GIbPSs 1.0 Manual Part 15: Documentation of export_seq_03.0.pl

5/22/2015 Andreas Hapke,
Institute of Anthropology, Johannes Gutenberg University Mainz, Germany, ahapke2@gmail.com

This program exports sequence data of all selected loci and individuals. For each selected locus, it produces three files: an alignment of all selected alleles as nonidentical sequences, an alignment with two sequences per individual, and an additional text file. Alignments are in FASTA, Nexus or PHYLIP format. The additional text file complements the alignment of nonidentical sequences with a tab-delimited table. The table contains information about the number of copies of each allele in specific individuals. The program is further able to assign individuals to groups specified in a user file and to include group memberships in all outfiles.

Usage

The program must reside within the main database directory together with the other programs of the GIbPSs package. Before you export sequences with **export_seq**, make a selection of data with **data_selector**. Your selection must fulfill one condition: It may not contain any genotype with more than two alleles. Use this option in **data_selector** to meet the condition: `2 select loci: 7 Numb of alleles in allele-richest ind.: min: 1 max: 2`. Before and/or after you use this option, you may use any further selection criteria offered by **data_selector**.

Call **export_seq** as follows to export sequences of all selected loci in FASTA format:

```
export_seq_03.0.pl
```

or

```
perl export_seq_03.0.pl
```

The following command flags enable you to select another alignment format and to make further modifications of the output. Defaults are given in parentheses.

```
-f     (f) alignment format: f: FASTA, n: Nexus, p: PHYLIP
-mi    (0) 1: include individuals with missing genotypes, 0: don't
-m     (N) character for missing genotypes
-indf ()   name of a user file with group assignments
           or other additional information for each individual
-rs    ()  random number seed
```

Outfiles and output options

Please make a selection of loci with **data_selector** before you call **export_seq** if you do not wish to get thousands of outfiles. The program produces outfiles in directory `export` in the main database directory. Example: You export sequences of locus 1234 in FASTA format. Outfiles have these names: `1234_nid.fas`, `1234_indseq.fas`, `1234_indall.txt`, `export_seq_settings.txt`. The program overwrites the first three files when they already exist. It appends data to the fourth file when it already exists. The extension `.fas` changes to `.nex` when you select Nexus format and to `.phy` when you select PHYLIP format. File `export_seq_settings.txt` contains the settings used by the program including the random number seed and, in case of a problem, error messages.

Example file `1234_nid.fas`

```
>465
ACTTTTGACCGATAG
>932
ACTCTTGACCGATAG
```

There is one sequence for each allele of locus 1234 in your selection of data. IDs are popall_IDs.

Example file `1234_indseq.fas`

```
>ind01_1
ACTCTTGACCGATAG
>ind01_2
ACTTTTGACCGATAG
>ind02_1
ACTCTTGACCGATAG
>ind02_2
ACTCTTGACCGATAG
```

There are two sequences per individual in your selection. IDs consist of the individual ID and `_1` or `_2`.

Random number seed -rs

The program randomly assigns the two alleles of a heterozygous individual to sequences `_1` and `_2`. You must provide the random number seed used by the program with flag `-rs` if you wish to reproduce the same file a second time or if you wish to export the same sequences in another format. You find the random number seed in file `export_seq_settings.txt`.

Example file `1234_indall.txt`

```
poplocID  popall_ID  ind    group   n_copies

1234      465        ind01  group1  1

1234      932        ind01  group1  1

1234      932        ind02  group1  2
```

This file contains a tab-delimited table. For each allele of locus 1234, it lists the individuals that carry this allele and the number of copies. Individual ind02 is homozygous for allele 932. Per default, all individuals are assigned to the same group with name group1. You can use flag `-indf` to assign individuals to different groups with the aid of a user file.

Flag `-indf`: format of the user file:

The file must contain one line per individual with two entries separated by TAB: individual ID and a string. Example:

```
ind01 groupA
ind02 groupB
```

This changes the sequence IDs in file `1234_indseq.fas` to

```
>ind01_groupA_1
>ind01_groupA_2
>ind02_groupB_1
>ind02_groupB_2
```

File `1234_indall.txt` then contains the appropriate groupIDs in column `group`.

Missing genotypes, -mi -m

If you export data from several loci, it can happen that genotypes of some selected individuals are missing at a given locus. They are missing because they have not been genotyped or because you have filtered them away with **data_selector**. Per default, the program does not include these individuals into files `*_indseq.fas`. That means these files may contain different numbers of sequences for different loci. Use `-mi 1` if you wish to have the same number of sequences and identical sequence IDs in all files `*_indseq.fas`. Missing sequences consist of the character specified with `-m`, per default: `N`. Example:

```
>ind01_1
NNNNNNNNNNNNNNN
>ind01_2
NNNNNNNNNNNNNNN
>ind02_1
ACTCTTGACCGATAG
>ind02_2
ACTCTTGACCGATAG
```

Nexus format: -f n example:

```
#NEXUS

Begin data;
      Dimensions ntax=4 nchar=15;
      Format datatype=dna gap=- missing=N;
      Matrix
ind01_1 NNNNNNNNNNNNNNN
ind01_2 NNNNNNNNNNNNNNN
ind02_1 ACTCTTGACCGATAG
ind02_2 ACTCTTGACCGATAG
      ;
End;
```

PHYLIP format: -f p example:

```
 4 15
ind01_1      NNNNNNNNNNNNNNN
ind01_2      NNNNNNNNNNNNNNN
ind02_1      ACTCTTGACCGATAG
ind02_2      ACTCTTGACCGATAG
```

Usually, individual IDs may not have more than 10 characters in this format. **export_seq** pads individual IDs with spaces to a length of 10 or to the length of the longest occurring ID if it is longer than 10 characters. The program does not truncate IDs. Your IDs may be longer than 10 characters, particularly when you use flag `-indf` to add information to IDs. Avoid long IDs when you plan to use a program that requires IDs with a maximum length of 10. Following to the ID, there are three spaces, then the sequence without any spaces. The first line contains the number of sequences and the number of characters in each sequence.

Infiles

The program reads the following infiles. Some of them are optional. The program produces an error message and stops execution when it cannot open a file that it needs.

Infile	produced by
user file provided with flag `-indf`	user
`individuals.txt`	indloc
`popall.txt`	poploc
`*_indpopall.txt`	indpoploc
`*_indpoploc.txt`	indpoploc
`export/sel_ind.txt`	data_selector
`export/sel_loc.txt`	data_selector
`export/sel_gt.txt`	data_selector

*_ stands for an individual ID