Bioinformatics?

Christian Storm Pedersen (cstorm@birc.au.dk)

Bioinformatics Research Center (BiRC), Aarhus University

Bioinformatics Research Centre (BiRC)

- A research center **established in 2001** focusing on research and education in bioinformatics. Co-located with MBG.

- **~35 researchers (6 permanent staff)** and **technical staff with** diverse backgrounds:

  - biology, molecular biology,

  - computer science, statistics, and

  - bioinformatics.
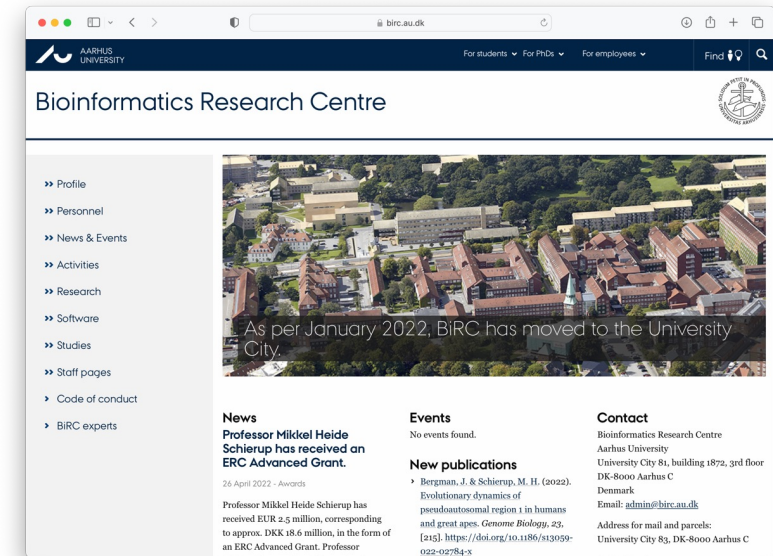
https://birc.au.dk

Biological questions	Computational problems	Data analysis
	Models of biological systems	Algorithms and programs

Abstraction   Algorithms   Automation

Biological questions   Computational problems   Data analysis

Models of biological systems   Algorithms and programs

Abstraction   Algorithms   Automation

Biological questions   Computational problems   Data analysis
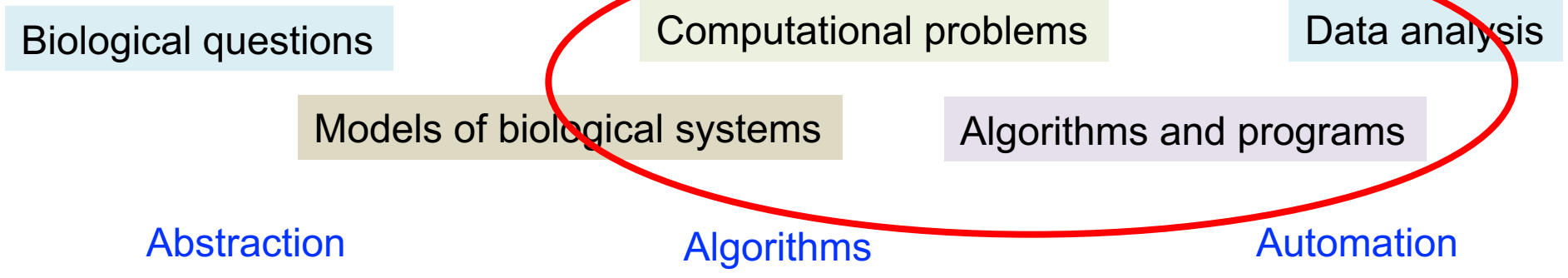
Models of biological systems   Algorithms and programs

Abstraction   Algorithms   Automation

Comparison of multiple biological sequences

Biological sequences (DNA, RNA, and proteins) can be modeled as strings over finite alphabets. The evolutionary relatedness of species can be inferred by comparing their biological sequences. Many algorithms exist for this purpose, and typically involves computing an alignment of the two or more strings. Computing a so-called multiple sequence alignment (MSA) is an important part of many bioinformatics analyses (https://en.wikipedia.org/wiki/Multiple_sequence_alignment), but in most scenarios it is a computational hard problem, and many heuristics and approximation algorithms therefore exist.

The aim of this project is to describe, implement and experiment with algorithms for computing MSAs. More specifically, to understand and describe approximation algorithms and heuristics for computing MSAs that are based on combining optimal pairwise alignments in a certain order such as Gusfield's 2- approximation algorithm (https://link.springer.com/article/10.1007/BF02460299) and heuristics based on (minimum) spanning trees (e.g. https://ieeexplore.ieee.org/abstract/document/1527521). This involves reading about and understanding algorithms for computing MSAs from selected papers and books, making efficient implementations of (a subset of) such algorithms in a programming language of your choice, and making experimental comparisons of the properties of the implemented algorithms using suitable test data, e.g. simulated genomic data in standard formats such as FASTA format for sequences and Phylip format for distance matrices.

