stable, as is always sufficient for the required task. The memory allocation is less stable in the nodes with a limited memory (Xeon D and Pentium J), as GATK might requires more memory than what initially allocated to perform the calculation. The exception to this behavior is the "mapping" step, that uses a fixed amount of memory independently from the available one (between 5 and 7 GB). This is due to the necessity of loading the whole human reference genome (version hg19GRCh37) to align each individual read to it. All the other steps do not require the human reference genome but can work on the individual reads, allowing greater flexibility in memory allocation.

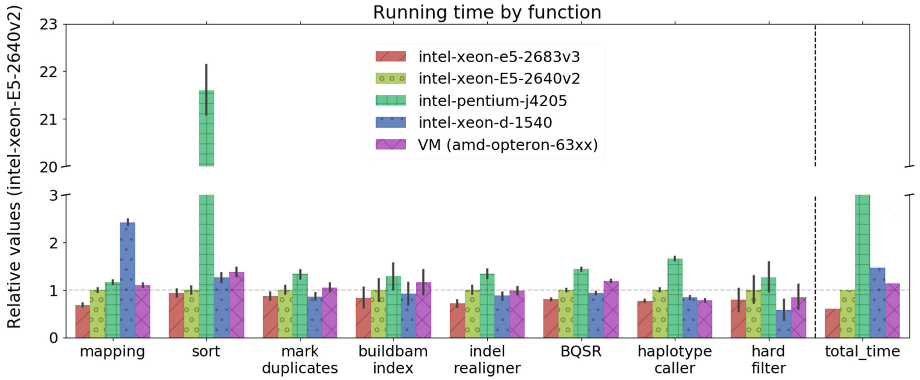

Fig. 3. Time elapsed per step of the pipeline, and total elapsed time. In the sorting step, Pentium J is 20 times slower than the reference, probably due to the limited cache size.

As can be seen in Figs. 3 and 4, this increase of memory consumption does not correspond to a proportional improvement of the time elapsed in the computation.

The elapsed time for each step and for the whole pipeline can be seen in Fig. 3. It can be seen that there is a non consistent trend in the behavior of the different environments. Aside from the most extreme low power machine, the pentium J, the elapsed times are on average higher for the low power and slightly higher for the cloud node, but the time for the individual rule can vary. In the sorting step, Pentium J is 20 times slower than the reference. This is probably due to the limited cache and memory size of the pentium J, that are both important factors determining the execution time of a sorting algorithm and are both at least four to six times smaller than the other machines. The HPC machine, the Xeon E52683, is consistently faster than the reference node.

The energy consumption per step can be seen in Fig. 4. The low power machines are consistently less than half the baseline consumption. Even considering the peak of consumption due to the long time required to perform the sorting, the most efficient low power machine, the pentium J, consumes 40% of the reference, and the Xeon D consumes 60% of the reference. The HPC machine, the Xeon E52683, have consumption close to the low power nodes, balancing out the higher energy consumption with a faster execution speed. The virtual machine has the highest consumption despite the fact that the execution time of the whole pipeline is comparable to the reference due to the high TDP compared to its execution time.

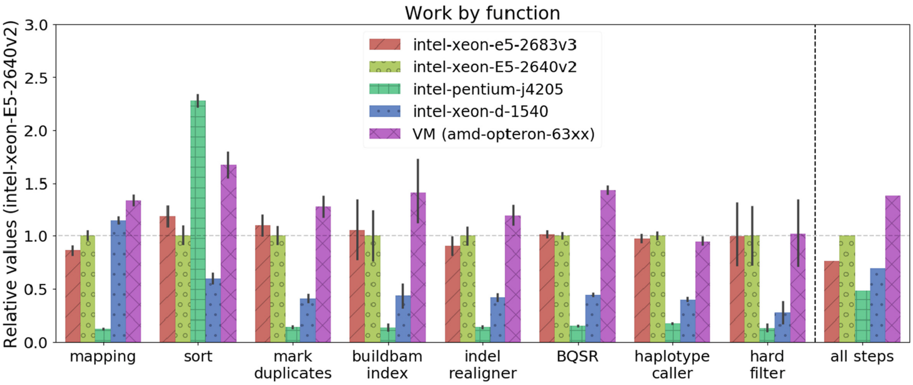

Fig. 4. Energy consumption per pipeline step and on the whole pipeline. Energy consumption is estimated as the time taken by the step, multiplied by the number of cores used in the step and the power consumption per core (TDP divided by the available cores).

