Abstract

In the last decade, bioinformatics has become an indispensable branch of modern science research, experiencing an explosion in financial support, developed applications and data collection. The growth of the datasets that are emerging from research laboratories, industry, the health sector, etc., are increasingly raising the levels of demand in computing power and storage. Processing biological data, in the large scales of these datasets, often requires the use of High Performance Computing (HPC) resources, especially when dealing with certain types of omics data, such as genomic and metagenomic data. Such computational resources not only require substantial investments, but they also involve high maintenance costs. More importantly, in order to keep good returns from the investments, specific training needs to be put in place to ensure that wasting is minimized. Furthermore, given that bioinformatics is a highly interdisciplinary field where several other domains intersect (such as biology, chemistry, physics and computer science), researchers from these areas also require bioinformatics-specific training in HPC, in order to fully take advantage of supercomputing centers. In this document, we describe our experience in training researchers from several different disciplines in HPC, as applied to bioinformatics under the framework of the leading European bioinformatics platform ELIXIR, and analyze both the content and outcomes of the course.
1. Introduction

In recent years, bioinformatics has undoubtedly become a ‘Big Data’ area of science, requiring increasingly larger computational platforms, known as supercomputers, to extract knowledge from raw data. Such is the case of several fields, for instance in medical [1], comparative genomics [2], molecular biology [3] or chemistry [4]. This scenario has led to an enormous footprint being left in the academic and industrial domains, mostly by influencing institutions to adopt and implement expensive computing centers [5] that are able to process such large amounts of data in shorter running times. Furthermore, this supercomputing trend has even reached out to the political arena, where actions are being taken by international and governmental agencies in order to standardize and promote centralized High Performance Computing in the form of accessible and shared infrastructures (e.g. the European Union, see [6] and [7]).

However, it is estimated that less than 1% of computer scientists have strong expertise in the programming of these massively parallel architectures [8]. This number drops even lower when researchers have no background in computer science, which then introduces an additional obstacle for these platforms to be fully exploited by different domains. Bioinformatics, however, is one of the fields that is able to make use of this supercomputing power, particularly for processing the huge amounts of data generated in genomics and metagenomics studies [9]. Consequently, ELIXIR, the European platform leading the coordination of Life Science resources throughout Europe [10], has been supporting the emerging HPC trend by offering a series of HPC courses targeted specifically at bioinformatics researchers [11]. Such instructive courses are part of the ELIXIR-EXCELERATE [12], Work Package 11 Training Programme, which aims to deliver and improve researchers’ skills to effectively exploit data, tools, standards and the computing infrastructures.

Generally, HPC is taught in a computer scientist-based environment [13], with trivial and numerical use cases that exemplify the benefits of parallelism in terms of increased performance, reduced complexity, higher efficiency, etc. However, such use cases do not represent a ‘real-life’ outcome for researchers from other domains. For instance, from a comparative genomics perspective, it would be interesting to provide domain-related metrics such as how many genomes can be compared at once, of which size and how long it will take to compare them (as opposed to illustrating parallelism with improvements in e.g. throughput). Hence, it is of interest for the researching community to be able to adapt and design HPC courses specific to
other domains, facilitating comprehension through illustrative, domain-related use cases.

In this document, we describe and discuss the deployment, methodology, results and outcomes of the HPC Train-the-Researcher course\(^1\) that was held in the University of Malaga, which addressed the complexities of parallel programming, with emphasis on genome-scale sequence-comparison algorithms use cases. Furthermore, we present this manuscript as a description of the bridging between two distant disciplines from the perspective of teaching a mixed-background audience. We conclude the report with several recommendations and hints obtained through instant and post-course feedback that aim to enhance the outcomes of future training courses from both an organizational, teaching and learning perspective.

2. Methods

In this section, course set up, main targets, required resources and the methodology followed to obtain quality metrics will be described. Additionally, the authors declare that the participants of the course were asked to give feedback and gave their verbal consent for the data to be captured and used in research. The feedback was anonymized, with no personal details shared with a third party. Further details on the given feedback can be found in the Supplementary Material.

