॥ सा विद्या या विमुक्तये ॥

स्वामी रामानंद तीर्थ मराठवाडा विद्यापीठ, नांदेड
'ज्ञानतीर्थ', विष्णुपुरी, नांदेड – ४३१ ६०६ (महाराष्ट्र राज्य) भारत
SWAMI RAMANAND TEERTH MARATHWADA UNIVERSITY, NANDED
'Dnyanteerth', Vishnupuri, Nanded - 431 606 (Maharashtra State) INDIA
स्वामी रामानंद तीर्थ मराठवाडा विद्यापीठ, नांदेड Established on 17th September, 1994, Recognized By the UGC U/s 2(f) and 12(B), NAAC Re-accredited with'B++' grade

Fax : (02462) 215572
Phone: (02462)215542
Academic-1 (BOS) Section
website: srtmun.ac.
E-mail: bos@srtmun.ac.

विज्ञान व तंत्रज्ञान विद्याशाखे अंतर्गत राष्ट्रीय शैक्षणिक धोरण २०२० नुसार पदव्यूत्तर द्वितीय वर्षाचे अभ्यासक्रम (Syllabus) शैक्षणिक वर्ष २०२४–२५ पासून लागू करण्याबाबत.

परिपत्रक

या परिपत्रकान्वये सर्व संबंधितांना कळविण्यात येते की, या विद्यापीठा अंतर्गत येणा–या सर्व संलग्नित महाविद्यालयामध्ये शैक्षणिक वर्ष २०२४–२५ पासून राष्ट्रीय शैक्षणिक धोरणानुसार पदव्यूत्तर द्वितीय वर्षाचे अभ्यासक्रम लागू करण्याच्या दृष्टीकोनातून विज्ञान व तंत्रज्ञान विद्याशाखे अंतर्गत येणा–या अभ्यासमंडळांनी तयार केलेल्या पदव्यूत्तर द्वितीय वर्षाच्या अभ्यासक्रमांना मा. विद्यापरिषदेने दिनांक १५ मे २०२४ रोजी संपन्न झालेल्या बैठकीतील विषय क्रमांक १५/५९–२०२४ च्या ठरावाअन्वये मान्यता प्रदान केली आहे. त्यानुसार विज्ञान व तंत्रज्ञान विद्याशाखेतील खालील एम. एस्सी द्वितीय वर्षाचे अभ्यासक्रम (Syllabus) लागू करण्यात येत आहेत.

1) M. Sc. II year Biotechnology (Affiliated College)
2) M. Sc. II year Biotechnology (Campus)
3) M. Sc. II year Bioinformatics (Sub Campus Latur)
4) M. Sc. II year Bioinformatics (Affiliated College)
5) M. Sc. II year Clinical Research (Affiliated College)
6) M. Sc. II year Botany (Campus)
7) M. Sc. II year Herbal Medicine
8) M. Sc. II year Boany (Affiliated College)
9) M. Sc. II year Geology (Campus)
10) M. Sc. II year Dairy Science
11) M. Sc. II year Electronics
12) M. Sc. II year Environmental Science
13) M. Sc. II year Environmental Science (Campus)
14) M. Sc. II year Geography (Campus)
15) M. Sc. II year Applied Mathematics
16) M. Sc. II year Mathematics
17) M. Sc. II year Mathematics (Campus)
18) M. Sc. II year Microbiology
19) M. Sc. II year Microbiology (Campus)
20) M. Sc. II year Statistics
21) M. Sc. II year Statistics (Campus)

सदरील परिपत्रक व अभ्यासक्रम प्रस्तुत विद्यापीठाच्या **www.srtmun.ac.in** या संकेतस्थळावर उपलब्ध आहेत. तरी सदरील बाब ही सर्व संबंधितांच्या निदर्शनास आणून द्यावी, ही विनंती.

'ज्ञानतीर्थ' परिसर,
विष्णुपुरी, नांदेड – ४३१ ६०६.
जा.क्र.:शै–१/एनइपी/विवत्रंविपदवी/२०२४–२५/९०८
दिनांक १२.०६.२०२४

डॉ. सरिता लोसरवार
सहा.कुलसचिव
शैक्षणिक (१–अभ्यासमंडळ) विभाग

प्रत : १) मा. आधिष्ठाता, विज्ञान व तंत्रज्ञान विद्याशाखा, प्रस्तुत विद्यापीठ.
२) मा. संचालक, परीक्षा व मुंल्यमापन मंडळ, प्रस्तुत विद्यापीठ.
३) मा. प्राचार्य, सर्व संबधित संलग्नित महाविद्यालये, प्रस्तुत विद्यापीठ.
४) मा. संचालक, सर्व संकुले परिसर व उपपरिसर, प्रस्तुत विद्यापीठ
५) सिस्टीम एक्सपर्ट, शैक्षणिक विभाग, प्रस्तुत विद्यापीठ. याना देवून कळविण्यात येते की, सदर परिपत्रक संकेतस्थळावर प्रसिध्द करण्यात यावे.

SWAMI RAMANAND TEERTH MARATHWADA UNIVERSITY, NANDED.

STRUCTURE AND SYLLABUS OF TWO YEAR MASTERS PROGRAM IN SCIENCE

(R-2023)

UNDER

NATIONAL EDUCATION POLICY (NEP 2020)

In

SUBJECT: BIOINFORMATICS

FACULTY OF SCIENCE AND TECHNOLOGY

M. Sc. Second Year

AFFILIATED COLLEGES

With Effect From June 2024

From the Desk of the Dean, Faculty of Science and Technology:

Swami Ramanand Teerth Marathwada University, Nanded, enduring to its vision statement "***Enlightened Student: A Source of Immense Power***", is trying hard consistently to enrich the quality of science education in its jurisdiction by implementing several quality initiatives. Revision and updating curriculum to meet the standard of the courses at national and international level, implementing innovative methods of teaching-learning, improvisation in the examination and evaluation processes are some of the important measures that enabled the University to achieve ***the 3Es, the equity, the efficiency*** and ***the excellence*** in higher education of this region. To overcome the difficulty of comparing the performances of the graduating students and also to provide mobility to them to join other institutions the University has adopted the cumulative grade point average (CGPA) system in the year 2014-2015. Further, following the suggestions by the UGC and looking at the better employability, entrepreneurship possibilities and to enhance the latent skills of the stakeholders the University has adopted the Choice Based Credit System (CBCS) in the year 2018-2019 at graduate and post-graduate level. This provided flexibility to the students to choose courses of their own interests. To encourage the students to opt the world-class courses offered on the online platforms like, NPTEL, SWAYM, and other MOOCS platforms the University has implemented the credit transfer policy approved by its Academic Council and also has made a provision of reimbursing registration fees of the successful students completing such courses.

SRTM University has been producing a good number of high calibre graduates; however, it is necessary to ensure that our aspiring students are able to pursue the right education. Like the engineering students, the youngsters pursuing science education need to be equipped and trained as per the requirements of the R&D institutes and industries. This would become possible only when the students undergo studies with an updated and evolving curriculum to match global scenario.

Higher education is a dynamic process and in the present era the stakeholders need to be educated and trained in view of the self-employment and self-sustaining skills like start-ups. Revision of the curriculum alone is not the measure for bringing reforms in the higher education, but invite several other initiatives. Establishing industry-institute linkages and initiating internship, on job training for the graduates in reputed industries are some of the important steps that the University would like to take in the coming time. As a result, revision of the curriculum was the need of the hour and such an opportunity was provided by the New Education Policy 2020. National Education Policy 2020 (NEP 2020) aims at equipping students with knowledge, skills, values, leadership qualities and initiates them for lifelong learning. As a result the students will acquire expertise in specialized areas of interest, kindle their intellectual curiosity and scientific temper, and create imaginative individuals.

The curriculum given in this document has been developed following the guidelines of NEP-2020 and is crucial as well as challenging due to the reason that it is a transition from general science based to the discipline-specific-based curriculum. All the recommendations of the ***Sukanu Samiti*** given in the **NEP Curriculum Framework-2023** have been followed, keeping the disciplinary approach with rigor and depth, appropriate to the comprehension level of learners. All the Board of Studies (BoS) under the Faculty of Science and Technology of this university have put in their tremendous efforts in making this curriculum of international standard. They have taken care of maintaining logical sequencing of the subject matter with proper placement of concepts with their linkages for better understanding of the students. We take this opportunity to congratulate the Chairman(s) and all the members of various Boards of Studies for their immense contributions in preparing the revised curriculum for the benefits of the stakeholders in line with the guidelines of the **Government of Maharashtra regarding NEP-2020**. We also acknowledge the suggestions and contributions of the academic and industry experts of various disciplines.

We are sure that the adoption of the revised curriculum will be advantageous for the students to enhance their skills and employability. Introduction of the mandatory ***On Job Training, Internship program*** for science background students is praise worthy and certainly help the students to imbibe first hand work experience, team work management. These initiatives will also help the students to inculcate the workmanship spirit and explore the possibilities of setting up of their own enterprises.

Dr. M. K. Patil
Dean
Faculty of Science and Technology

Preamble: The National Education Policy 2020 (NEP 2020) is formulated to revamp education system and lay down road map for new India. This policy is framed based on the fundamental pillars of access, equity, quality, affordability, and accountability and seeks to transform India into a thriving knowledge society and a global knowledge superpower.

Some of the important features of National Education Policy are increasing Gross enrolment ratio in higher education, Holistic and multidisciplinary education with multiple entry/exit options, Establishment of academic bank of credit, Setting up of multidisciplinary education and research Universities and National Research Foundation, Expansion of open and distance learning to increase gross enrolment ratio, Internationalization of education, Motivated, energized and capable faculty, Online and digital education and Effective governance and leadership.

As per the National Education Policy, the Government of Maharashtra has proposed a model curriculum framework and an implementation plan for the State of Maharashtra. It is to suggest and facilitate the implementation of schemes and programs, which improve not only the level of academic excellence but also improve the academic and research environment in the state. The proposed curriculum framework endeavours to empower the students and help them in their pursuit for achieving overall excellence.

In view of NEP priority and in-keeping with its vision and mission, process of updating the curriculum is initiated and implemented in SRTM University at UG and PG level from the academic year 2023-2024.

Bioinformatics is an interdisciplinary field that combines biology, computer science, and statistics to analyze and interpret biological data. It involves the application of computational techniques and algorithms to store, organize, analyze, and visualize biological information, particularly genomic and proteomic data. Bioinformatics utilizes various computational tools and techniques to extract meaningful information from biological data. These include database management systems, data mining algorithms, machine learning methods, and statistical analysis. Researchers and scientists in the field of Bioinformatics use these tools to gain insights into biological processes, understand disease mechanisms, discover new drugs and therapies, and enhance our understanding of evolutionary relationships among species. Overall, Bioinformatics plays a vital role in advancing biological research by enabling the integration and analysis of large and complex datasets, ultimately contributing to discoveries and breakthroughs in fields such as genomics, proteomics, evolutionary biology, and personalized medicine.

Keeping in mind, BOS in Biotechnology and Bioinformatics prepared the curriculum to ensure up-to-date level of understanding of Bioinformatics. Studying Bioinformatics prepares the students for their career working either in educational institutions or industries in which they can be directly involved in the teaching, research and development. Also, to ensure uniform curriculum and its quality at UG/PG level, curriculum of different Indian Universities, syllabus of NET, SET, MPSC, UPSC and the UGC model curriculum are referred to serve as a base in updating the same.

