

Course Specifications

Valid as from the academic year 2019-2020

Course size (nominal values; actual values may depend on programme)
Credits 5.0 Study time 150 h Contact hrs 45.0 h

Course offerings in academic year 2020-2021

A (semester 1) English

Lecturers in academic year 2020-2021

De Neve, Wesley

KR01 lecturer-in-charge

Van Messem, Arnout

KR01 co-lecturer

Offered in the following programmes in 2020-2021

[Bachelor of Science in Molecular Biotechnology](#)

crdts offering

5 A

Teaching languages

English

Keywords

Algorithmic design, Algorithms, Bioinformatics, Creative problem solving, Complexity analysis, Datastructures, Machine Learning, Programming, Python, R

Position of the course

The field of bioinformatics was born after biologists discovered how to sequence (digitize) DNA, raising the need for mathematical and computational techniques to decipher the language of DNA, RNA, and proteins. As a result, bioinformatics has become an important part of modern biology, often facilitating new insights and new (data-driven) approaches, driving further biological developments.

Primarily taking a computational point-of-view, this course aims at introducing students to the design, implementation, and analysis of standard algorithms in the field of bioinformatics, including exhaustive search algorithms, recursive algorithms, divide-and-conquer algorithms, greedy algorithms, graph algorithms, dynamic programming algorithms, machine learning algorithms (shallow and deep), and randomized algorithms. These algorithms and related datastructures (e.g., lists, tuples, sets, dictionaries, graphs, hash tables, and trees) are studied in the context of problems like pattern matching, genome rearrangements, DNA sequencing, DNA sequence alignment, regulatory motif finding, genome annotation (structural and functional), and/or medical image analysis.

Contents

Based on capita selecta from the course handbook used, this course deals with the following computational topics, making use of pseudocode and Python/R:

- *algorithms and complexity, including asymptotic notation;*
- *exhaustive search algorithms;*
- *recursive algorithms;*
- *divide-and-conquer algorithms;*
- *greedy algorithms;*
- *graph algorithms;*
- *dynamic programming algorithms;*
- *machine learning algorithms (shallow and deep); and*
- *randomized algorithms.*

The theory is brought into practice through six series of mostly computer-based exercises.

Initial competences

An understanding of basic probability theory and the behaviour of common functions studied in calculus (e.g., linear functions, logarithmic functions, exponential functions).

Prior experience in at least one high-level programming language (e.g., Python, R).

Informatics (O000096).

Final competences

- 1 Understand the biological motivation of common bioinformatics problems, as well as their mathematical and computational abstraction.*
- 2 Be familiar with the standard algorithms and datastructures used in bioinformatics.*
- 3 Design and implement algorithms for new bioinformatics problems using standard algorithmic techniques.*
- 4 Compare the effectiveness and time/memory efficiency of different algorithmic techniques in solving different bioinformatics problems.*

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

Demonstration, lecture, seminar: practical PC room classes

Learning materials and price

Handbook: Philip Compeau and Pavel A. Pevzner. Bioinformatics Algorithms: An Active Learning Approach. Active Learning Publishers; 3rd edition (2018), ISBN-13: 978-0990374633. About \$100.

Slides shown during the lectures will be made available on Minerva/Ufora (in English), together with additional learning materials (e.g., background information and links to relevant websites).

Free digital tools like Eclipse, PyCharm, and RStudio for writing and debugging source code.

Students are required to have a personal laptop for use in this course.

References

Mark Lutz (2009). Learning Python: Powerful Object-Oriented Programming. Fourth Edition. O'Reilly Media, ISBN-13: 978-0596158064.

Mark Pilgrim (2009). Dive into Python. CreateSpace, ISBN-13: 978-1441413024. Free download @ <http://diveintopython.org>.

Hans Peter Langtangen (2009). A Primer on Scientific Programming with Python. Springer, ISBN-13: 978-3642024740.

Tony Gaddis (2009). Starting Out with Python. Pearson Education - Addison Wesley, ISBN-13: 978-0321549419.

Michael H. Goldwasser (2007). Object-Oriented Programming in Python. Prentice Hall, ISBN-13: 978-0136150312.

Jason Kinser (2008). Python for Bioinformatics. Jones & Bartlett Publishers, ISBN-13: 978-0763751869.

Sebastian Bassi (2009). Python for Bioinformatics. Chapman & Hall, ISBN-13: 978-1584889298.

