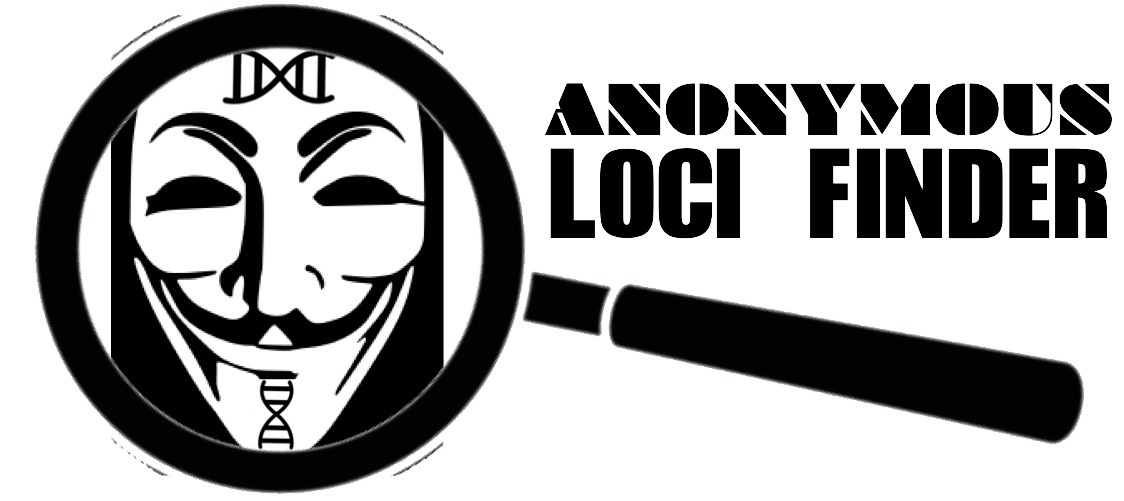
Universidade Federal do Rio de Janeiro

Instituto de Bioquímica Médica Leopoldo de Meis

**Lampada**

LAboratório Multidisciplinar Para Análise de Dados

****

**Alfie 1.0**

**a package for nuclear anonymous loci prediction and phylogenomic analysis using complete genomes**

Igor Rodrigues da Costa

W Bryan Jennings

Francisco Prosdocimi

Rio de Janeiro, 2015

**SUMMARY**

**1. Introduction**

**1.1. What is alfie? (page 3)**

**1.2. How it works**

**1.3. What to use Alfie for?**

**2. Quick Start (page 5)**

**2.1. Quick install**

**2.2. Simple run**

**2. Installation (page 5)**

**3. Parameters (page 12)**

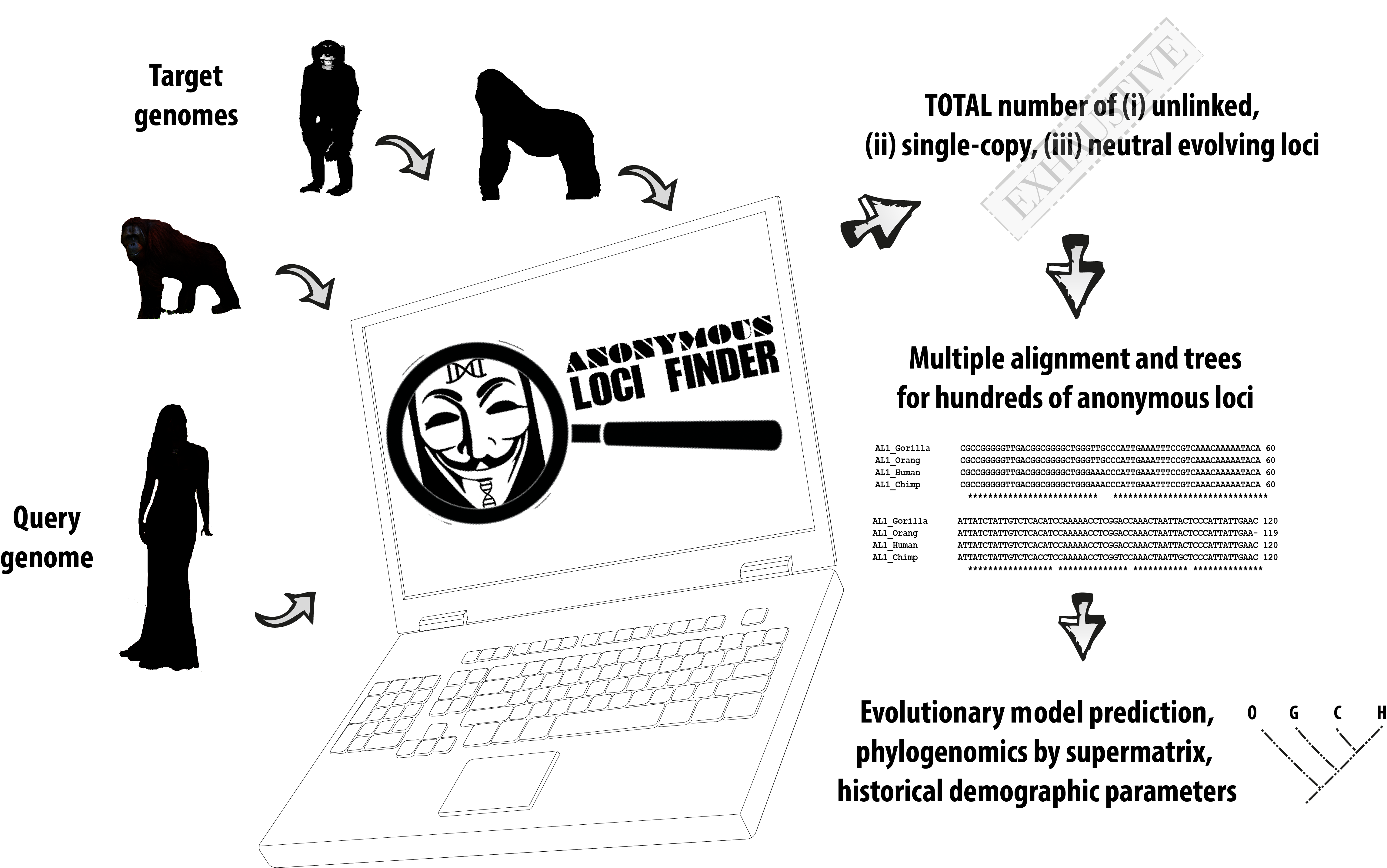
**4. Running the script (page 21)**

**5. Results (page 22)**

**Thank you for downloading Alfie !**

1. **Introduction**
   1. **What is alfie?**

Alfie is an open-source python package containing a pipeline that searchs for single-copy, neutral evolving and non-linked anonymous loci. In other words, it looks for non-coding loci with a set of characteristics which makes them perfect for **phylogenomics** and **populational genetic** analyses. The goal of alfie is to be a complete, user-friendly and flexible anonymous loci predictor.



* 1. **How it works**

Alfie utilises a sequential modular approach, with each step of the pipeline being stored in a single module that can be excecuted with greater flexibility in standalone mode. Many of those steps require external programs, such as BLAST or CLUSTALW. To reduce the complexity of the steps the whole pipeline can be run automatically using the afie.py script.