2.1. Aims, goals and organisation of the course

The HPC course was held from 6th-7th April, 2017 in Malaga, Spain, as part of the Train-the-Researcher subtask of the ELIXIR-EXCELERATE project. The course was aimed at introducing the participants to the complexities of parallel programming, with an emphasis on genome-scale comparison algorithms.

The theoretical aspects of this course covered (a) the parallel programming background, with an overview of the architectures and programming models; and (b) the basis for genome-scale sequence comparison algorithms, and how parallel schemes can be used to speed up the computation.

The practical aspects were organized to master the concepts of data distribution and balancing using a Map-Reduce [14] strategy and internal coding with MPI [15] and OpenMP [16]. Theory lessons were strongly interleaved with practical sessions in order to reinforce learning and keep high levels of attention [17]. Due to the broad technical level of the course, an a priori survey was conducted to select the

\(^1\)https://otorreno.github.io/hpc-excelerate/.
participants of the course based on their experience in bioinformatics and computer science, their career stage and gender equality.

The course agenda, spanning two days, was split as follows:

- The first day introduced HPC, with basic concepts, access to clusters and scheduling theory. This was then followed with a practical session with exercises on OpenMP and MPI. The day ended with an introduction to sequence database search and dynamic/static data distribution, with associated practical exercises.

- The second day provided the participants with the necessary background on the multiple genome comparison application GECKO [18]. This was then followed by a practical exercise where attendees had to calculate speedup in groups, based on the different parallelization levels. Discussion of results between groups of participants was initiated to account for differences in time executions, in order to notice the impact of heavy computational load in clusters.

Additionally, the course also covered visualization tools that were used in order to aid with the understanding of the domain-dependent results produced by the parallel executions (i.e. the output files of the bioinformatics applications). This was performed using GECKO-MGV [19], a web-based tool to analyze results of pairwise and multiple sequences comparison software. The use of GECKO-MGV was to provide the attendees with a visual understanding of the concepts used during the course, allowing students to interact with the results, ease up the learning curve and motivate them to participate in the learning process.

The full agenda of the course is available through the ELIXIR portal and the ELIXIR Slovenia (ELIXIR-SI) eLearning Platform (EeLP). Further details on the agenda, criteria of the selection process and material can be found within the Supplementary Material.

### 2.2. Deployment of the course

In preparation for the course to take place, computational and logistical resources had to be sourced and allocated. The local logistical resources were provided by the University of Malaga, which included:

- A classroom with a computer, projector and a chalkboard for detailed explanations.

- A broadband internet connection to ensure access to the clusters and to the servers which contained the associated course material.

---

2 [https://www.elixir-europe.org/events/elixir-excelerate-hpc-train-researcher-course](https://www.elixir-europe.org/events/elixir-excelerate-hpc-train-researcher-course).

3 [https://elixir.mf.uni-lj.si/enrol/index.php?id=21](https://elixir.mf.uni-lj.si/enrol/index.php?id=21).
- Video (camera, microphone and speakers) equipment to enable live streaming over the EeLP and ELIXIR-SI Video Conference (VC) system based on ELIXIR Guidelines for synchronous e-learning courses.

Virtual logistical resources were provided by using the ELIXIR-SI EeLP and ELIXIR-SI VC portal. EeLP was used as a virtual single-entry point for tools and services needed for the HPC course. EeLP was thus used for (1) user management, (2) communication/community tools (forum, chat, feedback survey, etc.), (3) course materials, (4) access to the computing resources via embedded Linux terminal and (5) for linking the provided live stream via ELIXIR-SI VC. Live stream, as a separate video stream and content/presentation stream, was produced using the ELIXIR-SI VC portal, maintained by the Arnes Slovenian e-Infrastructures member of ELIXIR-SI. Additionally, the computing resources were provided by ELIXIR-SI through the SLING platform (SLovenian Initiative for National Grid), which included the reservation of a 3-node cluster made up of:

- 96 cores (32 cores/node).
- 192 GB of memory RAM (64GB/node).
- 1.5TB disk space (500GB/node).

