



## Organisation Advanced bachelor of Bioinformatics





# Preface

---

This document contains information about the organisation and structure of the Advanced bachelor of Bioinformatics programme. It also gives more detailed information about the different course units, with a general objective, details about ECTS, contact hours, lecturers and learning goals per course unit.

Should you have any more questions concerning this educational programme, a possible collaboration, study or traineeship exchange, please do not hesitate to contact us ([bit@howest.be](mailto:bit@howest.be)).

Team BIT@HOWEST

**Howest University of Applied Sciences**  
**Campus Brugge Station – BST5**  
Rijselstraat 5 - 8200 Bruges, Belgium  
Tel.: +32 50 38 12 77  
Fax: +32 50 38 11 71  
<https://www.howest.be/en/bioinformatics>

# Index

---

1	Course unit BIT01 Linux operating systems.....	6
2	Course unit BIT02 Web technologies .....	8
3	Course unit BIT03 Structural bioinformatics .....	10
4	Course unit BIT04 Databases and data management.....	11
5	Course unit BIT05 Scripting .....	13
6	Course unit BIT06 Comparative genomic analysis .....	14
7	Course unit BIT07 Machine learning .....	16
8	Course unit BIT08 Data analysis, visualization and biostatistics using R.....	18
9	Course unit BIT09 High throughput analysis.....	19
10	Course unit BIT10 Systems biology .....	20
11	Course unit BIT11 Traineeship in Belgium or abroad .....	22

# Introduction

---

The programme of the Advanced bachelor of Bioinformatics is organised in one year (60 ECTS). A bioinformatician requires a profound knowledge of both bioinformatics, programming and informatics. Each course unit is therefore linked to one of these roles (bioinformatician, programmer and informatician).

To improve the study ability, the course units are organised using a quad-semester system (table 1). In this system, there are 4 blocks or periods of 6 to 8 weeks each. In each block, 3 to 4 course units are covered, followed by an examination period. In the last block, a traineeship – without a bachelor's thesis – is performed, after which the traineeship subject is presented to a jury of the labour market.

In the first period (semester 1, period A), the course units BIT 01 Linux operating systems, BIT 02 Web technologies and BIT 03 Structural bioinformatics are taught and evaluated.

In the second period (semester 1, period B), the course units BIT 04 Databases and data management, BIT 05 Scripting, BIT 06 Comparative genomic analysis and BIT 09 High throughput analysis (part 1) are taught and evaluated.

And in the third period (semester 2, period C), the course units BIT 07 Machine Learning, BIT 08 Data analysis, visualization and biostatistics using R, BIT 09 High throughput analysis (part 2) and BIT 10 Systems biology are the last course units which are taught and evaluated. At the end, there is the traineeship.

Table 1: organisation of the Advanced bachelor of Bioinformatics programme (colors indicate the different roles: blue: informatician, red: programmer, green: bioinformatician)

	Period	Module	ECTS
Semester 1	A	BIT 01 Linux operating systems	5
		BIT 02 Web technologies	5
		BIT 03 Structural bioinformatics	5
	B	BIT 04 Databases and data management	5
		BIT 05 Scripting	5
		BIT 06 Comparative genomic analysis	5
BIT 09 High throughput analysis		5	
Semester 2	C	BIT 07 Machine Learning	5
		BIT 08 Data analysis, visualization and biostatistics using R	5
		BIT 10 Systems biology	5
		BIT 11 Traineeship in Belgium or abroad	10
	D		

# Information per course unit

---

## 1 Course unit BIT01 Linux operating systems

---

### General objective

Linux is a free and open source operation system, which is frequently used in bioinformatics. A ton of scientific software is developed to run in a Linux/UNIX environment. Therefore different course units require Linux to install domain-specific software. Next to this, in Linux the most common used programming languages (e.g. Python, Perl, C) are already installed and ready to use. The application of bioinformatic tools for saving, organising and analysing molecular biological data and automising them by Bash shell-scripting are easier in a Linux environment.

The course unit BIT01, Linux operating systems, starts with installing Fedora, a popular Linux distribution within the domain of bioinformatics. This is followed by dealing with Linux (UNIX) commands, so they can be used in a terminal. These commands allow the user to work with files, manage processes, manipulate input or output, adapt the work environment, ... A next step teaches the student to draw up simple "bash" scripts.