The comments or suggestions from all teachers, students and other stakeholders are welcome for upbringing this curriculum.

Salient Features:

The syllabus of M Sc Bioinformatics has been framed to meet the requirement of Choice Based Credit System under NEP 2020. The courses offered here in will train and orient the students in the specific fields of Bioinformatics.

The Core Courses deals with Basics of Bioinformatics, Biochemistry, Cell and Molecular Biology, Programming in C, Databases and Tools in Bioinformatics, Immunology and Parasite Bioinformatics, Chemoinformatics and Drug Design, Python in Programming, Genomics and Proteomics, Perl Programming, Applications of Bioinformatics.

Apart from the core courses, the Department Specific Elective Courses deal with Statistics, Mathematics, Structural Bioinformatics, Programming in C++, Advanced Techniques in Bioinformatics, Programming in Java, Database Management System and Biological Sequence Analysis. These courses offered during this program are designed with the aim of imparting specific skills to the students which will lead to their employability. There are also two Research Projects in third and fourth semester respectively. This would help students to lay a strong foundation in the field of Bioinformatics.

Overall after completion of this course, students will acquire fundamental knowledge of applications of Bioinformatics.

Program Educational Objectives:

The Objectives of this program are:

PEO1: To offer postgraduate program in Bioinformatics based on the needs of industries, academic and research institutions worldwide.

PEO2: To expose the students to the different emerging fields of Bioinformatics.

PEO3: To update curriculum by introducing recent advances in the subject that enable the students to successfully face NET, SET, MPSC, UPSC and other competitive examinations.

PEO4: To train and orient the students so as to develop human resource for the educational institutes and other organizations.

PEO5: To inculcate analytical and application-oriented abilities to create active and frontline researchers and human resource for the industries.

PEO6: To develop specific skills amongst students for their employability and for the development of their own enterprises.

Program Outcomes:

The Outcomes of this program are:

PO1: This program will expose the students to the different emerging fields of Bioinformatics.

PO2: This will provide an updated curriculum with recent advances in the subject that enable the students to face NET, SET, MPSC, UPSC and other competitive examinations successfully.

PO3: This program shall train and orient the students so as to develop human resources for educational institutes and other organizations.

PO4: This program shall train and orient the students so as to develop active and frontline researchers and human resources for the industries.

PO5: This will also develop specific skills amongst students for their employability and for the development of their own enterprises.

Prerequisite:

The students seeking admission to this program should have knowledge of B Sc in Life Sciences, Bioinformatics, Biotechnology, Genetics, Computer Science, or a related discipline. The optional courses are offered to the students registered for post-graduate programs. Such students should have a basic knowledge of Bioinformatics and willing to gain additional knowledge in the field of Bioinformatics.

The students seeking admission to this program should have cleared B Sc or B Pharm or B Sc Agri from any statutory University.

Dr. Sunita D. Lohare
Chairman, BOS in Biotechnology and Bioinformatics,
Swami Ramanand Teerth Marathwada University,
Nanded 431606.

Details of the Board of Studies Members in the subject Biotechnology and Bioinformatics under the Faculty of Science & Technology, S.R.T.M. University, Nanded.

Sr No	Name of the Member	Designation	Sr No	Name of the Member	Designation
1	Dr Sunita Dhundiraj Lohare, Shri Havgiswami Mahavidyalaya, Udgir, Dist -Latur Mob 9284161504	Chairman	2	Dr Babasaheb S Surwase School of Life Sciences SRTM University, Nanded 431606. Mob 9075829767	Member
3	Dr Pratap V. Deshmukh Nagnath Arts, Commerce and Science College, Aundha Nagnath, Dist. Hingoli Mob 9637202024	Member	4	Dr Komal S. Gomare Dept of Biotechnology, Dayanand Science College, Latur Mob 9284238413	Member
5	Dr Vaibhav D. Deshpande, General Manager, Quality Corporate Office, Wockhardt, Mumbai Mob 9100988260	Member		--	
	Invitee Members				
6	Dr Laxmikant Kamble School of Life Sciences, SRTM University, Nanded 431606. Mob: 8669695555	Member	7	Dr M M V Baig Dept of Biotechnology, Yeshwant Mahavidyalaya, Nanded. Mob 9422170641	Member
8	Dr Sanjog T. Thul Environmental Biotechnology and Genomics Division, National Environmental and Engineering Research Institute (CSIR-NEERI). Nagpur. Mob 9881877072	Member	9	Dr Prashant Thakare Department of Biotechnology, SGB Amravati University, Amravati. Mob: 9822222822	Member
10	Dr Shivraj Hariram Nile Department of Food Science and Agriculture, National Agri-Food Biotechnology Institute (NABI), Mohali, Punjab. Mob 9561740707	Member	11	Dr Arun Ingale School of Life Sciences, North Maharashtra University, Umavinagar, Jalgaon. Mob: 9822708707	Member
12	Dr Makarand N. Cherekar Dept. of Biotechnology & Bioinformatics, MGM's College of CS and IT, Nanded. Mob: 9421454254	Member	13	Mr. Rameshwar S. Belnor Dept. of Biotechnology & Bioinformatics, MGM's College of CS and IT, Nanded. Mob: 9096430300	Member

Swami Ramanand Teerth Marathwada University, Nanded

Faculty of Science & Technology

Credit Framework and Structure of Two Year PG Program (NEP 2020)

Subject: M Sc Bioinformatics (Affiliated Colleges) (R-2023)

Year & Level	**Sem**	**Major Subject**		**RM**	**OJT / FP/CS (3-Cr)**	**Research Project**	**Practicals (1-Cr)**	**Credits**	**Total Credits**
		(DSC- 4 Cr)	(DSE- 3 Cr)						
1	1	SBIOC-401 Basics of Bioinformatics SBIOC-402 Biochemistry SBIOC-403 Cell and Molecular Biology	SBIOE-401 Statistics **OR** SBIOE-403 Mathematics	SVECR 401 Research Methodology (3-Cr)	--		SBIOP-401 Lab Course in Basics of Bioinformatics SBIOP-402 Lab Course in Biochemistry SBIOP-403 Lab Course in Cell and Molecular Biology SBIOE-402 Lab Course in Statistics **OR** SBIOE-404 Lab Course in Mathematics	22	44
	2	SBIOC-451 Programming in C SBIOC-452 Databases and Tools in Bioinformatics SBIOC-453 Immunology and Parasite Bioinformatics	SBIOE-451 Structural Bioinformatics **OR** SBIOE-453 Programming in C++	---	SBIOX-451 (O/F/C)	--	SBIOP-451 Lab Course in Programming in C SBIOP-452 Lab Course in Databases and Tools in Bioinformatics SBIOP-453 Lab Course in Immunology and Parasite Bioinformatics SBIOE-452 Lab Course in Structural Bioinformatics **OR** SBIOE-454 Lab Course in Programming in C++	22	
Exit option: Exit Option with PG Diploma in Basic Bioinformatics (After 2024-25)									
2	3	SBIOC-501 Chemoinformatics and Drug Designing SBIOC-502 Python Programming SBIOC-503 Genomics and Proteomics	SBIOE-501 Advanced Techniques in Bioinformatics **OR** SBIOE-503 Programming in Java	--	--	Research Project SBIOR-501 (4-Cr)	SBIOP-501 Lab Course in Chemoinformatics and Drug Designing SBIOP-502 Lab Course in Genomics and Proteomics SBIOE-502 Lab Course in Advanced Techniques in Bioinformatics **OR** SBIOE-504 Lab Course in Programming in Java	22	44
	4	SBIOC-551 Perl Programming SBIOC-552 Applications of Bioinformatics	SBIOE-551 Database Management System **OR** SBIOE-553 Biological Sequence Analysis	SVECP-551 Publication Ethics (2-Cr)	--	Research Project SBIOR-552 (6-Cr)	SBIOP-551 Lab Course in Perl Programming SBIOP-552 Lab Course in Applications of Bioinformatics SBIOE-552 Lab Course in Database Management System **OR** SBIOE-554 Lab Course in Biological Sequence Analysis	22	
Total Credits		44	12	05	03	10	14		**88**

DSE indicates Department Specific Elective Course. Bioinformatics student, in a particular semester, can opt either of these courses **OR** a course offered by the program of other Departments. DSC- Department Specific Core, OJT- On Job Training, FP- Field Project, CS- Case Study, RM- Research Methodology, Cr- Credit, VEC- Value Education Course, R- Revision, Credits of four semesters = 88, Total Marks of all four Semesters = 2200

M. Sc. Second Year Semester III (Level 7.0)

Teaching Scheme

Subject	Course Code	Course Name	Credits Assigned			Teaching Scheme (Hrs/ week)	
			Theory	Practical	Total	Theory (Hrs/ Week)	Practical (Hrs/ Week/Batch)
Major	SBIOC-501	Chemoinformatics and Drug Designing	04	--	04	04	--
	SBIOC-502	Python Programming	04	--	04	04	--
	SBIOC-503	Genomics and Proteomics	04	--	04	04	--
Elective (DSE)	SBIOE-501 SBIOE-503	Advanced Techniques in Bioinformatics **OR** Programming in Java	03	--	03	03	--
Research Project	SBIOR-501	Research Project	--	04	04	--	08
DSC Practical	SBIOP-501	Lab Course in Chemoinformatics and Drug Designing	--	01	01	--	02
	SBIOP-502	Lab Course in Genomics and Proteomics	--	01	01	--	02
DSE Practical	SBIOE-502 SBIOE-504	Lab Course in Advanced Techniques in Bioinformatics **OR** Lab Course in Programming in Java	--	01	01	--	02
Total Credits			**15**	**07**	**22**	**15**	**14**

M. Sc. Second Year Semester III (Level 7.0)

Examination Scheme

[20% Continuous Assessment (*CA)* and 80% End Semester Assessment (*ESA*)]

Subject	Course Code	Course Name	Theory				Practical		Total
			Continuous Assessment (CA)			ESA			
			Test I	Test II	Avg of (T1+T2)/2	Total	CA	ESA	
Major	SBIOC-501	Chemoinformatics and Drug Designing	20	20	20	80	--	--	100
	SBIOC-502	Python Programming	20	20	20	80	--	--	100
	SBIOC-503	Genomics and Proteomics	20	20	20	80	--	--	100
Elective (DSE)	SBIOE-501 SBIOE-503	Advanced Techniques in Bioinformatics **OR** Programming in Java	15	15	15	60	--	--	75
Research Project	SBIOR-501	Research Project	--	--	--	--	20	80	100
DSE Practical	SBIOP-501	Lab Course in Chemoinformatics and Drug Designing	--	--	--	--	05	20	25
	SBIOP-502	Lab Course in Genomics and Proteomics	--	--	--	--	05	20	25
DSE Practical	SBIOE-502 SBIOE-504	Lab Course in Advanced Techniques in Bioinformatics **OR** Lab Course in Programming in Java	--	--	--	--	05	20	25

M. Sc. Second Year Semester IV (Level 7.0)

Teaching Scheme

Subject	Course Code	Course Name	Credits Assigned			Teaching Scheme	
			Theory	Practical	Total	Theory (Hrs/ Week)	Practical (Hrs/ Week/Batch)
Major	SBIOC-551	Perl Programming	04	--	04	04	--
	SBIOC-552	Applications of Bioinformatics	04	--	04	04	--
Elective (DSE)	SBIOE-551 SBIOE-553	Database Management System **OR** Biological Sequence Analysis	03	--	03	03	--
Publication Ethics	SVECP-551	Publication Ethics	02	--	02	02	--
Research Project	SBIOR-551	Research Project	--	06	06	--	12
DSC Practical	SBIOP-551	Lab Course in Perl Programming	--	01	01	--	02
	SBIOP-552	Lab Course in Applications of Bioinformatics	--	01	01	--	02
DSE Practical	SBIOE-552 SBIOE-554	Lab Course in Database Management System **OR** Lab Course in Biological Sequence Analysis	--	01	01	--	02
Total Credits			**13**	**09**	**22**	**13**	**18**

