Universitat Autònoma de Barcelona

2023/2024

Programming in Bioinformatics

Code: 42401
ECTS Credits: 6

Degree	Type	Year	Semester
4313473 Bioinformatics	OB	0	1

Contact

Name: Javier Panadero Martinez

Email: javier.panadero@uab.cat

Teaching groups languages

You can check it through this link. To consult the language you will need to enter the CODE of the subject. Please note that this information is provisional until 30 November 2023.

Teachers

Javier Panadero Martinez

Prerequisites

For the general development of the course, it is recommended to have a B2 level, or equivalent, of the English language.

For this module, it is recommended to have basic notions of computer usage in Linux (i.e., knowledge of common user tools and file manipulation).

Objectives and Contextualisation

General objectives of this module are the application of the core tools and basic techniques for development in this area of knowledge. Provide skills to successfully assume the adaptation to changing technologies and new paradigms emerging in this interdisciplinary field.

Competences

- Design and apply scientific methodology in resolving problems.
- Identify the biocomputing needs of research centres and companies in the biotechnology and biomedicine sectors.
- Possess and understand knowledge that provides a basis or opportunity for originality in the development and/or application of ideas, often in a research context.

- Student should possess the learning skills that enable them to continue studying in a way that is largely student led or independent.
- Use operating systems, programs and tools in common use in biocomputing and be able to manage high performance computing platforms, programming languages and biocomputing analysis.
- Work individually and as part of a team in an international and multidisciplinary context.

Learning Outcomes

1. Design and apply scientific methodology in resolving problems.
2. Design, analyse and evaluate the performance of parallel infrastructures and large volumes of data.
3. Identify the advantage and limitations of biocomputing and the importance of applying new computer technology in omic research.
4. Manage parallel platforms and biocomputing databases according to needs.
5. Possess and understand knowledge that provides a basis or opportunity for originality in the development and/or application of ideas, often in a research context.
6. Student should possess the learning skills that enable them to continue studying in a way that is largely student led or independent.
7. Use algorithms and statistical calculation techniques to manage large volumes of data.
8. Using R and Python programming languages in bioinformatics
9. Work individually and as part of a team in an international and multidisciplinary context.

Content

1. Linux (commands and shell scripting)

Basic commands, user management, software management, and file system
Text processing tools and data manipulation
Redirections, pipes, and filters
Shell scripting in Bash

2. Programming languages

Introduction to programming in Python in Bioinformatics
Variables, expressions, data types, operators, programming constructs, and contexts
Code reutilization: functions, modules, and subroutines
Recursive programming
Input/Output
Code debugging
Other programing languajes: R

3. Data structures and data processing

Basic data structures (including strings, lists, tuple, sets, and dictionaries)
Nested data structures and objects
Trees and graphs
Modelling and representing bioinformatics data
Basic bioinformatics data formats (including FASTQ, SAM, VCF)
Regular expressions

4. Algorithms in bioinformatics

Introduction to algorithm complexity
Divide and conquer algorithms
Combinatorial enumeration and backtracking
Dynamic programming

5. Bioinformatics libraries and tools

Data visualisation tools
Introduction to Biopython
Introduction to NumPy and Pandas

Methodology

The methodology will combine classroom work, supervised problem-solving in class, unsupervised work in the computing lab, homework from recommended readings and independent study student. It will use the virtual platform and asked for papers related to the thematic blocks.

Annotation: Within the schedule set by the centre or degree programme, 15 minutes of one class will be reserved for students to evaluate their lecturers and their courses or modules through questionnaires.

Activities

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Classroom work	20	0.8	1, 2, 4, 7, 6, 5
Problem solving (in class)	14	0.56	1, 7, 6, 5
Work in the computing lab	12	0.48	1, 4, 3, 7, 5, 9
Type: Supervised			
Performing lab work from recommending reading	15	0.6	1, 3, 6, 5, 9
Type: Autonomous			
Regular general work on the deliverables definition and materials given	83	3.32	1, 2, 7, 6, 9

Assessment

The methodology will combine classroom work, problem solving in the classroom, unsupervised work done in the computing lab and individual work from recommended readings. It will make use of the virtual platform and will make references to selected publications related to the thematic blocks. None of the individual assessment activities will account for more than 50% of the final mark.

Retake exam

To be eligible for the retake process, the student should have been previously evaluated in a set of activities equaling at least two thirds of the final score of the module. The teacher will inform the procedure and deadlines for the retake process. Please note that activtities performed within class cannot be recuperated.

Not valuable

The student will be graded as "Not Valuable" if the weight of the evaluation is less than 67% of the final score.

This subject/module does not offer a single assessment evaluation.

Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
Evaluation of work done during the module, presented by the user	10%	1	0.04	1, 3, 7, 6, 9
Final Exam	30%	1	0.04	1, 6, 5
Individual theoretical and practical tests	50%	2	0.08	1, 2, 4, 6, 5, 8
Laboratory work, possibly in groups	10%	2	0.08	1, 2, 4, 3, 7, 6, 9

Bibliography

- Rajkumar Buyya, "High Performance Cluster Computing: Programming and Applications", PH, 1999.
- Bell, Charles; Kindahl, Mats; Thalmann, Lars. "MySQL High Availability". O'Reilly, 2010.
- Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., Rapp, B. A. & Wheeler, D. L. (2002).
- GenBank. *Nucl. Acids Res.*, 30(1):17-20. URL http://nar.oupjournals.org/cgi/content/abstract/30/1/17.
- Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. (2000).The Protein Data Bank. *Nucleic Acids Res.*, 28(1):235-242.
- Bessant, C., Shadford, I., Oakley, D. "Building Bioinformatics Solutions with Perl, R and MySQL", Oxford University Press, 2009
- Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.-C., Estreicher, A., Gasteiger, E., Martin, M. J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S. & Schneider, M. .The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.*, 31(1):365-370.
- Christiansen, P., Wall, L., Orwant, J., "Programming Perl". 4th Edition, O'Reilly, 2012
- Mäkinen et al., Genome-Scale Algorithm Design: Biological Sequence Analysis in the Era of High-Throughput Sequencing. Cambridge Univ. Press, 2015.
- Matloff, N., "The Art of R Programming". No Starch Press Inc., 2011
- Lutz, M., "Learning Python", O'Reilly, 5th edition, 2013
- Siever, E., Figgins, S., "Linux in a nutshell" O'Reilly 2009.
- Sobell, M., "A Practical Guide to Linux. Commands, editors and shell programming". Prentice Hall, 2009.
- Tindall, James., Begining Perl for Bioinformatics. O´Reilly 2012.

Recommended websites

- http://mscbioinformatics.uab.cat
- https://cv.uab.cat

Search for bioinformatics and computer science topics in UAB library e-book resources:

- http://www.uab.cat/biblioteques/trobador
- http://pagines.uab.cat/bctdigital

Software

Linux (Ubuntu, Bash, linux-tools, etc)
Python 2.7/3.x
Jupyter Notebook / PyCharm

R/RStudio
Matplotlib/Seaborn
Numpy/Pandas