### Details

Semester (+ period)	1A
Education language	English
Role	Informatician
Level	Beginner
ECTS	5
# Contact hours	48

### Learning goals

#### Knowledge

- The student describes the type and origin of a Linux operation system and gives the most important applications
- The student knows the basic commands to work in the terminal:
  - The student knows the commands to navigate the map structure
  - The student knows the commands to search and work with files
  - The student knows the commands to secure files
  - The student knows the commands to manage and plan processes
  - The student knows the 'redirection' operators for input, output and errors
  - The student knows the commands to install programs (including (de)archiving and (de)compress files)
- The student knows the basic elements and structure of a (shell) script:
  - The student knows the different kinds of variables
  - The student knows the different steps to build up a script
  - The student knows the different rules that apply to convert input
  - The student knows the conditional rules and loop structures

## ***Insights***

- The student uses basic commands to work in the terminal
  - The student navigates the map structure using commands
  - The student searches files and works with them
  - The student adapts the security settings of files
  - The student manages and plans processes
  - The student redirects input, output and error using redirection operators
  - The student installs new programs and commands
- The student searches information in manuals and help sections to understand the syntax of a command
- The student builds a script with variables, conditional rules and loop structures

## ***Applying knowledge and insights***

- The student communicates with a remote server using the Secure Shell (SSH) network protocol and secure file transfer protocol (SFTP)
- The student makes use of conda to create a container for reproducible bioinformatics research
  - The student creates a self-contained environment using the conda package manager
  - The student installs bioinformatics software, including dependencies, from bioconda
- The student builds a docker image and runs it as a container for reproducible bioinformatics research

## 2 Course unit BIT02 Web technologies

---

### General objective

Nowadays scientists rely more and more on web based applications for analysing their data. These applications are easy accessible, do not require installation and can be consulted on several devices by accessing a simple URL.

There are two main objectives of this course unit. On the one hand, this course unit provides an introduction to programming. The student learns to automatically analyse data via basic programming skills. On the other hand, the course unit deals with some basics on how to analyse and visualize data through web based applications. The students learn how data is sent to a server and how it is analysed and visualised in a browser.

This course unit serves as a basis for subsequent programming course units.

### Details

Semester (+ period)	1A
Education language	English
Role	Programmer
Level	Beginner
ECTS	5
# Contact hours	48

### Learning goals

#### Knowledge

- The student knows different HyperText Markup Language (HTML) elements and their attributes
- The student knows the proper build-up of a valid HTML page
- The student describes the different Cascading Style Sheets (CSS) that can be used to design a webpage:
  - The student knows the different CSS selectors and knows how to position and style an element
  - The student knows the common used CSS attributes and their possible values
- The student knows the different Hypertext Preprocessor (PHP) constructs (e.g. variables, if/else constructs, loops, functions ...) and knows the correct syntax to create a valid PHP script
- The student knows what a version control system (VCS) is and the different steps to synchronise code with a remote (like github.com)

#### Insights

- The student creates a logical and structured program
- The student selects the correct method/algorithm to solve a problem



***Applying knowledge and insights***

- The student creates a valid HTML5 document using HTML elements and their attributes:
  - The student creates a syntactically correct Document Object Model Tree (DOM tree) using semantic correct elements and attributes
  - The student creates HTML forms and sends data via the correct HTTP method (GET, POST) to the backend PHP scripts.
  - The student loads up additional accessories (pictures, CSS, ...) and uses them correctly
- The student designs his webpage via Cascading Style Sheets (CSS):
  - The student designs the global structure of a webpage
  - The student designs specific elements, using CSS3 selectors
- The student creates dynamic web pages using Hypertext Preprocessor (PHP) as programming language:
  - The students creates a PHP script and executes it in command line
  - The student starts up a webserver (built in PHP or Apache)
  - The student creates dynamic and interactive websites by applying PHP, using (superglobal) variables, loops, if/else constructs, functions and others

## 3 Course unit BIT03 Structural bioinformatics

---

### General objective

Structural bioinformatics is an interdisciplinary science that connects structural biology and informatics which mainly deals with structure and function of proteins. First, the necessary basic knowledge about DNA and protein structure is being taught. In a second part the methods, used to experimentally determine structures, are being discussed. Information about these determined structures are world widely collected and saved in the Protein Data Base (PDB). In a (logical) next step, this database and other databases are being explored. The data that can be retrieved from these structural databases is necessary for molecular visualisation, the next part of this course unit. In this part, students will be acquainted with molecular visualisation software (PyMOL), where 3D structures can be analysed. In the end, structure validation, function, prediction and protein interactions are being introduced. Here all of the previously gained knowledge will be combined.