M. Sc. Second t Year Semester IV (Level 7.0)

Examination Scheme

[20% Continuous Assessment (*CA)* and 80% End Semester Assessment (*ESA*)]

Subject	Course Code	Course Name	Theory				Practical		Total
			Continuous Assessment (CA)			ESA			
			Test I	Test II	Avg of (T1+T2)/ 2	Total	CA	ESA	
Major	SBIOC-551	Perl Programming	20	20	20	80	--	--	100
	SBIOC-552	Applications of Bioinformatics	20	20	20	80	--	--	100
Elective (DSE)	SBIOE-551 SBIOE-553	Database Management System **OR** Biological Sequence Analysis	20	20	20	80	--	--	100
Publication Ethics	SVECP-551	Publication Ethics	10	10	10	40	--	--	50
Research Project	SBIOR-551	Research Project	--	--	--	--	30	120	150
DSC Practical	SBIOP-551	Lab Course in Perl Programming	--	--	--	--	05	20	25
	SBIOP-552	Lab Course in Applications of Bioinformatics	--	--	--	--	05	20	25
DSE Practical	SBIOE-552 SBIOE-554	Lab Course in Database Management System **OR** Lab Course in Biological Sequen Analysis	--	--	--	--	05	20	25

SBIOC-501 Chemoinformatics and Drug Designing
Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBIOC-501	Chemoinformatics and Drug Designing	04	--	04	--	04

Assessment Scheme

Course Code	Course Name	Theory				Practical		Total
		CA			ESA			
		Test I	Test II	Avg of (T1+T2)/2		CA	ESA	
SBIOC-501	Chemoinformatics and Drug Designing	20	20	20	80	--	--	100

Course pre-requisite:

- **Basic Chemistry**: A solid understanding of general chemistry concepts is essential for chemoinformatics and drug design. This includes knowledge of atomic structure, chemical bonding, molecular geometry, and basic principles of organic and inorganic chemistry.
- **Biology/Biochemistry**: Since drug design often involves targeting biological molecules such as proteins, enzymes, or receptors, a background in biology or biochemistry is crucial. Students should be familiar with concepts such as molecular biology, cell biology, bioenergetics, and metabolism.
- **Computer Science/Programming**: Chemoinformatics heavily relies on computational methods and tools for data analysis, molecular modeling, and structure-activity relationship (SAR) studies. Therefore, a good understanding of computer science principles and proficiency in programming languages such as Python, R, or C++ is advantageous.

Course objectives:

- **Understanding Chemical Data Analysis Techniques**: This objective aims to equip students with the knowledge and skills to analyze and interpret chemical data using various computational methods and tools. Students will learn how to preprocess, visualize, and analyze chemical data sets, including molecular structures, chemical properties, and biological activities, using techniques such as statistical analysis, machine learning, and molecular modeling.
- **Application of Chemoinformatics in Drug Discovery**: This objective aims to familiarize students with the role of chemoinformatics in the drug discovery process. Students will learn about virtual screening, ligand-based and structure-based drug design approaches, pharmacophore modeling, QSAR/QSPR analysis, and other chemoinformatics methods used to identify, optimize, and design novel drug candidates.

Course outcomes:

- Understanding of Chemoinformatics Principles: Students will demonstrate a thorough understanding of the principles and techniques of chemoinformatics, including molecular descriptors, molecular modeling, QSAR/QSPR methodologies, molecular docking, and pharmacophore modeling. They will be able to apply these principles to analyze molecular structures, predict their properties, and design novel drug candidates.
- **Proficiency in Computational Tools:** Students will gain proficiency in using computational tools and software commonly employed in chemoinformatics and drug design. They will be able to utilize these tools to perform tasks such as molecular visualization, structure-based drug design, ligand-based virtual screening, and molecular dynamics simulations.
- **Application of Chemoinformatics in Drug Discovery**: By the end of the course, students will be able to apply chemoinformatics techniques to various stages of the drug discovery process. They will demonstrate the ability to retrieve, process, and analyze chemical data from databases, predict the biological activities of compounds, optimize molecular structures for improved drug-likeness and bioavailability, and prioritize lead compounds for further experimental validation.

Curriculum Details:

Module No.	Unit No.	Topic	Hrs.
1.0	**1**	**Introduction to Chemoinformatics**	15
	1.1	Overview of Chemoinformatics: Definition, scope, and applications in drug discovery. Chemical Structure Representation: Basics of molecular structure representation, SMILES notation, molecular graphs, and molecular fingerprints.	
	1.2	Molecular Descriptors: Types of molecular descriptors (1D, 2D, and 3D), calculation methods, and their significance in compound characterization.	
	1.3	Chemical Databases and Data Retrieval: Introduction to chemical databases (e.g., PubChem, ChEMBL) and techniques for data retrieval and management.	
	1.4	Chemical Data Visualization: Techniques for visualizing chemical structures and properties, including molecular visualization software.	
2.0	**2**	**Molecular Modeling and Computational Chemistry**	15
	2.1	Molecular Mechanics: Principles of molecular mechanics, force fields, energy minimization, and molecular dynamics simulations.	
	2.2	Molecular Docking: Theory and principles of molecular docking, scoring functions, and applications in virtual screening and lead optimization.	
	2.3	Pharmacophore Modeling: Basics of pharmacophore modeling, pharmacophore identification, and its role in ligand-based drug design.	
	2.4	Quantitative Structure-Activity Relationship (QSAR): Principles of QSAR modeling, descriptor selection, model building, and validation.	
3.0	3	**Structure-Based Drug Design**	15
	3.1	Definition and scope of bioinformatics	
	3.2	History of bioinformatics	
	3.3	Nature of biological data.	
	3.4	Applications of bioinformatics in biological research	
4.0	4	**Applications of Chemoinformatics in Drug Discovery**	15
	4.1	Protein Structure and Function: Basics of protein structure, folding, and function. Structure Determination Techniques: Overview of X-ray crystallography, NMR spectroscopy, and other techniques for protein structure determination.	
	4.2	Protein-Ligand Interactions: Understanding protein-ligand binding modes, binding sites, and interaction types.	
	4.3	Structure-Based Virtual Screening: Principles and methodologies of	

		structure-based virtual screening, including molecular docking and pharmacophore-based approaches.	
	4.4	De Novo Drug Design: Basics of de novo drug design, fragment-based drug design, and rational drug design strategies.	
		Total	**60**

R

References

1. Bunin, B. A., Siesel, B., Morales, G. A., & Bajorath, J. (2007). *Chemoinformatics theory* (pp. 1-49). Springer Netherlands.
2. Pearlman, R. S. (2004). An Introduction to Chemoinformatics By Andrew R. Leach (GlaxoSmithKline, Stevenage, UK) and Valerie J. Gillet (University of Sheffield, UK). Kluwer Academic Publishers: Dordrecht. 2003. xvi+ 260 pp. $74.00. ISBN: 1-4020-1347-7.
3. Faulon, J. L., & Bender, A. (2010). *Handbook of chemoinformatics algorithms*. CRC press.
4. Zhou, S. F., & Zhong, W. Z. (2017). Drug design and discovery: principles and applications. *Molecules*, *22*(2), 279.
5. Gasteiger, J., & Engel, T. (Eds.). (2006). *Chemoinformatics: a textbook*. John Wiley & Sons.
6. Merz Jr, K. M., Ringe, D., & Reynolds, C. H. (Eds.). (2010). *Drug design: structure-and ligand-based approaches*. Cambridge University Press.

SBIOP-501 Lab Course in Chemoinformatics and Drug Designing

Molecular Structure Representation:

- Introduction to drawing chemical structures using software tools like ChemDraw or MarvinSketch.
- Converting chemical structures to SMILES notation and vice versa.
- Generating 2D and 3D molecular structures from SMILES strings.

Molecular Descriptors Calculation:

- Calculation of basic molecular descriptors such as molecular weight, number of atoms, and molecular formula.
- Calculation of physicochemical properties like LogP, LogD, polar surface area (PSA), and hydrogen bond donors/acceptors using software packages like RDKit or Open Babel.
- Comparing molecular properties between different compounds.

Chemical Database Searching:

- Querying chemical databases (e.g., PubChem, ChEMBL) to retrieve compound information based on specific criteria such as chemical structure, activity, or target.
- Filtering and refining search results based on various parameters like molecular weight, activity values, and chemical similarity.

Molecular Visualization:

- Visualizing molecular structures in 2D and 3D using molecular visualization software such as PyMOL, VMD, or Jmol.
- Analyzing molecular properties and interactions (e.g., hydrogen bonding, hydrophobic interactions) using visualization tools.

Molecular Docking:

- Performing molecular docking simulations using software like AutoDock or Vina.
- Docking small molecule ligands into protein binding sites and analyzing docking poses.
- Scoring and ranking docked poses based on binding affinity.

Pharmacophore Modeling:

- Generating pharmacophore models using software tools like Discovery Studio or MOE.
- Validating pharmacophore models using known active compounds and decoys.
- Virtual screening of compound libraries against pharmacophore models to identify potential hits.

Quantitative Structure-Activity Relationship (QSAR):

- Building QSAR models to predict biological activity or property using software like RDKit or QSAR Toolbox.
- Evaluating QSAR model performance using statistical metrics like R^2, Q^2, and RMSE.
- Applying QSAR models to predict activity of new compounds and prioritize them for experimental testing.

Structure-Based Virtual Screening:

- Conducting structure-based virtual screening using molecular docking software.
- Screening compound libraries against protein targets of interest and identifying potential lead compounds.
- Analyzing docking results and selecting top-ranked hits for further evaluation.

SBIOC-502 Python Programming
Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBIOC-502	Python Programming	04	--	04	--	04

SBIOC-502 Python Programming
Assessment Scheme

Course Code	Course Name	Theory				Practical		Total
		CA			ESA			
		Test I	Test II	Avg of (T1+T2)/2		CA	ESA	
SBIOC-502	Python Programming	20	20	20	80	--	--	100

Course pre-requisite:

- Basic Understanding of Computer Fundamentals
- Foundational Knowledge of Programming Concepts
- Comfort with Mathematics and Logic

Course objectives:

- To gain a solid understanding of fundamental programming concepts such as variables, data types, operators, control structures (if statements, loops), functions, and exception handling in Python.
- To develop proficiency in using Python's built-in data structures like lists, tuples, dictionaries, and sets to solve various programming problems.
- To explore commonly used Python libraries and modules for different purposes such as numerical computation (NumPy), data manipulation (Pandas), visualization (Matplotlib, Seaborn), and scientific computing (SciPy). Learn how to install, import, and utilize Python packages effectively within your programs.

Course outcomes:

- Students will demonstrate a solid understanding of fundamental programming concepts using Python, including variables, data types, operators, control flow structures (such as if statements, loops), functions, and basic input/output operations. They will be able to write Python programs to solve simple computational problems and implement algorithms to manipulate data.
- By the end of the course, students will be proficient in Python syntax and commonly used libraries for tasks such as data manipulation, file handling, and basic data visualization.
- They will demonstrate the ability to write clean, well-structured Python code that adheres to best practices and efficiently utilizes built-in functions and libraries like NumPy, pandas, and matplotlib.

