User Manual

ExRec: A PYTHON PIPELINE FOR GENERATING RECOMBINATION-FILTERED MULTI-LOCUS DATASETS

Sam McCarthy Potter[1] and W. Bryan Jennings[2]

[1]Department of Biology, Carleton College, Northfield, Minnesota 55057, USA.
Email: smccarthypotter@gmail.com

[2]Department of Evolution, Ecology, & Organismic Biology, University of California, Riverside, California 92521, USA. Email: bryan.jennings@ucr.edu

Table of Contents

S1. About this software package

The ExRec package allows users to easily obtain recombination-filtered datasets that are often required in coalescent-based phylogeography, population genomics, and shallow-scale phylogenomics studies. The package consists of five stand-alone python applications, which form a three-step pipeline (step 3 is optional). In step 1, the user executes a program script on the command line to batch-convert single-locus NEXUS or PHYLIP files into a concatenated partitioned interleaved NEXUS file, which is the standard input file for the package's main program script. In step 2, the user inputs the concatenated NEXUS file into the main program, which performs the following steps in automatic fashion: 1) conducts four-gamete tests (Hudson and Kaplan, 1985) on all loci; 2) truncates apparently recombined loci down to non-recombined sequence blocks (following Hey and Nielsen, 2004); and 3) outputs recombination-filtered NEXUS and PHYLIP files and a tab-delimited summary table that shows descriptive statistics for each locus and the results of the recombination-filtering analyses. The user can select whether to output the longest (following Hey and Nielsen, 2004) or randomly selected (following Hey and Wang, 2019) non-recombined block for each locus that showed evidence of one or more historical recombination events. The concatenated-loci output files, which are in NEXUS and PHYLIP formats, can then be used in popular software programs like *BPP* (Yang, 2015; Flouri *et al.*, 2018). In the optional step 3, two program scripts in the package allow the user to split the recombination-filtered concatenated data files into single-locus files in NEXUS and PHYLIP formats, which can be batch-input into phylogenetics programs thus facilitating summary methods species tree analyses (e.g., *ASTRAL* Mirarab *et al.*, 2014, 2016).

S2. Conventions followed in this user manual

The term "program application" refers to each of the five stand-alone Python 3 programs included in the ExRec package: *Nexcombine.py*, *Phycombine.py*, *FGT.py*, *Nexsplit.py*, and *Physplit.py*. The names of these applications are italicized throughout the document. The user runs each of the five applications from the command line. All command line syntax instructions below are shaded in gray. Datafile names are underlined, and folder names are in bold letters.

S3. Installation procedures

1. Python 3 must be installed on your computer. If your computer does not have Python 3, then you can install it from https://www.python.org/downloads/
2. Download the ExRec package from https://github.com/Sammcarthypotter/ExRec. The package includes: five program applications (i.e., *Nexcombine.py*, *Phycombine.py*, *FGT.py*, *Nexsplit.py*, and *Physplit.py*) and two test data sets (i.e., "finch loci" and "hominoid loci"). These programs can run on Macintosh, PC, or UNIX/LINUX computers.
3. Place the *Nexcombine.py*, *Phycombine.py*, *FGT.py*, *Nexsplit.py*, and *Physplit.py* program scripts into separate folders on your desktop or whichever location on your computer that you choose.

S4. Overview of running the program applications from the command line

1. Place your single-locus NEXUS files into a folder named **nexus_files** or your single-locus PHYLIP files into a folder named **phylip_files**. If you are working with NEXUS files, then place the folder **nexus_files** into the folder that contains *Nexcombine.py*. If instead you are working with PHYLIP files, then place **phylip_files** into the folder that contains *Phycombine.py*. Execute *Nexcombine.py* or *Phycombine.py* by entering the appropriate commands on the command line prompt (explained in section S5). After *Nexcombine.py* or *Phycombine.py* finishes, an output file named combineloci.nex will appear in the folder containing the program application and data folder. This file contains all loci together in a concatenated partitioned interleaved NEXUS file, which is the required input file for *FGT.py*.
2. Next, move the combineloci.nex file to the folder containing the *FGT.py* application. Execute *FGT.py* by entering the appropriate commands on the command line prompt (explained in section S5). When *FGT.py* finishes it run, it will output three files: a recombination-filtered concatenated partitioned interleaved NEXUS file named Trunc_combineloci.nex, a recombination-filtered concatenated PHYLIP file named Trunc_combineloci.txt, and a tab-delimited text file named Results Summary Table.txt.
3. If you would like to split the Trunc_combineloci.nex file into single-locus NEXUS files, then place the Trunc_combineloci.nex file inside the same folder as *Nexsplit.py*. If instead you would like to split the data in Trunc_combineloci.nex into single-locus PHYLIP files, then place the Trunc_combineloci.nex inside the same folder as *Physplit.py*. Execute *Nexsplit.py* or *Physplit.py* by entering the appropriate commands on the command line prompt (explained in section S5). When *Nexsplit.py* finishes, it should output a folder named **nexus_split_files** that contains all single-locus NEXUS files and when *Physplit.py* finishes it should output a folder named **phylip_split_files** that contains all single-locus PHYLIP files.

S5. Detailed instructions for running each program application from the command line

1. The *Nexcombine.py* application

Nexcombine.py converts a collection of single-locus NEXUS files into one concatenated NEXUS file—the standard input file for *FGT.py*. To use *Nexcombine.py*, you must input your data as single-locus NEXUS files in either "sequential" or "interleaved" formats. Below is an example of a single-locus NEXUS sequential file (note: most of the sequences are not shown):

```
#NEXUS

begin data;
        dimensions ntax=4 nchar=577;
        format datatype=dna missing=? gap=-;
        matrix
acuticauda          ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACGGGCACTGCAGGGTAGTTTATTTTGTTCAAATGTAGCGATA
hecki               ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATTTTGTTCAAATGTAGCGACA
cincta              ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATTTTGTTCAAATGTAGCGATA
guttata             ACTTCATGCAGGCCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATTTTGTTCAAATGTAGCGATA
;
end;
```