Mark G. Sobell (2012). A Practical Guide to Linux: Commands, Editors, and Shell Programming. Fourth Edition. Prentice Hall, ISBN-13: 978-0134774602.

William F. Punch, Richard Enbody (2017). The Practice of Computing using Python. Third Edition. Addison Wesley, ISBN-13: 978-0133085044.

Steven Haddock and Casey Dunn (2010). Practical Computing for Biologists. First Edition. Sinauer Associates, Inc, ISBN-13: 978-0878933914.

Ashley Shade, Tracy K. Teal (2015). *Computing Workflows for Biologists: A Roadmap*. PLOS Biology.

Pavel A. Pevzner (2004). *Educating Biologists in the 21st Century: Bioinformatics Scientists versus Bioinformatics Technicians*. *Bioinformatics*, Vol.20, No.14, pages 2159-2161.

Alejandra J. Magana, Manaz Taleyarkhan, Daniela Rivera Alvarado, Michael Kane, John Springer, and Kari Clase (2014). *A Survey of Scholarly Literature Describing the Field of Bioinformatics Education and Bioinformatics Educational Research*. *CBE—Life Sciences Education*, Vol. 13, pages 607-623.

Phillip Compeau and Pavel Pevzner (2014). *Bioinformatics Algorithms: An Active Learning Approach*. Active Learning Publishers, ISBN-13: 978-0990374602.

Neil C. Jones and Pavel A. Pevzner. *An Introduction to Bioinformatics Algorithms*. The MIT Press, ISBN-13: 978-0262101066.

Istvan Albert (2019). *The Biostar Handbook: A Beginner's Guide to Bioinformatics*. Self-published. Second Edition.

Course content-related study coaching

The theory lectures touch upon the biological motivation of common bioinformatics problems, as well as their mathematical and computational abstraction.

During the weekly supervised hands-on sessions, students themselves learn how to tackle computational challenges in the field of bioinformatics by solving a series of mandatory computer exercises, mostly on an individual basis. These exercises aim at bringing the theory into practice.

After each deadline, example solutions for all exercises are made available on Minerva/Ufora.

Example examinations of previous years are made available on Minerva/Ufora near the end of the teaching activities.

Information about the calculation of the different evaluation marks is communicated during the theory lectures at the beginning and near the end of the first-term teaching activities.

Announcements on Minerva/Ufora are used for counselling, giving feedback, and providing background information.

Through individual appointments scheduled via email, the lecturers and the teaching assistants are available for answering questions about the course in general (grading, examination), the theory, and the exercises.

Evaluation methods

end-of-term evaluation and continuous assessment

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

Written examination with open questions, written examination with multiple choice questions, skills test

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

Written examination with open questions, written examination with multiple choice questions, skills test

Examination methods in case of permanent evaluation

Assignment

Possibilities of retake in case of permanent evaluation

examination during the second examination period is not possible

Calculation of the examination mark

For the first-term examination period, both the continuous assessment (hands-on sessions; non-periodic evaluation; permanent evaluation) and the end-of-term assessment (examination; periodic evaluation) account for 50% of the final examination mark. To qualify for passing, both the mark of the continuous assessment and the mark of the end-of-term assessment should be higher than or equal to 8/20 (40%). If that is

not the case, the final examination mark will be subject to an upper limit of 7/20. Note that the end-of-term assessment consists of a written closed-book part, accounting for 60% of the mark, and a computer-based open-book part, accounting for 40% of the mark.

The continuous assessment cannot be retaken during the resit examination period. Therefore, the examination mark for the resit examination period is calculated twice. For the first calculation, the mark of the continuous assessment, as obtained for the first-term examination period, accounts for 50% of the examination mark and the mark of the resit assessment, as obtained during the resit examination period, accounts for the remaining 50% of the examination mark. For the second calculation, the examination mark is equal to the mark of the resit assessment, as obtained during the resit examination period (that is, the mark of the continuous assessment, as obtained for the first-term examination period, is ignored). The final examination mark for the resit examination period is then equal to the maximum of the above two calculations. Note that the resit assessment consists of a written closed-book part, accounting for 50% of the mark, and a computer-based open-book part, also accounting for 50% of the mark.

Students who eschew period aligned and/or non-period aligned evaluations for this course unit may be failed by the examiner.