Federal State Autonomous Educational Institution of Higher Education "Moscow Institute of Physics and Technology (National Research University)"

APPROVED
Head of the Phystech School of Biological and Medical Physics
D.V. Kuzmin

Work program of the course (training module)

course:	Algorithms of Bioinformatics/Алгоритмы биоинформатики
major:	Applied Mathematics and Physics
specialization:	Applied Bioinformatics/Прикладная биоинформатика
	Phystech School of Biological and Medical Physics
	Chair of Bioinformatics and Systems Biology
term:	1
qualification:	Master

Semester, form of interim assessment: 1 (fall) - Grading test

Academic hours: 45 AH in total, including:
lectures: 30 AH.
seminars: 15 AH.
laboratory practical: 0 AH.

Independent work: 45 AH.

In total: 90 AH, credits in total: 2

Authors of the program:

A.S. Kasyanov, candidate of physics and mathematical sciences
V.Y. Makeev, doctor of physics and mathematical sciences

The program was discussed at the Chair of Bioinformatics and Systems Biology 04.06.2020

Annotation

The goal of the discipline is to give basic understanding of basic principles of bioinformatics algorithms and practical skills in applying bioinformatics methods for analyzing and interpreting biological data.

1. Study objective

Purpose of the course

Give basic understanding of basic principles of bioinformatics algorithms and practical skills in applying bioinformatics methods for analyzing and interpreting biological data.

Tasks of the course

- give an idea of the basic principle of bioinformatics algorithms;
- make students familiar with modern understanding of efficient analysis of biological data;
- teach how to use the main software in the field;
- introduce basic algorithms and data formats for genome sequence analysis.

.

2. List of the planned results of the course (training module), correlated with the planned results of the mastering the educational program

Mastering the discipline is aimed at the formation of the following competencies:

Code and the name of the competence	Competency indicators
Pro.C-3 Use research and testing equipment (devices and installations, specialized software) in a selected subject field	Pro.C-3.3 Evaluate the accuracy of the experimental (numerical) results
	Pro.C-3.2 Conduct an experiment (simulation) using research equipment (software)
	Pro.C-3.1 Understand the operating principles of the equipment and specialized software

3. List of the planned results of the course (training module)

As a result of studying the course the student should:

know:

- basic data structures: hash table, suffix tree, suffix array;
- quick search for substrings in the string - naive algorithms, Knut-Maurice-Pratt, Rabin-Karp, kangaroo algorithm;
- the Burrows-Wheeler index and transformation;
- BLAST - indexing, Altschul-Karlin statistics;
- motifs in the genomes, search and identification of motifs, multiple local alignment;
- optimization methods; expectation maximization and Gibbs sampling;
- dynamic programming algorithms for finding the shortest path between two vertices in a directed acyclic graph and calculating the sum of weights over all paths (partition function);
- an algorithm for optimal segmentation of a sequence using dynamic programming;
- concept of a hidden Markov model, transition and emission probabilities, search for an optimal sequence of state transitions for a sequence generated by a hidden Markov model (Viterbi algorithm), calculation of transition probability at a given point (back-and-forth algorithm), use of a dynamic programming algorithm for analyzing hidden Markov chains;
- Fundamentals of Bayesian statistics, likelihood, maximum likelihood method, marginalization of distributions and marginal likelihood;
- estimation of parameters of the hidden Markov chain, Viterbi training, the Baum-Welch method;
- genome analysis methods based on hidden Markov chains, search for coding sequences, search for homogeneous chromatin domains.

be able to:

- design efficient algorithms for mastering of large amount of information;
- be able to model functional motifs in biological sequences.

master:

- of estimating the required time and space resources for software using basic algorithms;
- of planning and implementation of efficient bioinformatic analysis.