### Details

Semester (+ period)	1A
Education language	English
Role	Bioinformatician
Level	Beginner
ECTS	5
# Contact hours	48

### Learning goals

#### **Knowledge**

- The student describes the basic structural principles of proteins and nucleic acids
- The student explains the current methods to experimentally determine macromolecular structures
- The student gives methods to find out the functions and interactions of proteins within the domain of structural bioinformatics
- The student describes methods to predict protein structures, gives the different steps to create comparative homology models and knows the public databases where the models are stored

#### **Applying knowledge and insights**

- The student validates the quality of a macromolecular structure
- The student searches if whether or not structural information about a protein or nucleic acid is available
- The student analyses a macromolecular structure using PyMOL

## 4 Course unit BIT04 Databases and data management

---

### General objective

On a daily basis people are confronted with databases, including in biological research, where nowadays they could not be missed. Exchanging data and using different research results for further analysis requires structured (data)storage. Next to public accessible databases, it is an asset that a bioinformatician can assemble a decent database system on his own. The used system is highly dependable of the further applications. In addition, it's also important to manage data that isn't stored in a database in an orderly manner. This data should then be findable, accessible (if allowed), interoperable and reusable.

In this course unit, the importance and basic structure of a database system is taught. A separation is made between relational and NoSQL system, this is done by working with MySQL and MongoDB. In both systems, the student learns how to assemble a database, fill it with relevant data and retrieve, adjust and delete this data. Next to this, correct normalisation of data to construct a decent database model is being learned. Moreover the principles to retrieve data from public databases like the Ensembl REST API is also taught. In addition, the student learns how to subject all created files to version control by using the revision software git. Finally the student learns to store his data according to the FAIR principles and knows about the importance of GDPR and data management plans.

### Details

Semester (+ period)	1B
Education language	English
Role	Informatician
Level	Beginner
ECTS	5
# Contact hours	48

### Learning goals

#### Knowledge

- The student gives the different types of public accessible bioinformatics databases and describes some alternative data retrieving methods for these databases
- The student describes the structure of the most common file formats within public databases like the structure of a GFF and a GenBank file
- The student describes the basic concepts and elements concerning databases and the submersive technology
  - The student gives the general structure of a database system
  - The student explains the different levels of a database system
- The student describes the general features of a database management system and enumerates the differences between a SQL and NoSQL system
- The student enumerates the different components of Structured Query Language
- The student names the different column types within a relational database
- The student describes the FAIR principles for biological datasets and scripts and gives examples of how to implement these principles

- The student explains what a data management plan is and the importance of GDPR for these plans

### ***Insights***

- The student selects the appropriate data base system based upon a problem
- The student constructs a relational database model and determines whether a relational or non-relational system is being used
- The student creates a database based upon a series of information
  - The student constructs a database with correlating tables
  - The student completes the tables in a database with relevant data
- The student determines whether data is in accordance with the FAIR principles

### ***Applying knowledge and insights***

- The student performs search commands within a database databank
  - The student retrieves data from a relational database
  - The student retrieves data from a NoSQL database
- The student normalizes data to construct a database model and applies correctly the normalisation rules
- The student subjects his files to version control and saves the files in a GIT repository
- The student queries online databases based upon a problem
  - The student retrieves data from the correct bio informatics database
  - The student retrieves data using Ensembl REST API

## 5 Course unit BIT05 Scripting

---

### General objective

Finding an answer to a biological question in the work field often requires tailored experiments and analyses. Next to developing new laboratory techniques, this also implies software innovations. Tasks like processing, archiving and visualising data are inseparable from an experimental set up.

Whether the bioinformatician is an independent researcher or has a supporting function, he must be able to: a) rapidly recognize core computer related problems and solve these problems, b) apply an efficient and goal oriented solution and c) deal with the rising quantity of data generated by high throughput analyses.

The competences to execute and automatize these tasks using Python scripts are being taught in this course unit. The acquired skills are essential to become a bioinformatician and data scientist.