Example of a single-locus NEXUS interleaved file (bottom of file not shown):

```
#NEXUS
begin data;
        dimensions ntax=4 nchar=1002;
        format datatype=dna missing=? gap=- interleave;
matrix
Gorilla --ATTGCATGGTTATACTGTATTTCTATCATGGCAGAGGTCCCTGGTACAGGAAGATGATTTGATAAACA
Homo    -AATTGCATGGTTATACTATATTTCTATCATGGCAGAGGTCCCTGGTACAGGAAGATGATTTGATAAACA
Pan     -AATTGCATGGTTATACTGTATTTCTATCATGGCAGAGGTCCCTGGTACAGGAAGATGATTTGATAAACA
Pongo   AAATTACATGGCTATACTATATTTCTTTCA-GGCAGAGGTCCCTGGTACAGGAAGATGATTTGATAAACA

Gorilla CTTGTTGAATGAAAAAGCCAGTATGCAGAGGTTTATGAGAAGGAAGCAGTTGCTCACTTAATAATTTTTA
Homo    CTTGTTGAATGAAAAAGCCAGTATGCAGAGGTTTATGAGAAGGAAGCAGTTGCTCACTTAATAATTTTTA
Pan     CTTGTTGAATGAAAAAACCAGTATGCAGAGGTTTATGAGAAGGAAGCAGTTGCTCACTTAATAATTTTTA
Pongo   CTTGTTGAATGAAAAAGCCAGTATGCAGAGGTTTATGAGAAGGAAGCAGTTGCTCACTTAATAATTTTTA

Gorilla AATGCTTTCCCCAAATGAACCTAAGATTAGAAAACAAAGACAAGAATGAAAAGCTCTGAGACATGTAAAA
Homo    AATGCTTTCCCCAAATGAACCTAAGATTAGAAAACAAAGACAAGAATGAAAGGCTCTGAGACATGTAAAA
Pan     AATGCTTTCCCCAAATGAACCTAAGATTAGAAAACAAAGACAAGAATGAAAGGCTCTGAGACATGTAAAA
Pongo   AATGTTTTCCCCAAATGAACCTAAGATTAGAAAACAAAGACAAGAATGAAAGGCTTTGAGACATGTAAAA
```

Running *Nexcombine.py* on the command line

To run *Nexcombine.py*, you must first place the *Nexcombine.py* application and a folder containing the single-locus NEXUS files into the same folder. The folder containing the single-locus NEXUS files must be named **nexus_files** and there cannot be any other files present in the folder. Next, cd to the folder containing *Nexcombine.py* and **nexus_files**. On the command line, enter the following commands to run the program:

```
>python3 Nexcombine.py
```

After the program starts running, it will print the names of all the single-locus NEXUS files on the screen and then list the first and last sites of each locus in brackets, thus giving the length (in bp) of each locus. The output file will be named combineloci.nex, which contains the concatenated partitioned loci in

NEXUS interleaved format. An example combineloci.nex file is shown below (note: only the top part of file is showing):

```
#NEXUS
begin data;
        dimensions ntax=4 nchar=292169;
        format datatype=dna missing=? gap=- interleave;
matrix
Gorilla --ATTGCATGGTTATACTGTATTTCTATCATGGCAGAGGTCCCTGGTACAGGAAGATGATTTGATAAACA
Homo    -AATTGCATGGTTATACTATATTTCTATCATGGCAGAGGTCCCTGGTACAGGAAGATGATTTGATAAACA
Pan     -AATTGCATGGTTATACTGTATTTCTATCATGGCAGAGGTCCCTGGTACAGGAAGATGATTTGATAAACA
Pongo   AAATTACATGGCTATACTATATTTCTTTCA-GGCAGAGGTCCCTGGTACAGGAAGATGATTTGATAAACA

Gorilla CTTGTTGAATGAAAAAGCCAGTATGCAGAGGTTTATGAGAAGGAAGCAGTTGCTCACTTAATAATTTTTA
Homo    CTTGTTGAATGAAAAAGCCAGTATGCAGAGGTTTATGAGAAGGAAGCAGTTGCTCACTTAATAATTTTTA
Pan     CTTGTTGAATGAAAAAACCAGTATGCAGAGGTTTATGAGAAGGAAGCAGTTGCTCACTTAATAATTTTTA
Pongo   CTTGTTGAATGAAAAAGCCAGTATGCAGAGGTTTATGAGAAGGAAGCAGTTGCTCACTTAATAATTTTTA

Gorilla AATGCTTTCCCCAAATGAACCTAAGATTAGAAAACAAAGACAAGAATGAAAAGCTCTGAGACATGTAAAA
Homo    AATGCTTTCCCCAAATGAACCTAAGATTAGAAAACAAAGACAAGAATGAAAGGCTCTGAGACATGTAAAA
Pan     AATGCTTTCCCCAAATGAACCTAAGATTAGAAAACAAAGACAAGAATGAAAGGCTCTGAGACATGTAAAA
Pongo   AATGTTTTCCCCAAATGAACCTAAGATTAGAAAACAAAGACAAGAATGAAAGGCTTTGAGACATGTAAAA
```

You will see that the bottom of the combineloci.nex file contains a "charset block," which lists all loci by their names (within the single quotation marks) and their coordinates within the data matrix:

```
Gorilla CAAGGGCAAAAATCAAAACAGCAATAACACTAAGATTATTAAATTATCAAATGGGGAAAGATGAGAACTT
Homo    CAAGGGCAAAAATCAAAACAGCAATAACACTAAGATTATTAAATTATCAAATGGGGAAAGAAGAGAACTT
Pan     CAAGGGCAAAAATCAAAACAGCAATAACACTAAGATTATTAAATTATCAAATGGGGAAAGAAGAGAACTT
Pongo   CAAGGGCAAAAATCAAAACAGCAATCACCCTAAGATTATTAAATTATCAAATGGGGAAAGAAGAGAACTT

Gorilla TTAAAATACAACAACATAG
Homo    TTAAAATACAACAACATAG
Pan     TTAAAATACAACAACATAG
Pongo   TTAAAATACAACAA-----

;
end;
#nexus
begin sets;
charset '1' = 1-1002;
charset '5' = 1003-2011;
charset '9' = 2012-3003;
charset '37' = 3004-4002;
charset '50' = 4003-5003;
charset '66' = 5004-6006;
charset '84' = 6007-7007;
charset '106' = 7008-7982;
charset '114' = 7983-8982;
```

Before using your combineloci.nex file in *FGT.py*, it is very important to open this file in a text editor and check that it is free of any obvious errors. One simple check that can be done is to open the first two single-locus NEXUS files and compare them to the first two loci in the combineloci.nex file, ensuring that the new file contains the correct "ntaxa," "nchar," and "charset" values and shows full length sequences.

2. The *Phycombine.py* application

Phycombine.py converts a collection of single-locus PHYLIP files into a concatenated NEXUS file named combineloci.nex for input into *FGT.py*. Although all PHYLIP files show the number of sequences and the number of sites at the top of the files followed by the aligned sequences, it is important to understand that many variants of this format exist. PHYLIP files either have sequence names in the "strict" format (i.e., maximum of 10 characters long) or "relaxed" format (i.e., names can be up to 256 characters long) and they can be in either the "sequential" or "interleaved" formats. Accordingly, to ensure that *Phycombine.py* produces an error-free combineloci.nex file, you must use the correct command line syntax depending on which PHYLIP variant files are being input into *Phycombine.py*. Table 1 lists various PHYLIP file types that are compatible with *Phycombine.py* as well as the command line syntax required to process each one.

Table 1. Summary of PHYLIP format variants that *Phycombine.py* can process. Flags can be lower- or upper-case and their order of input on the command line does not affect the analysis. Note, other PHYLIP format variants may work in *Phycombine.py* but you should check output files for errors.

