Title	Aims/Description	Requirements	Responsible
Pair hidden Markov models for sequence alignment	This project is about implementing methods for global and local pairwise alignment using pair hidden Markov models (HMMs). The aim of the project is to implement the basic algorithms (Viterbi, forward and backward) and training methods for pair HMMs, and conduct experiments that validates the theoretical running times.	Basic knowledge of algorithms, data structures and programming corresponding to introductory level classes in these topics.	Christian N. S. Pedersen
Space efficient implementations of dynamic programming based methods	This project is about the techniques for reducing the space consumption of dynamic programming based algorithms for sequence alignment described in the paper "Powell, D., Allison, L., & Dix, T. (1999). A versatile divide and conquer technique for optimal string alignment. Information Processing Letters, 70(3), 127–139". The aim of the project is to implement a space efficient version of global (and local) pairwise sequence alignment using affine gap cost.	Basic knowledge of algorithms, data structures and programming corresponding to introductory level classes in these topics.	Christian N. S. Pedersen
Visualization of basic sequence alignment algorithms	This project is about implementing a set of programs for visualizing key aspects of the basic algorithms for pairwise sequence alignment. There are several algorithms to focus on, eg global alignment with linear gap cost using Hirschberg's method to reduce space consumption. The aim is to implement a tool (or framework) for teaching purposes that visualizes and animates the execution of such an algorithm.	Basic knowledge of algorithms, data structures and programming (including GUI programming) corresponding to introductory level classes in these.	Christian N. S. Pedersen
Do genes associated with the same disease tend to interact?	Description: 1. Create a set of genes believed to be associated with a given disease: Look at the genome wide association catalogue (http://www.genome.gov/gwastudies/) and find the set of genes that are close to SNPs associating with the disease. This set of genes can be augmented with disease-genes reported in OMIM (http://www.ncbi.nlm.nih.gov/omim) 2. Create a set of random genes taking into account that it is more likely to find associations with long genes than short genes 3. Are the proteins produced by the disease-genes more likely to interact with each other than the random genes? (information on interactions can be taken from http://thebiogrid.org/ or a similar database) 4. Do the disease-genes cluster more in pathways / GO categories than the set of random genes?	Simple scripting/ programming skills	Søren Besenbacher
Probabilistic Modelling of Population Histories	Population histories can be modelled using the Isolation-with-migration model (IM-model) or one of its sub-models, e.g. the Isolation model or the Island model. The parameters of interest are the population sizes, split time and migration rates. When the IM-model is paired with a mutation-model, it is possible to infer these parameters from genomic data by using numerical maximum-likelihood. In this project, the student will derive estimators in simple cases and be presented with derived expressions in more complicated ones, which the student will then implement to perform analysis of a simulated data-set.	The student should understand probability theory, specifically, the concepts conditional probability and calculating probabilities using probability densities, and have some programming experience.	Lars Nørvang Andersen
Protein domain interactions	Analyze multi-domain proteins with known structure, to collect information on which type of interactions (electrostatic, hydrogen-bonds, hydrophobic) are present between protein domains, to elucidate what keeps macro-structures together.	Basic programming skills. Basic chemistry/molecular biology.	Lea Thøgersen
Developing a parallel reduction of the backward and posterior decoding algorithms for hidden Markov models	The student(s) should reuse the ideas from the parallel reduction of the forward and Viterbi algorithms to develop a parallel reduction of the backward and posterior decoding algorithms for hidden Markov models. The student(s) should furthermore implement the algorithms and make a few experiments showing the running time, when using multiple cores.	(Level ++) Algorithms and Data Structures 1. Experience in implementing algorithms. Knowing hidden Markov models in advance from Pattern Recognition in Bioinformatics will be an advantage	Andreas Sand
Implementing the parredHMM algorithms for General Purpose GPUs	Implementing the parredForward and parredViterbi hidden Markov model algorithms for General Purpose GPUs. Experimenting with the implementations, comparing to the CPU implementation of the algorithms and HMMlib.	Algorithms and Data Structures 1. Experience in implementing algorithms.	Andreas Sand

		Experience with CUDA or OpenCL.	
Computing the triplet distance of binary trees in time O(n log(n))	To make an implementation in C++ of the O(n log(n)) time algorithm for finding the triplet distance between a pair of rooted binary trees. To make experiments that show pros/cons of this algorithm compared to the O(n log^2(n)) time algorithm.	Algorithms and Data Structures 1. Experience in implementing algorithms. Experience with C++	Andreas Sand
Computing the all-pairs triplet distance	To develop an algorithm, inspired by the all-pairs quartet distance algorithm and the O(n log^2(n)) time algorithm for computing the triplet distance, for computing the all-pairs triplet distance of a set of rooted binary trees. To make an implementation in C++ of this algorithm and to make experiments with this implementation, confirming its theoretical running time.	Algorithms and Data Structures 1. Experience in implementing algorithms. Experience with C++	Andreas Sand
GapGlu	Genomes are being sequenced with rapid speed due to advancement in next generation sequencing technology. Short reads (50-500 bp) produced from these technologies often results in challenges when performing the de novo genome assembly. Repetitive regions often leads to unfilled regions (gaps) in the genomes as read originating from repeat region can map to multiple places. Here we target to solve this problem using mate-pair information of sequencing libraries.	Basic programming skills in Python/Perl. Understanding of genome structure	Vikas Gupta
Network structure of pairwise interaction in human disease association studies	Description: Ref: http://www.biomedcentral.com/1471-2105/12/364, http://bioinformatics.oxfordjournals.org/content/26/1/30.long Pairwise interaction has been heavily studied in GWAS studies, but only a few study have a focus on the network structure. The reference uses Information Gain to characterize relation of 2 snps, and conclude that disease related network shows distinctive structure compared with random network. The aim of the project is to explore different scoring methods of two SNPs. Scores based on rank or odds ratio may provide change the picture, and an overlap with existing function modules can be interesting. Furthermore, high order interactions can be detected (logic regression).	C/python, R, basic stats.	Yu Qian
Predictor based data-mining methods to detect epistatic interactions	Genome-wide association study focuses on genetic variants from individual to individual among a cohort of cases and controls. Some common complex diseases such as cancer and diabetes are caused by multiple genetic variants, and detecting high-order epistasis (gene-gene interactions) is challenging due to enormous number of possible SNP combinations. MDR (Multifactor dimensionality reduction) is a nonparametric method to identify interactions among discrete variables that influence a binary outcome. However, it is only applied to small-scale analysis and tend to output false positives or results that are difficult to interpret. The aim of the project is to use existing PPI network information, combined with tree-based supervised machine learning method to detect more interpretable results and higher order interactions.	C/C++, python, a little knowledge of machine learning	Yu Qian