



18245 - HERRAMIENTAS DE PROGRAMACIÓN EN BIOQUÍMICA Y BIOLOGÍA MOLECULAR

Información de la asignatura

Código - Nombre: 18245 - HERRAMIENTAS DE PROGRAMACIÓN EN BIOQUÍMICA Y BIOLOGÍA MOLECULAR

Titulación: 531 - Graduado/a en Bioquímica

Centro: 104 - Facultad de Ciencias

Ámbito: Bioquímica y biotecnología

Curso Académico: 2025/26

1. Detalles de la asignatura

1.1. Materia

Herramientas de Programación en Bioquímica y Biología Molecular (Itinerario de Biomedicina Molecular / Itinerario de Biología Molecular y sus Aplicaciones)

1.2. Carácter

Optativa

1.3. Nivel

Grado (MECES 2)

1.4. Curso

4

1.5. Semestre

Primer semestre

1.6. Número de créditos ECTS

6.0

1.7. Idioma

Español

1.8. Requisitos previos

Ninguno.

1.9. Recomendaciones

Prerrequisitos para cursar el módulo:

- Conocimientos de Biología Molecular como los adquiridos en los primeros cursos del Grado en Bioquímica o del

Código Seguro de Verificación:		Fecha:	01/09/2025	1/6
Firmado por:	Esta guía docente no estará firmada mediante CSV hasta el cierre de actas			
Url de Verificación:		Página:	1/6	

- Grado en Biología
- Conocimientos básicos de informática a nivel de usuario.

1.10. Requisitos mínimos de asistencia

La asistencia a las clases teóricas y de resolución de casos prácticos es muy recomendable.

1.11. Coordinador/a de la asignatura

Luis Peso Ovalle

<https://autoservicio.uam.es/paginas-blancas/>

1.12. Competencias y resultados del aprendizaje

1.12.1. Competencias / Resultados del proceso de formación y aprendizaje

1. Competencias generales

CG5.- Haber desarrollado las habilidades de aprendizaje necesarias para emprender estudios posteriores de especialización con un alto grado de autonomía, incluyendo la capacidad de asimilación de las distintas innovaciones científicas y tecnológicas que se vayan produciendo en el ámbito de las Biociencias Moleculares.

CG4.- Capacidad para transmitir información, ideas, problemas y soluciones dentro del área de la Bioquímica y Biología Molecular, incluyendo la capacidad de comunicar aspectos fundamentales de su actividad profesional a otros profesionales de su área, o de áreas afines, y a un público no especializado.

2. Competencias específicas.

CE1.- Entender las bases físicas y químicas de los procesos biológicos a nivel celular y molecular y conocer las herramientas empleadas para investigarlas y adquirir las habilidades matemáticas, estadísticas e informáticas para obtener, analizar e interpretar datos de sistemas biológicos.

3. Competencias transversales

Entre las competencias transversales, esta asignatura enfatiza la siguiente:

CT7.- Capacidad de utilizar las herramientas informáticas básicas para la comunicación, la búsqueda de información, y el tratamiento de datos en su actividad profesional.

1.12.2. Resultados de aprendizaje

- Entender los componentes fundamentales de un ordenador y sus relaciones (CPU, RAM, disco duro, periféricos).
- Entender las diferencias entre distintos tipos y formatos de ficheros, y utilizar editores apropiados para los distintos tipos de ficheros; comprender la organización de un sistema de ficheros y manipular y moverse entre los mismos desde la línea de comandos.
- Programar en dos lenguajes de alto nivel (Python y R) para llevar a cabo tareas de complejidad media para la manipulación de datos incluyendo su lectura desde otros ficheros y su escritura a ficheros en distintos formatos y su representación gráfica.
- Entre los problemas que los estudiantes serán capaces de resolver se incluyen saber encontrar motivos (motifs) de longitud determinada en ficheros en formato FASTA descargados de internet, poder manipular secuencias de DNA o proteínas, realizar sencillos análisis estadísticos repetitivos en conjuntos masivos de datos, o producir representaciones gráficas de calidad de las búsquedas o de los análisis previos.
- Saber instalar y configurar nuevo software y librerías.
- Acceder a bases de datos y llevar a cabo "queries" sencillas.

1.12.3. Objetivos de la asignatura

El objetivo fundamental de esta asignatura es dotar al estudiante de herramientas y conocimientos de programación de utilidad en bioquímica y biología molecular y biología.

1.13. Contenidos del programa

Tema 0. Presentación de la asignatura. Objetivos, estructura y método docente y de evaluación del curso. Contenidos. Introducción a la línea de comandos y los editores de texto para programación.

Código Seguro de Verificación:		Fecha:	01/09/2025	2/6
Firmado por:	Esta guía docente no estará firmada mediante CSV hasta el cierre de actas			
Url de Verificación:		Página:	2/6	

BLOQUE I. Python

Tema P1. Variables, expresiones, operadores en python. Tipos. Variables. Operadores y precedencia. Expresiones. Estructuras de datos.