PHYLIP format variant	**required flags**	**command line syntax**
STRICT SEQUENTIAL	none (defaults)	>python3 Phycombine.py
STRICT SEQUENTIAL UPPER	u	>python3 Phycombine.py u
STRICT SEQUENTIAL WRAPPED	w	>python3 Phycombine.py w
STRICT SEQUENTIAL UPPER WRAPPED	u w	>python3 Phycombine.py u w
STRICT INTERLEAVED	none (defaults)	>python3 Phycombine.py
STRICT INTERLEAVED GAPPED	none (defaults)	>python3 Phycombine.py
RELAXED SEQUENTIAL	r	>python3 Phycombine.py r
RELAXED SEQUENTIAL WRAPPED	r w	>python3 Phycombine.py r w
RELAXED INTERLEAVED	r	>python3 Phycombine.py r
RELAXED INTERLEAVED GAPPED	r	>python3 Phycombine.py r

Descriptions of each PHYLIP format variant are provided below. Only the beginning part of each file type is shown.

STRICT SEQUENTIAL: sequence name is in PHYLIP strict name format followed by the sequence in sequential format. The sequence must start at the 11th character space. Required flags: none (defaults)

```
       4      562
hecki      ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACGGGCACTGCAGGGTAGTTTATTTTGTTCAAAT
acuticaudaACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATTTTGTTCAAAT
cincta     ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATTTTGTTCAAAT
guttata    ACTTCATGCAGGCCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATTTTGTTCAAAT
```

STRICT SEQUENTIAL UPPER: sequence name is in PHYLIP strict name format on one line and then the sequence starts at the beginning of the next line in sequential format. Required flags: u

```
       4      562
hecki
ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACGGGCACTGCAGGGTAGTTTATTTTGTTCAAATGT
acuticauda
ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATTTTGTTCAAATGT
cincta
ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATTTTGTTCAAATGT
guttata
ACTTCATGCAGGCCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATTTTGTTCAAATGT
```

STRICT SEQUENTIAL WRAPPED: sequence name is in PHYLIP strict name format followed by the sequence in sequential format. The sequence starts at the 11th character space and then wraps around to the lines below. Required flags: w

```
       4      562
hecki      ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACGGGCACTGCAGGGTAGTTTATT
TTGTTCAAATGTAGCGATAGCACAAACCTGGTTATGGGTCACATTCAAGAGGCAGACAGT
GAATTAGATGGTGGGGGTTAAAAAATGTTATTGAGGATACTTTTATGAGCGAAAAACCCA
CTGAACATTACCCAGGGACCTGGGCAGGCCTGTTGGGCGTGTCATGAGTTCCATTCCAAA
AGTTTGGCAGAAGAAAACAGGCAATAGGTAGCTTCAGAGAAGCAGCCAGTCATCATTTTC
CTCAGGGCATTTGAGCTCCTGGTTCCTCAGGCTGTAGATGAGGGGGTTCAGGGATGGAGG
CACCACCGAGTACAGAACTGACACTGCCAGATCTAGGGATGGGGAGGAGAGGGAGGGGGC
TTTAGGTAGGCAAAGGCTGCGGTGCAGAGAAACAGGGAGAGCACAGCCAGGTGACGGAGG
CAGGTAGAAAAGGCTTTGTGCCGTCTCTGCTCAGAGGGAATCCTCAGCACAGCCCTGAAG
ATCTCCACATAGGAGAAAACCA
acuticaudaACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATT
TTGTTCAAATGTAGCGACAGGACAAACCTGGTTATGGGTCACATTCAAGAGGCAGACAGT
GAATTAGATGGTGGGGGTTAAAAAATGTTATTGAGGATACTTTTATGAGCGAAAAACCCA
CTGAACATTACCCAGGGACCTGGGCAGGCCTGTTGGGCGTGTCATGAGTTCCATTCCAAA
AGTTTGGCAGAAGAAAACAGGCAATAGGTAGCTTCAGAGAAGCAGCCAGTCATCATTTTC
CTCAGGGCATTTGAGCTCCTGGTTCCTCAGGCTATAGATGAGGGGGTTCAGGGATGGAGG
CACCACCGAGTACAGAACTGACACTGCCAGATCTAGGGATGGGGAGGAGAGGGAGGGGGC
TTTAGGTAGGCAAAGGCTGCGGTGCAGAGAAACAGGGAGAGCACAGCCAGGTGACGGAGG
CAGGTAGAAAAGGCTTTGTGCCGTCTCTGCTCAGAGGGAATCCTCAGCACAGCCCTGAAG
ATCTCCACATAGGAGAAAACCA
cincta     ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATT
TTGTTCAAATGTAGCGATAGCACAAACCTGGTTATGGGTCACATTCAAGAGGCAGACAGT
GAATTAGATGGTGGGGGTTAAAAACTGTTATTGAGGATACTTTTATGAGCGAAAAACCCA
CTGAACATTACCCAGGGACCTGGGCAGGCCTGTTGGGCGTGTCATGAGTTCCATTCCAAA
AGTTTGGCAGAAGAAAACAGGCAATAGGTAGCTTCAGAGAAGCAGCCAGTCATCATTTTC
CTCAGGGCATTTGAGCTCCTGGTTTCTCAGGCTGTAGATGAGGGGGTTCAGGGATGGAGG
CACCACCAAGTACAGAACTGACACTGCCAGATCTAGGGATGGGGAGGAGAGGGAGGGGGC
TTTAGGTAGGCAAAGGCTGTGGTGCAGAGAAACAAGGAAAGCACAGCCAGGTGACGGAGG
CAGGTAGAAAAGGCTTTGTCCCGTCTCTGCTCAGAGGGAATCCTCAGCACAGCCCTGAAG
ATCTCCACATAGGAGAAAACCA
```