Using as input a single query genome and its associated GTF file, alfie finds all anonymous regions distant from genomic features that might be target of natural selection to avoid the effect of genetic linkage. By default, this value is defined as 200Kb. After the initial anonymous regions have been found, they are split in small pieces of given size (default 1Kb). Each piece is searched against subject and query genomes to filter duplications and find orthologous regions in all genomes.

Alfie performs an exhaustive search, *i. e*., it finds every possible anonymous regions using the specified parameters in the genomes analyzed. Using default parameters and closelly related vertebrate genomes, alfie will usually find 100-1000 anonymous loci. Each ortholog group of anonymous region is written in NEXUS, PHYLIP, ALN and FASTA formats. Alfie will also concatenatenate all the alignments into a large file for phylogenomic analysis. The package comes with support for running modeltest, phylogenomics and population genetics software.

* 1. **What to use Alfie for?**

Alfie was developed to find hundreds to thousands of independent, nuclear markers among whole complete genomes and facilitate population genomics analyses. The basic user dataset would be four to ten genomes of a close-related taxa, we recommend no more than 20 million years of divergence to allow precise ortholog assignment. In the test case, we found about 300 loci for the human genome and expand the search to find orthologs to these markers in hominoid genomes (chimpanzee, gorilla and orang utan).

It is strongly recommended that at least one of the genomes have been extensively studied and annotated, presenting a comprehensive GTF file that describes the precise location of the main features targeted by natural selection, such as genes and regulatory elements.