These computational resources were allocated for a period of 6 months, in order to enable participants to experiment further with virtual follow-up and other experiments, after the course had closed. In addition, EeLP was used to enrich the traditional (i.e., face-to-face) learning environment. Besides classical learning features (e.g., personalized dashboard, collaborative tools and activities, calendar, file management, notifications, and track progress), EeLP enables automatic user authorization, and embedded Linux terminal. EeLP also simplifies the allocation of HPC resources, which strongly facilitates the organisation and deployment of the course.

Additionally, a registration form was implemented to select candidate students based on their preparation and expertise, enabling both a better uniformization of the background of attendees and a higher level of adaptation from the instructor’s point of view.

### 2.3. Instructional methodology

In this section, we introduce the didactic methodology followed throughout the course. In particular, we briefly present the outline of the syllabus regarding the current pedagogical context.
At a first level, the content of the course is structured into two categories, namely (1) theory lectures and (2) practical sessions. These two approaches were scheduled in alternate order, with theory lectures providing the necessary scaffolding background for the practical sessions. Such organisation of the content follows the First Principles of Instruction [20], which has been widely applied in educational and instructional theory [21]. In particular, the course content was designed to promote (1) task-centered fashion, since students had to tackle real use cases (i.e. demonstration) that were meaningful and related to their bioinformatic background, (2) activation of knowledge by employing incremental lectures and exercises and (3) the application of learned features by integrating them into their environment.

Furthermore, the gap between the two approaches was furtherly complemented with punctual learning methodologies in order to reinforce knowledge transfer. Specifically:

- **Active learning.** The practical sessions were defined as a two-phase exercise in which students had to (1) execute a guided template and (2) making small changes that were necessary to achieve a goal set a priori. This configuration was designed to force students to be experimentally involved, as opposed to passive practical approaches where students follow execution guides [22].

- **Reflective learning.** The final practical sessions were designed to promote practice reflection [23] of students upon what they had learned previously. This approach was scheduled as part of the latter practical session, where students were requested to devise new parallelisation improvements within a given piece of code. Groups were then formed to increase peer interactivity [24] and time was given to enable reflection and discussion of ideas, as well as a final time slice for implementation, testing and defense of achieved results.

### 2.4. Feedback procedure

Two feedback procedures were implemented to retrieve the opinion of attendees. These followed two different schemes: (1) instant feedback, with the ultimate goal of online overwatching of the performance of the attendees and (2) short-term, post-course feedback for a more detailed evaluation of the overall course.

The instant feedback was collected using the “show of hands” feedback method [25]. The instructor would ask the question “How confident are you about the current subject?”, giving choices of the different levels, and every attendee would raise their hands depending on the level of confidence they had about the last subject taught. This procedure helped to adjust the pace of delivery of the course material while periodically stimulating the participants to reflect on their own individual learning experience, thus contributing to their building of justifiable self-confidence. Conversely, the short-term, post-course feedback was collected by an online,
anonymous survey that was made up of questions that covered the participants' research area, satisfaction of the course and topics and future use of the covered resources. A feedback survey was originally developed by the ELIXIR Training Platform’s Quality and Impact Coordinator, via consultation with the ELIXIR Training Coordinators, and an extended survey based on this one was provided by the ELIXIR-SI node to include questions related to the internals of the course. The possible answers on this survey were categorical (Yes/No) and numerical using a Likert scale [26] from 1 to 5, enabling the production of rich metrics. The survey was implemented and distributed to participants electronically on the ELIXIR-SI EeLP, using the EeLP embedded survey module.

3. Results and discussion

In this section, the results that were obtained from the course are analyzed (i.e. the output) through both sources of feedback (instant and short-term) and personal experience from the instructors. Advantages and weaknesses of the methodology are identified in order to generate a series of guidelines and necessary steps to be considered for future events. Regarding the course, full attendance was achieved, and the audience research background revealed a variety of fields, such as mechatronics, mathematics, robotics, metaheuristics, biology or computer science. Further details on the audience can be found in the Supplementary Material.