STRICT SEQUENTIAL UPPER WRAPPED: sequence name is in PHYLIP strict name format on one line and then the sequence starts at the beginning of the next line in sequential format. Sequence wraps around to the lines below. Required flags: u w

```
       4      562
hecki
ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACGGGCACTGCAGGGTAGTTTATT
TTGTTCAAATGTAGCGATAGCACAAACCTGGTTATGGGTCACATTCAAGAGGCAGACAGT
GAATTAGATGGTGGGGGTTAAAAAATGTTATTGAGGATACTTTTATGAGCGAAAAACCCA
CTGAACATTACCCAGGGACCTGGGCAGGCCTGTTGGGCGTGTCATGAGTTCCATTCCAAA
AGTTTGGCAGAAGAAAACAGGCAATAGGTAGCTTCAGAGAAGCAGCCAGTCATCATTTTC
CTCAGGGCATTTGAGCTCCTGGTTCCTCAGGCTGTAGATGAGGGGGTTCAGGGATGGAGG
CACCACCGAGTACAGAACTGACACTGCCAGATCTAGGGATGGGGAGGAGAGGGAGGGGGC
TTTAGGTAGGCAAAGGCTGCGGTGCAGAGAAACAGGGAGAGCACAGCCAGGTGACGGAGG
CAGGTAGAAAAGGCTTTGTGCCGTCTCTGCTCAGAGGGAATCCTCAGCACAGCCCTGAAG
ATCTCCACATAGGAGAAAACCA
acuticauda
ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATT
TTGTTCAAATGTAGCGACAGGACAAACCTGGTTATGGGTCACATTCAAGAGGCAGACAGT
GAATTAGATGGTGGGGGTTAAAAAATGTTATTGAGGATACTTTTATGAGCGAAAAACCCA
CTGAACATTACCCAGGGACCTGGGCAGGCCTGTTGGGCGTGTCATGAGTTCCATTCCAAA
AGTTTGGCAGAAGAAAACAGGCAATAGGTAGCTTCAGAGAAGCAGCCAGTCATCATTTTC
CTCAGGGCATTTGAGCTCCTGGTTCCTCAGGCTATAGATGAGGGGGTTCAGGGATGGAGG
CACCACCGAGTACAGAACTGACACTGCCAGATCTAGGGATGGGGAGGAGAGGGAGGGGGC
TTTAGGTAGGCAAAGGCTGCGGTGCAGAGAAACAGGGAGAGCACAGCCAGGTGACGGAGG
CAGGTAGAAAAGGCTTTGTGCCGTCTCTGCTCAGAGGGAATCCTCAGCACAGCCCTGAAG
ATCTCCACATAGGAGAAAACCA
cincta
ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATT
TTGTTCAAATGTAGCGATAGCACAAACCTGGTTATGGGTCACATTCAAGAGGCAGACAGT
GAATTAGATGGTGGGGGTTAAAAACTGTTATTGAGGATACTTTTATGAGCGAAAAACCCA
CTGAACATTACCCAGGGACCTGGGCAGGCCTGTTGGGCGTGTCATGAGTTCCATTCCAAA
AGTTTGGCAGAAGAAAACAGGCAATAGGTAGCTTCAGAGAAGCAGCCAGTCATCATTTTC
CTCAGGGCATTTGAGCTCCTGGTTTCTCAGGCTGTAGATGAGGGGGTTCAGGGATGGAGG
CACCACCAAGTACAGAACTGACACTGCCAGATCTAGGGATGGGGAGGAGAGGGAGGGGGC
TTTAGGTAGGCAAAGGCTGTGGTGCAGAGAAACAAGGAAAGCACAGCCAGGTGACGGAGG
CAGGTAGAAAAGGCTTTGTCCCGTCTCTGCTCAGAGGGAATCCTCAGCACAGCCCTGAAG
ATCTCCACATAGGAGAAAACCA
guttata
ACTTCATGCAGGCCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATT
TTGTTCAAATGTAGCGATAGTACAAACCTGGTTATGGGTCACATCCAAGAGGCAGACAGT
GAATTAGATGGTGGGGGTTAAAAAATATTATTGAGGATACTTTTATGAGCGAAAAACCCA
CTGAACATTTCCCAGGGACCTGGGCAGGCCTGTTGGGCGTGTCATGAGTTCCATTCCAAA
AGTCTGGCAGAAGAAAACAGGAAATAGGTAGCTTCAGAGAAGCAGCCAGTCATCATTTTC
CTCAGGGCATTTGAGCTCCTGGTTCCTCAGGCTGTAGATGAGGGGGTTCAGGGCTGGAGG
CACCACCGAGTACAGAACTTACACTGCCAGATCCAGGGATGGGAGGAGAGGGAAGGGGGC
```

STRICT INTERLEAVED: sequence name is in PHYLIP strict name format followed by the sequence in interleaved format. The sequence must start at the 11th character space. Required flags: none (defaults)

```
       4      562
hecki       ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACGGGCACTGCAGG
acuticaudaACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGG
cincta      ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGG
guttata     ACTTCATGCAGGCCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGG

            GTAGTTTATTTTGTTCAAATGTAGCGATAGCACAAACCTGGTTATGGGTC
            GTAGTTTATTTTGTTCAAATGTAGCGACAGGACAAACCTGGTTATGGGTC
            GTAGTTTATTTTGTTCAAATGTAGCGATAGCACAAACCTGGTTATGGGTC
            GTAGTTTATTTTGTTCAAATGTAGCGATAGTACAAACCTGGTTATGGGTC

            ACATTCAAGAGGCAGACAGTGAATTAGATGGTGGGGGTTAAAAAATGTTA
            ACATTCAAGAGGCAGACAGTGAATTAGATGGTGGGGGTTAAAAAATGTTA
            ACATTCAAGAGGCAGACAGTGAATTAGATGGTGGGGGTTAAAAACTGTTA
            ACATCCAAGAGGCAGACAGTGAATTAGATGGTGGGGGTTAAAAAATATTA
```

STRICT INTERLEAVED GAPPED: sequence name is in PHYLIP strict name format followed by the sequence in interleaved format. The sequence must start at the 11th character space. The sequences contain a single-character gap after every ten nucleotides. The sequence on each line can be any length. Required flags: none (defaults)

```
       4      562
hecki      ACTTCATGCA GGTCTCAGAC TGGAATCCCA GATAAAACGG GCACTGCAGG
acuticaudaACTTCATGCA GGTCTCAGAC TGGAATCCCA GATAAAACAG GCACTGCAGG
cincta     ACTTCATGCA GGTCTCAGAC TGGAATCCCA GATAAAACAG GCACTGCAGG
guttata    ACTTCATGCA GGCCTCAGAC TGGAATCCCA GATAAAACAG GCACTGCAGG

           GTAGTTTATT TTGTTCAAAT GTAGCGATAG CACAAACCTG GTTATGGGTC
           GTAGTTTATT TTGTTCAAAT GTAGCGACAG GACAAACCTG GTTATGGGTC
           GTAGTTTATT TTGTTCAAAT GTAGCGATAG CACAAACCTG GTTATGGGTC
           GTAGTTTATT TTGTTCAAAT GTAGCGATAG TACAAACCTG GTTATGGGTC

           ACATTCAAGA GGCAGACAGT GAATTAGATG GTGGGGGTTA AAAAATGTTA
           ACATTCAAGA GGCAGACAGT GAATTAGATG GTGGGGGTTA AAAAATGTTA
           ACATTCAAGA GGCAGACAGT GAATTAGATG GTGGGGGTTA AAAACTGTTA
           ACATCCAAGA GGCAGACAGT GAATTAGATG GTGGGGGTTA AAAAATATTA
```

RELAXED SEQUENTIAL: sequence name is in PHYLIP relaxed name format followed by at least one space followed by the sequence in sequential format. Required flags: r

```
       4      562
Poephila_hecki ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACGGGCACTGCAGGGTAGTTTATTTTG
Poephila_acuticauda ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTA
Poephila_cincta ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATTTT
Poephila_guttata ACTTCATGCAGGCCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATTT
```