4 Discussion and Conclusions

Bioinformatic pipelines are one of the most important uses of biomedical big data and, at the same time, one of the hardest to optimize, both for their extreme requisites and the constant change of the specification, both in input-output data format and program API.

This makes the task of pipeline optimization a daunting one, especially for the final target of the results; physicians and biologists could lack the technical expertise (and time) required to optimize each new version of the various softwares of the pipelines. Moreover, in a verified pipeline updating the software included without a long and detailed crossvalidation with the previous one is often considered a bad practice: this means that often these pipelines are running with underperforming versions of each software.

Clinical use of these pipelines is growing, in particular with the rise of the concept of "personalized medicine", where the therapy plan is designed on the specific genotype and phenotype of the individual patient rather than on the characteristic of the overall population. This would increase the precision of the therapy and thus increase its efficacy, while cutting considerably the trial and error process required to identify promising target of therapy. This requires the pipelines to be evaluated in real time, for multiple subjects at the same time (and potentially with multiple samples per subject). To perform this task no single node is powerful enough, and thus it is necessary to use clusters. This brings the need to evaluate which is the most cost and time efficient node that can be employed.

In the cost assessment there are several factors that need to be considered aside of the initial setup cost, namely cost for running the server and opportunity cost for obsolescence. Scaled on medium sized facilities, such the one that could be required for a hospital, this cost could quickly overcome the setup cost. This cost does also include not only the direct power consumption of the nodes, but also the required power for air conditioning to maintain them in the working temperature range. Opportunity costs are more complex, but do represent the loss of possibility of using the most advanced technologies due to the cost of the individual node of the cluster. Higher end nodes require a significant investment, and thus can not be replaced often.

With this perspective in mind, we surmise that energy efficient nodes present an interesting opportunity for the implementation of these pipelines. As shown in this work, these nodes have a low cost per subject, paired with a low setup cost. This makes them an interesting alternative to traditional nodes as a workhorse node for a cluster, as a greater number of cores can be bought and maintained for the same cost.

Given the high variability of the performances in the various steps, in particular with the sorting and mapping steps, it might be more efficient to employ a hybrid environment, where few high power nodes are used for specific tasks, while the bulk of the computation is done by the energy efficient nodes. This is true even for those steps that can be massively parallelized, such as the mapping, as they benefit mainly from a high number of processors rather than few powerful ones. In this work we focused only on CPUs computation, but another possibility could be an hybrid-parallelization approach in which the use of a single GPU accelerator can improve the parallelization of the slower steps. Each pipeline workflow requires its own analyses and tuning to reach the best performances and the right parallelization strategy based on the use which it is intended but a low energy node approach is emerging as a good alternative to the more expensive and common solutions.

References

1. Behjati, S., Tarpey, P.S.: What is next generation sequencing? Arch. Dis. Child. Educ. Pract. Edition **98**(6), 236–238 (2013). http://ep.bmj.com/lookup/doi/10.1136/archdischild-2013-304340
2. Castellani, G., et al.: Systems medicine of inflammaging. Brief. Bioinform. **17**(3), 527–540 (2016). https://doi.org/10.1093/bib/bbv062
3. Cesini, D., et al.: Power-efficient computing: experiences from the COSA project. Sci. Program. **2017** (2017). http://www.sciencedirect.com/science/article/pii/S0092867400816839
4. Cibulskis, K., et al.: Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. Nat. Biotechnol. **31**(3), 213–219 (2013). http://www.nature.com/doifinder/10.1038/nbt.2514
5. Hanahan, D., Weinberg, R.A.: The hallmarks of cancer. Cell **100**(1), 57–70 (2000). http://www.sciencedirect.com/science/article/pii/S0092867400816839
6. McKenna, A., et al.: The genome analysis toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res. **20**(9), 1297–1303 (2010). https://doi.org/10.1101/gr.107524.110
7. Pooley, R.: Bridging the culture gap. No. 767 (2005)
8. Shendure, J., Ji, H.: Next-generation DNA sequencing. Nat. Biotechnol. **26**(10), 1135–1145 (2008). http://www.nature.com/doifinder/10.1038/nbt1486
9. do Valle, Í.F., et al.: Optimized pipeline of MuTect and GATK tools to improve the detection of somatic single nucleotide polymorphisms in whole-exome sequencing data. BMC Bioinform. **17**(S12), 341 (2016). http://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-1190-7
10. Zook, J.M., et al.: Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. Nat. Biotechnol. **32**, 246 (2014)
11. Zwolak, M., Di Ventra, M.: Colloquium: physical approaches to DNA sequencing and detection. Rev. Mod. Phys. **80**(1), 141–165 (2008)