# Manual for the version 1.14

## Contents:

## Introduction

Elloreas is a targeted genome assembler, which means it takes a starter and then iteratively extends it with sequencing reads. This starter may be a contig from another assembly or just a random read. The principle of operation of Elloreas is similar to the principles of many other similar tools, like TASR, Mapsemsler2, NOVOPlasty or DistributedNucleatingAssembler. However, while those assemblers are designed for short reads, Elloreas was created for long reads, produced by sequencing machines of PacBio and Oxford Nanopore.

I created Elloreas to assemble the mitochondrial genome of Fagopyrum esculentum. A feature of this genome is that it consists of 10 circular chromosomes. Moreover, they can recombine, merging with each other or splitting into several smaller chromosomes. Multiple alternative forms of chromosomes coexist within a single plant, this is called "structural heteroplasmy". You can read more about this in an upcoming paper "Mitochondrial genome of Fagopyrum esculentum and the genetic diversity of extranuclear genomes in buckwheat" where Elloreas is described. I decided that Elloreas may be useful for other bioinformaticians, this is why I deposited it on GitHub.

## How to run Elloreas.

A simple version, which will often be enough:

```
perl elloreas.pl --reads reads.fastq --starter starter.fasta
```

A slightly more complicated version:

```
perl elloreas.pl --reads reads.fastq --starter starter.fasta --
sequencing_technology oxford_nanopore --
minimum_length_of_aligned_read_part 8000
```

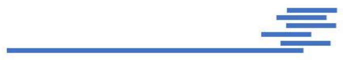
For a full list of parameters with suggestions on how to use them run

```
perl elloreas.pl --help
```

## How Elloreas works.

1) A starter is taken. It can be a contig from another assembly or a random read.

2) Reads are mapped to the edge of the starter.



3) Elloreas finds clusters of reads with similar sequences. There will be several clusters represented by many reads if there are several alternative extensions on the edge. Usually only one alternative extension is represented by many reads, while other alternative extensions are scarcely represented as they arise from reads with many sequencing errors.

4) The starter is elongated using a consensus sequence of the cluster represented by the largest number of reads.

5) Go to the next iteration (point “2)”).

At each iteration Elloreas produces a list of alternative extensions.

A more detailed description of the algorithm is also present in this document (How Elloreas works (a detailed description)).

## What is Elloreas used for?

Usually de novo genome assemblers cannot assemble each chromosome in a whole contig. They produce many contigs which are fragmented in places which are created by "forks" in the assembly graph. Forks are places with two or more alternative extensions, which usually arise because of repeats in genomes, like transposons. You can read more about forks, for example, here. Elloreas is useful to:

1. Understand which alternative extensions exist for your contig. Then you can choose the extension you like and assemble the genome by Elloreas further using this particular extension.
2. Assemble whole genomes. To do this, use a random read as a starter. Elloreas in not very fast - expect a rate of starter elongation of approximately 1000 bp/minute. Therefore, I don't recommend to use it for genomes larger than bacterial.

## Requirements:

Perl
R
minimap2
samtools
BLAST+
bedtools
USEARCH
MAFFT
EMBOSS
LASTZ

I tested Elloreas with the following versions, but it should work with others as well: Perl 5.28, R 3.5.1, minimap2 2.17, samtools 1.9, BLAST+ 2.9.0, USEARCH 11.0.667, MAFFT 7.402, EMBOSS 6.6.0, LASTZ 1.04. Paths to binaries of all these programs should be available through the environment variable $PATH.

## Output of Elloreas

The main files produced by Elloreas are:

1. elloreas_logs.txt - this file contains the basic information of how Elloreas worked. The first thing you may want to look at when Elloreas finishes is the end of this file - it provides the most important information of the Elloreas run.
2. history_of_alternative_extensions.txt - this file contains the list of alternative extensions present in each iteration of elongation. Only the top 5 most represented by reads extensions are listed there.
3. contig_from_the_final_iteration.fasta - this is the final contig produced by Elloreas.
4. coverage_plot_for_the_contig_from_the_final_iteration.jpeg and dotplot_for_the_contig_from_the_final_iteration.jpeg - these two figures represent a read coverage distribution and a dotplot for the final contig produced by Elloreas.