Curriculum Details:

Module No.	Unit No.	Topic	Hrs.
1.0	Unit I	**Introduction to Python Programming**	15
	1.1	Overview of Python: History, features, and applications.	
	1.2	Setting up Python Environment: Installation of Python interpreter, text editors, and integrated development environments (IDEs).	
	1.3	Python Syntax: Basics of Python syntax, variables, data types (integers, floats, strings, booleans), and basic arithmetic operations.	
	1.4	**Data Structures:** Lists and Tuples: Understanding lists and tuples, indexing, slicing, list methods, and tuple immutability. Dictionaries and Sets: Introduction to dictionaries and sets, key-value pairs, dictionary methods, and set operations.	
2.0	Unit II	**Control Flow, Functions and File handling**	15
	2.1	Control Flow: Introduction to control flow structures such as if statements	
	2.2	Control flow structures such as for loops, while loops, and their usage in Python programming.	
	2.3	Functions: Defining and calling functions, function arguments, return values, and local vs. global scope.	
	2.4	File Input/Output: Reading from and writing to files in Python, file modes, handling file exceptions, and file manipulation techniques.	
3.0	Unit III	**Advanced Python Concepts**	15
	3.1	Object-Oriented Programming (OOP) Basics: Introduction to OOP concepts such as classes, objects, attributes, and methods.	
	3.2	Inheritance and Polymorphism: Understanding inheritance, method overriding, and polymorphism in Python classes.	
	3.3	Exception Handling: Handling errors and exceptions using try-except blocks, raising exceptions	
	3.4	Exception hierarchy in Python.	
4.0	Unit IV	**Python Libraries and Applications**	15
	4.1	Introduction to Python Libraries: Overview of commonly used Python libraries such as NumPy, pandas, and matplotlib.	
	4.2	Data Manipulation with pandas: Basics of data manipulation and analysis using pandas DataFrame, data indexing, filtering, aggregation, and visualization.	
	4.3	Introduction to NumPy: Basics of numerical computing with NumPy arrays, array manipulation, mathematical operations, and broadcasting.	
	4.4	Basic Data Visualization: Introduction to data visualization using matplotlib, plotting graphs, histograms, scatter plots, and customizing plots.	
		Total	**60**

References

1. Matthes, E. (2023). Python crash course: A hands-on, project-based introduction to programming. no starch press.
2. McKinney, W. (2022). Python for data analysis. " O'Reilly Media, Inc.".
3. Lutz, M. (2013). Learning python: Powerful object-oriented programming. " O'Reilly Media, Inc.".
4. Eidelman, A. (2020). Python data science handbook by jake VANDERPLAS (2016). Statistique et Société, 8(2), 45-47.
5. Python Official Documentation: The official Python documentation (https://docs.python.org/3/) is an invaluable resource for learning Python syntax and basic programming concepts.
6. Python Documentation on File Handling: The Python documentation includes a section on file input/output operations (https://docs.python.org/3/tutorial/inputoutput.html), which covers reading from and writing to files, file modes, and exception handling.
7. Python Documentation on Classes: The Python documentation includes a tutorial on classes and object-oriented programming (https://docs.python.org/3/tutorial/classes.html), which provides a comprehensive overview of Python's OOP features.
8. Official Documentation for NumPy, pandas, and matplotlib: The official documentation for these libraries (NumPy: https://numpy.org/doc/, pandas: https://pandas.pydata.org/docs/, matplotlib: https://matplotlib.org/stable/contents.html) provides comprehensive guides, tutorials, and examples for using these libraries for data manipulation and visualization.

SBIOC-503 Genomics and Proteomics
Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBIOC-503	Genomics and Proteomics	04	--	04	--	04

SBIOC-503 Genomics and Proteomics
Assessment Scheme

Course Code	Course Name	Theory				Practical		Total
		CA			ESA			
		Test I	Test II	Avg of (T1+T2)/2		CA	ESA	
SBIOC-503	Genomics and Proteomics	20	20	20	80	--	--	100

Course pre-requisite:

- **Foundations of Molecular Biology**: This knowledge forms the basis for understanding genomic and proteomic analyses.
- Proficiency in biochemistry is crucial for comprehending the chemical principles underlying genomic and proteomic techniques.
- Familiarity with basic laboratory techniques and protocols is important for students to appreciate the experimental methodologies used in genomics and proteomics research

Course objectives:

- To gain a foundational understanding of genomic and proteomic principles, including DNA structure and function, gene expression, protein synthesis, post-translational modifications, and protein structure-function relationships.
- To develop skills in analyzing genomic and proteomic data using bioinformatics tools and software.
- To understand the applications of genomics and proteomics in biomedical research and biotechnology, including drug discovery, personalized medicine, agricultural biotechnology, and biomarker discovery.

Course outcomes:

- Students will develop a solid understanding of the fundamental principles, concepts, and technologies in genomics and proteomics.
- By the end of the course, students will gain practical experience with common experimental techniques used in genomics and proteomics research.
- Students will learn how to apply genomics and proteomics techniques to address biological questions and solve real-world problems in biomedical research.

Curriculum Details:

Module No.	Unit No.	Topic	Hrs.
1.0	**Unit I**	**Introduction to Genomics**	15
	1.1	Overview of Genomics: Definition, scope, and significance in biological research.	
	1.2	Central Dogma of Molecular Biology: Understanding DNA replication, transcription, and translation processes.	
	1.3	Genome Structure and Organization: Basics of genome organization in prokaryotes and eukaryotes, including chromosomes, genes, and non-coding regions.	
	1.4	Genomic Technologies: Introduction to DNA sequencing techniques, including Sanger sequencing, next-generation sequencing (NGS), and third-generation sequencing (e.g., PacBio, Nanopore).	
2.0	**Unit II**	**Functional Genomics**	15
	2.1	Gene Expression Analysis: Introduction to gene expression profiling techniques, including microarrays and RNA sequencing (RNA-seq).	
	2.2	Regulatory Genomics: Understanding gene regulation mechanisms, including transcriptional regulation, post-transcriptional regulation, and epigenetics.	
	2.3	Comparative Genomics: Basics of comparative genomics, genome annotation, and evolutionary genomics.	
	2.4	Genome-wide Association Studies (GWAS): Principles of GWAS, identifying genetic variants associated with diseases, and interpreting GWAS results.	
3.0	**Unit III**	**Introduction to Proteomics**	15
	3.1	Basics of Proteomics: Definition, scope, and applications in biological research.	
	3.2	Protein Structure and Function: Understanding protein structure, folding, domains, and motifs.	
	3.3	Protein Analysis Techniques: Introduction to protein purification methods, gel electrophoresis (e.g., SDS-PAGE), and Western blotting.	
	3.4	Mass Spectrometry in Proteomics: Principles of mass spectrometry, peptide fragmentation, and protein identification and quantification using MS-based proteomics.	
4.0	**Unit IV**	**Functional Proteomics and Systems Biology**	15
	4.1	Protein-Protein Interactions (PPIs): Techniques for studying protein-protein interactions, including yeast two-hybrid (Y2H) assays and co-immunoprecipitation (Co-IP).	
	4.2	Structural Proteomics: Basics of structural proteomics, protein structure determination methods (e.g., X-ray crystallography, NMR spectroscopy), and structure-function relationship studies.	
	4.3	Systems Biology Approaches: Introduction to systems biology concepts, network analysis, and modeling approaches for studying biological systems.	
	4.4	Integration of Genomics and Proteomics Data: Methods for integrating genomics and proteomics data to gain insights into biological processes, including pathway analysis and functional annotation.	

		Total	**60**

Reference:

1. Brown, T. A. (2023). Genomes 5. CRC Press.
2. Primrose, S. B., & Twyman, R. (2009). Principles of genome analysis and genomics. John Wiley & Sons.
3. Lesk, A. M. (2017). Introduction to genomics. Oxford University Press.
4. Kakidani, H., & Ptashne, M. (1988). GAL4 activates gene expression in mammalian cells. Cell, 52(2), 161-167.
5. Korpelainen, E., Tuimala, J., Somervuo, P., Huss, M., & Wong, G. (2014). RNA-seq data analysis: a practical approach. CRC press.
6. Rosenberg, M. S. (Ed.). (2009). Sequence alignment: methods, models, concepts, and strategies. Univ of California Press.
7. Twyman, R. (2004). Principles of proteomics. Taylor & Francis.
8. Palsson, B. Ø. (2011). Systems biology: simulation of dynamic network states. Cambridge University Press.

SBIOP-502 Lab Course in Genomics and Proteomics

1. **DNA Sequence Analysis:**
    - Practical exercise: Use bioinformatics tools (e.g., BLAST) to perform sequence similarity searches and identify homologous sequences.
    - Learning outcome: Gain proficiency in sequence analysis techniques and understand sequence conservation and divergence.
2. **Genome Assembly:**
    - Practical exercise: Assemble short DNA sequencing reads into contigs using bioinformatics software (e.g., SPAdes).
    - Learning outcome: Understand the process of de novo genome assembly and the challenges associated with assembling sequencing data.
3. **RNA-seq Data Analysis:**
    - Practical exercise: Analyze RNA-seq data to identify differentially expressed genes between experimental conditions using bioinformatics tools (e.g., DESeq2).
    - Learning outcome: Gain experience in RNA-seq data analysis, including quality control, read alignment, and differential expression analysis.
4. **Genome-Wide Association Studies (GWAS):**
    - Practical exercise: Perform a GWAS analysis to identify genetic variants associated with a phenotype of interest using publicly available genotype data and statistical software (e.g., PLINK).

- Learning outcome: Understand the principles of GWAS and learn how to conduct association tests and interpret GWAS results.

5. **Western Blot Analysis:**
    - Practical exercise: Perform Western blot experiments to detect and quantify specific proteins in cell lysates using antibodies and gel electrophoresis.
    - Learning outcome: Learn experimental techniques for protein detection and quantification and understand the principles of Western blotting.
6. **Protein-Protein Interaction Analysis:**
    - Practical exercise: Analyze protein-protein interaction networks using computational predictions from protein interaction databases.
    - Learning outcome: Understand the principles of protein-protein interaction networks and learn how to analyze and interpret interaction data.

SBIOE-501 Advanced Techniques in Bioinformatics
Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBIOE-501	Advanced Techniques in Bioinformatics	03	--	03	--	03

Assessment Scheme

Course Code	Course Name	Theory				Practical		Total
		CA			ESA			
		Test I	Test II	Avg of (T1+T2)/2		CA	ESA	
SBIOE-501	Advanced Techniques in Bioinformatics	15	15	15	60	--	--	75

Course pre-requisite:

- Basic knowledge of bioinformatics concepts, including sequence analysis, database searching, and molecular biology principles.
- Understanding of statistical concepts such as probability distributions, hypothesis testing, regression analysis, and statistical significance.
- Basic understanding of molecular biology principles, including DNA structure and function, gene expression regulation, and genetic variation.

Course objectives:

- The course aims to familiarize students with advanced bioinformatics tools, algorithms, and methodologies used for analyzing complex biological data.
- The course aims to provide hands-on experience with state-of-the-art bioinformatics software tools and databases.
- The course aims to enable students to apply advanced bioinformatics techniques to address complex biological problems.

