

## BASIC DETAILS:

<b>Subject:</b>	APRENDIZAJE AUTÓNOMO Y EVOLUCIÓN		
<b>Id.:</b>	33376		
<b>Programme:</b>	DOBLE GRADO EN FARMACIA Y BIOINFORMÁTICA. PLAN 2018		
<b>Module:</b>	BIOINFORMÁTICA		
<b>Subject type:</b>	OBLIGATORIA		
<b>Year:</b>	2	<b>Teaching period:</b>	Segundo Cuatrimestre
<b>Credits:</b>	6	<b>Total hours:</b>	150
<b>Classroom activities:</b>	63	<b>Individual study:</b>	87
<b>Main teaching language:</b>	Inglés	<b>Secondary teaching language:</b>	Castellano
<b>Lecturer:</b>	RANERA BELTRAN, BEATRIZ (T)	<b>Email:</b>	branera@usj.es

## PRESENTATION:

The main goal of this subject is to explore the algorithms for bioinformatics purposes and their computational application in order to understand the mechanisms of biological database searching, protein and nucleic acid alignments, perform predictions of bidimensional structures, identify genes and detect regulatory sequences. Statistics and machine learning will be included to support the basis of the algorithms that are based on them.

Search algorithms as greedy or exhaustive will be studied to find repeated sequence as motifs or in the DNA mapping.

Dynamic programming will be used to analyse the local and global alignments of amino acid and nucleotides sequences based on similarity or distance calculations. Softwares, as GENESCAN, will be studied to see a direct application of the statistical Likelihood Ratio.

Identification of DNA, proteins as well as genome assembly will be assessed through graphical algorithms, for what Euler's theorem will be introduced.

A machine learning approximation will be done with Hidden Markov Models as long as the Monte Carlo sampling. The searching of gene regulatory sequence, crucial structures in epigenetics, the CpG islands need from the HMM with the decoding algorithms of Viterbi and Baum-Welch. This part will also include stochastic models and log odd ratio.

More complex applications of dynamic programming and stochastic models will be studied through the determination of the RNA secondary structure. Structural alignments require from HMM and stochastic context free grammar to make probabilistic competitive approximations.

Algorithms were developed in Python and R languages, being chosen whichever suits better for the computational requirements.

## PROFESSIONAL COMPETENCES ACQUIRED IN THE SUBJECT:

<b>General programme competences</b>	G01	Use learning strategies autonomously for their application in the continuous improvement of professional practice.
	G02	Perform the analysis and synthesis of problems of their professional activity and apply them in similar environments.
	G03	Cooperate to achieve common results through teamwork in a context of integration, collaboration and empowerment of critical discussion.
	G04	Reason critically based on information, data and lines of action and their application on relevant issues of a social, scientific or ethical nature.
	G05	Communicate professional topics in Spanish and / or English both orally and in writing.
	G06	Solve complex or unforeseen problems that arise during the professional activity within any type of organisation and adapt to the needs and demands of their professional environment.
	G07	Choose between different complex models of knowledge to solve problems.
	G09	Apply information and communication technologies in the professional field.
	G10	Apply creativity, independence of thought, self-criticism and autonomy in the professional

		practice.
<b>Specific programme competences</b>	E02	Develop the use and programming of computers, databases and computer programs and their application in bioinformatics.
	E03	Apply the fundamental concepts of mathematics, logic, algorithmics and computational complexity to solve problems specific to bioinformatics.
	E04	Program applications in a robust, correct, and efficient way, choosing the paradigm and the most appropriate programming languages, applying knowledge about basic algorithmic procedures and using the most appropriate types and data structures.
	E05	Implement well-founded applications, previously designed and analysed, in the characteristics of the databases.
	E06	Apply the fundamental principles and basic techniques of intelligent systems and their practical application in the field of bioinformatics.
	E07	Apply the principles, methodologies and life cycles of software engineering to the development of a project in the field of bioinformatics.
	E12	Apply the principles and techniques of protein computational modelling to predict their biological function, their activity or new therapeutic targets (Structural Bioinformatics, Computational Toxicology).
	E13	Apply omics technologies for the extraction of statistically significant information and for the creation of relational databases of biodata that can be updated and publicly accessible to the scientific community.
	E14	Use programming languages, most commonly used in the field of Life Sciences, to develop and evaluate techniques and/ or computational tools.
	E15	Infer the evolutionary history of genes and proteins through the creation and interpretation of phylogenetic trees.
	E16	Plan linkage and association studies for medical and environmental purposes.
	E17	Induce complex relationships between samples by applying statistical and classification techniques.
	E18	Apply statistical and computational methods to solve problems in the fields of molecular biology, genomics, medical research and population genetics.
	E21	Apply computational and data processing techniques for the integration of physical, chemical and biological concepts and data for the description and/ or prediction of the activity of a substance in a given context.

#### PRE-REQUISITES:

Statistics is needed to understand the underlying knowledge of bioinformatics algorithms. Programming skills in R and Python would be necessary in order to understand and to code properly the bioinformatics algorithms explained. Knowledge in molecular biology and structural features of the main biomolecules are essential.