3.1. Analysis of the strengths and weaknesses

There are several conclusions that can be drawn from the experience of the course, taking into account the anonymous feedback (surveys composed of open-answer questions and ratings), as well as data from the instant feedback (confidence levels) that took place throughout the course. These include:

3.1.1. New programming paradigm

One of the most difficult steps to overcome when learning HPC is to understand the parallel programming paradigm, i.e., programmers, especially those who do not possess a background in computer science, usually think of their programs as a sequential execution of tasks, especially in bioinformatic applications where step-by-step workflows and pipelines are very common. However, HPC introduces new concepts such as thread synchronization and load balancing [27] which require programmers to think about several other collateral effects (e.g. the order in which instructions are executed, how to deal with blocking mechanisms, exclusive and critical code, etc.). Fully understanding these concepts is a key point as it helps the student to get a better grasp of the difference between programming paradigms.
Therefore, it becomes very important to focus on these ground concepts to make attendees get the most out of them.

### 3.1.2. Incremental difficulty

HPC follows a curve-like learning scheme, where future taught content is based on previously taught content. Therefore, it is essential that some concepts are learned prior to learning further concepts, with content building up together in a sequential manner. This fact is supported by Fig. 1, which depicts that concepts related to HPC are mostly concentrated on the top of the chart (lower understanding rating) whereas domain concepts (such as “Database search” or “GECKO sequential execution”) are concentrated on the higher rating part of the chart. Therefore, it is important that the basic HPC concepts are taught first, in order to ensure that the students are able to understand the higher-level concepts.

### 3.1.3. Theory-practice balance

The amount of theory taught should be well balanced with the amount of practical lessons. Theoretical parts of the course are attention-intensive for students, whereas practical examples that typically require more time to complete, strongly help attendees to consolidate the theoretical knowledge by integrating it into the environmental reality of students (see [22]). Furthermore, the interleaved order of lectures and practical sessions enabled students to apply previously demonstrated theory, as opposed to employing a sequential two-parts course (e.g. first only theory, then practice). Consequently, theory lectures should be distributed well alongside

**Fig. 1.** Users submitted ratings from 1 to 5, with 1 being the lowest and 5 being the highest, to describe the level of confidence regarding each session, as retrieved from the instant-feedback method. The x-axis shows the percentage of each rating from the total. Notice that the categorical y-axis is sorted from low to high rating.
practical examples in order to maximize the laying of foundational knowledge on which further learning can build upon.

3.1.4. Adequate examples

During the practical examples, it was noticed that since they are somewhat computationally-intensive (although only toy executions), they allowed students to observe, in real-time, the difference between running applications in a sequential manner versus in a parallel fashion. Although this can be difficult to set up, mostly due to limited resources, it certainly becomes interesting to design courses considering two-phase time frames, with the theory being taught while the executions are performed. Such schemes are a simple matter of timing executions with theory lectures, allowing students to notice the speedup whilst easing off the heavy theory load.

3.1.5. Simple but non-trivial examples

Along the course, it was noted that to illustrate features in HPC it was important to use examples that were easy to understand, and which therefore reduced intrinsic cognitive load to the minimum [28, 29], effectively enabling students to focus into the incorporation of new learning schemas. Furthermore, avoiding switching between domains or introducing new domain-related concepts can improve the attention span due to reduction of extraneous cognitive load. On the other hand, these examples should have well-defined goals since (according to goal-setting theory [30]), learning performance can drop substantially as a function of the specificity and clarity of tasks. Such scenario can easily occur due to the integration of different scientific fields (i.e. biology/bioinformatics with High Performance Computing). For example, students can find it hard to understand how a map-reduce strategy works if they do not understand what they are trying to achieve in the first place (e.g. comparative genomics). Therefore, examples should be detailed and justified enough before students can focus in the domain being taught.

3.2. Notes and recommendations for multidisciplinary bioinformatic courses

In this section, we discuss and address from a general perspective both notes and recommendations for the design and deployment of a multidisciplinary bioinformatic course. We briefly include here key aspects for the design of the course:

- Given the interdisciplinary nature of the course, it is appropriate to adapt the content to address the “how-to” rather than the “why”, since students will be much more interested in applying their new skills to their careers and jobs.
Lightness of the “why”: The introduction of the domain being learnt should be light and general, while still wide enough to enable interested students in learning further concepts.