RELAXED SEQUENTIAL WRAPPED: sequence name is in PHYLIP relaxed name format followed by at least one space followed by the sequence in sequential format. Sequence wraps around to the lines below. Required flags: r w

```
       4      562
Poephila_hecki ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACGGGCACTGCAGGGTAGTTTATT
TTGTTCAAATGTAGCGATAGCACAAACCTGGTTATGGGTCACATTCAAGAGGCAGACAGT
GAATTAGATGGTGGGGGTTAAAAAATGTTATTGAGGATACTTTTATGAGCGAAAAACCCA
CTGAACATTACCCAGGGACCTGGGCAGGCCTGTTGGGCGTGTCATGAGTTCCATTCCAAA
AGTTTGGCAGAAGAAAACAGGCAATAGGTAGCTTCAGAGAAGCAGCCAGTCATCATTTTC
CTCAGGGCATTTGAGCTCCTGGTTCCTCAGGCTGTAGATGAGGGGGTTCAGGGATGGAGG
CACCACCGAGTACAGAACTGACACTGCCAGATCTAGGGATGGGGAGGAGAGGGAGGGGGC
TTTAGGTAGGCAAAGGCTGCGGTGCAGAGAAACAGGGAGAGCACAGCCAGGTGACGGAGG
CAGGTAGAAAAGGCTTTGTGCCGTCTCTGCTCAGAGGGAATCCTCAGCACAGCCCTGAAG
ATCTCCACATAGGAGAAAACCA
Poephila_acuticauda ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATT
TTGTTCAAATGTAGCGACAGGACAAACCTGGTTATGGGTCACATTCAAGAGGCAGACAGT
GAATTAGATGGTGGGGGTTAAAAAATGTTATTGAGGATACTTTTATGAGCGAAAAACCCA
CTGAACATTACCCAGGGACCTGGGCAGGCCTGTTGGGCGTGTCATGAGTTCCATTCCAAA
AGTTTGGCAGAAGAAAACAGGCAATAGGTAGCTTCAGAGAAGCAGCCAGTCATCATTTTC
CTCAGGGCATTTGAGCTCCTGGTTCCTCAGGCTATAGATGAGGGGGTTCAGGGATGGAGG
CACCACCGAGTACAGAACTGACACTGCCAGATCTAGGGATGGGGAGGAGAGGGAGGGGGC
TTTAGGTAGGCAAAGGCTGCGGTGCAGAGAAACAGGGAGAGCACAGCCAGGTGACGGAGG
CAGGTAGAAAAGGCTTTGTGCCGTCTCTGCTCAGAGGGAATCCTCAGCACAGCCCTGAAG
ATCTCCACATAGGAGAAAACCA
Poephila_cincta ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGGGTAGTTTATT
TTGTTCAAATGTAGCGATAGCACAAACCTGGTTATGGGTCACATTCAAGAGGCAGACAGT
GAATTAGATGGTGGGGGTTAAAAAACTGTTATTGAGGATACTTTTATGAGCGAAAAACCCA
CTGAACATTACCCAGGGACCTGGGCAGGCCTGTTGGGCGTGTCATGAGTTCCATTCCAAA
AGTTTGGCAGAAGAAAACAGGCAATAGGTAGCTTCAGAGAAGCAGCCAGTCATCATTTTC
CTCAGGGCATTTGAGCTCCTGGTTTCTCAGGCTGTAGATGAGGGGGTTCAGGGATGGAGG
CACCACCAAGTACAGAACTGACACTGCCAGATCTAGGGATGGGGAGGAGAGGGAGGGGGC
TTTAGGTAGGCAAAGGCTGTGGTGCAGAGAAACAAGGAAAGCACAGCCAGGTGACGGAGG
CAGGTAGAAAAGGCTTTGTCCCGTCTCTGCTCAGAGGGAATCCTCAGCACAGCCCTGAAG
ATCTCCACATAGGAGAAAACCA
```

RELAXED INTERLEAVED: sequence name is in PHYLIP relaxed name format followed by at least one space followed by the sequence in interleaved format. Sequence on each line can be any length (e.g., 50 nucleotides long). Required flags: r

```
       4      562
Poephila_hecki              ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACGGGCACTGCAGG
Poephila_acuticauda         ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGG
Poephila_cincta             ACTTCATGCAGGTCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGG
Poephila_guttata            ACTTCATGCAGGCCTCAGACTGGAATCCCAGATAAAACAGGCACTGCAGG

                            GTAGTTTATTTTGTTCAAATGTAGCGATAGCACAAACCTGGTTATGGGTC
                            GTAGTTTATTTTGTTCAAATGTAGCGACAGGACAAACCTGGTTATGGGTC
                            GTAGTTTATTTTGTTCAAATGTAGCGATAGCACAAACCTGGTTATGGGTC
                            GTAGTTTATTTTGTTCAAATGTAGCGATAGTACAAACCTGGTTATGGGTC

                            ACATTCAAGAGGCAGACAGTGAATTAGATGGTGGGGGTTAAAAAATGTTA
                            ACATTCAAGAGGCAGACAGTGAATTAGATGGTGGGGGTTAAAAAATGTTA
                            ACATTCAAGAGGCAGACAGTGAATTAGATGGTGGGGGTTAAAAACTGTTA
                            ACATCCAAGAGGCAGACAGTGAATTAGATGGTGGGGGTTAAAAAATATTA
```

RELAXED INTERLEAVED GAPPED: sequence name is in PHYLIP relaxed name format followed by at least one space followed by the sequence in interleaved format. The sequences contain a single-character gap after every ten nucleotides. The sequence on each line can be any length. Required flags: r

```
       4      562
Poephila_hecki              ACTTCATGCA GGTCTCAGAC TGGAATCCCA GATAAAACGG GCACTGCAGG
Poephila_acuticauda         ACTTCATGCA GGTCTCAGAC TGGAATCCCA GATAAAACAG GCACTGCAGG
Poephila_cincta             ACTTCATGCA GGTCTCAGAC TGGAATCCCA GATAAAACAG GCACTGCAGG
Poephila_guttata            ACTTCATGCA GGCCTCAGAC TGGAATCCCA GATAAAACAG GCACTGCAGG

                            GTAGTTTATT TTGTTCAAAT GTAGCGATAG CACAAACCTG GTTATGGGTC
                            GTAGTTTATT TTGTTCAAAT GTAGCGACAG GACAAACCTG GTTATGGGTC
                            GTAGTTTATT TTGTTCAAAT GTAGCGATAG CACAAACCTG GTTATGGGTC
                            GTAGTTTATT TTGTTCAAAT GTAGCGATAG TACAAACCTG GTTATGGGTC

                            ACATTCAAGA GGCAGACAGT GAATTAGATG GTGGGGGTTA AAAAATGTTA
                            ACATTCAAGA GGCAGACAGT GAATTAGATG GTGGGGGTTA AAAAATGTTA
                            ACATTCAAGA GGCAGACAGT GAATTAGATG GTGGGGGTTA AAAACTGTTA
                            ACATCCAAGA GGCAGACAGT GAATTAGATG GTGGGGGTTA AAAAATATTA
```

Running *Phycombine.py*

To run *Phycombine.py*, you must first place the *Phycombine.py* application and a folder containing the single-locus PHYLIP files into the same folder. The folder containing the single-locus files must be named **phylip_files** and there cannot be any other files present in this folder. Next, cd to the folder containing *Phycombine.py* and **phylip_files**. On the command line, enter the commands to run the program using the syntax provided in Table 1. After starting *Phycombine.py*, the program will first print the names of the single-locus PHYLIP files to the screen and then it will print the name of the program followed by "STRICT," "STRICT UPPER," "STRICT WRAPPED," "STRICT UPPER WRAPPED," RELAXED, or RELAXED WRAPPED and the flags used to indicate the program settings for the input files. When finished, *Phycombine.py* will output the data in the combineloci.nex file. Before using this newly generated combineloci.nex file in *FGT.py*, it is very important to first open the file in a text editor and check that it is free of any obvious errors. Incorrectly generated combineloci.nex files may lack most or all the sequences, or the sequences may be oddly arranged. You can also easily spot flawed combineloci.nex files if the numbers of sequences and/or sites for each locus are wrong. These flawed files, which in our experience are easy to notice, can be produced when we input the wrong command line syntax for the input files. Thus, it is essential that you use the correct command line syntax when running *Phycombine.py*.