## Caveats:

1. Long reads usually contain a large percent of sequencing errors, namely short indels and substitutions of one nitrogen base by another. Some of these errors may persist in your assembly. Therefore, after you assemble the sequence, I highly recommend to polish it with programs such as Pilon or Racon. Don't forget to include all sequences from your

genome during polishing. Otherwise, there may be mistakes. For example, if you polish a plant's mitochondrial genome but don't add the plastid genome during polishing, some of plastid reads will map to the mitochondrial genome (because these two genomes have homologous sequences), which may make mistakes. During the assembly stage, the plastid genome won't be a big problem, because those homologous sequences create forks, which are reported by Elloreas, so you can choose the mitochondrial extension and ignore the plastid one.

---

## Frequently asked questions:

1. How to cite Elloreas?

   Cite the paper "Mitochondrial genome of Fagopyrum esculentum and the genetic diversity of extranuclear genomes in buckwheat" (currently under review) where Elloreas was first described.

2. What's the difference between this manual and the readme at https://github.com/shelkmike/Elloreas?

   The only difference is that this manual contains not only a short description of the algorithm of Elloreas, but also an additional detailed description (How Elloreas works (a detailed description)).

3. Elloreas extends the contig only in the 3' direction ("to the right"). Why not make it to extend the contig to the left too?

   One of the most important features of Elloreas is that it provides a user with a list of forks, when they are present. Making a user to examine forks on 3' and 5' ends simultaneously may confuse the user. If you need to extend the contig leftward, just take the reverse complement version of the contig, so left becomes right. If you're assembling a circular chromosome, you won't even need to do this, because during the extension the right end will join the left end.

4. Can Elloreas be used with short reads?

   Yes, it can. Minimap2, the read mapper which Elloreas uses, is capable of mapping short reads too. However, Elloreas will use paired-end reads as single-end reads. This is why I recommend to use dedicated targeted assemblers for short reads, like NOVOPlasty or TASR.

5. There was a fork at some iteration. Elloreas chose one of alternative extensions, but I want to try another one. What should I do?

   In the file history_of_alternative_extensions.txt find the respective iteration. You will see a list of alternative extensions for this iteration. Elloreas always chooses the extension supported by the highest number of reads. Take the file contig_from_the_final_iteration.fasta, remove everything starting with this most-supported extension and instead paste the extension you want to try. Then, use this newly formed contig as a starter.

6. Where can I ask questions about Elloreas?

   You can use the "Issues" section on GitHub (https://github.com/shelkmike/Elloreas/issues)

7. How fast is Elloreas?

   I tested it using 22 CPU cores for several datasets each containing about a million reads and the rate of contig elongation was on the order of 1 kilobase per minute.

8. Will Elloreas run on Windows or Mac?

   I didn't test it on Mac, but I think it will work if you can install all the required programs. To run Elloreas on Windows use a virtual Linux machine, for example Ubuntu run through VirtualBox (https://www.osboxes.org/ubuntu/).

9. Many de novo assemblers create assembly graphs. Why can't they be used to detect forks in the assembly instead of Elloreas?

   You can use them. For example, Spades produces FASTG files with assembly graphs which can be visualised with Bandage. However, in my experience, de novo assemblers often don't report all forks. Probably, this is because there are some inner requirements of how represented by reads a branch should be to report this branch in a FASTG file. Elloreas, instead, produces a more detailed list of alternative extensions arising during the assembly process.

10. Well, finally, will Elloreas be useful for me?

    I don't know. It definitely was useful for ME. You may give it a try.

---

## How Elloreas works (a detailed description).

Will appear here soon