### Details

Semester (+ period)	1B
Education language	English
Role	Programmer
Level	Beginner
ECTS	5
# Contact hours	48

### Learning goals

#### *Applying knowledge and insights*

- The student uses the most important datatypes in Python
- The student uses loop structures and functions to organize his Python script and to execute calculations and repetitive tasks
- The student applies the basic principles of programming in Python
- The student reads or selects and writes or deposes data in files or a database (system)
- The student uses different bioinformatics related libraries and frameworks during scripting

## 6 Course unit BIT06 Comparative genomic analysis

---

### General objective

Comparative genomic analysis is driven by sequencing different genomes, including human, mice, yeast, chimpanzee ... Comparing the human genome with those of other organisms allows us to gain new insights in structure and function of human genes. With these new insights new strategies to cure human diseases can be developed. Next to this, comparative genomic analysis is a powerful tool to study evolutionary changes between organisms, resulting in identification of conserved gene regions between species and species specific genes (allowing to have different characteristics per organism) ([www.genome.gov](http://www.genome.gov), 2016).

Within this course unit, following things are dealt with: firstly, the most important bioinformatics (sequence) databases. Next the focus is on using online software resources, like BLAST and similar programs, to align sequences and execute online searches. The student is being taught to efficiently build phylogenetic trees and to explain genomic structures based on evolution. Finally the (human) genetic variation and the biomedical use of these applications are linked to this.

In an assignment, the installation and use of popular bioinformatics tools (like BLAST, HMMER and/or PHYLIP) in Linux are examined.

### Details

Semester (+ period)	1B
Education language	English
Role	Bioinformatician
Level	Beginner
ECTS	5
# Contact hours	48

### Learning goals

#### Knowledge

- The student describes the basic concepts and elements of comparative genomic analysis
  - The student defines comparative genomic analysis
  - The student knows the most important, public bioinformatics databases
  - The student knows the most important genomic browsers
- The student describes the method to execute a “pairwise sequence alignment”
  - The student defines homologous/orthologous and paralogous sequences
  - The student describes the difference between “global alignment” and “local alignment”
  - The student enumerates the different BLAST programs
  - The student describes alternative programs and algorithms to solve problems and limitations of the standard BLAST programs
- The student describes the different steps to construct a phylogenetic tree

- The student enumerates the different “multiple sequence alignment” algorithms with their advantages and limitations
- The student explains the corrections performed by the different nucleotide substitution models
- The student describes the different phylogenetic tree construction algorithms with their advantages and limitations
- The student defines natural genetic variation and describes causes and consequences of this genetic variation

### ***Insights***

- The student selects the right bioinformatic database based upon a problem and/or research question (including genome browsers)
- The student executes a pairwise sequence alignment
  - The student determines the relation (homologous/orthologous/paralogous) between sequences
  - The student executes a BLAST search
  - The student selects alternative programs to solve issues of standard BLAST programs
- The student constructs a phylogenetic tree starting with individual sequences
  - The student executes a multiple sequence alignment and explains the chosen algorithm
  - The student makes use of bioinformatics software to construct a phylogenetic tree and explains the chosen algorithm
- The student researches the link between natural genetic variation and genetically determined disorders

### ***Applying knowledge and insights***

- The student installs and executes (simple) bioinformatics software in command line
- The student works fluently with colleagues, helps or asks for help wherever possible or needed, does more than just the bare minimum ...

## 7 Course unit BIT07 Machine learning

---

### General objective

During the last years, the major progress in genomics and proteomics has resulted in a substantial amount of biological data. Often, this data can be analysed using existing tools or scripts. However, in some cases the data is too complex to analyse it using a fixed ruleset. This is where machine learning and, associated with it, artificial intelligence come in to play.

Within this course unit, the student gets acquainted with the different machine learning techniques. The most used programming language for machine learning is Python. Using Jupyter Notebook, an open-source web application running Python in the background, the student first explores the data and adapts it if necessary. Next off, the different machine learning techniques like classification, clustering and dimensionality reduction are discussed. The student learns to create models for each of these techniques and tries to improve them by hyperparameter tuning. Finally, the models are tested using, for the model, unseen data. A conclusion about these results is formulated. The focus of this course unit is, as much as possible, focused on bioinformatics applications. However, examples from other disciplines are also discussed.