4. Content of the course (training module), structured by topics (sections), indicating the number of allocated academic hours and types of training sessions

4.1. The sections of the course (training module) and the complexity of the types of training sessions

№	Topic (section) of the course	Types of training sessions, including independent work			
		Lectures	Seminars	Laboratory practical	Independent work
1	Basic data structures: hash table, suffix tree, suffix array	4	2		2
2	Quick search for substrings in a string - naive algorithms, Knut-Maurice-Pratt, Rabin-Karp, kangaroo algorithm	2	1		2
3	Burrows-Wheeler Index and Transformation	2	1		4
4	BLAST - Altschul-Karlin indexing	4	1		4
5	Motifs in genomes, search and identification of motifs, multiple local alignment	2	2		4
6	Algorithms of dynamic programming for finding the shortest path between two vertices in a directed acyclic graph and calculating the sum of weights over all paths (partition function)	4	1		5
7	Applications of dynamic programming algorithms. Algorithm for optimal sequence segmentation using dynamic programming	2	1		4
8	Hidden Markov Models.	2	1		4
9	Fundamentals of Bayesian statistics, likelihood, maximum likelihood method, marginalization of distributions and marginal likelihood	2	1		4
10	Optimization Methods: Expectation Maximization Gibbs Sampling	2	1		4
11	Estimation of parameters of a Hidden Markov Mode, Viterbi training, Baum-Welch method	2	1		4
12	Genome functional annotation methods based on Hidden Markov Model, search for coding sequences, search for homogeneous chromatin domains	2	2		4
AH in total		30	15		45
Exam preparation		0 AH.			
Total complexity		90 AH., credits in total 2			

4.2. Content of the course (training module), structured by topics (sections)

Semester: 1 (Fall)

1. Basic data structures: hash table, suffix tree, suffix array

Hash table, suffix tree, suffix array, search complexity in each case.

2. Quick search for substrings in a string - naive algorithms, Knut-Maurice-Pratt, Rabin-Karp, kangaroo algorithm

Naïve algorithm, Knut-Maurice-Pratt, Rabin-Karp, kangaroo algorithm. Estimates of time and space efficiency. Selection depending on the length of the motif to search. Wildcards. Implementation.

3. Burrows-Wheeler Index and Transformation

Barrows-Wheeler Index and Transformation. Evaluation of the search complexity. Problem with indels. Use in the BWA and Bowtie programs read mapping.

4. BLAST - Altschul-Karlin indexing

Indexing, dependence of the key length on the alphabet, using the BLAST index in proteomics problems, comparing the BLAST and Smith-Waterman approaches in the search for local alignments. Altschul-Karlin Statistics. Extreme values distribution. Gumbel distribution. High scoring paths. P-value and E-value.

5. Motifs in genomes, search and identification of motifs, multiple local alignment

Motifs in genomes, search and identification of motifs, multiple local alignment. Representations of motives: consensus string, matrix of positional weights, Bayesian network. Tuset-Varre algorithm for calculating the probability of observing the motive in a random sequence. Algorithms for constructing multiple local alignments and identifying motives: the Stormo greedy algorithm, MEME, Gibbs sampler, ChIPmunk.

6. Algorithms of dynamic programming for finding the shortest path between two vertices in a directed acyclic graph and calculating the sum of weights over all paths (partition function)

Dynamic programming algorithms for finding the shortest path between two vertices in a directed acyclic graph (Bellman-Ford) and calculating the sum of weights over all such paths (partition).

7. Applications of dynamic programming algorithms. Algorithm for optimal sequence segmentation using dynamic programming

Applications of dynamic programming algorithms. Smith-Waterman local alignment algorithm. Smith-Waterman matrix and corresponding graph. Examples of paths. The algorithm of optimal segmentation of a sequence into domains that are homogeneous in composition. Graphs representations.

8. Hidden Markov Models.

The concept of a Hidden Markov Model, transition and emission probabilities, the search for the optimal sequence of transitions between states for the sequence generated by a Hidden Markov Model (Viterbi algorithm), the calculation of the transition probability at a given point (back and forth algorithm), the use of a dynamic programming algorithm for analyzing Hidden Markov Model.

9. Fundamentals of Bayesian statistics, likelihood, maximum likelihood method, marginalization of distributions and marginal likelihood

Likelihood, maximum likelihood method, marginalization of distributions and marginal likelihood. Sequential Bayes Estimation. Dirichlet integral. Dirichlet mixture. Conjugate Distributions. The role of prior distributions.

