



COURSE DESCRIPTION CARD - SYLLABUS

Course name

Structural bioinformatics

Course

Field of study

Bioinformatics

Area of study (specialization)

Level of study

Second-cycle studies

Form of study

full-time

Year/Semester

1/2

Profile of study

general academic

Course offered in

Polish

Requirements

compulsory

Number of hours

Lecture

15

Laboratory classes

15

Other (e.g. online)

Tutorials

Projects/seminars

Number of credit points

3

Lecturers

Responsible for the course/lecturer:

prof. dr hab. inż. Marta Szachniuk

Wydział Informatyki i Telekomunikacji

Responsible for the course/lecturer:

Prerequisites

The person starting the 2nd-cycle studies should have attained the learning objectives defined for the 1st-cycle studies in the resolution of the PUT Academic Senate – the learning objectives are available at the website of the Faculty of Computing and Telecommunications.

The student starting this module should have the basic knowledge about structural biology, theory of algorithms, programming, data bases and combinatorial algorithms. Moreover, he / she should be able to solve the fundamental problems in the area of programming and analysis of biological data.

In addition, with respect to the social skills, the student should show such attitudes as honesty, responsibility, perseverance, curiosity, creativity, manners, and respect for other people.

Course objective

1. The transfer of knowledge concerning (i) algorithms and optimization methods used to solve fundamental problems of modern biology and bioinformatics of RNA structures, and (ii) currently best tools for biological data acquisition, storage, processing and analysis.



2. The acquaintance of students with an architecture of most common tools used in structural bioinformatics, including both, new and precursory projects. The emphasis is placed on pros and cons of the solutions, their limitations (mostly unknown to the typical users), technologies used in implementation, quality and accuracy of obtained results, their reliability and ways of evaluation and verification.
3. The development of students' skills in the area of mathematical modeling of structural biology problems, in solving computational problems of RNA structural biology using both, simple and advanced bioinformatic algorithms, in testing novel computational methods.

Course-related learning outcomes

Knowledge

1. The student has an extended knowledge of the mathematical models used to represent structures of nucleic acid molecules on different levels of their architecture.
2. The student knows methods, techniques and tools used in the process of solving complex tasks in the analysis of RNA structures, mainly of engineering nature.
3. The student knows an architecture and the functional details of the most popular structural data repositories and algorithms for structure processing (structure prediction, structure determination, structure evaluation and comparison, clustering, etc).
4. The student is familiar with issues in modeling and analysis of biological structures based on theoretical foundations.
5. The student has a detailed knowledge of the design of structure prediction processes, structure comparison and structure modeling.
6. The student has a detailed, based on solid theoretical foundation, knowledge of optimization and efficient algorithms used in RNA structural bioinformatics; knows the most important (from the historical point of view) and most popular algorithms to compute and predict structures, search for common substructures, align structures.
7. The student knows the trends in development of computational methods for crucial problems of structural biology and is aware of the approaches proposed for solving novel problems in structural bioinformatics.

Skills

1. The student uses and interprets the information acquired from the literature concerning the general problems of RNA structural bioinformatics and from specialist journals (scientific publications in Bioinformatics, BMC Bioinformatics, Nucleic Acids Research, etc; internet services and portals from the bioinformatics area); interprets and evaluates their contents.
2. The student is able to use advanced techniques and computational tools (e.g. GenBank, BLAST, RNA FRABASE, FARFAR, ModeRNA, RNAComposer, RNAstructure, Eterna, mfold, RNAfold, RNAinverse) for solving biological problems and can estimate their usefulness.



3. Under the supervision of a teacher, the student can plan and solve research tasks (e.g. modeling of biomolecule structure with the use of bioinformatic tools, evaluating of RNA model quality and accuracy, clustering of structures based on their similarity) using the known computational methods.
4. The student is able to design and develop computer software to solve selected structural bioinformatics problems using appropriate methods, techniques and tools.
5. The student prepares the presentation of obtained results in Polish and English (reports, multimedia presentations) and discusses them in the group.
6. The student can estimate the usefulness of new achievements of structural bioinformatics, especially in the area of structural data processing and analysis.

Social competences

1. The student understands the necessity of learning (during the lectures students are introduced with the latest achievements of structural bioinformatics; students have the possibility to present an interesting subject concerning the latest solutions in the field).
2. The student can co-operate and work in a group (laboratory gives a chance to work in groups).
3. The student understands the necessity of reading the scientific and popular journals in order to gain and extend bioinformatics knowledge.
4. The student has a creative attitude in professional and social activities.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment

a) verification of assumed learning objectives related to lectures:

- answers to the questions regarding the material from previous lectures

b) verification of assumed learning objectives related to tutorials / laboratory classes:

- evaluation of the student's preparation to laboratory exercises (input test) and evaluation of the ability to perform the laboratory exercises
- continuous evaluation, on every laboratory (oral tests), rewarding the gain of skills necessary to execute the laboratory exercises
- evaluation of the report of the exercises (prepared partially during the laboratory and partially at home); the assessment covers also the ability to work in a group
- evaluation of the exercises' execution during the laboratories

Total assessment

a) verification of assumed learning objectives related to lectures:



- evaluation of the knowledge during the written test. The test is composed of 5 questions; the student can obtain 3 points for an answer to each question; 20% of the test is prepared as multiple choice questions; the remaining part has the form of an open questions; all questions concern the problems introduced during the lectures and practised during laboratory exercises. In order to get positive mark, one should obtain at least 8 points.

