MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

| **Module Information**  **معلومات المادة الدراسية** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Module Title** | **Bioinformatics Programming** | | | | **Module Delivery** | | |
| **Module Type** | Core | | | | * **☒ Theory** * **☐ Lecture** * **☒ Lab** * **☐ Tutorial** * **☐ Practical** * **☐ Seminar** | | |
| **Module Code** | ITC320130 | | | |
| **ECTS Credits** | 5.00 | | | |
| **SWL (hr/sem)** | 125 | | | |
| **Module Level** | | 2 | **Semester of Delivery** | | | | 4 |
| **Administering Department** | | BID | **College** | BMIC | | | |
| **Module Leader** | Dr. Zainab N. Al-Qudsy | | **e-mail** | [dr.zainab.n.yousif@uoitc.edu.iq](mailto:dr.zainab.n.yousif@uoitc.edu.iq) | | | |
| **Module Leader’s Acad. Title** | | Lecturer | **Module Leader’s Qualification** | | | | Ph.D. |
| **Module Tutor** |  | | **e-mail** |  | | | |
| **Peer Reviewer Name** | | jwan k alwan | **e-mail** | jwanism@uoitc.edu.iq | | | |
| **Scientific Committee Approval Date** | | 18/6/2023 | **Version Number** | | | 1.0 | |

| **Relation with other Modules**  **العلاقة مع المواد الدراسية الأخرى** | | | |
| --- | --- | --- | --- |
| **Prerequisite module** | Computer Programming II / BMI121 | **Semester** | 2 |
| **Co-requisites module** | None | **Semester** |  |

| **Module Aims, Learning Outcomes and Indicative Contents**  **أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية** | |
| --- | --- |
| **Module Aims**  **أهداف المادة الدراسية** | This module introduces the study of current areas in which computer science approaches have been applied in the field of genomics and the construction of algorithms based on computational and statistical analysis of DNA sequence data, including sequence alignment and denovo assembly. The main objectives of this module are:   1. Study the basic concepts of algorithms used in analyzing DNA sequence data. 2. Learn about past and present DNA sequencing technology and how it works. 3. Recognize genomes as sequences, and read sub-sequences of the genome, How to read a genome and analysis it using Python, Understand how DNA is copied. 4. Working with sequencing reads and matching them with the original genome that exists and lacks a source by means of alignments and overlaps. 5. Understanding naive matching, synthetic reads matching, real reads matching. |
| **Module Learning Outcomes**  **مخرجات التعلم للمادة الدراسية** | 1. Exploration of algorithms for analyzing DNA sequencing data. 2. Discuss DNA sequencing technology, it’s past and present, and how it works. 3. Understanding the main concepts and advantages of Genetic sequence analysis and where it is used all over the field of life science. 4. Recognizing how genetic sequencing are developed from past to present. 5. Study how to download FASTA files and parsing a genome. 6. Understanding the main concepts of second-generation sequencers. 7. Study how to obtain Sequencing reads from FASTQ file format and Analyzing reads by position. 8. Explanation of the most important methods of analyzing genetic sequence with reference genome in order to match real reads. 9. Learn useful and flexible new algorithms for solving the exact and approximate matching problems by studying a Boyer-Moore, a fast and very widely used algorithm for exact matching. 10. Discussion of read alignment by learning about algorithms that solve both the edit distance problem and related bio sequence analysis problems, like global and local alignment. 11. Learn numerous methods to solve the assembly and alignment problems. |
| **Indicative Contents**  **المحتويات الإرشادية** | The module includes four main requirements that the student must complete in order to successfully pass the course.  **1. Readings:** Students must weekly read each lecture before presenting it in the classroom in order to be able to interact and discuss. The content of the course includes four main parts, and each part includes topics that are illustrated in weekly syllabus, which includes:  **Part 1 – Matching DNA sequencing**  **Part 2 – Preprocessing, indexing and approximate matching**  **Part 3 – Edit distance, assembly, overlaps**  **Part 4 – Algorithms for assembly**  **2. Discussion:** We will use discussion as the main form of interaction in the class. Students’ responses to the weekly readings, their individual assignments, and their thoughtful responses to their classmates' posts show their level of understanding. Their active participation in the discussions is the best way to get the most out of the course!  **3. Oral Presentations:** The purpose of this assignment is to allow students to explore a topic in more detail for each lecture and to share the results with their classmates. Each student is required to submit a short report in slideshow format (10 slides not including title and reference slides) on a topic relevant to one of the course lectures. The proposed topics are presented in the classroom, but the student can choose other related topics (but after the approval of the teacher). Presentations should be based on scientific sources of information (be sure to include an appropriate list of references). And we should delve deeper into an interesting topic for each section. Try to use non-text materials in your presentation (videos or online examples, tables, charts, and graphs) as a way to group and present the main ideas and themes. If some text is necessary, please limit it to very short paragraphs and bulleted lists. Although not a requirement, all presentations will be posted to Google Classroom and a resource for other class participants. The student is expected to answer the questions of his classmates.  **4.** **Project:** This assignment requires the submission project; the student presents a project to solve a new problem that employs all the algorithms that were used during the semester. |