10. Optimization Methods: Expectation Maximization Gibbs Sampling

Expectation maximization. Cluster separation problem. Selection initial values. Convergence assessment. Multiple local alignments via expectation maximization (MEME). Gibbs sampling. Detailed balance. Convergence assessment problem.

11. Estimation of parameters of a Hidden Markov Mode, Viterbi training, Baum-Welch method

Viterbi training, Baum-Welsh approach, the role of dynamic programming and Bayesian estimation

12. Genome functional annotation methods based on Hidden Markov Model, search for coding sequences, search for homogeneous chromatin domains

Functional annotation methods based on Hidden Markov Mhains, search for coding sequences, search for homogeneous chromatin domains.

5. Description of the material and technical facilities that are necessary for the implementation of the educational process of the course (training module)

Equipment needed for lectures and seminars: computer and multimedia equipment (projector, sound system)

6. List of the main and additional literature, that is necessary for the course (training module) mastering

Main literature

Provided at the department:

1. Neil C. Jones, Pavel A. Pevzner An Introduction to Bioinformatics Algorithms 2004 Book (J&P)
2. Pavel A. Pevzner. Computational Molecular Biology 2000 Book
3. Phillip Compeau, Pavel Pevzner, Bioinformatics Algorithms: An Active Learning Approach 2014 Book
1. Durbin, R., Eddy, S., Krogh, A., Mitchison, G. Biological sequence analysis, Cambridge University Press, 1998.
2. Borodovsky, M., Ekisheva, S. Problems and solution in biological sequence analysis. Cambridge University Press, 2006.
3. Pevzner, P.A., Shamir, R. Bioinformatics for Biologists. Cambridge University Press, 2011

Additional literature

Provided at the department:

Gusfield D. Algorithms on Strings, Trees, and Sequences. University of California Davis.

7. List of web resources that are necessary for the course (training module) mastering

Scientific bibliographic and patent databases in the field of physico-chemical biology, available on the Internet in free mode - Science Citation Index (Web of Science), Medline (PubMed), Scientific Electronic Library (NEB), Russian Patent DB of FGU FIPS and American USPAFULL patent database; email addresses of major scientific publishers who provide access to the full text of current and archival issues of these journals.

8. List of information technologies used for implementation of the educational process, including a list of software and information reference systems (if necessary)

For some of the lessons, you will need Zoom. Google Drive to access course materials. The presence of smartphones / laptops during classes is encouraged to participate in interactive exercises.

9. Guidelines for students to master the course

A student who studies discipline must, on the one hand, master a general conceptual apparatus, and on the other hand, must learn to apply theoretical knowledge in practice.

As a result of studying the discipline, the student should know the basic definitions of the discipline, be able to apply this knowledge to solve various problems.

Successful learning requires:

- visits to all classes provided by the curriculum for the discipline;
- conducting the abstract of occupations;
- intense independent work of the student.

Independent work includes:

- reading recommended literature;
- study of educational material, preparation of answers to questions intended for self-study;
- solving problems offered to students in the classroom;
- preparation for performance of tasks of the current and intermediate certification.

An indicator of possession of the material is the ability to answer questions on discipline topics without an outline.

It is important to achieve an understanding of the material being studied, and not its mechanical memorization. If it is difficult to study individual

topics, questions, you should seek advice from the teacher.

Intermediate control of students' knowledge in the form of problem solving in accordance with the subject of classes is possible

SUPPLEMENT

Assessment funds for course (training module)

major:	Applied Mathematics and Physics
specialization:	Applied Bioinformatics/Прикладная биоинформатика Phystech School of Biological and Medical Physics Chair of Bioinformatics and Systems Biology
term:	1
qualification:	Master

Semester, form of interim assessment: 1 (fall) - Grading test

Authors:

A.S. Kasyanov, candidate of physics and mathematical sciences
V.Y. Makeev, doctor of physics and mathematical sciences