### Details

Semester (+ period)	2C
Education language	English
Role	Informatician
Level	Advanced
ECTS	5
# Contact hours	48

### Learning goals

#### Knowledge

- The student gives an overview of the different subdomains of AI and the different kinds of machine learning and deep learning techniques
  - The student describes supervised learning and the corresponding machine learning techniques
  - The student describes unsupervised learning and the corresponding machine learning techniques
  - The student describes reinforcement learning
  - The student gives the different subdomains of AI by giving a (biological) example and describes the different kinds of machine learning techniques

#### Insights

- The student explores the structure of a (biological) data set
  - The student describes the different kinds of variables (including the class) and gives an example for each kind of variable
  - The student studies the information value of variables by analysing the distribution of measurements for one or more variables



- The student describes outliers and missing values and removes them if necessary
- The student prepares a (biological) data set for analysis by adapting the file format or by converting variables
  - The student selects a biological data set and converts the rough data set to a file format, suitable for analysis with a machine learning algorithm
  - The student describes and applies different techniques to convert variables, like normalisation and discretisation of variables

### ***Applying knowledge and insights***

- The student uses a linear regression algorithm to construct a regression model of a (biological) dataset
- The student uses a logistic regression algorithm to construct a classification model of a (biological) dataset
- The student uses the Naïve Bayes algorithm, based upon probability analysis, to construct a classification model of a (biological) data set with class
- The student uses a decision tree algorithm to construct a classification model of a (biological) data set with class
- The student uses an unsupervised algorithm, to divide (biological) data in meaningful subsets or to reduce the dimensionality
- The student uses dimensionality reduction to create a smaller set of principle variables and combines this with supervised learning techniques
- The student compares classification methods for one or more (biological) data sets
- The student uses the neural network algorithm to construct a classification model that divides multidimensional input in meaningful subsets

## 8 Course unit BIT08

### Data analysis, visualization and biostatistics using R

---

#### General objective

Knowledge of data analysis and data visualization is essential for a bioinformatician to allow critical analysis of experimental data. During other course units, scripting languages were already introduced. R has become a major statistic programming language within the domain of bioinformatics and data science with a broad applicability. Therefore, learning this scripting language is an essential part within the educational programme of an advanced bachelor of bioinformatics.

#### Details

Semester (+ period)	2C
Education language	English
Role	Programmer
Level	Advanced
ECTS	5
# Contact hours	48

#### Learning goals

##### *Applying knowledge and insights*

- The student knows and applies the different data types and structures in R
- The student selects and applies the correct visualization technique for graphical representation of variables in a dataset
- The student searches, installs, tests and applies R and Bioconductor packages
- The student conducts data manipulations on data sets and writes R code to solve a problem
- The student applies the basic knowledge about statistics and executes a descriptive statistical analysis to evaluate biological and biomedical data sets
- The student applies the different statistical tests for analysing one or more variables in a dataset and reports the statistical results in a brief and scientifically approved manner

## 9 Course unit BIT09 High throughput analysis

---

### General objective

In this course unit the analysis of big data sets, generated by the newest experimental methods, is being taught. Because of the ongoing evolution of analysis methods within the domain of life science, more and more data can be generated during an experiment. Therefore, it is important for a bioinformatician to gain insights about this. In this way, a bioinformatician becomes essential for the contemporary scientific research, where he or she gets a specific role in setting up data analysis pipelines.

### Details

Semester (+ period)	1B + 2C
Education language	English
Role	Bioinformatician
Level	Advanced
ECTS	5
# Contact hours	48

### Learning goals

#### ***Applying knowledge and insights***

- The student independently searches, installs and applies software that can specifically be used to analyse big biological data sets
- The student decides, based upon the gained knowledge concerning the different sorts of high-throughput experiments, how to manipulate a large biological data set and how to analyse it with the necessary techniques and software
- The student connects logically a number of analysis steps with each other using an analysis pipe line
- The student critically interprets the data used in and the results generated by a high throughput analysis
- The student reports the results of a high throughput analysis in a brief and scientifically approved manner

## 10 Course unit BIT10 Systems biology

---

### General objective

In Systems biology the 'system' of biological components is studied and computational modelled. A living system is dynamic and has many interactions between different biological components, from genes and proteins to metabolites, ... and others. Studying these dynamic interactions deflects from the reductionist manner of working and allows to understand and model the system and its complex nature.

