# Assignment 1:  Abstract

Modern Sequencing machines generate a large amount of data that must be processed. However, the existing algorithms and best tools are not perfect and produce archives.

We will use compression algorithm called FQSqueezer that sequences data and processes single- and paired-end reads of variable lengths, and what was mentioned above was based on the dynamic Marco encoder algorithm.

The disadvantage of the proposed method are large memory and time requirements.

# Introduction

Illumina's instruments generate the vast majority of available data at a low cost compared to other sequencers.

The sequenced reads are short but with excellent quality, on the other side The PacBio or Oxford Nanopore 3rd generation instruments can produce much longer readings, but with worse quality and lower throughput.

Unfortunately, items like short-distance repetitions of sections of data are rare in read sets, so researchers looked for other options. They initially concentrated on the compression of bases to collect reads from near regions of genomes.

Our algorithm” FQSqueezer” make use of the ideas from the prediction by partial matching (PPM)19,20 and dynamic Markov coder (DMC)21 general-purpose methods.

Some suggested techniques: the organization of the huge dictionaries, design of the PPM-like estimation of probabilities, use of a custom DMC as the stage following the PPM, the technique for prediction and correction of sequencing errors, technique of ordering the reads and making use of shared prefixes.

The main aspect of FQSqueezer is its compression ratio, usually much better than of the state-of-the-art competitors, i.e., FaStore12, Spring14, and Minicom15. however, our tool has  some drawbacks in terms of speed and memory usage" it is a few times slower than the mentioned competitors in compression and much slower in decompression".

# Assignment 2 : Related Works

Given that the amount of data that needs to be sequenced and stored is large, and this will consume a lot of effort and money, it is not surprising that a large amount of research has been conducted to solve this problem.

The first step was Using gzip, (which is a general purpose compressor). Reducing the file size by nearly 3x was impressive, but the deluge of data demanded more. The biggest drawback to Gzip is that it is mainly built for text data, or more specifically, data with repetitive textual types.

The next step was to create specialized algorithms that consider the different types of redundancy found in FASTQ 3 files.

As the details of the proposed algorithms differ, researchers have begun to look for alternatives. They initially focused on pressing the law. The basic idea was to rearrange the data in order to capture readings from regions close to the genome, this may seem like a loss of data.

In the following years, other researchers explored the concept of using minimizers, i.e., short substrings of sequences, to find reads from close regions.

In 12, it is illustrated how the reads are grouped from slightly larger genome regions. However, three recent research papers describing HARC13, Spring14 and Minicom15 have had much better results. Attempts differ in detail, but they are all based on the same principles.