1. Competencies formed during the process of studying the course

Code and the name of the competence	Competency indicators
Pro.C-3 Use research and testing equipment (devices and installations, specialized software) in a selected subject field	Pro.C-3.3 Evaluate the accuracy of the experimental (numerical) results
	Pro.C-3.2 Conduct an experiment (simulation) using research equipment (software)
	Pro.C-3.1 Understand the operating principles of the equipment and specialized software

2. Competency assessment indicators

As a result of studying the course the student should:

know:

- basic data structures: hash table, suffix tree, suffix array;
- quick search for substrings in the string - naive algorithms, Knut-Maurice-Pratt, Rabin-Karp, kangaroo algorithm;
- the Burrows-Wheeler index and transformation;
- BLAST - indexing, Altschul-Karlin statistics;
- motifs in the genomes, search and identification of motifs, multiple local alignment;
- optimization methods; expectation maximization and Gibbs sampling;
- dynamic programming algorithms for finding the shortest path between two vertices in a directed acyclic graph and calculating the sum of weights over all paths (partition function);
- an algorithm for optimal segmentation of a sequence using dynamic programming;
- concept of a hidden Markov model, transition and emission probabilities, search for an optimal sequence of state transitions for a sequence generated by a hidden Markov model (Viterbi algorithm), calculation of transition probability at a given point (back-and-forth algorithm), use of a dynamic programming algorithm for analyzing hidden Markov chains;
- Fundamentals of Bayesian statistics, likelihood, maximum likelihood method, marginalization of distributions and marginal likelihood;
- estimation of parameters of the hidden Markov chain, Viterbi training, the Baum-Welch method;
- genome analysis methods based on hidden Markov chains, search for coding sequences, search for homogeneous chromatin domains.

be able to:

- design efficient algorithms for mastering of large amount of information;
- be able to model functional motifs in biological sequences.

master:

- of estimating the required time and space resources for software using basic algorithms;
- of planning and implementation of efficient bioinformatic analysis.

3. List of typical control tasks used to evaluate knowledge and skills

During the current control, the student should be able to answer the following questions:

1. Basic data structures: hash table, suffix tree, suffix array.
2. Algorithms for searching substrings in the string: naive, Knut-Maurice-Pratt, Rabin-Karp.
3. The Burrows-Wheeler Index and Transformation.
4. BLAST: indexing and searching locally aligned sections.
5. BLAST: weights of locally aligned sections, Gumbel distribution, Altschul-Karlin statistics, P-value and E-value.
6. Representations of motifs in genomes: consensus row, matrix of positional weights, Bayesian network.
7. Tuset-Varre algorithm for calculating the probability of meeting a motive in a random sequence.
8. Algorithms for the construction of multiple local alignments and identification of motives: the Stormo greedy algorithm, MEME, Gibbs sampling, ChIPmunk.

9. Dynamic programming algorithm for finding the shortest path between two vertices in a directed acyclic graph (Bellman-Ford).

10. Modifications of dynamic programming algorithms for locally alignment and segmentation of sequences into blocks of uniform composition.

11. The concept of a Hidden Markov Model, transition and emission probabilities.

12. Viterbi algorithm for searching the optimal sequence of state transitions for the sequence generated by the hidden Markov model

13. Algorithm “forward-backward” calculation of the probability of transition in a Hidden Markov Chain at a given point

15. Basics of Bayesian statistics. Prior probability distribution. Marginalization.

16. Expectation Maximization Algorithms.

17. Estimation of the parameters of the hidden Markov chain using the Viterbi method.

18. Estimation of parameters of the hidden Markov chain using the Baum-Welch algorithm.

19. Search for coding sequences using hidden Markov chains. GeneMark program.

20. Search for sites with a specific state of chromatin using hidden Markov chains. Ernst-Kellys approach.

During the class, interactive discussions can take place in the course chats, which will be homework. It is possible to perform patent search as an independent task. Successful completion of all tasks in the course and the completion of control slices of knowledge gives an advantage in the differential credit.