Consider the computational aspects, particularly cluster size and capacity and connection, since it is very easy to overlook these details and they can cause severe interruptions in class, such as undefined behaviour of executions due to limited resources or aborted connections which cause work needing to be redone. Working in pairs or separate executions by groups is recommended if there is not enough computing power.

Consider performing an initial survey on the expertise for the domain and select candidates upon interest and career.

Consider which metrics are suitable for measuring the desired outcome (e.g. short and long term, online feedback, etc.) in respect to the set goals and funding institutions.

Consider if a follow up for students is beneficial and if it should be optional or mandatory. If follow up is included, consider also for how long the computational resources should be online after the course has been deployed.

On the other hand, regarding the deployment of the course:

Avoid long lectures without immediate practical sessions and always attempt to accompany new theory concepts with practical examples to promote activation of knowledge.

Practice is a must since it is applied to a particular domain (parallelism to biology). In particular:

- Practice must include templates but also self-doing (reflective learning).
- Interleave practice with theory lectures regularly, i.e. not all at once sequentially, taking into account that our working memory is very limited.
- Although difficult, set rewards based on acceleration of programs, since these are useful in their domain and directly represent an improvement in their careers.

Collaborative tasks should be included as they provide pedagogical benefits from two perspectives:

- Learning perspective: cooperative learning [31] is known to help consolidate knowledge by discussing, engaging in peer-critique and demonstrating newly acquired skills.
Organisational perspective: Cooperation can reduce the amount of executions which will typically take place on a rented cluster and therefore computing power will be limited.

Additionally, it is recommended to have several other people involved in the team (i.e. assistants) when deploying courses that make use of command line programs, coding or compiling, since students will naturally make mistakes that can be difficult to find, extremely time consuming and whose learning value is nearly zero at the given context, such as an execution not working because they are working from a different directory. For further details, see the Supplementary Material.

3.3. Post-course feedback survey

A collection of questions that aimed to measure the outcome and satisfaction of the attendees was employed. These questions are useful to evaluate how well the allocated resources were utilized, and hence can be used as a generalized metric. The following questions were included:

- Overall rating of the course (1–5): This question aims to obtain a metric at the personal satisfaction level that sums up the overall impression left from the course, which includes feelings about the accommodation to the taught content, etc.

- Sufficient networking time (Yes/No): This question is intended for researchers and students to evaluate (besides the lessons) whether there was enough time to liaise with other fellow attendees and/or the trainers to establish connections and strengthen working relationships.

- Expectations met (1–5): This question rates whether attendees, after finishing the course, believed or not that it had fulfilled their pre-course expectations, i.e. they had received the theory and practical lessons that initially attracted them towards the course.

- Recommendation rating (Yes/No): The recommendation rating evaluates not only the satisfaction of the attendee at a personal level, but also makes him or her rate whether attendees would suggest this as a suitable course for their peers to attend. Therefore, this question attempts to make attendees to think in a social fashion how would the course impact other people they know.

- Course organization rating (1–5): This question evaluates the organization, schedules and timing of the lessons, the practical sessions, coffee and lunch breaks. This is a critical aspect, since a bad organization can turn a successful course into a dense, frustrating experience.
The ratings for all these questions are collected and represented in Fig. 2. Each pie chart shows the ratings for a particular question, in percentages over the students that attended the course. Although the questions regarding overall rating and rate of recommendation appear to be similar, these two are of special interest in combination: the overall rating will often be dependent on the knowledge and expertise of the student itself (and hence a biased estimator), whereas the recommendation rate includes the hidden question “Would your colleagues take any benefit from the course?”, since it is an empathetic question that attempts to extrapolate the individual experience towards their peers.

Therefore, the combination of the two rates reduces the bias of the student, since it forces to think what is appropriate at both personal and peer level, and therefore gives a better estimate of the course value for a more general audience. Supporting this hypothesis, it can be observed that the rating of expectations met is almost a mirror of the overall rating of the course, since the expectations are likely to be biased in the same way that the overall rating is (i.e. the rating given on a course will depend mostly on whether my personal expectations are met or not).