| **Learning and Teaching Strategies**  **استراتيجيات التعلم والتعليم** | |
| --- | --- |
| **Strategies** | The main strategy that will be adopted in the delivery of this module by encouraging students to participate in discussions, while improving and expanding their critical thinking skills. This will be achieved through discussions during the weekly lectures and after the oral presentations by answering the questions of their colleagues. Enhancing the principle of teamwork by participating in the implementation of the laboratory Assignments and developing the student skills in programming using Python by implementing challenged project assignments. |

| **Student Workload (SWL)**  **الحمل الدراسي للطالب** | | | |
| --- | --- | --- | --- |
| **Structured SWL (h/sem)**  **الحمل الدراسي المنتظم للطالب خلال الفصل** | 63 | **Structured SWL (h/w)**  **الحمل الدراسي المنتظم للطالب أسبوعيا** | 4 |
| **Unstructured SWL (h/sem)**  **الحمل الدراسي غير المنتظم للطالب خلال الفصل** | 62 | **Unstructured SWL (h/w)**  **الحمل الدراسي غير المنتظم للطالب أسبوعيا** | 4 |
| **Total SWL (h/sem)**  **الحمل الدراسي الكلي للطالب خلال الفصل** | 125 | | |

| **Module Evaluation**  **تقييم المادة الدراسية** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **As** | | **Time/**  **Number** | **Weight (Marks)** | **Week Due** | **Relevant Learning Outcome** |
| **Formative assessment** | **Quizzes** | 2 | 10% (10) | 4,10 | LO #1,2,3,4 and 9 |
| **Project Assignment** | 1 | 10% (10) | 12 | All |
| **Lab. Assignment** | 1 | 10% (10) | Continuous | All |
| **Seminar** | 1 | 10% (10) | The student chooses the week and the topics | All |
| **Summative assessment** | **Midterm Exam** | 2hr | 10% (10) | 7 | LO # 1-9 |
| **Final Exam** | 3hr | 50% (50) | 16 | All |
| **Total assessment** | | | 100% (100 Marks) |  |  |

| **Delivery Plan (Weekly Syllabus)**  **المنهاج الاسبوعي النظري** | |
| --- | --- |
| **Week** | **Material Covered** |
| **Week 1** | **Part 1 – Matching DNA sequencing**   * Course introduction * genetic sequencing past and present * DNA sequence validation using Python * Counting DNA Nucleotides using Python * DNA to RNA using Python * Translating RNA into Protein using Python * Counting Point Mutations using Python |
| **Week 2** | * Manipulating DNA strings * Downloading and parsing a genome |
| **Week 3** | * How DNA gets copied * How second-generation sequencers work Sequencing reads in FASTQ format * Working with sequencing reads * Analyzing reads by position |
| **Week 4** | * Read alignment * Naive exact matching * Matching artificial reads * Matching real reads |
| **Week 5** | **Part 2 – Preprocessing, indexing and approximate matching**   * Boyer-Moore basics * Diversion: Repetitive elements |
| **Week 6** | * Implementing Boyer-Moore * Preprocessing |
| **Week 7** | **Midterm Exam** |
| **Week 8** | * Indexing and the k-mer index * Ordered structures for indexing * tables for indexing * Variations on k-mer indexes |
| **Week 9** | * Genome indexes used in research * Approximate matching, Hamming and edit distance * Pigeonhole principle and implementation |
| **Week 10** | **Part 3 – Edit distance, assembly, overlaps**   * Solving the edit distance problem * Using dynamic programming for edit distance |
| **Week 11** | * A new solution to approximate matching * Meet the family: global and local alignment |
| **Week 12** | * Read alignment in the field * Assembly: working from scratch |
| **Week 13** | * First and second laws of assembly * Overlap graphs |
| **Week 14** | **Part 4 – Algorithms for assembly**   * The shortest common superstring problem * Lecture: De Bruijn graphs and Eulerian walks |
| **Week 15** | * Lecture: Greedy shortest common superstring * Lecture: Third law of assembly: repeats are bad * Lecture: When Eulerian walks go wrong |