Course outcomes:

- Students will develop proficiency in using advanced bioinformatics tools and techniques for analyzing complex biological data.
- By the end of the course, students will be able to integrate and analyze multi-omics data sets, including genomics, transcriptomics, proteomics, and metabolomics data.
- Students will demonstrate the ability to apply advanced bioinformatics techniques to address complex biological questions and solve real-world problems in biomedical research.

Curriculum Details:

Module No.	Unit No.	Topic	Hrs.
1.0	**Unit I**	**Next-Generation Sequencing (NGS) Data Analysis**	11
	1.1	Overview of NGS Technologies: Principles of NGS platforms (e.g., Illumina, Oxford Nanopore), sequencing workflows, and data generation.	
	1.2	NGS Data Preprocessing: Quality control, read trimming, and adapter removal.	
	1.3	Read Alignment and Variant Calling: Alignment algorithms (e.g., BWA, Bowtie), variant calling methods (e.g., GATK, Samtools), and variant annotation.	
	1.4	Transcriptome Analysis: RNA-seq data analysis pipelines, differential gene expression analysis, alternative splicing detection, and functional enrichment analysis.	
2.0	**Unit II**	**Structural Bioinformatics and Molecular Modeling**	12
	2.1	Protein Structure Prediction: Introduction to protein structure prediction methods (homology modeling, ab initio modeling), structure validation, and model refinement.	
	2.2	Molecular Docking: Principles of molecular docking, docking algorithms (e.g., Autodock, GOLD), scoring functions, and applications in drug discovery.	
	2.3	Molecular Dynamics Simulations: Basics of molecular dynamics simulations, force fields, simulation software (e.g., GROMACS, AMBER), and analysis of simulation trajectories.	
	2.4	Protein-Ligand Interaction Analysis: Binding site prediction, protein-ligand docking, and analysis of protein-ligand interactions using computational tools.	
3.0	**Unit III**	**Systems Biology and Network Analysis**	11
	3.1	Introduction to Systems Biology: Principles of systems biology	
	3.2	Biological networks (metabolic networks, signaling networks, gene regulatory networks), and data integration.	
	3.3	Network Construction and Visualization: Construction of biological networks from omics data, network visualization tools (e.g., Cytoscape), and network analysis techniques.	
	3.4	Network-Based Analysis of Biological Data: Network-based approaches for pathway analysis, functional module identification, and disease gene prioritization.	
4.0	**Unit IV**	**High-Throughput Data Analysis and Omics Integration**	11
	4.1	Multi-Omics Data Integration: Integration of genomics, transcriptomics, proteomics, and metabolomics data, data preprocessing, and normalization.	
	4.2	Machine Learning in Bioinformatics: Introduction to machine learning techniques (e.g., classification, clustering, regression) and their applications in bioinformatics.	

	4.3	Bioinformatics Pipelines and Workflows: Design and implementation of bioinformatics pipelines and workflows for high-throughput data analysis using workflow management systems (e.g., Snakemake, Nextflow).	
	4.4	Emerging Trends in Bioinformatics: Overview of emerging trends and technologies in bioinformatics research, such as single-cell sequencing, spatial transcriptomics, and long-read sequencing.	
		Total	**45**

References

1. Buffalo, V. (2015). Bioinformatics data skills: Reproducible and robust research with open source tools. " O'Reilly Media, Inc.".
2. Korpelainen, E., Tuimala, J., Somervuo, P., Huss, M., & Wong, G. (2014). RNA-seq data analysis: a practical approach. CRC press.
3. Rangwala, H., & Karypis, G. (2010). Introduction to protein structure prediction. Introduction to Protein Structure Prediction: Methods and Algorithms, 1-13.
4. Kukol, A. (Ed.). (2008). Molecular modeling of proteins (Vol. 443). Totowa, NJ:: Humana Press.
5. Palsson, B. Ø. (2006). Systems biology: properties of reconstructed networks. Cambridge university press.
6. Rocha, L. M. (2022). On the feasibility of dynamical analysis of network models of biochemical regulation. Bioinformatics, 38(14), 3674-3675.
7. Abu-Jamous, B., Fa, R., & Nandi, A. K. (2015). Integrative cluster analysis in bioinformatics.
8. Larranaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., ... & Robles, V. (2006). Machine learning in bioinformatics. Briefings in bioinformatics, 7(1), 86-112.
9. Zomaya, A. Y., & Pan, Y. (2009). Machine learning in bioinformatics. John Wiley & Sons.
10. Buffalo, V. (2015). Bioinformatics data skills: Reproducible and robust research with open source tools. " O'Reilly Media, Inc.".

SBIOE-502 Lab Course in Advanced Techniques in Bioinformatics

1. **NGS Data Quality Control and Preprocessing**:
   - Practical exercise: Use FastQC to assess the quality of raw sequencing reads, perform read trimming and adapter removal using tools like Trimmomatic or Cutadapt.
   - Learning outcome: Understand the importance of data quality control and preprocessing steps in NGS data analysis.
2. **Read Alignment and Variant Calling**:

- Practical exercise: Use tools such as BWA or HISAT2 for read alignment to a reference genome, followed by variant calling using GATK or Samtools.
- Learning outcome: Gain proficiency in aligning NGS reads to a reference genome and identifying genetic variants.

3. **Protein Structure Prediction**:
    - Practical exercise: Use homology modeling software (e.g., MODELLER) to predict the 3D structure of a protein based on its amino acid sequence and known homologous structures.
    - Learning outcome: Learn how to predict protein structures computationally and evaluate model quality.
4. **Molecular Docking**:
    - Practical exercise: Perform molecular docking simulations to predict the binding mode of a small molecule ligand to a protein target using software like Autodock Vina or GOLD.
    - Learning outcome: Understand the principles of molecular docking and its applications in drug discovery.
5. **Network Construction and Visualization**:
    - Practical exercise: Use Cytoscape to construct and visualize biological networks (e.g., protein-protein interaction networks, gene regulatory networks) from experimental data.
    - Learning outcome: Gain hands-on experience in network construction and visualization.
6. **Network-Based Analysis of Biological Data**:
    - Practical exercise: Perform functional enrichment analysis on a biological network to identify enriched biological pathways or functional modules using tools like EnrichmentMap in Cytoscape.
    - Learning outcome: Learn how to analyze biological networks to extract meaningful biological insights.
7. **Multi-Omics Data Integration:**
    - Practical exercise: Integrate genomics, transcriptomics, and proteomics data sets to identify coordinated changes in gene expression and protein abundance using statistical methods and visualization tools.
    - Learning outcome: Gain proficiency in integrating multi-omics data sets and extracting biological insights from integrated data analysis.
8. **Machine Learning in Bioinformatics**:
    - Practical exercise: Apply machine learning algorithms (e.g., random forest, support vector machines) to predict biological outcomes (e.g., disease classification, protein function prediction) using omics data sets.
    - Learning outcome: Learn how to apply machine learning techniques to solve bioinformatics problems and interpret model predictions.

SBIOE-503 Programming in Java
Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBIOE-503	Programming in Java	03	--	03	--	03

Assessment Scheme

Course Code	Course Name	Theory				Practical		Total
		CA			ESA			
		Test I	Test II	Avg of (T1+T2)/2		CA	ESA	
SBIOE-503	Programming in Java	15	15	15	60	--	--	75

Course pre-requisite:

- **Basic Understanding of Programming Concepts**: Students should have a basic understanding of fundamental programming concepts such as variables, data types, control structures (e.g., loops, conditional statements), functions, and basic problem-solving skills.
- **Familiarity with Object-Oriented Programming (OOP)**: Since Java is an object-oriented programming (OOP) language, students should have some familiarity with OOP principles such as classes, objects, inheritance, encapsulation, and polymorphism.
- Students should have a basic understanding of computer science concepts such as algorithms, data structures (e.g., arrays, lists, queues, stacks), and memory management.

Course objectives:

- To provide students with a solid understanding of the fundamental concepts of the Java programming language.
- Java is an object-oriented programming language, and one of the key objectives of the course is to teach students the principles of object-oriented programming (OOP) and how to apply them in Java.
- To help students develop practical programming skills by working on hands-on coding exercises

Course outcomes:

- Students will demonstrate proficiency in Java syntax and programming constructs, including variables, data types, control flow structures (such as loops and conditionals), methods, classes, objects, inheritance, polymorphism, and exception handling.
- Students will develop a solid understanding of object-oriented programming (OOP) concepts and principles, including encapsulation, inheritance, polymorphism, and abstraction.
- Students will be able to apply Java programming skills to develop practical software applications.

Curriculum Details:

Module No.	Unit No.	Topic	Hrs.
1.0	**Unit I**	**Introduction to Java Basics**	11
	1.1	Introduction to Java: History, features, and applications.	
	1.2	Setting up Java Development Environment: Installation of JDK, setting up IDEs (such as IntelliJ IDEA, Eclipse), and configuring development environment.	
	1.3	Java Syntax and Structure: Basics of Java syntax, data types, variables, operators, expressions, and statements.	
	1.4	Control Flow: Conditional statements (if, else-if, switch), looping constructs (for, while, do-while), and branching statements (break, continue, return).	
2.0	**Unit II**	**Object-Oriented Programming (OOP) in Java**	12
	2.1	Introduction to Object-Oriented Programming (OOP) Concepts: Classes, objects, methods, encapsulation, inheritance, and polymorphism.	
	2.2	Class and Object Creation: Defining classes, creating objects, constructors, and instance variables.	
	2.3	Inheritance and Polymorphism: Extending classes, method overriding, dynamic method dispatch, and abstract classes.	
	2.4	Encapsulation and Access Modifiers: Access modifiers (public, private, protected, default), getters and setters, and data hiding.	
3.0	**Unit III**	**Java Standard Library and Data Structures**	11
	3.1	Java Standard Library: Overview of Java API documentation, packages, and commonly used classes (e.g., String, Math, Scanner).	
	3.2	Collections Framework: Introduction to Java collections (lists, sets, maps), interfaces, and implementations (ArrayList, LinkedList, HashSet, HashMap).	
	3.3	Exception Handling: Handling exceptions using try-catch blocks, throwing exceptions, and creating custom exceptions.	
	3.4	File Handling: Reading from and writing to files, file I/O operations using FileReader, FileWriter, BufferedReader, and BufferedWriter.	
4.0	**Unit IV**	**Advanced Java Topics**	11
	4.1	Multithreading and Concurrency: Introduction to threads, creating and managing threads, synchronization, and inter-thread communication.	
	4.2	Networking in Java: Basics of networking, sockets programming (TCP/IP and UDP), client-server communication, and network protocols (HTTP, FTP).	
	4.3	GUI Programming with JavaFX: Introduction to JavaFX framework, creating graphical user interfaces (GUIs), event handling, and layout management.	
	4.4	Introduction to Java Database Connectivity (JDBC): Basics of database programming in Java, connecting to databases, executing SQL queries, and handling result sets.	

		Total	**45**

References

1. Schildt, H. (2006). Java: a beginner's guide. McGraw-Hill, Inc..
2. Sierra, K., & Bates, B. (2005). Head First Java: A Brain-Friendly Guide. " O'Reilly Media, Inc.".
3. Poo, D. C., Kiong, D., & Ashok, S. (2008). Object-oriented programming and Java. Springer Science & Business Media.
4. Schildt, H. (2014). Java: the complete reference. McGraw-Hill Education Group.
5. Goetz, B. (2006). Java concurrency in practice. Pearson Education.
6. Harold, E. R. (2004). Java network programming. " O'Reilly Media, Inc.".
7. Bai, Y. (2011). Practical database programming with Java. John Wiley & Sons.

