



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### **DIT743 Computational Methods in Bioinformatics, 7.5 credits**

Beräkningsmetoder i bioinformatik, 7,5 högskolepoäng

*Second Cycle*

---

#### **Confirmation**

This course syllabus was confirmed by Department of Computer Science and Engineering on 2020-12-18 and was last revised on 2022-11-15 to be valid from 2023-08-28, autumn semester of 2023.

*Field of education:* Science 100%

*Department:* Department of Computer Science and Engineering

#### **Position in the educational system**

The course is offered within several programmes. It is also a single subject course at the University of Gothenburg.

The course can be part of the following programmes: 1) Computer Science, Master's Programme (N2COS), 2) Mathematical Sciences, Master's Programme (N2MAT), 3) Applied Data Science Master's Programme (N2ADS) and 4) Computer Science, Bachelor's Programme (N1COS)

#### *Main field of studies*

Computer Science

Data Science

#### *Specialization*

A1F, Second cycle, has second-cycle course/s as entry requirements

A1F, Second cycle, has second-cycle course/s as entry requirements

#### **Entry requirements**

To be eligible for the course, the student should have successfully completed 60 credits of studies in Computer Science, Software Engineering, Data Science, Mathematics, Mathematical Statistics, or equivalent. Furthermore, the student should have successfully completed a course in Programming (DIT012 Imperative Programming

with Basic Object-orientation, DIT042 Object-oriented Programming, DIT143 Functional programming, or equivalent) and a basic course in discrete mathematic (DIT980, DIT856 or equivalent).

Applicants must prove knowledge of English: English 6/English B or the equivalent level of an internationally recognized test, for example TOEFL, IELTS.

### **Learning outcomes**

On successful completion of the course the student will be able to:

#### *Knowledge and understanding*

- describe and summarise problems that have been addressed in the bioinformatics literature, and computational approaches to solving them

#### *Competence and skills*

- design and implement computational solutions to problems in bioinformatics

#### *Judgement and approach*

- critically discuss different bioinformatics methods that address the same task or related tasks, and to discuss differences in the tasks addressed, or differences in the computational approaches
- identify situations where the same computational methods are applied in addressing different problems, even across different application areas

### **Course content**

This course demonstrates how computational methods that have been presented in other computing courses can be applied to solve problems in an application area. We look at problems related to the analysis of biological sequence data (sequence bioinformatics) and macromolecular structures (structural bioinformatics).

Computing scientists need to be able to understand problems that originate in areas that may be unfamiliar to them, and to identify computational methods and approaches that can be used to solve them. Biological concepts needed to understand the problems will be introduced.

Reading research articles is valuable training for scientists and researchers. Developing skill in reading research articles is useful preparation for future scientific studies, and at the same time their own scientific writing can be improved. Therefore, in this course, research articles are used as the main reference material, in particular to show how to present ideas and methods, and how to critically evaluate them.

Computational methods and concepts featured in this course include: dynamic programming; heuristic algorithms; graph partitioning; image skeletonisation, smoothing and edge detection; clustering; sub-matrix matching; geometric hashing; constraint logic programming; Monte Carlo optimisation; simulated annealing; self-avoiding walks.

Biological problems featured in this course include: sequence alignment; domain assignment; structure comparison; comparative modelling; protein folding; fold recognition; finding channels; molecular docking; protein design.

### *Sub-courses*

#### **1. Assignments (*Inlämningsuppgifter*), 7.5 credits**

Grading scale: Pass with distinction (5), Pass with credit (4), Pass (3) and Fail (U)

### **Form of teaching**

Lectures and programming assignments.

*Language of instruction:* English

### **Assessment**

The course is examined by individual programming assignments and written assignments.

If a student, who has failed the same examined component twice, wishes to change examiner before the next examination, a written application shall be sent to the department responsible for the course and shall be granted unless there are special reasons to the contrary (Chapter 6, Section 22 of Higher Education Ordinance).

In cases where a course has been discontinued or has undergone major changes, the student shall normally be guaranteed at least three examination occasions (including the ordinary examination) during a period of at least one year from the last time the course was given.

### **Grades**

The grading scale comprises: Pass with distinction (5), Pass with credit (4), Pass (3) and Fail (U).

The grading scale comprises Fail (U), 3, 4 or 5.

To pass the course, all mandatory components must be passed. To earn a higher grade

than Pass, a higher weighted average from the grades of the components is required.

**Course evaluation**

The course is evaluated through meeting after the course between teachers and student representatives. Further, an anonymous questionnaire is used to ensure written information. The outcome of the evaluations serves to improve the course by indicating which parts could be added, improved, changed or removed.

**Additional information**

The course is a joint course together with Chalmers.

Course literature to be announced the latest 8 weeks prior to the start of the course.

It is recommended to have taken an introductory course in data structures beforehand. Familiarity with some basic chemistry concepts (including atoms and molecules, chemical bonding) is useful.

The course replaces the course DIT741 Computational Methods in Bioinformatics, 7.5 credits. The course cannot be included in a degree which contains DIT741. Neither can the course be included in a degree which is based on another degree in which the course DIT741 is included.