Within this course unit, the basic principles of a system biological research strategy, the different kinds of 'omics' and the most important high throughput technologies are being discussed. The focus is on the one hand on constructing and simulating a biochemical model and on the other hand on the analysis of protein and genetic interaction networks. To construct and simulate a biochemical model, Cell Designer software is being used. Analysis of the interaction networks is done in Cytoscape. On top of analysing the construction or topology of an interaction network, the interaction networks are being enriched using text mining, gene ontology, expression data and other functional annotation data.

Finally, the student gets acquainted with an open, web based platform with computational tools to analyse high throughput data within the life science research.

### Details

Semester (+ period)	2C
Education language	English
Role	Bioinformatician
Level	Advanced
ECTS	5
# Contact hours	48

### Learning goals

#### **Knowledge**

- The student describes the basic principles of a systems biology research strategy, the different kinds of 'omics' and the most important high throughput technologies

#### **Insights**

- The student searches in relevant databases for information of a biological or biochemical pathway and downloads this information in a correct file format
- The student describes the most important protein-protein interactions and how they were determined

#### **Applying knowledge and insights**

- The student uses pathway simulation software, like Cell Designer, to construct and simulate a biochemical model
- The student downloads interaction data from a database and constructs a protein or genetic interaction network in network visualisation software, like Cytoscape
- The student uses basic functionalities of network visualisation software to analyse and manipulate the construction (nodes and edges) of interaction networks

- The student uses text mining and gene ontology to construct an interaction network and to analyse these for common features
- The student identifies complexes and modules in interaction networks and enriches interaction networks with expression data to determine clusters based on co-expression
- The student makes use of (public) functional annotation data to infer interaction networks and to analyse existing networks
- The student uses an open, web based platform with computational tools to analyse high throughput data within life science research

## 11 Course unit BIT11 Traineeship in Belgium or abroad

---

### General objective

The final step in the advanced bachelor programme is the actual experience in the work field: the traineeship. The objective is to apply several learning outcomes in work situations. Because of its size in duration and ECTS and as it is the ideal step-up in the work field, the traineeship is the perfect method to function as a bioinformatician in a complex authentic situation.

The self-directed behaviour is being stimulated by letting the student draw up a reflection portfolio. During the traineeship there is reflection about personal development by using the learning outcomes, the formative intermediary evaluation, the received feedback and the intercultural and international aspects when being a bioinformatician.

During the traineeship, all useful information is recorded in an electronic traineeship notebook. Git is to be used as the version-control system for tracking changes in computer files. Preferably the Howest Git server is to be used. In case a Git repository of the work placement must be used, a clone of this repository must be committed to the Howest Git server, by the end of the traineeship. All useful information, including results, is recorded in a documentation system, which is described in a Traineeship Documentation Plan.

The abstract summarizes the most essential parts of the traineeship.

To conclude the traineeship, the traineeship project is being presented to a judge (consisting of representatives from the work field) followed by a defence.

### Details

Semester (+ period)	2D
Education language	English
Role	Bioinformatician, informatician, programmer, New Young Professional
Level	Specialized
ECTS	10
# Contact hours	295 nCU traineeship + 5 nCU reflection task

### Learning outcomes

- BIT0100 The advanced bachelor in bioinformatics autonomously clarifies and solves a biological problem by selecting a relevant programming language and by applying it efficiently to develop an own program
- BIT0200 The advanced bachelor in bioinformatics autonomously adjusts existing programs and data structures according to the in the work field expected programming skills to contribute to an improvement of the application possibilities
- BIT0300 The advanced bachelor in bioinformatics manages, processes and retrieves biological complex data in a user-friendly manner by using existing or still to be developed database or software structures
- BIT0400 The advanced bachelor in bioinformatics actively follows the present developments in the work field of bioinformatics including ethical features of the matter to formulate solutions for a given biological problem

- BIT0500 The advanced bachelor in bioinformatics chooses, depending on the given biological problem, the relevant software and efficiently uses this software to clarify and solve the given biological problem
- BIT0600 The advanced bachelor in bioinformatics autonomously develops a multidisciplinary perspective on bioinformatics which joins biological and computational skills in practical applications within an authentic context
- BIT0700 The advanced bachelor in bioinformatics reports transparent and correct about research data and analysis results with the proper technical terminology. He/she cooperates constructively, respectfully and in a team-oriented way in an intraprofessional and interprofessional context