Tema P2. Ejecución condicional, recursión, iteración. Operadores lógicos. Flujo de ejecución (if). Ejecución condicional y alternativa. Recursión. Iteración (for, while).

Tema P3. Lectura y escritura de datos. Estructura de árbol de directorios. Lectura de ficheros de disco. “Parseado” de ficheros. Escritura de ficheros. Manejo de argumentos pasados por línea de comando.

Tema P4. Funciones y modularidad. Funciones: argumentos, valores devueltos, void functions. Composición de funciones. Flujo de ejecución. Variables locales y globales. Importar módulos.

Tema P5. Expresiones regulares. Qué son y para qué sirven las expresiones regulares. Expresiones regulares en Python.

Tema P6. Programación orientada a objetos. Clases, funciones, métodos, herencia. Biopython. Acceso a bases de datos y búsquedas sencillas.

BLOQUE II. R.

Tema R1. Variables, expresiones, operadores en R. Tipos. Variables. Operadores y precedencia. Expresiones. Estructuras de datos. Vectores. Tipos de datos. Datos categóricos.

Tema R2. Ejecución condicional, recursión, iteración en R. Operadores lógicos. Flujo de ejecución (if). Ejecución condicional y alternativa. Recursión. Iteración (for, while).

Tema R3. Lectura y escritura de datos en R. Estructura de árbol de directorios. Lectura de ficheros de disco. “Parseado” de ficheros. Escritura de ficheros. Manejo de argumentos pasados por línea de comando. Organización de datos en forma de “Data Matrix”.

Tema R4 Funciones y modularidad en R. Funciones: argumentos, valores devueltos, void functions. Composición de funciones. Variables locales y globales. Importar librerías. CRAN y Bioconductor.

Tema R5. Visualización de datos en R. Sistema “base” de gráficos en R. Selección de visualizaciones de datos. Parámetros básicos de los gráficos: color, etiquetas, ejes, leyenda. Introducción a ggplot (“grammar of graphics plot”). Sintaxis en ggplot2. Gráficos interactivos. Gráficos por categorías (facets).

Tema R6. Transformación de datos en R. Transformaciones básicas: sort, merge, cbind, subset. Transformaciones avanzadas: familia apply, aggregate.

Tema R7. Desarrollo de proyectos en RStudio. Elaboración de informes de análisis de datos con R-Markdown y Quarto.

Tema R8. Aplicaciones de R específicas para bioquímica y biología molecular. Análisis de datos -ómicos.

1.14. Referencias de consulta

(Los bloques I, II, III, y V seguirán muy de cerca Haddock y Dunn, con algún material de Murrell. El bloque IV utilizará sobre todo Matloff y Zuur et al. Para todos los temas hay abundante, excelente, y gratuita bibliografía. Incluimos algunos de nuestros favoritos. Los enlaces a Software Carpentry contienen estupendo material sobre Python y el bloque III.).

Código Seguro de Verificación:		Fecha:	01/09/2025	3/6
Firmado por:	Esta guía docente no estará firmada mediante CSV hasta el cierre de actas			
Url de Verificación:		Página:	3/6	

Bessant, C., Shadforth, I., Oakley, D. 2008 Building Bioinformatics Solutions: With Perl, R and MySQL. Oxford University Press

Ceder, Vernon L. 2nd revised edition 2010. The Quick Python Book. Manning Publications

Downey, A. 2008. Think Python: How to Think like a Computer Scientist. Green Tea Press.
<http://www.greenteapress.com/thinkpython/thinkpython.html>

Grolemund, G, Wickham, H. 2016. R for Data Science. O'Reilly. (See also <http://r4ds.had.co.nz/>)

Haddock, S. H. D., Dunn, C. W. 2011. Practical computing for biologists. Sinauer.

Kabacoff, R.I. R in Action, 2nd edition. Manning Publications.

Lott SF. 2010. Bulding Skills in Python. http://homepage.mac.com/s_lott/books/python.html

Matloff, N. 2011. The art of R programming. No Starch Press.

Murrell P. 2009. Introduction to Data Tecnologies. Chapman & Hall. Disponible también desde <http://www.stat.auckland.ac.nz/~paul/ltDT/>

Paradis E. 2005. R for beginners. http://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf

Peng, R. 2015. R Programming for Data Science. <https://leanpub.com/rprogramming>

Peng, R. et al. 2017. Mastering software development in R. Leanpub, <https://leanpub.com/msdr>. (See also <https://bookdown.org/rdpeng/RProgDA/>)

Software Carpentry, Versión 3. http://software-carpentry.org/3_0/

Software Carpentry, Versión 4. http://software-carpentry.org/4_0/

van Rossum G. 2011. Python Tutorial. <http://docs.python.org/tutorial/> .

Wickham, H. 2014. Advanced R. Chapman and Hall/CRC (see also <http://adv-r.had.co.nz>)

Zelle, John M. 2nd edition 2010. Python Programming: An Introduction to Computer Science. Franklin, Beedle & Associates Inc.

Zuur, A. F., Ieno, E. N. Meesters, E. H. W. G. 2009. A beginner's guide to R. Springer.