3. The *FGT.py* application

FGT.py automatically carries out the four-gamete filtering procedures on all input loci in the combineloci.nex file and then outputs three files: a recombination-filtered concatenated partitioned interleaved NEXUS file named Trunc_combineloci.nex, a recombination-filtered concatenated PHYLIP file named Trunc_combineloci.txt, and a tab-delimited text file named Results Summary Table.txt. *FGT.py* can perform two different four-gamete filtering analyses: 1) it can output the longest presumably non-recombined sequence block for each apparently recombined locus (current best practice, first suggested by Hey and Nielsen, 2004); or 2) it can output a randomly selected non-recombined sequence block when there are at least two presumably non-recombined blocks for an apparently recombined locus (an approach first suggested by Hey and Wang, 2019).

Option 1: output the longest non-recombined sequence blocks ("longest block mode")

To run *FGT.py* in the mode that outputs the longest non-recombined sequence blocks, you first need to place the combineloci.nex file in the same folder as the *FGT.py* application. Then cd to the folder containing *FGT.py* and combineloci.nex before entering the following commands on the command line:

```
>python3 FGT.py combineloci.nex ? - ms
```

After entering the program name "FGT.py" on the command line, you must then enter a single space followed by the name of the input file, which will usually be combineloci.nex. You can input other file names provided that the file name exactly matches the name of the file. After entering the name of the input file, you must then enter a single space, a question mark, another single space, a hyphen, another single space, and the letters "ms." The "?", "–", and "ms" flags instruct *FGT.py* to carry out the four-gamete tests under the most stringent conditions because most datasets will have missing data at some sites (indicated by "?" in the matrices), alignment gaps (indicated by "–" in the matrices) and will be missing entire sequences for some loci (indicated by sequences of question marks). Thus, the "?" and "–" flags instruct *FGT.py* to ignore sites with missing data and alignment gaps (following Rozas *et al.*, 2017), respectively, while the "ms" flag instructs the program to ignore missing sequences while conducting four-gamete tests. These settings should yield identical four-gamete test results to those obtained using the program *DNAsp* (Rozas *et al.*, 2017).

While *FGT.py* is running, each locus name will appear on the computer screen in the order that it is processed. When *FGT.py* finishes, a timer on the screen will indicate how long the program ran and three files will be output into the *FGT.py* folder: Trunc_combineloci.nex, Trunc_combineloci.txt, and Results Summary Table.txt. The recombination-filtered data are thus contained in the concatenated NEXUS and PHYLIP files Trunc_combineloci.nex and Trunc_combineloci.txt, respectively. You can utilize *Nexsplit.py* or *Physplit.py* in the ExRec package to split the Trunc_combineloci.nex file into its component single-locus NEXUS or PHYLIP files, respectively, for use in batch-mode phylogenetic analyses thereby facilitating summary methods species tree analyses (see below). You can also immediately input the Trunc_combineloci.txt file into species delimitation/historical demography analyses (e.g., the software *BPP*; Yang, 2015; Flouri *et al.*, 2018). Note that because *FGT.py* arranges the sequence/locus identification information for each sample (sequence) in the Trunc_combineloci.txt file using the *BPP* input data file convention (i.e., locus_name^sequence_name), this file can be immediately used in *BPP* analyses without any further modifications. The user will only need to construct an IMAP.txt file to map the sequences to their populations or species names to run in *BPP* (see Yang, 2015).

Results Summary Table.txt is a text file that contains descriptive statistics about each locus and the results of the four-gamete tests/recombination filtering procedures. The table shows for each locus: locus name, starting length (bp), length (bp) excluding gaps/missing data at sites, S (number of segregating sites), list of sites that violate the infinite sites model, R_M (minimum number of recombination events; Hudson and

Kaplan, 1985), pairs of sites that had recombination event(s) within them, sites that define the longest non-recombined block, and the length of the retained longest non-recombined block (bp). You can copy and paste this file into a spreadsheet for further metanalyses of the data (Fig. 1).

	A	B	C	D	E	F	G	H	I	J
1	Locus number	Locus name	starting length (bp)	Length excluding gaps\missing data	S	Infinite sites no	Rm	Locations of recombinantion events	longest block	final length (bp)
2	1	'Pa_01'	577	562	25	81	0	[[0, 0]]	[1, 577]	577
3	2	'Pa_02'	613	611	25	Zero	0	[[0, 0]]	[1, 613]	613
4	3	'Pa_03'	317	300	14	196	0	[[0, 0]]	[1, 317]	317
5	4	'Pa_04'	512	512	8	Zero	0	[[0, 0]]	[1, 512]	512
6	5	'Pa_05'	590	590	20	Zero	0	[[0, 0]]	[1, 590]	590
7	6	'Pa_06'	544	544	10	Zero	0	[[0, 0]]	[1, 544]	544
8	7	'Pa_07'	500	500	23	Zero	0	[[0, 0]]	[1, 500]	500
9	8	'Pa_08'	573	572	20	447	0	[[0, 0]]	[1, 573]	573
10	9	'Pa_09'	659	650	22	Zero	0	[[0, 0]]	[1, 659]	659
11	10	'Pa_11'	681	616	26	Zero	2	[[271, 339], [339, 429]]	[1, 271]	271
12	11	'Pa_12'	797	598	11	Zero	0	[[0, 0]]	[1, 797]	797
13	12	'Pa_13'	520	520	9	Zero	1	[[379, 383]]	[1, 379]	379
14	13	'Pa_15'	468	465	6	Zero	0	[[0, 0]]	[1, 468]	468
15	14	'Pa_16'	481	473	21	Zero	0	[[0, 0]]	[1, 481]	481
16	15	'Pa_17'	216	216	4	Zero	0	[[0, 0]]	[1, 216]	216
17	16	'Pa_18'	620	611	20	Zero	0	[[0, 0]]	[1, 620]	620
18	17	'Pa_19'	608	600	19	Zero	0	[[0, 0]]	[1, 608]	608
19	18	'Pa_20'	339	338	50	21, 46, 169	2	[[168, 212], [212, 326]]	[1, 168]	168
20	19	'Pa_21'	591	590	24	189, 564	0	[[0, 0]]	[1, 591]	591
21	20	'Pa_22'	640	639	22	Zero	0	[[0, 0]]	[1, 640]	640
22	21	'Pa_23'	657	643	10	399	0	[[0, 0]]	[1, 657]	657
23	22	'Pa_24'	561	558	19	Zero	1	[[41, 244]]	[244, 561]	318
24	23	'Pa_25'	618	618	17	Zero	0	[[0, 0]]	[1, 618]	618
25	24	'Pa_26'	540	539	13	Zero	1	[[82, 208]]	[208, 540]	333
26	25	'Pa_27'	405	405	13	Zero	0	[[0, 0]]	[1, 405]	405
27	26	'Pa_29'	657	593	16	Zero	1	[[443, 637]]	[1, 443]	443
28	27	'Pa_30'	571	570	7	Zero	0	[[0, 0]]	[1, 571]	571

