GHENT UNIVERSITY

Course Specifications

Valid as from the academic year 2023-2024

Computational Biology (C003789)

Course size *(nominal values; actual values may depend on programme)*

Credits 6.0 **Study time 180 h**

Course offerings and teaching methods in academic year 2023-2024

A (semester 2)	Dutch	Gent	seminar
			lecture

Lecturers in academic year 2023-2024

Dawyndt, Peter	WE02	lecturer-in-charge

Offered in the following programmes in 2023-2024	**crdts**	**offering**
Bachelor of Science in Computer Science	6	A
Master of Science in Teaching in Science and Technology(main subject Mathematics)	6	A
Master of Science in Mathematics	6	A

Teaching languages

Dutch

Keywords

Computational methods, molecular biology, genome structure, gene prediction, sequence alignment, phylogenetics, comparative genomics, gene expression analysis

Position of the course

Where did SARS come from? Have we inherited genes from Neanderthals? How do plants use their internal clock? To what extent do bacterial species exchange genetic material? The genomic revolution in biology enables us to answer such questions. But the revolution would have been impossible without the support of powerful computational and statistical methods that enable us to exploit genomic data (1.3). Solving the current and future problems in this intriguing field (1.2) requires training of a next generation of researchers that are fluent in the languages of mathematics, computer science and biology (3.1,3.2,3.5).
Discover the world of computational genomics through a number of real case studies that are worked out in Matlab/Python. These cases will allow the students to gradually work themselves into the terminology and basic principles of molecular biology, and give them a flavour of the different computational techniques used in this ever-expanding research domain (2.1,2.2,2.4). This course provides a roadmap to navigate entry to this field (1.6,5.1,5.5). It guides the students through key achievements of bioinformatics, using a hands-on approach. Statistical sequence analysis, sequence alignment, hidden Markov models, gene and motif finding and more (1.1), are introduced in a rigorous yet accessible way.

Contents

- Introduction to cell biology
- Sequence statistics
- Gene finding
- Sequence alignment
- Probabilistic sequence models (hidden Markov models)
- Genetic variation within and between species
- Evolution: natural selection at the molecular level
- Phylogenetic analysis
- Whole genome comparisons
- Gene expression patterns
- Identification of regulatory sequences

Initial competences

An understanding of basic probability theory, algorithmic concepts and programming skills are essential. Prior experience in the Python programming language and being able to work with modules for scientific computing (numpy, scipy, matplotlib) are required. No previous knowledge of molecular biology will be assumed, as time will be spent throughout the course to elaborate on the biological background. It is primordial to have an certain interest in the biological aspects of the problem domain.

Final competences

1 Understand statistical and algorithmic approaches to sequence and gene expression analyis.
2 Understand why and how mathematics and computer science can play a role in modern biology.
3 Read and understand recent articles on genomic topics, and in particular assess the data-analysis aspects of them.
4 Be familiar with the standard tools and problems, and current and future goals of the research domain of computational biology.
5 Access and manipulate real genomic data.

Conditions for credit contract

Access to this course unit via a credit contract is determined after successful competences assessment

Conditions for exam contract

This course unit cannot be taken via an exam contract

Teaching methods

Seminar, Lecture

Extra information on the teaching methods

Guided practical sessions in PC room, with regular reporting of the executed analysis.

Learning materials and price

Course book: Bioinformatics algorithms: an active learning approach. Active Learning Publishers. 2 volumes: ISBN-13:978-0990374619 and ISBN-13:978-0990374626.

Computing cluster with pre-installed software that is required for the practical courses.
Additional material is made available through the electronic-learning environment Ufora
- Powerpoint slides used during lectures
- (publicly) available software tools
- scientific journals related to computational biology
- informative websites

Cost: 0 EUR

References

Durbin R., S. Eddy, A. Krogh en G. Mitchinson. Biological Sequence Analysis - Probabilistic models of proteins and nucleic acids, Cambridge University Press, 1998. (ISBN-13: 978-0521629713 | ISBN-10: 0-521-62971-3)
Felsenstein J., Inferring Phylogenies, Sinauer Associates, 2003. (ISBN-13: 978-0878931774 | ISBN-10: 0-878-93177-5)
Gusfield D., Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology, Cambridge University Press, 1997. (ISBN-13: 978-0521585194 | ISBN-10: 0-521-58519-8)
Jones N.C. en P.A. Pevzner, An Introduction to Bioinformatics Algorithms, MIT Press, Cambridge, MA, 2004. (ISBN-13: 978-0262101066 | ISBN-10: 0-262-10106-8)
Setubal C. en J. Meidanis, Introduction to Computational Molecular Biology, PWS Publishing, 1997. (ISBN-13: 978-0534952624 | ISBN-10: 0-534-95262-3)
Mount D.W., Bioinformatics: Sequence and Genome Analysis. Cold Spring Harbor Laboratory Press, 2001. (ISBN-13: 978-0879697129 | ISBN-10: 0-879-69712-1)

Course content-related study coaching

Algorithmic aspects of the computational biology that are touched upon during the lectures, are further illustrated by means of a series of case studies that are worked out in Matlab or Python. Some case studies are an integral part of the lectures, while others need further elaboration by the students during practical sessions in a PC-room or as obligatory projects. The electronic learning-environment Ufora plays a central role in the guided self-learning process of the students, as it enables the exchange of feedback and additional study material between lecturer and students.

Assessment moments

continuous assessment

Examination methods in case of periodic assessment during the first examination period

Examination methods in case of periodic assessment during the second examination period

Examination methods in case of permanent assessment

Assignment

Possibilities of retake in case of permanent assessment

examination during the second examination period is possible

Extra information on the examination methods

Students are assessed by the combination of a series of programming assignments (permanent evaluation: 60%) and an individual project (exam: 40%).

Calculation of the examination mark

Permanent evaluation (60%) and project elaborated in exam period (40%). At least half of the point must be obtained for both components: permanent evaluation and project elaborated in exam period. If at least half of the points was obtained for only one of the two components, the final score is based only on the other component, which is then weighted 100%.
For the second examination period, alternative assignments are indicated for the four assignments (other than those from the first examination period). The project just stays the same. Students can retake each of these assignments/project in the second examination period. If they don't retake an assignment, they keep their score for that assignment from the first exam period.