**2-Installation:**

Users can either download and install the entire source code from (i) github at the address <http://github.org/lampada/alfie.git> or download the docker container with all the dependencies included.

You can also look at the *testcase/* folder for an example of an input file.

Working versions for each required program can be found on the following links:

Blast: [http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE\_TYPE=Blast Docs&DOC\_TYPE=Download](http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=Blast%20Docs&DOC_TYPE=Download)

ClustalW: <http://sourceforge.net/projects/mira-assembler/>

PhyML: <http://www.atgc-montpellier.fr/phyml/binaries.php>

ModelTest: <https://code.google.com/p/jmodeltest2/>

You will also need a working **BioPython package**, which you can get from <http://biopython.org/wiki/Download>

**2.2. Testing the script**

$> python test.py

**4 – Flags to alter the program behavior:**

**4.1 – The -j flag:**

**4.2 – The -r flag:**

**4.3 – The -i flag:**

**5 – Running the script:**

*Two examples of simple usage:*

Example 1. Command line

>python mitoMaker.py -j test -i test.input -p 8 -r my\_ref.gb

This command line will run mitoMaker with job *test,* using test.input as the input file (please read section 3 for more info on how to build this file) using 8 threads whenever possible during assembly programs.

Example 2. Command line

>python mitoMaker.py -j test -i test.input -p 64 -r my\_ref.gb --skipmitobim --skiptrna

This command will run mitoMaker with job *test,* using test.input as the input file, using 64 threads whenever possible. It will also skip MITObim phase and the program tRNAscan-SE.

**6 – Results**

Inside the main folder the user chose to run the job, various files will be created and a final **result** folder. Here's a summary of the most important files created:

**7 – Standalone modules:**

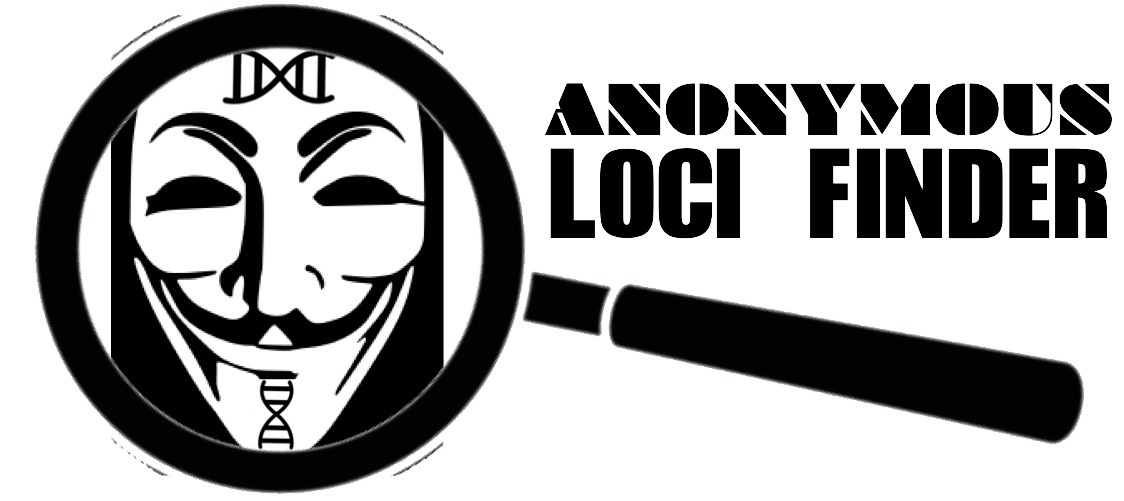
The script for circularization checking and for automatic annotation used by mitoMaker can also be called as standalones for expert users. This will be helpful if the user has some assemblies already performed that need to be annotated. Or in the case the user you prefer to use another method to assembly the reads.

**7.1. al\_circos.py:**

This script is used to check if there is evidence of circularization in a given assembly. In order to call it as a standalone, the user will need to provide the FASTA file on which you want to check for circularization and the script will print the results as a python tuple:

Thanks for using **Alfie!**

Please cite us:

****

Costa, Prosdocimi and Jennings (2015).

**in silico Phylogenomics Comes of Age**:   
Using Bioinformatic Algorithms to describe anonymous markers in whole Genome Datasets.