Building phylogenetic trees using neighbor-joining

Inferring the evolutionary relationships between a set of organisms is an important step in many biological or medical workflows. It is often referred to as building a phylogenetic tree and is often done by clustering the organisms according to estimates of their pairwise relationships. The neighbor-joining (NJ) method is a widely used method for constructing useful phylogenetic trees. Using the canonical NJ method by Saitou and Nei (https://academic.oup.com/mbe/article/4/4/406/1029664), it takes $O(n3)$ time to build a tree of n organisms if their pairwise relationships (distances) are known. There are many variations of neighbor- joining and many heuristics for speeding it up. RapidNJ (http://birc.au.dk/software/rapidnj/) is one such heuristic.

The aim of this project is to describe, implement and experiment with methods for building phylogenetic trees. More specifically, to understand and describe approaches such as RapidNJ and examine how their running time in practice compares to canonical neighbor-joining. This involves reading about and understanding algorithms for construction of phylogenetic trees using NJ methods from selected papers and books, making efficient implementations of (a subset of) such algorithms in a programming language of your choice, and making experimental comparisons of the properties of the implemented algorithms using algorithms using suitable test data, e.g. simulated genomic data in standard formats such as FASTA format for sequences, Phylip format for distance matrices, and Newick format for trees.

Finding tandem repeats in genomic data

Various non-random patterns in genomic DNA are usually indicators of biological function. Searching for patterns in genomes, that are often billions of characters long, is a non-trivial task, however, and requires efficient data structures and algorithms. This project aims at locating all so-called tandem repeats in a string, that is, strings that are repeated next to each other in the genome (https://en.wikipedia.org/wiki/Tandem_repeat).

A suffix tree (https://en.wikipedia.org/wiki/Suffix_tree) is an efficient index structure for strings that enable many pattern matching algorithms. A straightforward construction takes $O(n2)$, which obviously isn't a feasible solution if we need to search in strings that are hundreds of millions of characters long, but they can also be constructed in $O(n)$ using e.g. McCreight's algorithm (https://dl.acm.org/doi/abs/10.1145/321941.321946). Once we have a suffix tree, we can efficiently search for tandem repeats using e.g. Stoye and Gusfield's algorithm (https://www.sciencedirect.com/science/article/pii/S0304397501001219), locating all tandem repeats in $O(n \log n + z)$, where n is the length of the string and z is the number of tandem repeats identified.

The project involves working with genomic data in standard formats, reading about and understanding relevant algorithms and data structures, and making efficient implementations of (a subset of) such algorithms and data structures in a programming language of your choice.

Gene finding using hidden Markov models

When having large amount of genomic data available for a wide range of organisms, the next step is to infer non-trivial biological information from this data. One important property to infer is which parts of the genome encodes protein, i.e. infer the gene structure of (parts) the genome.

Gene finding can be formulated as the task of deciding for each position in the genome (a large string of millions of characters) whether the nucleotide (character) at that position is in a gene or not. This annotation task can be addressed in many computational manners but has successfully been addressed using machine learning and hidden Markov models, which allows to model useful biological properties in a probabilistic setting. One such method is GenScan (https://www.ncbi.nlm.nih.gov/pubmed/9149143) that is a gene finder for eukaryotic genomes (e.g. human genomes).
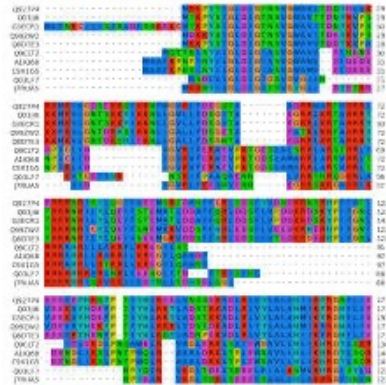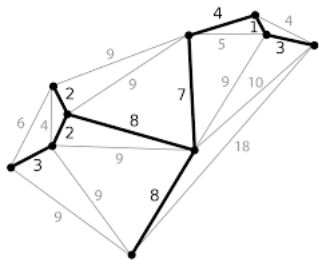
The aim of this project is to make a hidden Markov model-based gene finder for eukaryotic genomes and compare its performance to existing tools, e.g. GenScan. This involves working with genomic data in standard formats such as FASTA, reading about and understanding algorithms related hidden Markov models (training and decoding), and making time and space efficient implementations of (a subset of) such algorithms in a programming language of your choice.

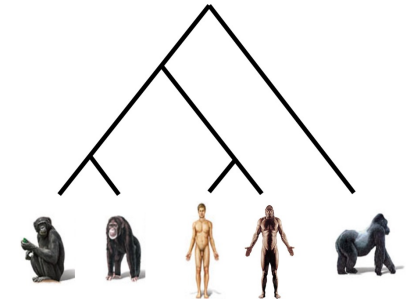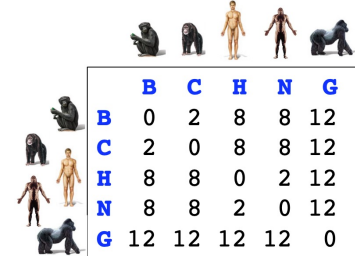## Comparison of multiple biological sequences

The evolutionary relatedness of species can be inferred by comparing their biological sequences.