Fig. 1. Summary table of descriptive statistics for all loci and results when *FGT.py* is run in longest block mode.

Option 2: output randomly chosen non-recombined sequence blocks ("random block mode")

To run *FGT.py* in the mode that outputs randomly chosen non-recombined sequence blocks, you first need to place the combineloci.nex file in the same folder as the *FGT.py* application. Then cd to the folder containing *FGT.py* and combineloci.nex before entering the following commands on the command line:

```
>python3 FGT.py combineloci.nex ? - ms r
```

Note that you must enter an "r" flag on the command line after you input the file name and any flags that are used. As before, single spaces must separate each flag. When you run *FGT.py* in the random block mode, the summary table will be identical to the table generated under the longest block mode except for the following differences: columns showing the pairs of sites that define presumably non-recombined sequence blocks, the pair of sites that define the randomly selected non-recombined block, and the length (bp) of the randomly selected block (Fig. 2).

	A	B	C	D	E	F	G	H	I	J	K
1	Locus number	Locus name	starting length (bp)	Length excluding gaps\missing data	S	Infinite sites no	Rm	Locations of recombinantion events	Locations without recombinantion events	Selected block	final length (bp)
2	1	'Pa_01'	577	562	25	81	0	[[0, 0]]	[[1, 577]]	[1, 577]	577
3	2	'Pa_02'	613	611	25	Zero	0	[[0, 0]]	[[1, 613]]	[1, 613]	613
4	3	'Pa_03'	317	300	14	196	0	[[0, 0]]	[[1, 317]]	[1, 317]	317
5	4	'Pa_04'	512	512	8	Zero	0	[[0, 0]]	[[1, 512]]	[1, 512]	512
6	5	'Pa_05'	590	590	20	Zero	0	[[0, 0]]	[[1, 590]]	[1, 590]	590
7	6	'Pa_06'	544	544	10	Zero	0	[[0, 0]]	[[1, 544]]	[1, 544]	544
8	7	'Pa_07'	500	500	23	Zero	0	[[0, 0]]	[[1, 500]]	[1, 500]	500
9	8	'Pa_08'	573	572	20	447	0	[[0, 0]]	[[1, 573]]	[1, 573]	573
10	9	'Pa_09'	659	650	22	Zero	0	[[0, 0]]	[[1, 659]]	[1, 659]	659
11	10	'Pa_11'	681	616	26	Zero	2	[[271, 339], [339, 429]]	[[1, 271], [430, 681]]	[1, 271]	271
12	11	'Pa_12'	797	598	11	Zero	0	[[0, 0]]	[[1, 797]]	[1, 797]	797
13	12	'Pa_13'	520	520	9	Zero	1	[[379, 383]]	[[1, 378], [383, 520]]	[383, 520]	138
14	13	'Pa_15'	468	465	6	Zero	0	[[0, 0]]	[[1, 468]]	[1, 468]	468
15	14	'Pa_16'	481	473	21	Zero	0	[[0, 0]]	[[1, 481]]	[1, 481]	481
16	15	'Pa_17'	216	216	4	Zero	0	[[0, 0]]	[[1, 216]]	[1, 216]	216
17	16	'Pa_18'	620	611	20	Zero	0	[[0, 0]]	[[1, 620]]	[1, 620]	620
18	17	'Pa_19'	608	600	19	Zero	0	[[0, 0]]	[[1, 608]]	[1, 608]	608
19	18	'Pa_20'	339	338	50	21, 46, 169	2	[[168, 212], [212, 326]]	[[1, 167], [326, 339]]	[326, 339]	14
20	19	'Pa_21'	591	590	24	189, 564	0	[[0, 0]]	[[1, 591]]	[1, 591]	591
21	20	'Pa_22'	640	639	22	Zero	0	[[0, 0]]	[[1, 640]]	[1, 640]	640
22	21	'Pa_23'	657	643	10	399	0	[[0, 0]]	[[1, 657]]	[1, 657]	657
23	22	'Pa_24'	561	558	19	Zero	1	[[41, 244]]	[[1, 41], [245, 561]]	[1, 41]	41
24	23	'Pa_25'	618	618	17	Zero	0	[[0, 0]]	[[1, 618]]	[1, 618]	618
25	24	'Pa_26'	540	539	13	Zero	1	[[82, 208]]	[[1, 81], [208, 540]]	[208, 540]	333
26	25	'Pa_27'	405	405	13	Zero	0	[[0, 0]]	[[1, 405]]	[1, 405]	405
27	26	'Pa_29'	657	593	16	Zero	1	[[443, 637]]	[[1, 443], [638, 657]]	[1, 443]	443
28	27	'Pa_30'	571	570	7	Zero	0	[[0, 0]]	[[1, 571]]	[1, 571]	571

Fig. 2. Summary table of descriptive statistics for all loci and results when *FGT.py* is run in random block mode.

4. The *Nexsplit.py* application

Nexsplit.py converts the Trunc_combineloci.nex file into single-locus NEXUS files in interleaved format. To use *Nexsplit.py*, you must place the Trunc_combineloci.nex file into the same folder as the *Nexsplit.py* script. After you cd to the folder containing *Nexsplit.py* and Trunc_combineloci.nex, type the following commands on the command line and then hit enter (or return):

```
>python3 Nexsplit.py
```

While the program is running, it will print the name of the input file in brackets on the screen. When finished, *Nexsplit.py* will output single-locus NEXUS files into a new folder named **nexus_split_files**.

5. The *Physplit.py* application

Physplit.py converts the Trunc_combineloci.nex file into single-locus PHYLIP files in relaxed sequential format. To use *Physplit.py*, you must place the Trunc_combineloci.nex file into the same folder as the *Physplit.py* application. After you cd to the folder containing *Physplit.py* and Trunc_combineloci.nex, type the following commands on the command line and then hit enter (or return):

```
>python3 Physplit.py
```

While the program is running, it will print the name of the input file in brackets on the screen. When finished *Physplit.py* will output single-locus PHYLIP files into a new folder named **phylip_split_files**.

