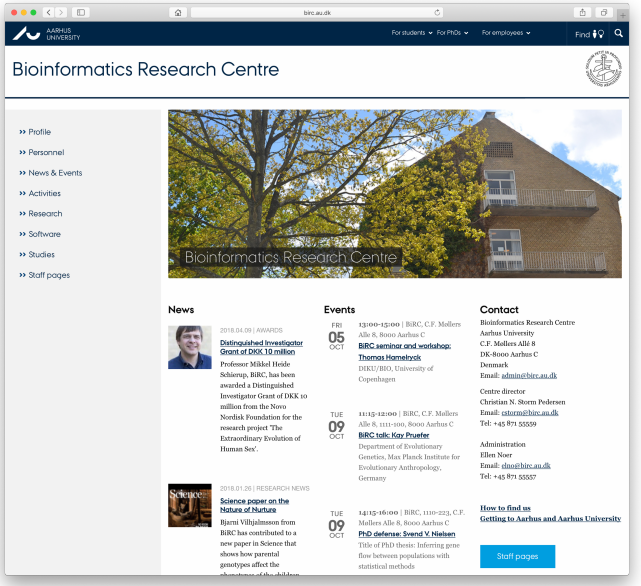
Bioinformatics?

Christian Storm Pedersen (cstorm@birc.au.dk) and Thomas Mailund (mailund@birc.au.dk)

Bioinformatics Research Center (BiRC), Aarhus University

Bioinformatics Research Centre (BiRC)

- An interdisciplinary research center **established in 2001** focusing on rsearch and education in bioinformatics.
- Organized as an **independent center** under the Faculty of Natural Sciences (NAT).
- **~35 researchers (7 permanent)** and **technical staff** with backgrounds in biology, computer science and statistics.



http://birc.au.dk

Biological questions

Computational problems

Data analysis

Models of biological systems

Algorithms and programs

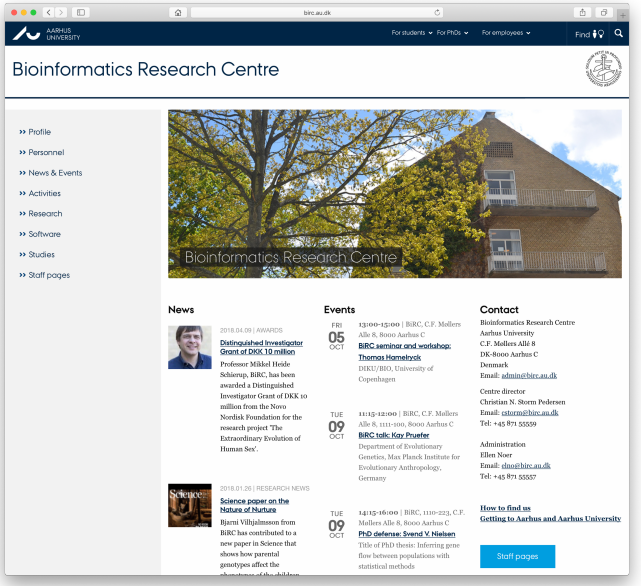
Abstraction

Algorithms

Automation

Bioinformatics Research Centre (BiRC)

- An interdisciplinary research center **established in 2001** focusing on research and education in bioinformatics.
- Organized as an **independent center** under the Faculty of Natural Sciences (NAT).
- **~35 researchers (7 permanent)** and **technical staff** with backgrounds in biology, computer science and statistics.



http://birc.au.dk

Biological questions

Computational problems

Data analysis

Models of biological systems

Algorithms and programs

Abstraction

Algorithms

Automation

Large-scale pattern matching - Read mapping

Area: Bioinformatics, String Algorithms, Algorithmic Engineering

When sequencing the genome of an organism an important step is to map (match) a very large number of reads (small strings of ~50 characters) to a reference genome (a much larger string of millions of characters). The aim is to find out where the reads match with few errors, e.g. match within a short edit- or Hamming-distance.

There are many algorithms and data structures to explore ranging from simple exact pattern matching algorithm like Knuth-Morris-Pratt and Boyer-Moore to more advanced methods based on hashing or data structures such suffix trees, suffix arrays, and the Burrow-Wheeler transform (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3375638/). There are (of course) also a lot of existing read mapping tools such, e.g. BWA (http://bio-bwa.sourceforge.net and http://sfg.stanford.edu/mapping.html).