Building phylogenetic trees using neighbor-joining

Inferring evolutionary relationships between organism by using clustering methods to build phylogenetic trees.
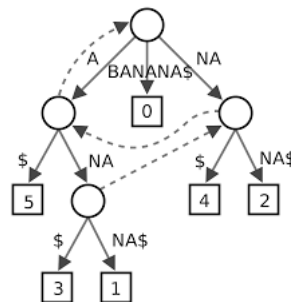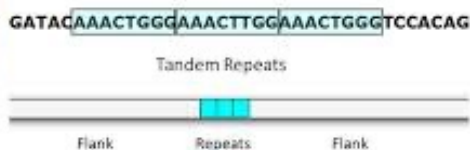

Finding tandem repeats in genomic data

Various non-random patterns in genomic DNA can be indicators of biological function. Searching for patterns in large genomes is a non-trivial task
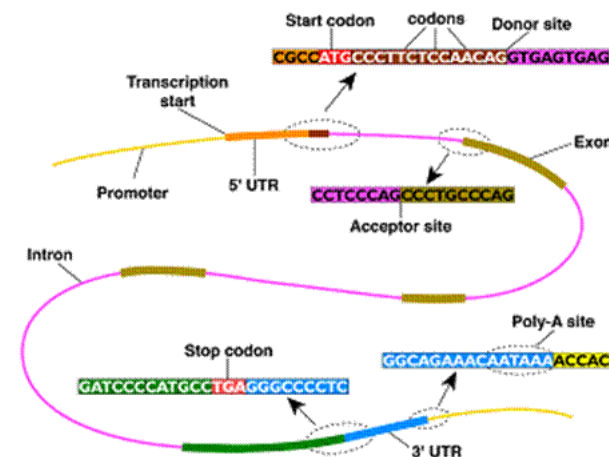

Gene finding using hidden Markov models

Inferring non-trivial biological patterns such as gene structure) in genomes using machine learning.

Comparison of multiple biological sequences

The evolutionary relatedness of species can be inferred by comparing their biological sequences.

Sequence1   –TCAGGA–TGAAC–––––
Sequence2   ATCACGA–TGAACC–––
Sequence3   ATCAGGAATGAATCC––
Sequence4   –TCACGATTGAATCGC–
Sequence5   –TCAGGAATGAATCGCM
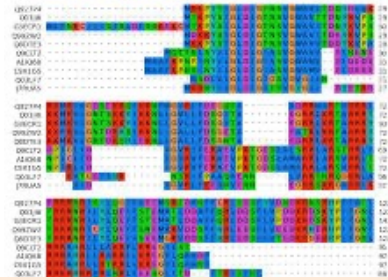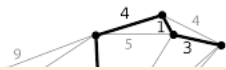
Building phylogenetic trees using neighbor-joining

Inferring evolutionary relationships between organism by using clustering methods to build phylogenetic trees.

	B	C	H	N	G
B	0	2	8	8	12
C	2	0	8	8	12
H	8	8	0	2	12
N	8	8	2	0	12
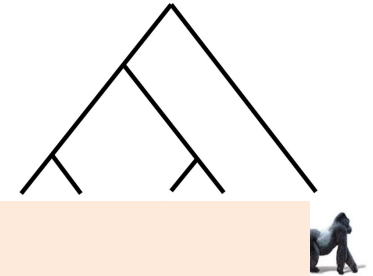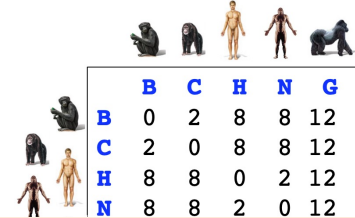
BSc projects in bioinformatics involves:

Reading about and **understanding algorithms** and data structures with applications for biological data analysis, **making efficient implementations** in a programming language of your choice, and **making experiments** to examine their computational properties in practice, and the relevance of the solutions they yield. This involves finding and/or generating **suitable test data**.

Various
indicat
large ge

Tandem Repeats

• They serve as useful markers for identification because they are found in great abundance throughout the human genome.

GATAC AAACTGGG AAACTTGG AAACTGGG TCCACAG

Tandem Repeats

Flank   Repeats   Flank

BANANA$

A   NA
0

$ NA $ NA$
5   4   2

$ NA$
3   1

Start codon   codons   Donor site
CGCCATGCCCTTCTCCAACAGGTGAGTGAG

Transcription start

Promoter   5' UTR   CCTCCCAGCCCTGCCCAG
Acceptor site

Intron   Exon

Stop codon   GGCAGAAACAATAAAACCAC   Poly-A site
GATCCCCATGCCTGAGGGCCCCTC

3' UTR