

Computational Challenges in Bioinformatics (C003711)

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

Credits 6.0 Study time 180 h Contact hrs 60.0 h

Course offerings and teaching methods in academic year 2020-2021

A (semester 2)	English	seminar: coached	15.0 h
		exercises	
		self-reliant study	5.0 h
		activities	
		project	10.0 h
		lecture	30.0 h

Lecturers in academic year 2020-2021

Dawyndt, Peter	WE02	lecturer-in-charge
Fostier, Jan	TW05	co-lecturer

Offered in the following programmes in 2020-2021

	crdts	offering
Brugprogramma Master of Science in Bioinformatics (main subject Engineering)	6	A
Master of Science in Bioinformatics (main subject Engineering)	6	A
Master of Science in Computer Science	6	A
Master of Science in Computer Science Engineering	6	A
Master of Science in Computer Science Engineering	6	A
Exchange Programme in Bioinformatics (master's level)	6	A
Exchange Programme in Computer Science (master's level)	6	A

Teaching languages

English

Keywords

Performance, compute intensive applications, big data, hardware acceleration, data structures and algorithms, bioinformatics

Position of the course

The objective of this course is to focus on existing computational challenges in the field of bioinformatics and the techniques that are applied in scientific literature to deal with those challenges. Both compute intensive and data intensive ('big data') problems are considered. Even though the main focus of the course is on the software aspects, certain hardware acceleration techniques are covered. The goal of this course is aimed at students that are interested in software development in the field of bioinformatics.

Contents

- Sequence alignment
 - Needleman-Wunsch and Smith-Waterman algorithm, advanced variants
 - Multiple sequence alignment
 - Implementation aspects and hardware acceleration
- Gene prediction
 - Markov models
 - Viterbi algorithm
- String matching and applications
 - Data structures and algorithms for string matching: suffix trees, suffix arrays, Burrows-Wheeler transformation, FM index
 - Applications to read mapping
- Graph-based algorithms and applications

- De Bruijn graphs
- Application to genome assembly

Initial competences

Advanced programming skills in Java, Python or C/C++, advanced knowledge of data structures and algorithms

Final competences

- 1 To understand the commonly used data structures and algorithms that are commonly used in bioinformatics applications.
- 2 To understand the most important computationally intensive problems in bioinformatics and the existing solutions (or heuristics) to solve them.
- 3 Being able to estimate the runtime and memory footprint based on the computational complexity of the underlying algorithm and the size of the problem.
- 4 Being able to design advanced algorithms and software implementations based on standard data structures and algorithms.
- 5 To take into account the computational feasibility when critically assessing several algorithmic approaches.
- 6 To spend enough time on the design and the computational complexity evaluation prior to the actual implementation of software.
- 7 To understand the tradeoffs between performance, memory footprint and accuracy.

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

Lecture, project, self-reliant study activities, seminar: coached exercises

Learning materials and price

- Slides available on Ufora
- Scientific papers will be distributed through Ufora
- Example source code available on Ufora

References

- "Biological Sequence Analysis", Richard Durbin, Cambridge University Press, 1998
- "Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology", Dan Gusfield, 2001
- "Bioinformatics Algorithms: Sequence Analysis, Genome Rearrangements, and Phylogenetic Reconstruction", Enno Ohlebusch, Oldenbusch-Verlag, 2013.

Course content-related study coaching

- By e-mail or after appointment
- Additional information through Ufora

Evaluation methods

end-of-term evaluation and continuous assessment

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

Written examination, open book examination, oral examination

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

Written examination, open book examination, oral examination

Examination methods in case of permanent evaluation

Skills test, report

Possibilities of retake in case of permanent evaluation

examination during the second examination period is not possible

Extra information on the examination methods

- Periodic evaluation: open-book examination (exercises), oral examination with written preparation
- Permanent evaluation: graded homework assignments (source code + report)

Calculation of the examination mark

For both examination periods, the final mark will be composed as follows: 25% homework assignments, 75% exam. For both examination periods, a minimum score of

50% is required for the examination part. If this condition is not fulfilled, the final mark for this course will be limited to 9/20.