2. Metodologías docentes y tiempo de trabajo del estudiante

2.1. Presencialidad

	#horas
Porcentaje de actividades presenciales (mínimo 33% del total)	48,5
Porcentaje de actividades no presenciales	101,1

2.2. Relación de actividades formativas

Actividades presenciales	Nº horas
Clases teóricas en aula	0

Código Seguro de Verificación:		Fecha:	01/09/2025	4/6
Firmado por:	Esta guía docente no estará firmada mediante CSV hasta el cierre de actas			
Url de Verificación:		Página:	4/6	

Seminarios	0
Clases prácticas en aula	
Prácticas clínicas	
Prácticas con medios informáticos	41
Prácticas de campo	
Prácticas de laboratorio	
Prácticas externas y/o practicum	
Trabajos académicamente dirigidos	
Tutorías	2
Actividades de evaluación	5
Otras	

Todas las clases son teórico-prácticas y requieren el uso de ordenador. Se recomienda que los estudiantes traigan su propio equipo pero, si no fuera posible, se les proporcionará un ordenador de la universidad.

Bloque I:

Este bloque se desarrollará utilizando clases teórico-prácticas en aula combinados con la resolución de ejercicios de programación, tanto individuales como grupales. Se evaluará tanto los ejercicios presentados, como la corrección del ejercicio de otros compañeros.

Bloque II:

Este bloque se desarrollará mediante metodología de Aprendizaje Basado en Proyectos (PBL). Los proyectos planteados tendrán como objetivo elaborar métodos para el análisis de datos derivados de experimentos de bioquímica y biología molecular, que puedan ser reutilizados fácilmente. Se evaluarán tanto la entrega final como las entregas parciales y la corrección del trabajo de otros compañeros.

3. Sistemas de evaluación y porcentaje en la calificación final

3.1. Convocatoria ordinaria

La asignatura incluye un examen parcial y un examen final por cada bloque de la asignatura (python y R). Cada estudiante puede decidir si presentarse al solo al parcial, solo al final o a ambos exámenes en cada uno de los bloques de la asignatura. La nota del bloque en este apartado será la máxima alcanzada en los dos exámenes (los exámenes no presentados se contabilizarán con valor de cero).

Además, se evaluarán los problemas, casos prácticos de programación y comentarios a los programas de otros grupos, así como el proyecto presentado en el bloque II. Las actividades no realizadas, o entregadas fuera de plazo, se contabilizarán con valor de cero en los cálculos de notas.

La nota final del curso se calcula de la forma siguiente

- Resolución de ejercicios, casos prácticos de programación en Python y su evaluación: 30%.
- Examen bloque I (python): 20% calculado como máximo(parcial_python , final_python)
- Proyecto de R: 30% (además de la presentación final, se evaluarán las entregas parciales)
- Examen bloque II (R): 20% calculado como máximo(parcial_r , final_r)

La evaluación de las entregas prácticas en ambos bloques se realizará mediante entregas grupales, según los grupos de trabajo asignados en cada caso. Todos los miembros del grupo tendrán la misma calificación, que podrá ser ponderada por la contribución realizada por cada miembro, estimada mediante una encuesta en la que cada uno de los miembros del grupo evaluará de manera argumentada a los demás miembros y/o preguntas por parte del profesor a cada uno de los miembros del equipo sobre el trabajo realizado.

3.1.1. Relación actividades de evaluación

Actividad de evaluación	%
Examen final (máximo 70% de la calificación final o el porcentaje que figure en la memoria)	40
Evaluación continua	60

Código Seguro de Verificación:		Fecha:	01/09/2025	5/6
Firmado por:	Esta guía docente no estará firmada mediante CSV hasta el cierre de actas			
Url de Verificación:		Página:	5/6	

3.2. Convocatoria extraordinaria

Todos los estudiantes que no superen la asignatura en la convocatoria ordinaria serán evaluados en la convocatoria extraordinaria por un único examen que tendrá una puntuación máxima de 10. La nota final será la media entre este examen (con un peso del 60%) y la nota obtenida en la resolución de ejercicios y casos prácticos de programación (peso 40%). Los estudiantes tendrán la opción de completar los casos prácticos de programación en Python y/o el apartado(s) del Proyecto de R que no hayan realizado durante el curso antes del examen final extraordinario.

3.2.1. Relación actividades de evaluación

Actividad de evaluación	%
Examen final (máximo 70% de la calificación final o el porcentaje que figure en la memoria)	60
Evaluación continua	40

4. Cronograma orientativo

Semanas	Bloque	Horas presenciales	Horas no presenciales
1-6	I	18	36
8	Eval. Bloque I	1.5	5
7-15	II	22.5	45
15	Eval. Bloque II	1.5	5
NA	final exam	1.5	10

Código Seguro de Verificación:		Fecha:	01/09/2025	6/6
Firmado por:	Esta guía docente no estará firmada mediante CSV hasta el cierre de actas			
Url de Verificación:		Página:	6/6	