| **Delivery Plan (Weekly Lab. Syllabus)**  **المنهاج الاسبوعي للمختبر** | |
| --- | --- |
| **Week** | **Material Covered** |
| **Week 1** | * How to install Jupyter Notebook on Windows * String definitions and Python examples * DNA sequence validation using Python * Counting DNA Nucleotides using Python * DNA to RNA using Python * Translating RNA into Protein using Python * Counting Point Mutations using Python |
| **Week 2** | * Manipulating DNA strings and Python examples |
| **Week 3** | * Read FASTA and FASTAQ files format. * Working with sequencing reads and Python examples |
| **Week 4** | * Matching artificial reads * Matching real reads and Python examples |
| **Week 5** | * Implementing Boyer-Moore |
| **Week 6** | * Implementing Boyer-Moore |
| **Week 7** | * Implementing a k-mer index |
| **Week 8** | * Implementing a k-mer index |
| **Week 9** | * Implementing the pigeonhole principle |
| **Week 10** | * Implementing dynamic programming for edit distance |
| **Week 11** | * Implementing global alignment |
| **Week 12** | * Overlaps between pairs of reads |
| **Week 13** | * Finding and representing all overlaps |
| **Week 14** | * Implementing shortest common superstring |
| **Week 15** | * Implementing greedy shortest common superstring * Practical: Building a De Bruijn graph |

| **Learning and Teaching Resources**  **مصادر التعلم والتدريس** | | |
| --- | --- | --- |
|  | **Text** | **Available in the Library?** |
| **Required Texts** | Wang, Xinkun. Next-generation sequencing data analysis. CRC Press, 2023.  DOI <https://doi.org/10.1201/9780429329180>  Next-Generation Sequencing Data Analysis | No |
| **Recommended Texts** | Datta, Somnath, and Dan Nettleton. Statistical analysis of next generation sequencing data. New York: Springer, 2014.  الحجم نتيجة الصورة لـ Datta, Somnath, and Dan Nettleton. Statistical Analysis of Next Generation Sequencing Data. New York Springer, 2014..: 120 × 185. المصدر: www.amazon.in  Bosserhoff, A., Kappelmann-Fenzl, M. (2021). Next Generation Sequencing (NGS): What Can Be Sequenced?. In: Kappelmann-Fenzl, M. (eds) Next Generation Sequencing and Data Analysis. Learning Materials in Biosciences. Springer, Cham. <https://doi.org/10.1007/978-3-030-62490-3_1>  الغلاف الأمامي | No |
| **Recommended Websites** | <https://www.coursera.org/learn/dna-sequencing>  <https://www.coursera.org/learn/introduction-genomics> | |

| **Grading Scheme**  **مخطط الدرجات** | | | | |
| --- | --- | --- | --- | --- |
| **Group** | **Grade** | التقدير | **Marks (%)** | **Definition** |
| **Success Group**  **(50 - 100)** | **A -** Excellent | **امتياز** | 90 - 100 | Outstanding Performance |
| **B -** Very Good | **جيد جدا** | 80 - 89 | Above average with some errors |
| **C -** Good | **جيد** | 70 - 79 | Sound work with notable errors |
| **D -** Satisfactory | **متوسط** | 60 - 69 | Fair but with major shortcomings |
| **E -** Sufficient | **مقبول** | 50 - 59 | Work meets minimum criteria |
| **Fail Group**  **(0 – 49)** | **FX –** Fail | **راسب (قيد المعالجة)** | (45-49) | More work required but credit awarded |
| **F –** Fail | **راسب** | (0-44) | Considerable amount of work required |
|  |  |  |  |  |
| **Note:** Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above. | | | | |