- discussion on the test results

b) verification of assumed learning objectives related to tutorials / laboratory classes:

- the verification of the skills and knowledge regarding the issues mentioned during the laboratories during the final written test

- the final evaluation is based on the weighted average, which includes the marks obtained for the projects, reports and activity during the laboratories.

Additional activities taken into account:

- discussion on the additional aspects of the topics,

- preparation of short presentation about a structural bioinformatics problem that was not presented by the lecturer,

- effectiveness knowledge gained during problem solving,

- remarks helping to improve the teaching materials or the teaching process.

Programme content

The program of the lecture encompasses the following issues. The first lecture generally introduces the subject, in particular it discusses the crucial issues of modern RNA structural bioinformatics, their proposed solutions, and their characteristics from the point of view of requirements, limitations and computational complexity. The second lecture introduces to the data bases of RNA structures. The students learn about selected, most popular structural repositories, data schemes, database architecture, interface applications, meta-languages defined to generate advanced queries and data formats used to save structural information. The succeeding lecture is devoted to the most popular models and formats proposed for RNA secondary structure representation, secondary motifs and the methods of their visualization. During the lecture, the students learn the methodology of mathematical modeling of structural data and issues, including their optimization for the following analysis of the problems and considering the completeness of modeled information. The fourth lecture refers to the basic methods of secondary structure prediction. It presents the first, historical algorithms for RNA secondary structure prediction (the methods performance is shown based upon the selected cases) as well as the novel methods. Within the next lecture, the fundamental approaches to the tertiary structure modeling are discussed, including the general characteristics of the approaches, their drawbacks, advantages and usefulness. The selected modern methods for the RNA tertiary structure prediction are also presented. The sixth lecture is devoted to the algorithms for structural data comparison, similarity search, distance



and similarity measures, and data clustering. Examples for using various similarity measures are presented for different types of structures, and the most popular modern tools that enable structure comparison and analysis are shown. The visualization of bioinformatic data is a subject of the next lecture. It introduces the basic information about computational graphics and visualization methods implemented in various bioinformatic tools. The final lecture is devoted to students' presentations: student pairs present the latest issues appearing in structural bioinformatics, due to their choice and interests.

Laboratory exercises are strictly related to the lectures. The lecture is a theoretical introduction, while the laboratory allows for hands-on exercises related to the introduced topics and for implementation of the own solutions for simple bioinformatic problems.

Laboratory has the form of seven 2-hour exercises in the computational lab. Students obtain tasks to be solved, thus, practising the issues discussed during the lectures and preparing to use the available tools and programming libraries. Additionally, during the laboratory exercises, the students work in pairs and thus, they implement their projects (partially during the laboratory, partially as a homework), which solve the selected combinatorial problems within the field of structural biology. The program of laboratory exercises encompasses the following issues. During the first laboratory, the students learn to use the most common RNA databases. The goal is to learn about the relations between databases and the construction of advanced database queries. The students must generate simple and complicated queries, and prepare the obtained results for the remote access. The next laboratory is aimed at RNA secondary structures. The students learn the available tools for parsing and visualization of the secondary structures. The practical task is designing the own application for reading the selected data formats and an analysis of the information about the secondary structures of biological molecules. This subject is the introduction to the next one, concerning the methods of secondary structure prediction with dynamic programming based algorithms. The goal of the next laboratory exercise is project and implementation of one secondary structure algorithm. The forth topic refers to the tertiary structure modeling, model quality analysis and structure comparison. The students get familiar with the available tools, similarity measures and algorithms for structure evaluation. Within the laboratory, the students solve several tasks related to the topic (using available tools), run a series of computational experiments (from structure prediction with different input parameters, to results analysis) and summarize their work in written reports. The following subject is structure visualization. The laboratory exercises are devoted to practice with commonly used tools and analyze their functionality. During text laboratory student learn how to use different programming libraries with their work on the tertiary structures. The most important libraries (for several programming languages) are presented, along with use cases for the real data. The final laboratory is devoted to an analysis of trajectories showing the states of molecular dynamic simulations. The students learn about various formats of such data representation and about the methodology of these data acquisition and access. The task given to the students is preparing a simple project, which enables an analysis of structures and processing of the dynamic simulation trajectories.

Teaching methods



1. Lectures: multimedia presentation, blackboard presentation illustrated with examples, multimedia show.
2. Laboratory classes: solving exercises, practical exercises, running computational experiments, discussion, group work, multimedia show, workshops, games, case study.

Bibliography

Basic

1. J. Gu, P.E. Bourne, "Structural Bioinformatics"
2. P. Baldi, S. Brunak, "Bioinformatics: The Machine Learning Approach"
3. T. Schwede, M. Peitsch, "COmputational structural biology. Methods and applications."

Additional

Recent scientific publications from the area of structural bioinformatics.

Breakdown of average student's workload

	Hours	ECTS
Total workload	75	3,0
Classes requiring direct contact with the teacher	30	1,5
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) ¹	45	1,5

¹ delete or add other activities as appropriate