User guide for MicNeSs

August 7, 2015

Contents

1 Implementation **3**

2 Usage **3**

3 Output format **4**

4 Options **4**
4.1 Verbosity 4
4.2 SSRs detection 5
4.2.1 Minimum number of sequences 5
4.2.2 Minimum number of repeats 5
4.2.3 Motif size 5
4.2.4 Reverse complement 6
4.3 Genotyping 6
4.3.1 Substitution(s) 6
4.3.2 Maximum width of the distribution 6
4.3.3 Maximum asymmetry of the distribution 7

5 Examples **7**

6 Contact **7**

7 Citing MicNeSs **8**

This manual provides a tutorial for MicNeSs 1.0. MicNeSs genotypes individuals using microsatellites from a collection of (NGS) sequences. The method is described in Suez M. *et al.* (submitted).

1 Implementation

MicNeSs is an open source program written in Python 2.7 that can only be run as a command line. It is is available at `http://wwwabi.snv.jussieu.fr/public/micness/`. Importantly, MicNeSs relies on two standard python libraries: **scipy** and **numpy**, freely available from www.scipy.org/.

2 Usage

To run MicNeSs, simply type:

```
python2.7 MicNeSs.py inputfile1 inputfile2 inputfile3 ...
```

`python2.7` specifies the version of python. The current implementation does not work with python3.0 but can work with version prior to 2.7. Alternatively, you can edit the source and set the python in the first line (e.g. "#! /usr/bin python2.7")

MicNeSs has two input modes:

- As many `inputfiles` as individuals. Each `inputfile` is a fasta file that includes all sequences for a single individual. The sequences must not be aligned (no gap) but should include the locus of interest. There are as many input files as many individuals to be genotyped. The filename itself is used for formatting the result. MicNeSs assumes that each file is named as **IndividualName_LocusName.ext**. 'IndividualName' is different in each filename since it specifies the individual, 'LocusName' is identical for all inputfiles and 'ext' only refers to the file format (typically fst, fas or fasta). Please note that the filename has no influence on the genotyping and is only used for formatting the output.

- A single `inputfile`. This `inputfile` is a fasta file that include all sequences for all individuals. The sequences must not be aligned (no gap) but should include the locus of interest. MicNeSs assumes that the file is named as **LocusName.ext**, 'ext' only refers to the file format (typically fst, fas or fasta). Importantly each read is

named as > **IndividualName**. Please make sure do not forget the space between the chevron and the individual name.

```
python2.7 MicNeSs.py -h
```

will list all available options

3 Output format

The output reports:

- A header with the date, time and parameters used in the analysis.
- The motif of the microsatellite that is used for genotyping.
- A tab-delimited table (easily imported as a spreadsheet) where each line is:
  1. the name of the locus
  2. the name of the individual
  3. the genotype of the individual (*i.e.* two alleles). Each allele is characterized by its number of repetitions –r– and its number of substitutions –s–. A genotype looks like "(r,s) (r',s')"

4 Options

MicNeSs can be tuned through several different parameters. The default values corresponds to what we think would be the most common usage.

4.1 Verbosity

By default the verbosity is at 0. With this default value, MicNeSs print all that is described in the section "Output File". However, MicNeSs has 3 others level of verbosity that are set using:

```
python2.7 MicNeSs.py -v # inputfile1 ...
```

The first level **-v 1** reports:

- the pool of alleles with their mode (continuous), number of substitutions and the right and left standard deviations.
- The theoretical distribution that is under test and the one that is inserted at each round.

The second level **-v 2** also reports:

- the observed distribution of SSR length for each individual.
- The sequences with an undefined nucleotide (*i.e.* N).

The third level **-v 3** reports a lot of interesting and less interesting stuff.

4.2 SSRs detection

4.2.1 Minimum number of sequences

By default, files with less than 16 sequences are ignored to avoid poorly estimated distributions. Using the default value, we recommend to use at least 30X cover to minimize the number of files with less than 16 sequences (and consequently missing genotypes). However, this minimum number of sequences can be changed using the **-n** option.

```
python2.7 MicNeSs.py -n # inputfile1 ...
```

4.2.2 Minimum number of repeats

By default, we consider only microsatellite with at minimum 4 repeated motifs. This minimum can be changed using the **-l** option.

```
python2.7 MicNeSs.py -l # inputfile1 ...
```

4.2.3 Motif size

By default, mono-nucleotides are ignored; only motifs of size [2,5] are considered. However, mono-nucleotides can be included using the **-1** option.

```
python2.7 MicNeSs.py -1 inputfile1 ...
```

4.2.4 Reverse complement

By default, input files should contains homologous sequences in the same orientation. However, some sequences can be reverse-complemented using the **-r** option. When this option is turned on, all sequence which header line starts with '>R ...' (without space between the chevron and the R) are reverse-complemented before analysis:

```
python2.7 MicNeSs.py -r inputfile1 ...
```

4.3 Genotyping

4.3.1 Substitution(s)

By default, MicNeSs allows for at most 1 substitution. Consequently any SSR with 2 (or more) substitutions is considered as two (or more) overlapping SSRs among which only the longest is considered. This maximum number of substitution(s) can be changed through the **s** option.

```
python2.7 MicNeSs.py -s # inputfile1 ...
```

When this value is set to 0, only strict microsatellite are considered; this may lead to wrongly genotype individuals as homozygous (by pooling different alleles into a single category). On the other hand, a value higher than 1 could capture irrelevant motifs.

4.3.2 Maximum width of the distribution

By default, MicNeSs has an upper limit for right and left standard deviations at 5.0 times the mode of the distribution (a coefficient of variation like). As the fidelity of the polymerase is negatively correlated to the number of repeat, it seemed a natural way to set a maximum width for the length distribution. This maximal width can be changed using the **-b** option:

```
python2.7 MicNeSs.py -b #.# inputfile1 ...
```

Note that the programm is not very sensitive to the choice of this value.

4.3.3 Maximum asymmetry of the distribution

By default, the right and left standard deviations have a maximal ratio of 2.5. This ensures that the distribution cannot be too asymmetric (e.g. spread on one side and abrupt on the other). However, depending on the the type of microsatellites and the sequencing technique, this maximal asymmetry may need to be changed. This value can be changed using the **-a** option.

```
python2.7 MicNeSs.py -a #.# inputfile1 ...
```

The default value give us excellent results with mammal species. However, a smaller ratio ([1,1.3]) seems more adequate for species with shorter microsatellites (e.g. *Drosophila melanogaster* as shown in the manuscript).

5 Examples

We have provided two data sets that can be used to learn the usage of MicNeSs. The first set are *Drosophila* sequences of 1 locus (3r4m3) for 47 individuals. It runs in approximatively 15 seconds on a standard desktop. The option **-r** must be turned on (reverse-complement some sequences) :

```
python2.7 MicNeSs.py -r Drosophila/*3r4m3*
```

The second set are red deer sequences of 1 locus on 47 individuals. It runs in approximately 3 minutes. Again, the option -r must be turned on.

```
python2.7 MicNeSs.py -r RedDeer/*CSSM16*
```

6 Contact

To ask a question on MicNeSs, report a suggestion (e.g. why not including other options) or if you think you have discovered a bug (if any ?), please contact:

Marie Suez at `msuez@abi.snv.jussieu.fr`

7 Citing MicNeSs

M. Suez, A. Behdenna, S. Brouillet, P. Graca, D. Higuet and G. Achaz. MicNeSs: genotyping microsatellite loci from a collection of (NGS) sequences (submitted)