The aim of this project is to implement a read mapper and evaluate its performance, in terms of quality, running time, and space consumption. This involves working with genomic data in standard formats such as FASTA, FASTQ and SAM, reading about and understanding algorithms and data structures for exact and inexact pattern matching from selected papers and books, and making efficient implementations of (a subset of) such algorithms and data structures in a programming language of your choice.

Comparison of biological sequences using different gap costs

Area: Bioinformatics, String Algorithms, Algorithmic Engineering

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 strings under various scoring schemes (https://en.wikipedia.org/wiki/Sequence_alignment).

The aim of this project is to describe, implement and experiment with such classical alignment based measured using different types of gap costs. More specifically, to understand and described algorithms that makes it possible to use complex scoring schemes such as concave (or convex) gap cost functions (e.g. https://link.springer.com/article/10.1007/BF02459948), and investigate how their performance (running time and space consumption) compare to algorithms with simpler scoring schemes. This involves working with genomic data in standard formats such as FASTA, reading about and understanding alignment algorithms from selected papers and books, and making efficient implementations of (a subset of) such algorithms in a programming language of your choice.

Gene finding using hidden Markov models

Area: Bioinformatics, Machine Learning, Algorithmic Engineering

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 or not 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 sub set of) such algorithms in a programming language of your choice.

Building phylogenetic trees using neighbor-joining

Area: Bioinformatics, Clustering, Algorithmic Engineering

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 (as e.g. inferred by alignment methods).

The neighbor-joining (NJ) methods is a widely used method that for constructing useful phylogenetic trees. Using this method, it takes O(n3) time to build a tree of n organisms assuming that their pairwise relationships (distances) are known. There are many variations of neighbor- joining and heuristics for speeding it up. One such method is RapidNJ (http://birc.au.dk/software/rapidnj/).

The aim of this project is to describe, implement and experiment with methods for building phylogenetic trees. More specifically, to understand and described approaches such as RapidNJ and examine how their running time in practice compares to canonical neighbor-joining. This involves working with genomic data in standard formats such as FASTA, reading about and understanding algorithms for construction of phylogenetic trees using NJ methods from selected papers and books, and making efficient implementations of (a subset of) such algorithms in a programming language of your choice.

Large-scale pattern matching - Read mapping

Area: Bioinformatics, String Algorithms, Algorithmic Engineering

When sequencing the genome of an organism an important step is to map (match) a very large number of reads (small strings of ~50 characters) to a reference genome (a much larger string of millions of characters). The aim is to find out where the reads match with few errors, e.g. match within a short edit- or Hamming-distance.

There are many algorithms and data structures to explore ranging from simple exact pattern matching algorithm like Knuth-Morris-Pratt and Boyer-Moore to more advanced methods based on hashing or data structures such suffix trees, suffix arrays, and the Burrow-Wheeler transform (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3375638/). There are (of course) also a lot of existing read mapping tools such, e.g. BWA (http://bio-bwa.sourceforge.net and http://sfg.stanford.edu/mapping.html).

The aim of this project is to implement a read mapper and evaluate its performance, in terms of quality, running tim
standard formats such
algorithms and data st
and books, and making
structures in a prograr

Comparison of biological sequences using different gap costs

Area: Bioinformatics, String Algorithms, Algorithmic Engineering

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 strings under various scoring schemes (https://en.wikipedia.org/wiki/Sequence_alignment).

The aim of this project is to describe, implement and experiment with such classical alignment based measured using different types of gap costs. More specifically, to understand and described algorithms that makes it possible to use complex scoring schemes such as concave (or convex) gap cost functions (e.g. https://link.springer.com/article/10.1007/BF02459948), and investigate how their performance (running time and space consumption) compare to algorithms with simpler scoring schemes. This involves working with genomic data in standard formats such as FASTA, reading about and understanding alignment algorithms from selected
) such algorithms in a

BSc projects in bioinformatics

Reading about and understanding algorithms and data structures, making efficient implementations in a programming language of your choice, and making experiments in order to examine their computational properties in practice, and the relevance of the solutions they yield

Gen

Area: Bioinformatics, M

When having large am
step is to infer non-triv
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 or not 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 sub set of) such algorithms in a programming language of your choice.

