COURSE UNIT (MODULE) DESCRIPTION

Course unit (module) title	Code
GNU/Linux type operating systems	

Lecturer(s)	Department(s) where the course unit (module) is delivered
Coordinator: Saulius Gražulis **Readers: Andrius Merkys, Antanas Vaitkus**	Department of Protein-DNA Interactions, Institute of Biotechnology, Life Sciences Center, Saulėtekio av. 7, LT-10257 Vilnius, Lithuania

Study cycle	Type of the course unit (module)
Second cycle	Elective

Mode of delivery	Period when the course unit (module) is delivered	Language(s) of instruction
Face-to-face, self-study Lectures, seminars and practice		English

Requirements for students	
Prerequisites: an introductory programming course; an introductory computer architecture course; English B2 level	**Additional requirements (if any): acquaintance with non-decimal numeric systems (binary, octal)**

Course (module) volume in credits	Total student's workload	Contact hours	Self-study hours
5	132	64	68

Purpose of the course unit (module): programme competences to be developed
The systems biology course aims to teach students the skills to analyse and critically assess large volumes of biological data. Computers are absolutely necessary to handle such amounts of data efficiently. Currently, most popular and powerful computing systems for data analysis and computer simulations (computer clusters, HPC computers) are based on Unix or GNU/Linux operating systems, and good command of these systems is an essential ingredient for efficient learning and later for productive work in the systems biology field. The purpose of this module is to acquire the necessary skills in working with GNU/Linux operating systems and to perform tasks necessary for biomedical data analysis using these systems. Students should be able to use command line of the GNU/Linux systems efficiently, combine command line tools with visual GUI programs and to master basic data and computation management skills.

Learning outcomes of the course unit (module)	Teaching and learning methods	Assessment methods
1.1 Have general understanding about secure communication and confidential information management.	Lectures, hands-on tutorials, practical assignments	Completion of practical assignments; classroom discussions
2.1 Be able to perform duties within the deadlines and goals of a project	Hands-on tutorials, practical assignments	Completion of practical assignments; written examinations and/or quizzes in an automatic learning system

4.4 Be able to operate GNU/Linux operating systems	Lectures, hands-on tutorials, practical assignments	Completion of practical assignments; written examinations and/or quizzes in an automatic learning system
4.5 Be able perform practical calculations using modern high-performance open computing platforms	Lectures, hands-on tutorials, practical assignments	Completion of practical assignments; written examinations and/or quizzes in an automatic learning system
5.3 Be able to manage computation workflows for data analysis in the field of systems biology	Lectures, hands-on tutorials, practical assignments	Completion of practical assignments; written examinations and/or quizzes in an automatic learning system; data analysis report
6.2 Be able to gather and analyse information on subjects related to system biology with a critical approach, and to carry out a technological watch	Lectures, hands-on tutorials, practical assignments	Completion of practical assignments; data analysis report

Content: breakdown of the topics	**Contact hours**							**Self-study work: time and assignments**	
	Lectures	Tutorials	Seminars	Exercises	Laboratory work	Internship/work	**Contact hours**	**Self-study hours**	**Assignments**
1. Principles of GNU/Linux and Unix(R) architecture. History, impact and prevalence. Modern GNU/Linux OSes: Linux, *BSD, Solaris. Open source and proprietary systems. Influence on modern OSes (MacOS X, DOS/Windows, Android)	2		1	1			4	4	Discussion of GNU/Linux advantages and drawbacks
2. Starting to work with GNU/Linux: login, logout, passwords; terminal window, shell. Unix Shell. Introduction to the file system. Commands 'which', 'locate', 'pwd'. Shell wildcards.	2		1	1			4	4	GNU/Linux command practice: simple commands
3. GNU/Linux command line and basic commands (cd, cp, mv, pwd, mkdir, rmdir, cat, echo). Text editors. Command, parameter and options syntax. Getting help. Comfortable work with the command line (tab-expansions, abbreviations, aliases). Well-known aliases. Safely using 'rm', 'mv' and 'cp'.	2		1	1			4	4	GNU/Linux command practice: more advanced commands with options
4. Standard Unix input and output streams. Unix pipes and program composition. Filters. Commands 'grep', 'tr', 'cut', 'paste'. Microlanguages 'sed', 'awk'; using Perl as a microlanguage.	2		1	1			4	4	GNU/Linux command practice: composing commands into pipelines
5. Shell scripting. Shell languages: control structures, loops, variables. String substitution and expansion. Processing command line parameters in Shell. Recommendations on writing reusable and convenient Shell scripts.	2		1	1			4	4	Writing a simple shell script; writing a more advanced shell script with command line argument and option processing and with temporary file management.