SBIOE-504 Lab Course in Programming in Java

1. **Hello World Program**:
   - Practical exercise: Write a Java program that prints "Hello, World!" to the console.
   - Learning outcome: Familiarize students with the basic structure of a Java program and the syntax for writing and executing Java code.
2. **Basic Calculator Program**:
   - Practical exercise: Write a Java program that takes two numbers as input from the user and performs basic arithmetic operations (addition, subtraction, multiplication, division).
   - Learning outcome: Reinforce understanding of variables, data types, user input/output, and basic arithmetic operations in Java.
3. **Class and Object Creation:**
   - Practical exercise: Define a class representing a "Person" with attributes such as name, age, and email. Create objects of the "Person" class and demonstrate how to initialize object attributes and access them.
   - Learning outcome: Understand the concepts of classes, objects, and instance variables in Java.
4. **Inheritance and Polymorphism:**
   - Practical exercise: Define a super class "Shape" with methods for calculating area and perimeter. Create subclasses such as "Circle" and "Rectangle" that inherit from the "Shape" class and override the area and perimeter methods. Demonstrate polymorphism by invoking these methods on objects of different subclasses.
   - Learning outcome: Gain hands-on experience with inheritance, method overriding, and polymorphism in Java.
5. **Using Java Collections Framework:**

- Practical exercise: Create a program that demonstrates the usage of Java collections framework classes such as ArrayList, LinkedList, HashSet, and HashMap. Perform operations such as adding, removing, and iterating over elements in these collections.
- Learning outcome: Understand the usage of Java collections and their common operations.

6. **Exception Handling:**
    - Practical exercise: Write a program that reads input from the user and converts it to an integer. Handle possible exceptions (e.g., NumberFormatException) that may occur when converting the input string to an integer.
    - Learning outcome: Learn how to use try-catch blocks to handle exceptions and ensure robust error handling in Java programs.

SEMESTER IV

SBIOC-551 Perl Programming
Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBIOC-551	Perl Programming	04	--	04	--	04

Assessment Scheme

Course Code	Course Name	Theory				Practical		Total
		CA			ESA			
		Test I	Test II	Avg of (T1+T2)/2		CA	ESA	
SBIOC 551	Perl Programming	20	20	20	80	--	--	100

Course pre-requisite:

- Students should have a basic understanding of programming concepts and techniques. This includes familiarity with variables, data types, control structures (such as loops and conditional statements), functions or subroutines, and basic file input/output operations.
- Prior knowledge or familiarity with regular expressions will be beneficial for students to effectively use Perl's pattern matching capabilities.
- Students should be comfortable navigating the command line and running Perl scripts from the terminal. Additionally, proficiency with text editors for writing and editing Perl code is essential. Familiarity with editors like Vim, Emacs, or Sublime Text can be advantageous.

Course objectives:

- To develop proficiency in understanding and using Perl programming language syntax, including variables, data types, operators, control structures (loops, conditionals), subroutines, and built-in functions.
- To gain a thorough understanding of regular expressions and their usage for pattern matching and text processing in Perl.
- To learn how to use Perl for system administration tasks, automation, and scripting purposes.

Course outcomes:

- Students will be able to write Perl scripts to perform basic tasks such as text processing, file manipulation, and data manipulation using Perl language constructs effectively. This will also enhance students problem-solving abilities and approach to solve real-world problems using programming.
- Students will be proficient in using regular expressions to search, extract, and manipulate text data, enabling them to solve complex text processing tasks efficiently using Perl.
- Students will be able to write Perl scripts to automate system administration tasks, such as file and directory management

Curriculum Details:

Module No.	Unit No.	Topic	Hrs.
1.0	1	**Introduction to Perl Basics**	15
	1.1	Overview of Perl: History, features, and applications. Getting Started with Perl: Installation, setting up the development environment, and running Perl scripts.	
	1.2	Perl Syntax and Data Types: Scalars, arrays, hashes, variables, and basic operators.	
	1.3	Control Flow: Conditional statements (if, elsif, else), loops (for, foreach, while, until), and control flow constructs (next, last, redo).	
	1.4	Input and Output: Reading from and writing to files, standard input/output, filehandles, and basic file manipulation.	
2.0	2	**Perl Functions and Subroutines**	15
	2.1	Introduction to Functions: Defining and calling subroutines, passing arguments, and returning values.	
	2.2	Built-in Functions: Commonly used Perl functions for string manipulation (e.g., length, substr, split), array manipulation (e.g., push, pop, shift), and hash manipulation (e.g., keys, values, each).	
	2.3	File Handling Functions: Working with files and directories using Perl's filehandling functions (e.g., opendir, readdir, closedir, stat).	
	2.4	Regular Expressions: Introduction to regular expressions in Perl, pattern matching (m//), substitution (s///), and metacharacters.	
3.0	3	**Perl Modules and Libraries**	
	3.1	Introduction to Perl Modules: Overview of Perl modules and the CPAN (Comprehensive Perl Archive Network) repository.	
	3.2	Working with Standard Modules: Exploring commonly used Perl modules for various tasks, such as file handling (**File::Path**, **File::Copy**), system interaction (**Cwd**, **File::Temp**), and command-line argument parsing (**Getopt::Long**).	
	3.3	Introduction to Object-Oriented Programming (OOP) in Perl: Basics of object-oriented programming concepts in Perl, defining classes, creating objects, and accessing methods and attributes.	
	3.4	Exploring Additional Perl Modules: Introduction to additional Perl modules for specific tasks, such as database interaction (**DBI**), XML processing (**XML::Parser**, **XML::LibXML**), and web programming (**CGI**, **LWP**).	
4.0	4	**Advanced Perl Topics**	15
	4.1	Error Handling and Debugging: Handling errors and exceptions in Perl, using **die**, **warn**, and **eval**, and debugging techniques.	
	4.2	Advanced File Handling: Working with filehandles, file locking, file testing operators (-f, -d, -r)	
	4.3	Advanced file manipulation.	
	4.4	Interacting with External Programs: Running external commands and programs from within Perl scripts, using system(), exec(), and backticks (`).	
		Total	**60**

References

1. Schwartz, R. L., & Phoenix, T. (2011). Learning perl. " O'Reilly Media, Inc.".

2. Christiansen, T., & Torkington, N. (2003). Perl cookbook: Solutions & examples for Perl programmers. " O'Reilly Media, Inc.".
3. Schwartz, R. L., & Phoenix, T. (2012). Intermediate Perl: Beyond The Basics of Learning Perl. " O'Reilly Media, Inc.".
4. Ford, J. L. (2006). Perl Programming for the Absolute Beginner. Delmar Learning.
5. Wall, L., Christiansen, T., & Schwartz, R. L. (1999). Programming perl.
6. Langworth, I. (2005). Perl testing: a developer's notebook. " O'Reilly Media, Inc.".

SBIOP-551 Lab Course in Perl Programming

1. **Hello World Program:**
    - Write a Perl script that prints "Hello, World!" to the console.
2. **Simple Calculator:**
    - Write a Perl script that takes two numbers as input from the user and performs basic arithmetic operations (addition, subtraction, multiplication, division).
3. **Temperature Converter:**
    - Write a Perl subroutine that converts temperature from Celsius to Fahrenheit and vice versa. Test the subroutine with sample input values.
4. **Text Processing:**
    - Write a Perl script that reads a text file, counts the number of words in the file, and prints the result.
5. **File Manipulation:**
    - Write a Perl script that reads data from a text file, performs some data manipulation (e.g., search, replace), and writes the modified data to another file.
6. **Regular Expression Matching:**
    - Write a Perl script that uses regular expressions to search for patterns (e.g., email addresses, phone numbers) in a text file and prints the matches.
7. **Directory Listing:**
    - Write a Perl script that lists all files and directories in a specified directory.
8. **File Backup Utility:**
    - Write a Perl script that takes a directory path as input and creates a backup of all files in the directory by copying them to a specified backup directory.
9. **String Manipulation:**
    - Write a Perl script that prompts the user to enter a string and performs various string manipulation operations (e.g., concatenation, substring extraction, case conversion).
10. **Basic Database Interaction:**
    - Write a Perl script that connects to a MySQL database using DBI module, executes a simple SQL query (e.g., SELECT * FROM table), and prints the results.

SBIOC-552 Applications of Bioinformatics
Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBIOC-552	Applications of Bioinformatics	04	--	04	--	04

Assessment Scheme

Course Code	Course Name	Theory				Practical		Total
		CA			ESA			
		Test I	Test II	Avg of (T1+T2)/2		CA	ESA	
SBIOC-552	Applications of Bioinformatics	20	20	20	80	--	--	100

Course pre-requisite:

- Students should have a foundational understanding of biological concepts, including genetics, molecular biology, and biochemistry.
- students should have proficiency in programming languages commonly used in bioinformatics, such as Python, R, or Perl.
- Students should have a basic understanding of statistics, probability theory, and mathematical concepts relevant to bioinformatics, such as linear algebra and calculus.

Course objectives:

- To introduce students to a variety of bioinformatics tools, databases, and resources commonly used in biological research.
- To demonstrate the practical application of bioinformatics techniques in addressing biological questions and solving real-world problems.
- To foster understanding of the complementary relationship between bioinformatics and experimental biology, and the importance of interdisciplinary collaboration.

Course outcomes:

- Students will gain a comprehensive understanding of commonly used bioinformatics tools, databases, and resources.
- Students will develop proficiency in performing bioinformatics analyses and interpreting results.
- Students will understand the integration of bioinformatics approaches with biological research and its applications in various domains.

Curriculum Details:

Module No.	Unit No.	Topic	Hrs.
1.0	1	**Foundations of Bioinformatics**	15
	1.1	Introduction to Bioinformatics: Definition, scope, and applications in biological research.	
	1.2	Biological Databases and Resources: Overview of major biological databases (e.g., NCBI, UniProt, Ensembl) and bioinformatics resources for sequence, structure, and functional data.	
	1.3	Sequence Analysis: Basics of sequence alignment, sequence databases, sequence similarity searching (BLAST), and multiple sequence alignment.	
	1.4	Genome Annotation: Principles of genome annotation, gene prediction algorithms, functional annotation, and genome browsers.	
2.0	2	**Structural Bioinformatics and Molecular Modeling**	15
	2.1	Protein Structure Prediction: Introduction to protein structure prediction methods, homology modeling, threading, and ab initio modeling.	
	2.2	Protein-Ligand Interactions: Basics of molecular docking, protein-ligand binding modes, and structure-based drug design.	
	2.3	Protein Structure Analysis: Techniques for analyzing protein structures, including Ramachandran plots, solvent accessibility, and secondary structure prediction.	
	2.4	Molecular Dynamics Simulation: Introduction to molecular dynamics simulations, force fields, and simulation software (e.g., GROMACS, AMBER).	
3.0	3	**Functional Genomics and Transcriptomics**	15
	3.1	Gene Expression Analysis: Overview of microarray technology, RNA-seq, and other high-throughput gene expression profiling techniques.	
	3.2	Differential Gene Expression Analysis: Methods for differential gene expression analysis, normalization, statistical analysis, and visualization of gene expression data.	
	3.3	Functional Enrichment Analysis: Introduction to functional enrichment analysis methods (e.g., Gene Ontology enrichment analysis), pathway analysis, and network-based analysis of gene expression data.	
	3.4	Transcription Factor Binding Site Analysis: Prediction of transcription factor binding sites, motif discovery, and regulatory network analysis.	
4.0	4	**Applications of Bioinformatics in Biomedical Research**	15
	4.1	Pharmacogenomics and Personalized Medicine: Role of bioinformatics in pharmacogenomics, drug response prediction, and personalized medicine.	
	4.2	Disease Genomics: Genomic approaches to studying human diseases, genome-wide association studies (GWAS), and identification of disease-associated genetic variants.	
	4.3	Cancer Genomics: Bioinformatics analysis of cancer genomics data, identification of driver mutations, tumor heterogeneity, and therapeutic targets.	
	4.4	Metagenomics and Microbiome Analysis: Introduction to metagenomics, analysis of microbial communities, and bioinformatics tools for microbiome analysis.	