-joining

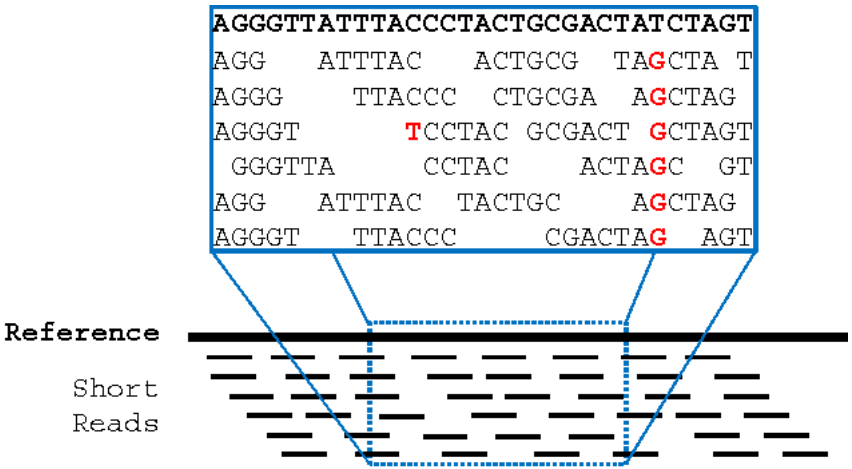
important step in
a phylogenetic tree,
and is often done by clustering the organisms according to estimates of their pairwise relationships (as e.g. inferred by alignment methods).

The neighbor-joining (NJ) methods is a widely used method that for constructing useful phylogenetic trees. Using this method, it takes O(n3) time to build a tree of n organisms assuming that their pairwise relationships (distances) are known. There are many variations of neighbor- joining and heuristics for speeding it up. One such method is RapidNJ (http://birc.au.dk/software/rapidnj/).

The aim of this project is to describe, implement and experiment with methods for building phylogenetic trees. More specifically, to understand and described approaches such as RapidNJ and examine how their running time in practice compares to canonical neighbor-joining. This involves working with genomic data in standard formats such as FASTA, reading about and understanding algorithms for construction of phylogenetic trees using NJ methods from selected papers and books, and making efficient implementations of (a subset of) such algorithms in a programming language of your choice.

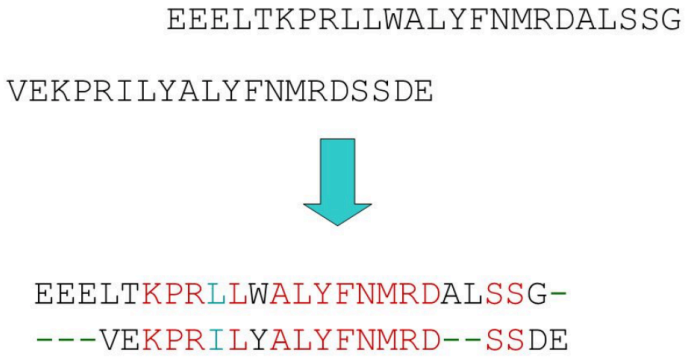
Large-scale pattern matching - Read mapping

Combining small fragments of DNA to an entire genome by large-scale (approximate) pattern matching using suffix trees or suffix arrays.



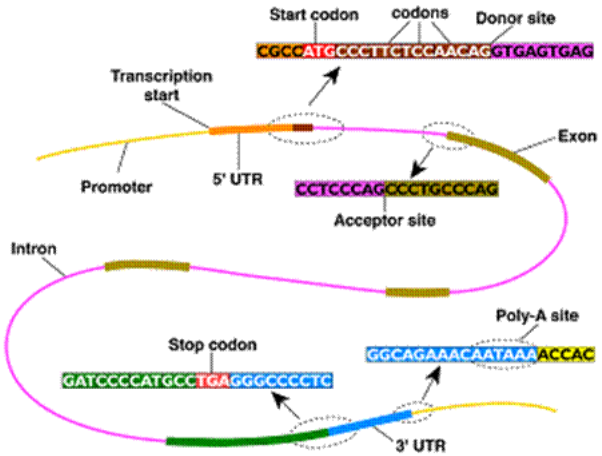
Comparison of biological sequences using different gap costs

Estimating the evolutionary distance between biological sequences by using distance measures between strings.



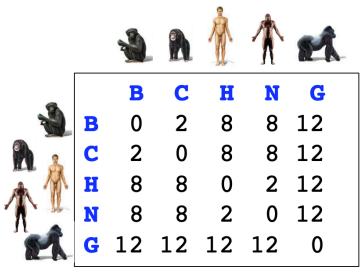
Gene finding using hidden Markov models

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



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
G	12	12	12	12	0