S6. Tutorials using example data

USING SINGLE-LOCUS NEXUS FILES AS INPUT DATA TO THE EXREC PIPELINE

In this first tutorial, you will analyze a dataset called "**finch_nexus,**" a folder which contains 27 anonymous DNA sequence loci from the study of Jennings and Edwards (2005). The single-locus files are in NEXUS sequential format.

1. Because each application is run separately and produces unique output files, we suggest that you organize your analyses by placing each of the five applications into new dedicated folders and then name each folder after the application. Here, we will name the one containing *Nexcombine.py* "Nexcombine."

2. Move the **finch_nexus** folder out of the **Sample data** folder and into the new **Nexcombine** folder.

3. Change the name of **finch_nexus** to **nexus_files** so that *Nexcombine.py* will recognize it as the input data folder.

4. Open the command line and then cd to the **Nexcombine** folder.

5. To run *Nexcombine.py*, simply type in the following commands followed by enter (or return):

```
>python3 Nexcombine.py
```

Once you execute the program, you should see the locus lengths (in base pairs) listed down the screen until the program finishes.

6. Now look inside the **Nexcombine** folder for the output file called "combineloci.nex." We suggest you open this file using a text editor so that you can be familiar with its format, which is a partitioned concatenated interleaved NEXUS format. There should be 27 loci in this file, which you can confirm by looking at the bottom of the file where each locus is listed [note: at first there may seem to be 30 loci, but loci Pa_10 and Pa_14 are missing because they only consisted of three sequences – remember that a minimum of four sequences are needed to perform a four-gamete test].

7. Move the combineloci.nex file into the **FGT** folder.

8. On the command line, cd to the **FGT** folder.

9. To run *FGT.py* in "longest block" (default) mode, type in the following commands followed by enter (or return):

```
>python3 FGT.py combineloci.nex ? - ms
```

Note that it is critically important to only use a single character whitespace between each flag otherwise the program may not process the data correctly.

10. Now look inside the **FGT** folder. You should see three output files. Two of them, Trunc_combineloci.nex and Trunc_combineloci.txt files, are the recombination-filtered data in concatenated NEXUS and PHYLIP formats. Using a text editor, open the two output data files to confirm that they appear correct. You can split this NEXUS file into single-locus NEXUS or PHYLIP files using *Nexsplit.py* or *Physplit.py*, respectively (see below). The PHYLIP file is formatted to work in the program *BPP* (Yang, 2015), but you will need to create an IMAP.txt file that links the sequence names to their populations or species (see the *BPP* manual and Yang, 2015). The Results Summary Table.txt file contains a summary of the data and results. You may want to copy and paste the summary table into a spreadsheet where you can perform metanalyses of the data.

11. Now, let's run *FGT.py* in "random block" mode. Before running *FGT.py* again you will need to place the output files from the previous analysis into a new folder so they don't get overwritten by the next analysis. Now, type in the following commands followed by enter (or return):

```
>python3 FGT.py combineloci.nex ? - ms r
```

As before, it is critical to only type a single blank whitespace between each flag.

12. Check in the **FGT** folder to see if the three output files are there and then use the text editor to quickly glance at each file to check that they are correct.

13. To split the recombination-filtered data into single-locus NEXUS files, first move a Trunc_combineloci.nex file from the **FGT** folder to the **Nexsplit** folder. Then type in the following commands followed by enter (or return):

```
>python3 Nexsplit.py
```

14. In the **Nexsplit** folder you should now see the output data in a new folder called **nexus_split_files**. Open this new folder to see that the single-locus NEXUS files are present.

15. To split the recombination-filtered data into single-locus PHYLIP files, first move a Trunc_combineloci.nex file to the **Physplit** folder. Then type in the following commands followed by enter (or return):

```
>python3 Physplit.py
```

16. In the **Physplit** folder you should now see the output data in a new folder called **phylip_split_files**. Open this new folder to see that the single-locus PHYLIP files are present.

USING SINGLE-LOCUS PHYLIP FILES AS INPUT DATA TO THE EXREC PIPELINE

In this second tutorial, you will analyze a dataset called "**hominoid_phylip**" folder, which contains 292 anonymous DNA sequence loci from the study of Costa *et al.* (2016). The single-locus files are in PHYLIP strict sequential upper format.

1. Verify that the *Phycombine.py* application is in a new folder called "**Phycombine**."

2. Move the **hominoid_phylip** folder out of the **Sample data** folder and into the **Phycombine** folder.

3. Change the name of **hominoid_phylip** to **phylip_files** so that *Phycombine.py* will recognize it as the input data folder.

4. Open the command line and then cd to the **Phycombine** folder.

5. Before you can run *Phycombine.py*, you must know the exact format variant of your phylip files so that you can use the correct syntax on the command line. These sample phylip files are in strict sequential upper format (see page 6), and therefore you must include a "u" flag in the commands (Table 1). Thus, to run *Phycombine.py*, type in the following commands followed by enter (return):

```
>python3 Phycombine.py u
```

Again, it is important that you only type a single blank whitespace to separate the name of the application from the u flag.

6. Now look inside the **Phycombine** folder for the output file called "combineloci.nex." We suggest you open this file using a text editor so that you can be familiar with its format, which is a partitioned concatenated interleaved NEXUS format. There should be 292 loci in this file, which you can confirm by looking at the bottom of the file where each locus is listed.

Steps 7-16 are the same as for the first tutorial.

S7. References

Costa,I.R. *et al.* (2016) In silico phylogenomics using complete genomes: a case study on the evolution of hominoids. *Genome Res.*, **26**, 1257–1267.

Flouri,T. *et al.* (2018) Species tree inference with BPP using genomic sequences and the multispecies coalescent. *Molecular Biology and Evolution*, **35**, 2585–2593.

Hey,J. and Nielsen,R. (2004) Multilocus methods for estimating population sizes, migration rates and divergence time, with applications to the divergence of *Drosophila pseudoobscura* and *D. persimilis*. *Genetics*, **167**, 747–760.

Hey,J. and Wang,K. (2019) The effect of undetected recombination on genealogy sampling and inference under an *isolation-with-migration* model. *Mol Ecol Resour*, **19**, 1593–1609.

Hudson,R.R. and Kaplan,N.L. (1985) Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics*, **111**, 147–164.

Jennings,W.B. and Edwards,S.V. (2005) Speciational history of Australian Grass Finches (*Poephila*) inferred from thirty gene trees. *Evolution*, **59**, 2033–2047.

Mirarab,S. *et al.* (2014) ASTRAL: genome-scale coalescent-based species tree estimation. *Bioinformatics*, **30**, i541–i548.

Mirarab,S. *et al.* (2016) Evaluating summary methods for multilocus species tree estimation in the presence of incomplete lineage sorting. *Syst Biol*, **65**, 366–380.

Rozas,J. *et al.* (2017) DnaSP 6: DNA sequence polymorphism analysis of large data sets. *Molecular Biology and Evolution*, **34**, 3299–3302.

Yang,Z. (2015) The BPP program for species tree estimation and species delimitation. *Current Zoology*, **61**, 854–865.