Regarding the sufficient networking time rating, this is a highly important question for international projects, as it reveals whether researchers could or not establish
contact with other fellow researchers. Such initial contacts (produced by sufficient networking time) is a first step for long-lasting collaborations, which is largely one of the ultimate goals between international, multidisciplinary research platforms. In the case of this particular training course, it can be observed that mostly every attendee agreed on the amount of break times. Nonetheless, this is a feature that can be adjusted online (increasing or decreasing the break time) by observing students and by balancing the amount of time dedicated to each lesson (see the Full Agenda for further details on the course organisation), and hence an improved outcome can be obtained. Lastly, the course organization rating provides a positive valuation of the internal organisation of the course, that is, students were generally satisfied with the organisational constraints such as resource allocation, time scheduling, location, etc.

Moreover, this scenario suggests that questions regarding the assessment of quality metrics in courses should be detailed and specific in such a way that there is as less overlap as possible between questions.

4. Conclusions

In this document, we have described a didactic method to tackle HPC training for researchers and students with virtually no background in computer science. We have analyzed the quality metrics employed to obtain an estimate of the satisfaction of the attendees, which are also employed as an overwatch metric for funding institutions in the form of feedback statistics. Furthermore, we have discussed how an HPC programme could be improved in order to ease the learning curve and how to avoid pitfalls that are difficult to predict. In general:

- We have observed a pattern of incremental difficulty in the lessons that comprise an introduction to HPC. Moreover, this is motivated by the large number of domains that can benefit from HPC, and hence the background and expertise of attendees might vary strongly and sufficiently to cause difficulties in order to keep the level of attention uniform. This suggests that courses should be broken down into small lessons with a time break between them, in order to facilitate the absorption of contents. Moreover, a precise description of the prerequisites needed to understand and follow the course will help possible candidate students situate themselves in the spectrum of target audiences. If possible, such description can be accompanied with material previous to the course, aimed at introducing the main ideas of the course in a general and simple way.

- Concepts must be consolidated through both theory and practice, but these should be carefully balanced. Theory lessons should be kept simple, with the aim of introducing concepts, whereas practice lessons should be aimed at
consolidating knowledge, and if possible, allowing trial and error scenarios, which help students understand the reasons and motivation of the theory that is being learnt.

- Use cases should be as real as possible (while keeping them controlled), avoiding scenarios that will not exemplify the benefit of the HPC techniques in use. However, using real-life examples might produce a considerable slowdown in the executions. To overcome this limitation, timing exercise executions with practical lessons are a key point to take into consideration.

- If the course is intended to teach HPC applied to a certain domain, the examples utilized in the use cases must also attempt to be simple in their understanding, in the sense that a short number of non-HPC concepts related to the applied domain should be used. That is, new concepts that are not essential to the learning process should be kept to a minimum in order to let students focus.

Additionally, we have also discussed that the background and expertise of the attendees plays a key role in the organization and success of the course. Consequently, it is suggested that pre-course surveys are performed on the background of incoming students, with the aim of adapting the complexity of the teaching lessons and the Use Cases to fit the scope of both the course and the students, as much as possible. Furthermore, it is important to keep in mind the target audience of the course (For what type of student was it designed?), since content must be adapted depending on attendees at both professional (developers, researchers, business people, etc.) and academic level (undergraduates, master students, doctorates, etc.). A certain student, e.g. a developer, might look for concrete things, such as algorithms or programming techniques, whereas a doctorate student might be looking for concepts and paradigms that could be applied to their own research. In this line, the goals of the students will therefore affect their dedication to the course, since a researcher that needs specific training for continuing his or her research will dedicate more effort than one which is just naturally curious or interested. Such division of goals is related to the effectiveness of a virtual follow-up. More dedicated students aiming to complete their research will probably make more use of a follow-up platform to consolidate knowledge, whereas curious students might not if their expectations were not highly met.

However, virtual follow-ups should be carefully planned and balanced: a constraining follow-up that requires extensive work by the student might keep him or her away from attending the initial course, whereas a voluntary follow-up might not attract assiduous students. In this line, it would also be beneficial to organize a short (e.g. from one to a few hours) virtual post-course video conference meeting with the participants between 14 days to 3 months after the course. This scenario would enable participants to ask additional questions (particularly if these could not be solved among other students in the electronic platforms). Additionally, it would
provide valuable feedback for upcoming courses, especially in the sense of discovering what points are students most troubled with. Long term feedback survey could also be done during such meeting.