4. Evaluation criteria

1. Basic data structures: hash table, suffix tree, suffix array.

2. Algorithms for searching substrings in the string: naive, Knut-Maurice-Pratt, Rabin-Karp.

3. The Burrows-Wheeler Index and Transformation.

4. BLAST: indexing and searching locally aligned sections.

5. BLAST: weights of locally aligned sections, Gumbel distribution, Altschul-Karlin statistics, P-value and E-value.

6. Representations of motifs in genomes: consensus row, matrix of positional weights, Bayesian network.

7. Tuset-Varre algorithm for calculating the probability of meeting a motive in a random sequence.

8. Algorithms for the construction of multiple local alignments and identification of motives: the Stormo greedy algorithm, MEME, Gibbs sampling, ChIPmunk.

9. Dynamic programming algorithm for finding the shortest path between two vertices in a directed acyclic graph (Bellman-Ford).

10. Modifications of dynamic programming algorithms for locally alignment and segmentation of sequences into blocks of uniform composition.

11. The concept of a Hidden Markov Model, transition and emission probabilities.

12. Viterbi algorithm for searching the optimal sequence of state transitions for the sequence generated by the hidden Markov model

13. Algorithm “forward-backward” calculation of the probability of transition in a Hidden Markov Chain at a given point

15. Basics of Bayesian statistics. Prior probability distribution. Marginalization.

16. Expectation Maximization Algorithms.

17. Estimation of the parameters of the hidden Markov chain using the Viterbi method.

18. Estimation of parameters of the hidden Markov chain using the Baum-Welch algorithm.

19. Search for coding sequences using hidden Markov chains. GeneMark program.

20. Search for sites with a specific state of chromatin using hidden Markov chains. Ernst-Kellys approach.

Can be left unchanged

The mark is excellent (10 points) - it is given to a student who has shown comprehensive, systematic, deep knowledge of the curriculum of the discipline, who has an interest in this subject area, has demonstrated the ability to confidently and creatively put them into practice in solving specific problems, and a free and proper substantiation of decisions.

The mark is excellent (9 points) - it is given to a student who has shown comprehensive, systematic, in-depth knowledge of the curriculum of the discipline and the ability to confidently put them into practice in solving specific problems, free and proper substantiation of the decisions made.

The mark is excellent (8 points) - given to a student who has shown comprehensive, systematic, in-depth knowledge of the curriculum of the discipline and the ability to confidently apply them in practice in solving specific problems, correct justification of decisions made, with some shortcomings.

A mark is good (7 points) - it is put up for a student, if he knows the material firmly, sets it up competently and in essence, knows how to apply the knowledge gained in practice, but does not competently substantiate the results obtained.

Evaluation is good (6 points) - it is put up to a student, if he knows the material firmly, sets it up correctly and in essence, knows how to apply this knowledge in practice, but admits some inaccuracies in the answer or in solving problems.

A mark is good (5 points) - it is given to a student, if he basically knows the material, correctly and essentially sets it out, knows how to apply this knowledge in practice, but allows a sufficiently large number of inaccuracies to answer or solve problems.

Grade satisfactorily (4 points) is given to a student who has shown the fragmented, fragmented nature of knowledge, insufficiently correct formulations of basic concepts, violations of the logical sequence in the presentation of program material, but at the same time he has mastered the main sections of the curriculum necessary for further education and can apply knowledge is modeled in a standard situation.

Grade satisfactorily (3 points) - given to a student who showed the fragmented, scattered nature of knowledge, making mistakes in formulating basic concepts, disrupting the logical sequence in presenting program material, poorly masters the main sections of the curriculum required for further education and even applies the knowledge gained in a standard situation.

The rating is unsatisfactory (2 points) - is given to a student who does not know most of the main content of the curriculum of the discipline, makes gross mistakes in the wording of the basic principles and does not know how to use this knowledge when solving typical tasks.

Unsatisfactory mark (1 point) - is given to a student who does not know the main content of the discipline's curriculum, makes gross errors in the wording of the basic concepts of the discipline and does not have any skills to solve typical practical problems.

5. Methodological materials defining the procedures for the assessment of knowledge, skills, abilities and/or experience

The student is given 30 minutes to prepare. Interview with a student on a differential oral test should not exceed one astronomical hour.