

Mikroskop — version 1.0

Installation and User Guide

Broňa Brejová, Tomáš Vinař

{bbrejova,tvinar}@cs.uwaterloo.ca

Contents

1	Introduction	1
2	Installing mikroskop	1
3	Using mikroskop	2
3.1	Creating list of loci	2
3.2	Description of diagrams	2
3.3	File formats	2
3.4	Fine-tuning pictures	4

1 Introduction

Mikroskop is a tool for visualization of gene annotations, such as results of gene finding programs. It creates diagrams of sequence features, including gene structures, GC content, EST alignments and other information.

Mikroskop requires Perl (version at least 5.8), and jgraph (www.cs.utk.edu/~plank/plank/jgraph/jgraph.html). The input should be in GTF format (genes.cs.wustl.edu/GTF2.html), quantitative information can also be supplied in a simple fasta-like format. The output is in encapsulated postscript (EPS), or optionally PDF format (requires epstopdf available at www.ctan.org/tex-archive/support/epstopdf/).

The program is distributed AS IS, under GPL license, in a hope that it will be useful.

2 Installing mikroskop

1. Download the file mikroskop-1.0.tgz and uncompress it.

```
tar xzf mikroskop-1.0.tgz
```

This will create an installation directory mikroskop-1.0

2. Change to the installation directory and run installation script as follows:

```
make install
```

By default, the script will install mikroskop in subdirectories of /usr/local. This usually requires root privileges. You can change the destination by setting one or more of the following variables:

prefix: use different directory instead of /usr/local; all other variables are set relative to the prefix, unless overridden by the user

bindir: directory for user executables (default: \$prefix/bin/)

libdir: additional files required by mikroskop (default: `$prefix/lib/mikroskop`)

docdir: documentation (default: `$prefix/share/doc/mikroskop`)

For example, `make install prefix=~/.mikroskop` will install mikroskop files in subdirectory mikroskop of your home directory. The user executables are placed in `~/.mikroskop/bin`.

3. Make sure that user executables are included in your path; by default, executables are placed in `/usr/local/bin` and are included in the system path.
4. If everything went well, you should be able to run executables: `mikroskop`, and `find-loci`; running these executables should display brief help messages.
5. You can now remove the installation directory.

3 Using mikroskop

The package includes sample files illustrating use of mikroskop. Copy the directory with sample files `$prefix/share/doc/mikroskop/sample` into a directory where you have writing permissions, change to that directory, and run:

```
mikroskop mikroskop.txt locuses.txt images
```

If everything is installed correctly, the script will create subdirectory “images”, and in it four PDF files, including the two shown in Figures 1 and 2. You can use files in the sample directory as examples of files that you need to create in order to use mikroskop.

Mikroskop generates multiple diagrams in a single run. All diagrams display the same features, but for different regions of one or several sequences. The first parameter in this example, file `mikroskop.txt`, describes the content of the figures, while the second file, `locuses.txt`, describes the regions of the sequence (loci) for which the diagrams will be created. The third parameter specifies the target directory in which the diagrams will be created.

3.1 Creating list of loci

File containing the list of loci specifies region for each diagram on a separate line. Each of these lines is in the following format:

```
<diagram_name> <sequence_name> <coordinates_from> <coordinates_to>
```

`<diagram_name>` is used as a base for the file name of the resulting diagram. `<sequence_name>` is the name of the sequence to which diagram relates (note that GTF format may contain information about annotation of several sequences; this sequence name corresponds to the first column in the GTF file). Finally, `<coordinates_from>` and `<coordinates_to>` describe the section of the sequence for which information should be displayed.

The list of loci can be created by your own methods, or by the script `find-loci`. This script automatically finds disjoint regions with overlapping genes from one or several GTF files. The script considers only rows of GTF files with `transcript_id` defined. Run `find-loci` executable to receive more detailed help message.

3.2 Description of diagrams

Mikroskop also needs a description of diagrams. This description is stored in a file (in above example `mikroskop.txt`), which specifies location of one or several GTF files or files that contain quantitative information in a simple fasta-like format. The file also contains description of how the information should be displayed. The format of the diagram description file and available options are displayed by running:

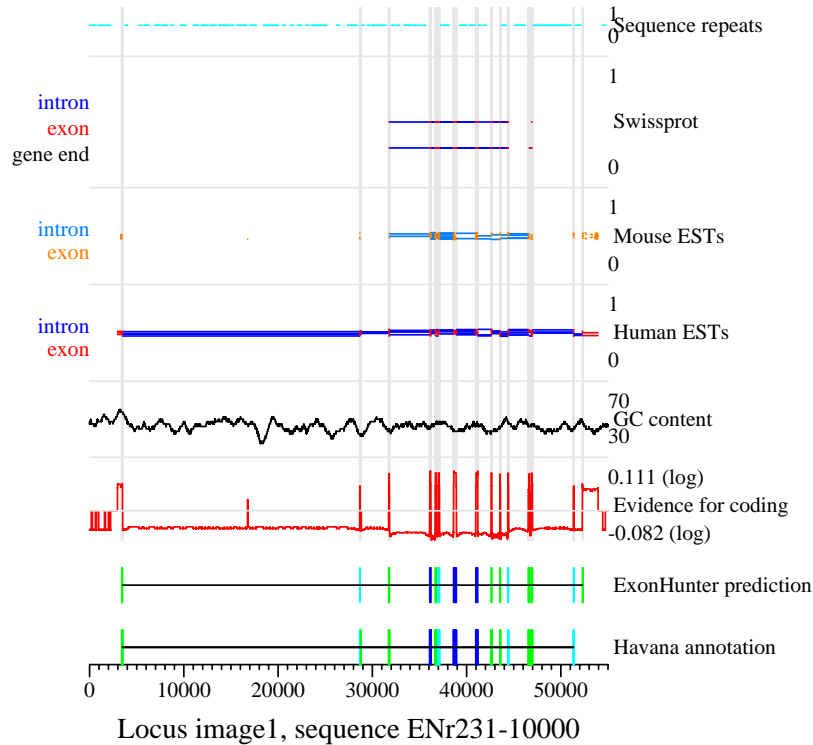


Figure 1: Two different annotations of the same gene and supporting evidence.

`mikroskop --help`

Two examples of the output of `mikroskop` are shown in Figures 1 and 2. The first image shows two annotations of the same gene in human genome. The HAVANA annotation is hand-curated, ExonHunter annotation is an automated prediction by ExonHunter gene finder. The second figure shows greater detail of one of the areas.

3.3 File formats

Mikroskop can draw sequence information stored in one of three supported formats: GTF, values and advisor files.

GTF format. GTF format is a standard format used to describe gene annotation. Specification can be found at genes.cs.wustl.edu/GTF2.html. It is possible, for example, to download various features of multiple genomes in GTF format from the UCSC genome browser. The lower two tracks of the Figures 1 and 2 display information from GTF files describing genes. Coding regions (exons) are displayed as colored boxes, introns as black lines. Green and blue shades are used on forward strand, red and yellow on the reverse strand. The color helps to distinguish reading frame: if two exons on the same x-coordinate have the same color, they have the same reading frame (which is the case for all exons in the two figures). Alternative isoforms and overlapping genes are simply drawn on top of each other. In Figure 2 we see that HAVANA annotation exhibits alternative splicing at the fourth exon from the left.

Tracks for repeats and EST and protein alignments also use information from GTF files. The third column of a GTF file contains so called “feature”, such as `exon`, `CDS`, or `intron`. We may choose to display arbitrary features in one track.

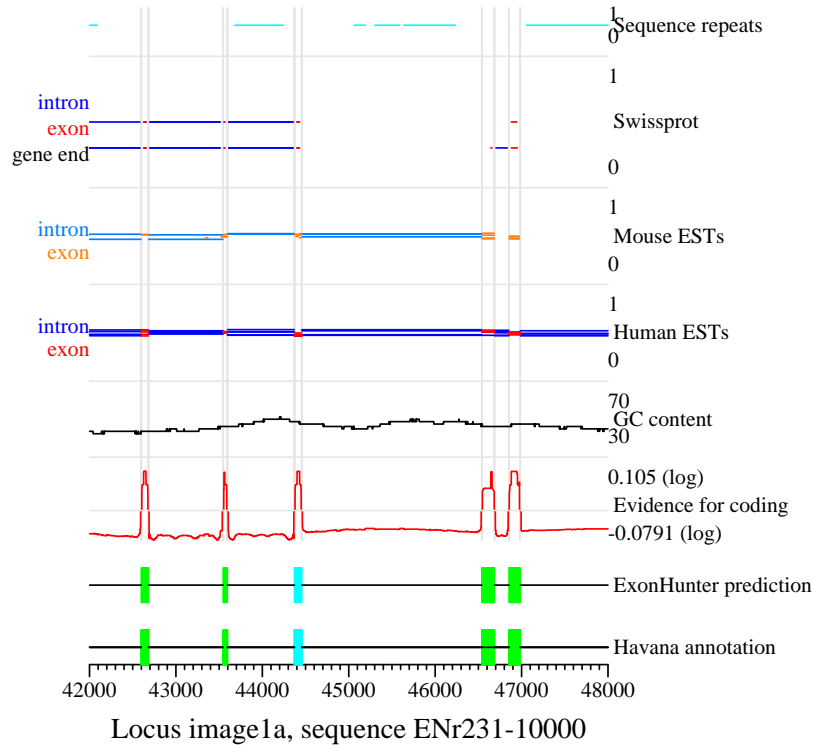


Figure 2: Detailed view of the part of the sequence in Figure 1.

Quantitative information in a simple fasta-like format. Such file contains one number for every position of the sequence (separated by whitespace or end of line). Name of each sequence is on a separate line, starting with >, as in fasta sequence files. In the two sample figures, tracks for GC content and numerical summary of evidence for coding region use information from a value file.

Advisor files. Finally, advisor files are internal files used in ExonHunter gene finder. Their role is similar to value files.

3.4 Fine-tuning pictures

To use mikroskop images in publications, further fine-tuning may be desirable. Instead of EPS or PDF images, mikroskop can produce a source code for the jgraph package. This source code can be then manually modified to achieve the desired effect. This method allows to change most aspects of the image, such as image title, fonts, x-axis labels, line widths, location of labels, etc. It is also possible to add or remove some elements of the figure. See documentation of jgraph for details of the jgraph format.