Furthermore, we have shown and discussed that surveys (both instant and post-course) are a key tool for the teachers and organizers to obtain information on how the course is developing and the impact it leaves behind. From the experience gathered in this course, the authors of this manuscript highly recommend carrying out at least two types of surveys, and if possible, all three of them:

- **Instant feedback.** This procedure is mostly aimed at obtaining information in real time, particularly to evaluate the confidence of students on the current subject, and hence, allow to make adjustments in upcoming sessions based on it.

- **Short-term feedback.** It is advisable to perform a short-term questionnaire right after the course finishes, as this will enable students to leave their opinions when they have completed the full course and remember the most from it, as opposed to post-course feedback.

- **Post-course feedback.** On the other hand, post-course feedback will help gather statistics of the long term impact of the course, generate metrics and overall enable the organizers (and moreover, funding institutions) to evaluate the impact and future continuity of the course. In this case, it is advisable to not overextend questionnaires as there is no obligement feeling from the students towards filling up the questionnaire.

Still, it is also recommended to include short *a priori* surveys on the expertise of students to get a grasp of the general picture in terms of their preparation and capabilities. If this information is taken into account, the adaption of the upcoming material to the needs of students could positively affect the overall outcome of the course.

**Declarations**

**Author contribution statement**

Esteban Pérez-Wohlfeil: Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Oscar Torreno: Conceived and designed the experiments; Performed the experiments; Contributed reagents, materials, analysis tools or data.

Louisa J. Bellis, Pedro L. Fernandes, Brane Leskosek: analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Oswaldo Trelles: Conceived and designed the experiments; Analyzed and interpreted the data.
Funding statement

This work was partially supported by the European project ELIXIR-EXCELERATE (grant no. 676559), the Spanish national projects Plataforma de Recursos Biomoleculares y Bioinformáticos (ISCIII-PT13.0001.0012) and RIRAAF (ISCIII-RD12/0013/0006), the Instituto de Investigación Biomédica de Málaga IBIMA and the University of Málaga.

Competing interest statement

The authors declare no conflict of interest.

Additional information

Supplementary content related to this article has been published online at https://doi.org/10.1016/j.heliyon.2018.e01057.

Acknowledgements

The authors would like to thank all the coworkers and colleagues from the ELIXIR platform that have contributed in the planning, development and deployment of the course.

References

[1] M. Vázquez, R. Arís, G. Houzeaux, R. Aubry, P. Villar, J. García-Barnés, D. Gil, F. Carreras, A massively parallel computational electrophysiology model of the heart, Int. J. Numer. Meth. Biomed. Eng. 27 (12) (2011) 1911–1929.

[2] L. Heshan, X. Ma, W. Feng, N. Samatova, Coordinating computation and I/O in massively parallel sequence search, IEEE Trans. Parallel Distrib. Syst. 22 (4) (2011) 529–543.

[3] P. Sander, S. Páll, R. Schulz, P. Larsson, P. Bjelkmar, R. Apostolov, M. Shirts, J. Smith, P. Kasson, D. Spoel, B. Hess, E. Lindahl, GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit, Bioinformatics 29 (7) (2013) 845–854.

[4] K. Michio, T. Nakajima, MPI/OpenMP hybrid parallel algorithm of resolution of identity second-order Möller–Plesset perturbation calculation for massively parallel multicore supercomputers, J. Chem. Theor. Comput. 9 (12) (2013) 5373–5380.
[5] K. Volodymyr, P. Trancoso, Trends in high-performance computing, Comput. Sci. Eng. 13 (3) (2011) 92–95.

[6] DIGIBYTE. EU ministers commit to digitising Europe with high-performance computing power. Available from: https://ec.europa.eu/digital-single-market/en/news/eu-ministers-commit-digitising-europe-high-performance-computing-power. (Last accessed 20 December 2017).

[7] Khalil, R. High Performance Computing and the Digital Single Market strategy review. Available from: https://ec.europa.eu/digital-single-market/en/blog/high-performance-computing-and-digital-single-market-strategy-review. (Last accessed 20 December 2017).