#### SUBJECT PROGRAMME:

##### Subject contents:

<b>1 - Introduction to bioinformatics algorithms</b>
1.1 - Algorithm definition. Types of algorithms: recursive, iterative, fast, slow. Bioinformatics algorithms and applications
1.2 - Programming in python. Object Orientated Programming
<b>2 - Motif Searching.</b>
2.1 - Exhaustive search / Bruteforce algorithms: restriction mapping. Coding in python bruteforce algorithms.
2.2 - Greedy algorithms: A greedy approach to motif finding.
<b>3 - Pairwise and Multiple Sequence Alignments.</b>
3.1 - Dynamic programming with Python. Pairwise Sequence Alignment.
3.2 - Dynamic Programming with Python. Multiple Sequence Alignment.
<b>4 - CpG islands search.</b>
4.1 - Stochastic models. Markov chains.
4.2 - Hidden Markov Models.
4.3 - Implementation HMM in Python
<b>5 - Finding genes</b>
<b>6 - DNA Sequencing.</b>

6.1 - Graph algorithms.
6.2 - Implementation in python for Genome assembly.
<b>7 - Matching reads to Reference Sequences</b>
7.1 - Combinatorial Pattern Matching.
7.2 - Implementation in Python.
<b>8 - RNA secondary structure.</b>
8.1 - RNA secondary structure.
8.2 - Zuker and Nussinov algorithms for prediction.
8.3 - Competitive probabilistic approximation.
8.4 - Structural alignments.

Subject planning could be modified due unforeseen circumstances (group performance, availability of resources, changes to academic calendar etc.) and should not, therefore, be considered to be definitive.

## TEACHING AND LEARNING METHODOLOGIES AND ACTIVITIES:

### Teaching and learning methodologies and activities applied:

The student must be concerned to practise everytime after a lecture the coding work proposed in order to not get behind the rest. Programming skills are trained through practise, for that reason it is highly recommended a continous studying of the subject, reproducing examples and doing the proposal exercises.

Magistral lectures will be alterned with practical in obligatory assistance sessions. The programming languages that are going to be used are Python and R. Practical lessons will be aimed to get fluency in coding in Python languages using laboratory work (real examples of sequences to be compared, assembled...) and cases analysis. Most of the time students will have to translate from pseudocode to the corresponding language the orders the programmes have to follow.

As the subject needs from logical intelligence application and coding practise, all works to hand will be individuals. Nevertheless, common sessions can be used to put in commons problems, doubts or any kind of difficulty students might find. Students will be supported by the teacher all the time through e-mail and tutor appointments.

By performing individual workds students must demonstrate their programming habilities and the capacities to use bioinformatics algorithms to solve the assigned cases. Moreover, students have to show understanding of the probabilistic issues behind of algorithms applied in each of the cases. Both coursework will be presented to the class, students must show the programme elaborated by themselves works come along with a clear and concise explanation.

### Student work load:

Teaching mode	Teaching methods	Estimated hours
<b>Classroom activities</b>	Master classes	25
	Practical exercises	15
	Practical work, exercises, problem-solving etc.	16
	Coursework presentations	1
	Workshops	4
	Extra-curricular activities (visits, conferences, etc.)	2
<b>Individual study</b>	Tutorials	3
	Individual study	24
	Individual coursework preparation	32
	Research work	15
	Compulsory reading	4
	Recommended reading	4

	Other individual study activities	5
	<b>Total hours:</b>	150

### ASSESSMENT SCHEME:

#### Calculation of final mark:

Written tests:	20 %
Individual coursework:	35 %
Final exam:	30 %
Presentaciones:	15 %
<b>TOTAL</b>	<b>100 %</b>

\*Las observaciones específicas sobre el sistema de evaluación serán comunicadas por escrito a los alumnos al inicio de la materia.

### BIBLIOGRAPHY AND DOCUMENTATION:

#### Basic bibliography:

JONES, Neil C.; PEVZNER, Pavel A. An introduction to bioinformatics algorithms. MIT press, 2004.  
 COMPEAU, Phillip; PEVZNER, P. A. Bioinformatics algorithms: an active learning approach, Vol. I. SI: ACTIVE LEARNING, 2015.  
 COMPEAU, Phillip; PEVZNER, P. A. Bioinformatics algorithms: an active learning approach, Vol. II. SI: ACTIVE LEARNING, 2015.  
 DURBIN, Richard, et al. Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge university press, 1998.  
 ROCHA, M., & Ferreira, P. G. (2018). Bioinformatics Algorithms: Design and Implementation in Python. Academic Press.

#### Recommended bibliography:

ANDERSON, James WJ, et al. Evolving stochastic context-free grammars for RNA secondary structure prediction. BMC bioinformatics, 2012, vol. 13, no 1, p. 78.  
 NEBEL, Markus E.; SCHEID, Anika. Evaluation of a sophisticated SCFG design for RNA secondary structure prediction. Theory in Biosciences, 2011, vol. 130, no 4, p. 313-336.  
 MODEL, Mitchell L. Bioinformatics Programming Using Python: Practical Programming for Biological Data. " O'Reilly Media, Inc.", 2009.  
 ROCHA, M., & Ferreira, P. G. (2018). Bioinformatics Algorithms: Design and Implementation in Python. Academic Press.

#### Recommended websites:

Documentación python	<a href="https://docs.python.org/2.7/">https://docs.python.org/2.7/</a>
Problemas python	<a href="http://rosalind.info/problems/locations/">http://rosalind.info/problems/locations/</a>
Bioconductor	<a href="http://bioconductor.org/">http://bioconductor.org/</a>
Algoritmos bioinformáticos	<a href="http://www.bioalgorithms.info">http://www.bioalgorithms.info</a>