		Total	**60**

Reference Books

1. Claverie, J. M., & Notredame, C. (2011). Bioinformatics for dummies. John Wiley & Sons.
2. Xiong J. Essential bioinformatics. Cambridge University Press; 2006.
3. Durbin, R., Eddy, S. R., Krogh, A., & Mitchison, G. (1998). Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge university press.
4. Zhumur Ghosh Bioinformatics: Principles and Applications
5. Zvelebil, M., & Baum, J. O. (2007). Understanding bioinformatics. Garland Science.
6. Rangwala, H., & Karypis, G. (2010). Introduction to protein structure prediction. Introduction to Protein Structure Prediction: Methods and Algorithms, 1-13.
7. Lesk, A. M. (2017). Introduction to genomics. Oxford University Press.
8. Lam, Y. W. F., & Scott, S. R. (Eds.). (2018). Pharmacogenomics: challenges and opportunities in therapeutic implementation.

SBIOP-552 Lab Course in Applications of Bioinformatics

1. **Database Search and Retrieval:**
    - Practical exercise: Use NCBI's BLAST tool to search for homologous sequences in a given database, analyze the results, and interpret sequence alignments.
2. **Genome Annotation:**
    - Practical exercise: Use a genome annotation tool (e.g., Prokka, MAKER) to annotate genes in a bacterial or eukaryotic genome sequence, and analyze the annotation output.
3. **Protein Structure Prediction:**
    - Practical exercise: Use the SWISS-MODEL or Phyre2 web server to predict the 3D structure of a protein sequence, visualize the predicted structure, and analyze structural features.
4. **Molecular Docking:**
    - Practical exercise: Perform molecular docking of a small molecule ligand to a protein target using software such as AutoDock or PyRx, analyze binding interactions, and predict binding affinity.
5. **RNA-seq Data Analysis:**
    - Practical exercise: Analyze RNA-seq data using bioinformatics tools such as HISAT2, StringTie, and DESeq2 to identify differentially expressed genes, visualize expression profiles, and interpret results.
6. **Functional Enrichment Analysis:**
    - Practical exercise: Perform Gene Ontology (GO) enrichment analysis of a set of differentially expressed genes using tools such as DAVID or Enrichr, and interpret biological functions associated with enriched terms.
7. **Pharmacogenomics Analysis:**
    - Practical exercise: Analyze pharmacogenomic data from public databases (e.g., PharmGKB) to identify genetic variants associated with drug response phenotypes, and interpret clinical implications.
8. **Cancer Genomics Analysis:**
    - Practical exercise: Analyze cancer genomic data (e.g., TCGA) using bioinformatics tools such as cBioPortal or UCSC Xena, identify somatic mutations, copy number alterations, and gene expression changes associated with cancer subtypes.

SBIOE-551 Database Management System
Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBIOE-551	Database Management System	03	--	03	--	03

Assessment Scheme

Course Code	Course Name	Theory				Practical		Total
		CA			ESA			
		Test I	Test II	Avg of (T1+T2)/2		CA	ESA	
SBIOE-551	Database Management System	15	15	15	60	--	--	75

Prerequisite:

- Students should have a foundational understanding of basic computer science concepts, including programming fundamentals, data structures, algorithms, and computer architecture.
- Familiarity with relational algebra and set theory is beneficial for understanding the theoretical foundations of database management systems.
- Students should have a basic understanding of SQL, the standard language for interacting with relational databases.

Course objectives:

- To provide students with a comprehensive understanding of fundamental concepts and principles underlying database management systems.
- To develop students' ability to design, implement, and manage relational databases effectively.
- To develop students' ability to write complex SQL queries and perform data manipulation tasks efficiently.

Course outcomes: This course will enable the students to get

- Students will demonstrate proficiency in designing, implementing, and managing relational databases.
- Students will develop effective querying and data manipulation skills using Structured Query Language (SQL).
- Students will gain a deep understanding of fundamental concepts and principles underlying database management systems.

Curriculum Details:

Module No.	Unit No.	Topic	Hrs.
1.0	1	**Introduction to Databases and Relational Model**	11
	1.1	Introduction to Databases: Definition, role, and importance of databases in computing.	
	1.2	Data Models: Overview of hierarchical, network, and relational data models.	
	1.3	Relational Model: Concepts of relations, tuples, attributes, keys, domains, and integrity constraints.	
	1.4	Relational Algebra and Calculus: Basic operations (selection, projection, join) and formal query languages.	
2.0	2	**Database Design and Normalization**	11
	2.1	Database Design Process: Requirements analysis, conceptual design, logical design, and physical design.	
	2.2	Entity-Relationship Model (ER Model): Entities, relationships, attributes, keys, and ER diagrams.	
	2.3	Functional Dependencies and Normalization: First, Second, and Third Normal Forms (1NF, 2NF, 3NF), Boyce-Codd Normal Form (BCNF), and normalization algorithms.	
	2.4	Denormalization: Reasons for denormalization, trade-offs, and best practices.	
3.0	3	**SQL and Relational Database Management Systems (RDBMS)**	11
	3.1	Structured Query Language (SQL): Basic SQL commands (SELECT, INSERT, UPDATE, DELETE), data definition language (DDL) commands (CREATE, ALTER, DROP), data manipulation language (DML) commands (INSERT, UPDATE, DELETE), and data control language (DCL) commands (GRANT, REVOKE).	
	3.2	Advanced SQL: Joins (INNER, OUTER, CROSS), set operations (UNION, INTERSECT, EXCEPT), subqueries, views, stored procedures, triggers, and transactions.	
	3.3	Relational Database Management Systems (RDBMS): Overview of popular RDBMS platforms such as MySQL, PostgreSQL, Oracle, SQL Server, and SQLite.	
	3.4	Database Administration: User management, security, backup and recovery, performance tuning, and monitoring.	
4.0		**Advanced Database Concepts and Emerging Technologies**	12
	4.1	Advanced Data Models: Overview of object-oriented, XML, and NoSQL data models.	
	4.2	Distributed Databases: Concepts, architectures, distributed transactions, and replication.	
	4.3	Big Data and NoSQL Databases: Introduction to NoSQL databases (document, key-value, column-family, graph databases), CAP theorem, and BASE properties.	
	4.4	Cloud Databases: Overview of cloud database services (e.g., Amazon RDS, Google Cloud SQL, Azure SQL Database), scalability, elasticity, and data privacy considerations.	
		Total	45

Reference Books

1. Garcia-Molina, H. (2008). Database systems: the complete book. Pearson Education India.
2. Ramakrishnan, R., & Gehrke, J. (2002). Database management systems. McGraw-Hill, Inc..
3. Hernandez, M. J. (2013). Database design for mere mortals: a hands-on guide to relational database design. Pearson Education.
4. Foster, E., & Godbole, S. (2022). Database systems: a pragmatic approach. Auerbach Publications.
5. Shields, W. (2019). SQL QuickStart Guide: The Simplified Beginner's Guide to Managing, Analyzing, and Manipulating Data With SQL. ClydeBank Media LLC.
6. Beaulieu, A. (2020). Learning SQL: Generate, manipulate, and retrieve data. O'Reilly Media.
7. Warren, J., & Marz, N. (2015). Big Data: Principles and best practices of scalable realtime data systems. Simon and Schuster.
8. Sadalage, P. J., & Fowler, M. (2013). NoSQL distilled: a brief guide to the emerging world of polyglot persistence. Pearson Education.

SBIOE-552 Lab Course in Database Management System

1. **Database Installation and Setup:**
    - Practical exercise: Install a relational database management system (e.g., MySQL, PostgreSQL) on your computer or use an online database service (e.g., AWS RDS, Google Cloud SQL).
    - Learning outcome: Familiarize students with the process of setting up a database environment for practice and experimentation.
2. **Entity-Relationship Modeling:**
    - Practical exercise: Use a diagramming tool (e.g., Lucidchart, Draw.io) to create entity-relationship diagrams (ERDs) for a given set of requirements.
    - Learning outcome: Develop students' skills in representing entities, attributes, relationships, and cardinality constraints in ER diagrams.
3. **Normalization Practice:**
    - Practical exercise: Given a set of unnormalized tables, normalize them to Third Normal Form (3NF) or Boyce-Codd Normal Form (BCNF) using normalization techniques.
    - Learning outcome: Reinforce students' understanding of normalization principles and techniques through hands-on practice.
4. **Schema Design and Creation:**
    - Practical exercise: Design a database schema based on a provided set of requirements and create the corresponding tables using Data Definition Language (DDL) commands in SQL.
    - Learning outcome: Develop students' skills in translating conceptual data models (ER diagrams) into physical database schemas.

5. **Basic SQL Queries:**

    - Practical exercise: Write SQL queries to perform basic CRUD operations (Create, Read, Update, Delete) on a set of tables (e.g., employees, customers).
    - Learning outcome: Enable students to practice SQL syntax and basic query writing skills.

6. **Advanced SQL Queries:**

    - Practical exercise: Write SQL queries involving joins, subqueries, aggregation functions, and conditional logic to answer complex business questions.
    - Learning outcome: Enhance students' proficiency in writing advanced SQL queries for data retrieval and analysis.

7. **NoSQL Database Setup and Usage:**

    - Practical exercise: Install a NoSQL database system (e.g., MongoDB, Cassandra) and perform basic data manipulation operations (e.g., insert, query) using the NoSQL query language.
    - Learning outcome: Introduce students to the concepts and usage of NoSQL databases and compare them with traditional relational databases.

8. **Cloud Database Deployment:**

    - Practical exercise: Deploy a database instance on a cloud platform (e.g., Amazon RDS, Google Cloud SQL) and connect to it using a SQL client.
    - Learning outcome: Familiarize students with the process of provisioning and managing databases in a cloud environment and understand cloud-specific database features and limitations.

SBIOE-553 Biological Sequence Analysis

Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBIOE-553	Biological Sequence Analysis	03	--	03	--	03

Assessment Scheme

Course Code	Course Name	Theory				Practical		Total
		CA			ESA			
		Test I	Test II	Avg of (T1+T2)/2		CA	ESA	
SBIOE-553	Biological Sequence Analysis	15	15	15	60	--	--	75

Course pre-requisite:

- Students should have a foundational understanding of basic biological concepts, including genetics, molecular biology, and biochemistry.
- Basic proficiency in programming and data analysis.
- Familiarity with basic bioinformatics concepts and techniques.

Course objectives:

- To provide students with a comprehensive understanding of the principles and techniques used in biological sequence analysis.
- To develop students' proficiency in using bioinformatics tools and software for sequence analysis.
- To enable students to apply sequence analysis techniques to address biological research questions and problems.