[8] Parsons, M. Multiple challenges for multicore processors. Available from: https://sciencenode.org/feature/opinion-multiple-challenges-multicore-processors.php. (Last accessed 20 December 2017).

[9] E. Van Dijk, H. Auger, Y. Jaszczyszyn, C. Thermes, Ten years of next-generation sequencing technology, Trends Genet. 30 (9) (2014) 418–426.

[10] L. Crosswell, J. Thornton, ELIXIR: a distributed infrastructure for European biological data, Trends Biotechnol. 30 (5) (2012) 241–242.

[11] Training eSupport System. ELIXIR EUROPE. Available from: https://tess.elixir-europe.org/events?q=HPC. (Last accessed 20 December 2017).

[12] ELIXIR-EXCELERATE Project. Available from: https://www.elixir-europe.org/about-us/how-funded/eu-projects/excelerate. (Last accessed 20 December 2017).

[13] ACM Joint Task Force, Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, Technical report, Association for Computing Machinery (ACM) IEEE Computer Society, 2013.

[14] J. Dean, S. Ghemawat, MapReduce: simplified data processing on large clusters, Commun. ACM 51 (1) (2008) 107–113.

[15] W. Gropp, E. Lusk, N. Doss, A. Skjellum, A high-performance, portable implementation of the MPI message passing interface standard, Parallel Comput. 22 (6) (1996) 789–828.

[16] L. Dagum, R. Menon, OpenMP: an industry standard API for shared-memory programming, IEEE Comput. Sci. Eng. 5 (1) (1998) 46–55.

[17] F. Korthagen, J. Kessels, B. Koster, B. Lagerwerf, T. Wubbels, Linking Practice and Theory: the Pedagogy of Realistic Teacher Education, Routledge, 2001.
[18] O. Torreno, O. Trelles, Breaking the computational barriers of pairwise genome comparison, BMC Bioinf. 16 (2015) 250.

[19] S. Diaz-del-Pino, P. Rodriguez-Brazzarola, E. Pérez-Wohlfeil, O. Trelles. Combining strengths for multi-genome visual analytics. Bioinf. Biol. Insights (In press).

[20] M. Merrill, M. David, First principles of instruction, Educ. Technol. Res. Dev. 50 (3) (2002) 43–59.

[21] C. Reigeluth, Instructional theory and technology for the new paradigm of education, RIED. Rev. Iberoam. Educ. Distancia 32 (2012) 1–18.

[22] S. Freeman, L. Sarah, E. Miles, M. Smith, N. Okoroafor, H. Jordt, M. Wenderoth, Active learning increases student performance in science, engineering, and mathematics, Proc. Natl. Acad. Sci. Unit. States Am. 111 (23) (2014) 8410–8415.

[23] N. Thompson, J. Pascal, Developing critically reflective practice, Reflective Pract. 13 (2) (2012) 311–325.

[24] S. Rodger, An interactive lecture approach to teaching computer science, ACM SIGCSE Bull. 27 (1) (1995) 278–282.

[25] G. Sisson, Hands-on Training: a Simple and Effective Method for on the Job Training, Berrett-Koehler Publishers, 2001.

[26] M. Matell, J. Jacoby, Is there an optimal number of alternatives for Likert scale items? Study I: reliability and validity, Educ. Psychol. Meas. 31 (3) (1971) 657–674.

[27] D. Kuck, G. Grest, S. McKay, W. Christian, High performance computing—challenges for future systems, Comput. Phys. 11 (3) (1997) 259–261.

[28] J. Sweller, Cognitive load theory, learning difficulty, and instructional design, Learn. InStruct. 4 (4) (1994) 295–312.

[29] P. Ginns, Integrating information: a meta-analysis of the spatial contiguity and temporal contiguity effects, Learn. InStruct. 16 (6) (2006) 511–525.

[30] E. Locke, Motivation through conscious goal setting, Appl. Prev. Psychol. 5 (2) (1996) 117–124.

[31] D. Johnson, R. Johnson, K. Smith, Active Learning: Cooperation in the College Classroom, Interaction Book Company, 1998.