6. GNU/Linux file system. Files, directories, root, symbolic links and hard links; special files and devices. Absolute and relative paths. Hard links and symbolic links; commands 'cp -a', 'cp -r', 'ln', rsync. Massive file processing; commands 'find', 'xargs', 'ls -R', 'seq'. Creating archives with 'zip' and 'tar'. File compression with 'gzip' and 'bzip2'. Recommendations on convenient project layout and data management.	2		1	1			4	4	GNU/Linux command practice: modifying the file system contents
7. Unix security model. Access rights. Executable programs. User groups. Commands 'chmod' and 'chown'. System administrator and her/his rights. Using setuid/setgid bits and programs.	2		1	1			4	4	GNU/Linux command practice: changing file permissions
8. GNU/Linux processes. Process and job control. Commands 'bg', 'fg', 'jobs', 'kill'. Starting program in a background (command modifier '&'). Hot-key sequence Ctrl-Z. Inter-process communication (signals, pipes, shared memory, message queues, semaphores). Possible problems and their solutions: file locking, temporary files, deadlock.	2		1	1			4	4	GNU/Linux command practice: managing jobs and processes
9. How Shell finds commands, files, programs and how loader finds libraries. Variables PATH, LD_LIBRARY_PATH and friends. Environment variables.	2		1	1			4	4	GNU/Linux command practice: configuring programs
10. GNU/Linux networking. Commands 'ssh', 'telnet', 'scp', 'rcp', 'rsync', 'wget/wput', 'curl'. Web servers and the REST interface.	2		1	1			4	5	GNU/Linux command practice: accessing remote computers
11. GNU/Linux programming environment: compilers, linkers, IDEs. General purpose languages (c, gcc, g++, tcc, javac, g77, gfortran, ada, chill, fpc, ...). Project assembly using Make, Ant, Maven. Application languages R, Gnuplot, Sage, Maxima, Python. Computational "notebooks". Writing, installing and using your own programs. Version control systems (Subversion, GIT, Mercurial, Bzr, RCS).	2		1	1			4	5	Writing a simple program Makefile
12. Managing computation workflows: Shell scripts, Makefiles. Using version control and Makefiles to manage computational workflows.	2		1	1			4	4	Building a Makefile to perform simple bioinformatics computations.
13. GNU/Linux administration basics: root account; installing and uninstalling programs and packages. Creating and removing user accounts. Good passwords. Commands 'useradd', 'userdel', 'usermod', 'chsh', 'chown', 'chgrp'. The /etc directory. Security basics. Virtual machines. Installing GNU/Linux.	2		1	1			4	4	Installing a GNU/Linux system on a virtual machine; creating user accounts, installing additional packages and managing them.
14. High performance computing (HPC): background jobs, queues. Queue management systems (Torque, Slurm; commands 'qsub', 'qrm', 'qstat'...). Computer clusters; SMP machines. Managing and sharing system resources; HPC "etiquette".	2		1	1			4	4	Writing a shell script for HPC job submission. Submitting a job to a cluster and getting its result.
15. GNU/Linux text processing facilities: nroff/troff, TeX, LaTeX, LibreOffice. Managing bibliographies with BibTeX and JabRef.	2		1	1			4	4	Writing a report in LaTeX

16. Secure communication and confidential information management: strong encryption, key generation, key signing, key management. Command 'gpg' and the GPG system. Using encryption for managing e-mail (Thunderbird Enigmail, Kleopatra, GPG). Encrypted file systems. Plausible denial. Program packages and system image signatures and checksums, their checks. Server certificates, their checks. Encrypted Web channels using SSL and TLS. Secure boot.	2		1	1			4	4	Creating a GPG public/private key pair; creating an ssh key pair. Establishing secure connection to a remote computer, sending and receiving a secure e-mail.
Total	**32**	**0**	**16**	**16**	**0**	**0**	**64**	**68**	

Assessment strategy	**Weight, %**	**Deadline**	**Assessment criteria**
Lecture quizzes	10	10 min. at the beginning of each practical.	4-question quiz covering several recent lectures (Blooms 1 an 2 level questions) using an electronic teaching environment (Moodle, Open edX or similar).
Intermediate quiz	15	mid-term	approx. 30-question quiz covering several recent lectures (Blooms 1 to 9 level questions) using an electronic teaching environment (Moodle, Open edX or similar).
Evaluation of practical assignments	50	After each practical according to the announced schedule	The results of an assigned practical exercise (Shell session logs, computation directory archives, self-created Shell scripts) are uploaded to the electronic teaching environment.
Oral and written report on the performed computations	10	end of term	Students provide a written (2 page) LaTeX report on the performed computations, emphasising the use of GNU/Linux tools in these computations, and provide a 5 min. Beamer slide-backed presentation of their report.
Final exam	15	end of term	approx. 30-question quiz covering several recent lectures (Blooms 1 to 9 level questions) using an electronic teaching environment (Moodle, Open edX or similar).
Total	100		The final mark is obtained summing up all points obtained for each task, quiz or assignment, dividing them by 100 and rounding using the math number rounding rules (.5 rounds to the larger integer). The maximum possible points add up to at least 1000, but to no more than 1300.

Author	**Year of publication**	**Title**	**Issue of a periodical or volume of a publication**	**Publishing place and house or web link**
Compulsary reading				
Arnold Robbins	2005	Unix in a Nutshell, 4th Edition	–	O'Reilly Media http://shop.oreilly.com/product/9780596100292.do
Jerry Peek, Shelley Powers, Tim O'Reilly, Mike Loukides	2002	Unix Power Tools, 3rd Edition	–	O'Reilly Media http://shop.oreilly.com/product/9780596003302.do Google Books https://books.google.lt/books?id=tDDb5zRoONwC&printsec=frontcover&hl=lt&source=gbs_atb#v=onepage&q&f=false
Randal K. Michael	2008	Mastering Unix Shell Scripting, 2nd Edition	–	Wiley http://shop.oreilly.com/product/9780470183014.do
Katherine Wrightson, Joseph Merlino	2006	Mastering UNIX	–	Wiley / Sybex http://shop.oreilly.com/product/

				9780782128178.do
Optional reading				
Pike, Rob; Kernighan, Brian W.	1983, 2009	The Unix Programming Environment	–	Prentice-Hall
Керниган Б.В., Пайк Р.	1992	UNIX - универсальная среда программирования	–	Финансы и статистика
Brown, Peter J.	1984	Starting with UNIX	–	Addison-Wesley
Браун П.	1987	Введение в операционную систему UNIX	–	Москва, Мир
S. R. Bourne	1983	The UNIX System	–	Addison-Wesley
Баурн С.	1986	Операционная система UNIX	–	Москва, Мир