Course outcomes:

- Students will demonstrate proficiency in performing pairwise and multiple sequence alignments and comparing biological sequences.
- Students will be able to analyze genetic variation and infer evolutionary relationships using sequence analysis methods.
- Students will understand the application of sequence analysis techniques in biomedical research and its impact on understanding disease mechanisms and personalized medicine.

Curriculum Details:

Module No.	Unit No.	Topic	Hrs.
1.0	1	**Introduction to Sequence Analysis and Alignment**	12
	1.1	Introduction to Biological Sequence Analysis: Definition, importance, and applications in molecular biology and bioinformatics.	
	1.2	Sequence Data Types: Overview of DNA, RNA, and protein sequences, sequence formats (e.g., FASTA, GenBank), and sequence databases.	
	1.3	Pairwise Sequence Alignment: Introduction to sequence alignment algorithms (e.g., Needleman-Wunsch, Smith-Waterman), scoring matrices (e.g., BLOSUM, PAM), and gap penalties.	
	1.4	Multiple Sequence Alignment: Concepts of multiple sequence alignment, progressive and iterative alignment methods, and software tools (e.g., Clustal Omega, MUSCLE).	
2.0	2	**Sequence Database Searching and Similarity Analysis**	11
	2.1	Sequence Database Searching: Principles of sequence database searching	
	2.2	BLAST algorithm, BLAST parameters (e.g., E-value, bit score), and interpretation of BLAST results.	
	2.3	Sequence Similarity Analysis: Analysis of sequence similarity scores, statistical significance, sequence identity	
	2.4	Similarity searches against specialized databases (e.g., Pfam, UniProt)	
3.0	3	**Phylogenetic Analysis and Evolutionary Relationships**	11
	3.1	Introduction to Phylogenetics: Concepts of evolutionary trees, phylogenetic inference methods	
	3.2	Tree-building algorithms (e.g., neighbor-joining, maximum likelihood).	
	3.3	Molecular Evolution: Principles of molecular evolution, molecular clocks, neutral theory, and models of sequence evolution (e.g., Jukes-Cantor, Kimura).	
	3.4	Phylogenetic Tree Visualization and Interpretation: Visualization of phylogenetic trees using tree-drawing software (e.g., FigTree), interpretation of tree topologies, and ancestral sequence reconstruction.	
4.0	4	**Advanced Topics in Sequence Analysis**	11
	4.1	Sequence Motif Discovery: Introduction to sequence motifs, motif detection algorithms (e.g., MEME, Gibbs sampling), and motif analysis tools (e.g., WebLogo).	
	4.2	Functional Annotation of Sequences: Methods for functional annotation of DNA and protein sequences, including domain identification, functional site prediction, and GO term enrichment analysis.	
	4.3	Next-Generation Sequencing (NGS) Data Analysis: Overview of NGS technologies (e.g., RNA-seq, ChIP-seq), data preprocessing, alignment, variant calling	

	4.4	Analysis of NGS data using bioinformatics pipelines (e.g., BWA, SAMtools, GATK).	
		Total	**45**

Reference Books

1. Durbin, R., Eddy, S. R., Krogh, A., & Mitchison, G. (1998). Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge university press.
2. Craig, P. (2009). Introduction to bioinformatics, by Arthur M. Lesk.
3. Compeau, P., & Pevzner, P. (2015). Bioinformatics algorithms: an active learning approach. (No Title).
4. Korf, I. (2003). BLAST/Ian Korf, Mark Yandell and Joseph Bedell;[foreword by Stephen Atschul]. An essential guide to the Basic Local Alignment Search Tool.
5. Bioinformatics Data Skills: Reproducible and Robust Research with Open Source Tools" by Vince Buffalo
6. Phylogenetic Trees Made Easy: A How-To Manual" by Barry G. Hall
7. Nei, M., & Kumar, S. (2000). Molecular evolution and phylogenetics. Oxford university press.
8. Worley, K. (2003). Sequence Analysis in a Nutshell: A guide to common tools and databases. The American Journal of Human Genetics, 73(5), 1213.
9. Next-Generation Sequencing Data Analysis" by Xinkun Wang and Marco A. Marra
10. Campbell, A. M., & Heyer, L. J. (2003). Discovering genomics, proteomics, and bioinformatics. San Francisco, CA: Benjamin Cummings.

SBIOE-554 Lab Course in Biological Sequence Analysis

1. **Pairwise Sequence Alignment:**
    - Practical exercise: Implement the Needleman-Wunsch algorithm for global sequence alignment and the Smith-Waterman algorithm for local sequence alignment in a programming language of choice (e.g., Python).
    - Learning outcome: Gain hands-on experience with dynamic programming algorithms for pairwise sequence alignment and understand the significance of alignment scores and gap penalties.
2. **Multiple Sequence Alignment:**
    - Practical exercise: Use a multiple sequence alignment tool (e.g., Clustal Omega) to align a set of protein sequences or DNA sequences related to a specific gene family.
    - Learning outcome: Understand the principles of multiple sequence alignment and gain experience in aligning homologous sequences to identify conserved regions and evolutionary relationships.

3. **BLAST Search:**
    - Practical exercise: Perform BLAST searches against a sequence database (e.g., NCBI nr database) using known protein or nucleotide sequences as queries and analyze the BLAST results.

- Learning outcome: Learn to use the BLAST algorithm for sequence database searching, interpret BLAST output, and identify homologous sequences with significant similarity.

4. **Sequence Similarity Analysis:**
   - Practical exercise: Calculate sequence identity and similarity scores between pairs of aligned sequences, and analyze the effect of different scoring matrices (e.g., BLOSUM, PAM) and gap penalties on alignment quality.
   - Learning outcome: Understand how sequence similarity scores are calculated, evaluate the significance of sequence similarity, and interpret sequence alignment results.
5. **Phylogenetic Tree Reconstruction:**
   - Practical exercise: Use phylogenetic tree reconstruction software (e.g., MEGA, PhyML) to build a phylogenetic tree from aligned DNA or protein sequences and visualize the tree.
   - Learning outcome: Gain experience in reconstructing phylogenetic trees using distance-based or maximum likelihood methods, interpret tree topologies, and understand the evolutionary relationships between sequences.
6. **Ancestral Sequence Reconstruction:**
   - Practical exercise: Use ancestral sequence reconstruction methods to infer ancestral sequences at internal nodes of a phylogenetic tree and analyze the evolutionary changes along specific lineages.
   - Learning outcome: Understand the principles of ancestral sequence reconstruction, apply ancestral sequence inference algorithms, and interpret evolutionary changes in ancestral sequences.
7. **Motif Discovery:**
   - Practical exercise: Use motif discovery software (e.g., MEME Suite) to identify conserved sequence motifs in a set of aligned DNA or protein sequences and analyze the discovered motifs.
   - Learning outcome: Gain hands-on experience with motif discovery algorithms, interpret sequence motifs, and understand their biological significance.
8. **Next-Generation Sequencing (NGS) Data Analysis:**
   - Practical exercise: Analyze next-generation sequencing data (e.g., RNA-seq, ChIP-seq) using bioinformatics pipelines (e.g., HISAT2, STAR, DESeq2) to preprocess data, align reads, and identify genomic features.

- Learning outcome: Learn to process and analyze NGS data, understand the workflow of NGS data analysis pipelines, and interpret genomic features identified from NGS experiments.

SVECP-551: Publication Ethics
Teaching Scheme

Course Code	Course Name	Teaching Scheme (Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SVECP-551	Publication Ethics	02	--	02	--	02

Assessment Scheme

Course Code	Course Name	Theory				Practical		Total
		CA			ESA			
		Test I	Test II	Avg of (T1+T2)/2		CA	ESA	
SVECP-551	Publication Ethics	10	10	10	40	--	--	50

Course pre-requisite:

- General awareness regarding publication basics

Course objectives:

- To know rules, issues, options, and resources for research ethics.
- To familiarize with various institutional ethics review boards/academic integrity guidelines.
- To understand the purpose and value of ethical decision-making.
- To have a positive disposition towards continued learning about research ethics

Course outcomes:

- To have a positive disposition towards continued learning about research philosophy & ethics.
- To know Rules, Regulations, Issues, Options, and Scientific Resources of Research Ethics.
- To learn the culture of fairness, honesty and integrity in academic communications and to understand the purpose and value of ethical decision-making.
- Avoid wasteful and duplicate publications & encourage original contributions to advance Academic Research and Scholarship.
- Acquiring knowledge & professional competence and expertise about Patents, Copyrights, and other forms of Intellectual Property Rights.
- To promote social good and prevent or mitigate societal hazards through innovative ideas, creativity and research advocacy

Curriculum Details:

odule No.	Unit No.	Topic	Hrs.
1.0		**Publication ethics**	**08**
	1.1	Publication ethics: definition, introduction and importance, Best practices/standards setting initiatives and guidelines: COPE, WAME, etc. Conflicts of interest	
	1.2	Publication misconduct: definition, concept, problems that lead to unethical behavior and vice verse, types	
	1.3	Violation of publication ethics, authorship and contributor ship	
	1.4	Identification of publication misconduct, complaints and appeals. Predatory publishers and journals	
2.0		**Open access publishing**	**07**
	2.1	Open access publications and initiatives.	
	2.2	SHERPA/RoMEO online resource to check publisher copyright and self-archiving policies	
	2.3	Software tool to identify predatory publications developed by SPPU	
	2.4	Journal finder/ journal suggestion tools viz. JANE	
3.0		**Publication misconduct**	**07**
	3.1	Subject specific ethical issues, FFP, authorship	
	3.2	Conflicts of interest	
	3.3	Complaints and appeals: examples and fraud from India and abroad	
	3.4	Use of plagiarism software like Turnitin, Urkund and other open source software tools.	
4.0		**Databases and research metrics**	**08**
	4.1	Databases: Indexing databases	
	4.2	Citation databases: Web of Science, Scopus, etc.	
	4.3	Research Metrics: Impact Factor of journal as per journal citation report, SNIP, SJR, IPP, Cite Score.	
	4.4	Metrics: h-index, g index, i10 index, altmetrics	
		Total	**30**

References:

1. Donna M. Mertens, Pauline E. Ginsberg The Handbook of Social Research Ethics, SAGE (2009).
2. Rose Wiles, Bloomsbury What are Qualitative Research Ethics? (2013).
3. Robin Levin Penslar, eds, Research Ethics: Cases and Materials, Indiana University Press (1995).
4. Gary Comstock, Research Ethics: A Philosophical Guide to the Responsible Conduct of Research, Cambridge University Press (2013)
5. Bird, A. Philosophy of Science. Routledge, 2006.
6. MacIntyre, Alasdair A Short History of Ethics London, 1967
7. P. Chaddah Ethics in Competitive Research: Do not get scooped; do not get plagiarized, 2018
8. National Academy of Sciences, National Academy of Engineering and Institute of Medicine. (2009).

9. On being a Scientist: A Guide to Responsible Conduct in Research. Third Edition. National Academies Press.
10. Resnik, D. B. What is ethics in research & why is it important. National Institute of Environmental Health Sciences, 2018.
    Retrieved from https www.nichs.nih.gov/research/resources/bioethics/whatis/index.cfm
11. Beall, J. Predatory publishers are corrupting open access. Nature, 2012.
    https://doi.org/10.1038/489179a
12. Indian National Science Academy (INSA), Ethics in Science Education, Research and Governance(2019), SBN:978-81-939482-1-7.
    http://www.insaindia.res.in/pdf/Ethics Book.pdf

******************************02.06.2024***************************************