Running hapferret

1) At present (5/2020) hapferret runs on a Mac (OS X, probably any version) in a terminal window.

2) At present (5/2020) hapferret takes no command line input. The names of the input and output files are fixed; to run it you need to create a folder and put the input files—and a copy of ferret—there. The output files will be created there also. So the strategy is to either create a new folder for each run—easy enough on a Mac—or to keep one folder but replace the input and move or rename the output after each run.

3) **Running**
   Open a terminal window, navigate to the folder containing ferret and the input files (just type "cd", space, and drag the folder to the terminal command line), and type ./hapferret. The required input files are: files.data.txt, hap_search_settings.txt, and the input data files specified in files.data.txt.

4) **Specifying input files**
   Input files and their formats are specified in a file with default (currently required) name "files_data.txt", which must be in the directory from which hapferret is run.

   Sample files_data.txt:

   wide format (all genotypes for an individual on one line)

   ```
   fileformat c
   genotype_file hla_ceu_haps_14long.txt
   ```

   Each line starts with an identifier, as shown. The two lines give the file format (here "c" for comma, specifying allele,allele format for each genotype), then the genotype file name.

   long format (one genotype—one locus for one individual—on each line)

   ```
   fileformat s
   genotype_file c22mge_ceu.txt
   var_info_file c22mge_ceu.i.txt
   ```

   Again each line starts with an identifier. The three lines give the file format ("s" stands for scan input) then the genotype file name, then the associated locus info file.

5) File format

HapFerret accepts two input formats (“wide” and “long” in database terminology).

Wide format has all genotype information for one individual on one line. Each line starts with a subject identifier, and has one genotype entry for each locus. This file must have a title line; the first entry, e.g. “PID”, identifies the column of subject IDs and is ignored. This is followed by the names of each locus. This format is specified by fileformat “c” in the files_data.txt file. Example:

```
PID rs136160 rs136161 rs713753 rs4419330 rs4350853 rs136168 rs2239785
UX19193     G,C     G,G     C,C     T,C     T,T     A,A     A,G
UX18501     G,C     C,G     T,C     T,T     T,T     A,G     A,G
UX19093     G,C     C,G     C,C     T,C     T,T     A,A     A,G
UX19209     G,G     G,G     C,C     T,T     T,T     A,G     A,G
UX19144     G,C     C,G     C,C     T,T     T,T     A,G     G,G
UX19222     G,G     C,G     T,C     T,T     T,T     A,G     A,G
UX19193     G,C     G,G     C,C     T,T     T,T     A,G     A,G
UX19101     C,C     G,G     C,C     T,T     T,T     G,G     G,G
UX19101     C,C     G,G     C,C     T,C     T,T     G,A     G,G
UX19127     C,C     G,G     C,C     T,T     T,T     G,G     G,G
UX19140     C,C     G,G     C,C     T,T     T,T     G,G     G,G
UX19209     G,C     G,G     C,C     T,C     T,T     A,G     A,G
UX18522     C,G     G,G     C,C     T,C     T,G     G,A     G,A
UX19138     C,C     G,G     C,C     T,T     T,T     G,G     G,G
```

These are standard .txt files, with entries divided by white space, one or more spaces or tabs. Note that the loci are given by the standard (rs #) name, and the genotypes are allele,allele, with the allele given as the actual base. The locus and the allele can be identified by any alphanumeric (without spaces, and without spaces between the allele names and the comma); e.g. an example with loci from the HLA region:

PID	DQA1	DQB1CA	DQB1	G51152	G496	TAP2	42154
simsub_1	501,501	4,14	301,201	15,14	13,5	3.2,4.2	3.3,1.2
simsub_2	102,201	12,4	602,303	11,15	13,13	4.2,4.2	2.1,3.3
simsub_3	201,103	3,12	201,603	15,11	6,19	4.2,3.2	3.3,3.3
simsub_4	201,101	7,10	201,503	15,13	5,6	4.2,4.2	1.2,3.3
simsub_5	103,101	12,12	603,501	11,3	13,13	4.2,3.2	2.1,2.2
simsub_6	101,301	12,10	501,302	3,10	6,14	3.2,4.2	2.2,1.2
simsub_7	301,301	5,10	301,302	15,10	13,14	4.1,4.2	2.1,3.3
simsub_8	301,101	8,10	302,503	10,13	13,6	4.2,3.2	3.2,3.3
simsub_9	301,201	10,7	302,201	10,15	13,5	4.2,4.2	1.2,3.3
simsub_10	501,102	3,12	301,602	15,11	13,14	4.2,4.2	2.1,3.3
simsub_11	201,101	5,12	201,501	15,3	5,13	3.2,3.2	2.1,2.2
simsub_12	101,102	12,12	501,602	3,11	13,19	3.2,4.2	2.2,3.3
simsub_13	201,102	3,12	201,604	15,12	5,14	4.2,4.2	3.3,3.3
simsub_14	103,501	12,14	603,201	11,14	19,19	3.2,4.2	3.3,2.1
simsub_15	201,501	7,5	201,301	15,15	13,3	3.2,4.2	3.3,3.2
simsub_16	101,101	12,12	501,501	3,3	13,6	4.1,3.2	3.3,2.2
simsub_17	103,501	12,2	603,301	11,15	19,13	3.2,4.1	3.3,4.2
simsub_18	501,301	14,8	201,302	14,10	19,13	4.2,4.1	2.1,4.2
simsub_19	301,102	8,12	302,602	10,11	14,14	4.2,4.2	2.1,3.3

Long format

Each line is one genotype: PID, race, polymorphism identifier, and the two alleles seen. Entries are separated by white space;here the two alleles are separated by white space and not by a comma. This is specified by fileformat "c" in the files_data.txt file. Example files_data file:

```
fileformat s
genotype_file c22mge_ceu.txt
var_info_file c22mge_ceu.i.txt
```

The third line, var_info_file, specifies the file with locus information

```
44O00768   1   rs3752462   C   C
44O00769   1   rs3752462   C   T
44O00774   1   rs3752462   T   T
44O00779   1   rs3752462   C   C
44O00781   1   rs3752462   C   T
44O00782   1   rs3752462   C   C
44O00785   1   rs3752462   C   T
44O00787   1   rs3752462   C   C
44O00793   1   rs3752462   C   T
44O00794   1   rs3752462   C   T
44O00797   1   rs3752462   C   T
44O00803   1   rs3752462   T   T
44O00813   1   rs3752462   C   T
44O00825   1   rs3752462   C   T
```

```
44O000837   1   rs3752462   C   T
44O000840   1   rs3752462   C   T
44O000841   1   rs3752462   C   T
44O000844   1   rs3752462   C   C
44O000846   1   rs3752462   C   T
44O000849   1   rs3752462   C   T
44O000852   1   rs3752462   T   T
44O000853   1   rs3752462   C   C
44O000855   1   rs3752462   C   C
44O000859   1   rs3752462   C   C
44O000860   1   rs3752462   C   C
44O000862   1   rs3752462   C   C
44O000868   1   rs3752462   C   C
44O000869   1   rs3752462   C   C
44O000872   1   rs3752462   C   T
```

Each line is one genotype: PID, race, polymorphism identifier, and the two alleles seen; entries separated by white space—here the two alleles are separated by white space and not by a comma.

Long format requires a locus info file, with each line giving a polymorphism id, followed by its genome coordinates (no chromosome info is used in the current version). The current algorithm needs this information to know the order of the loci, but doesn't use the actual position for the calculation.

```
rs1557529 35035474
rs2157256 35037606
rs2413396 35038030
rs5750250 35038428
rs3830104 35038569
rs4820229 35038699
rs3752462 35040128
rs8141971 35041308
rs5756152 35042417
rs9610489 35043476
rs2239784 35044580
rs1005570 35045219
rs12159211 35049108
rs8136336 35052479
rs16996672 35055916
rs16996677 35057228
rs11704382 35058098
rs4820234 35059020
```

6) Specifying parameters for haplotype inference

A file is also required specifying settings for the hap inference run, this has the (currently fixed) name "hap_search_settings.txt". Example content:

```
accept_params 1
accept_params 1
mode 1
blocksequence 0
race 1
disease_data 0
target_delta 0.00003
max_iterations 1000
full_hap_call 1
subseq_hap_call 0
max_subblock = 30
calc_hap_call_entropy 1
n_bootstrap_reps 0
```

These need not be in order.   The parameters input:

-accept_params (1 or 0) Should these parameters be used as input (1), or should the user queried to enter possible changes

-**mode** 1 (keep this setting, ignore this)

-blocksequence 0 (keep this setting, ignore this)

-**race** Used if genotype input file is long format, then this specifies which race—identifier in 2nd column of file—to use. Ignored in wide format.

-**disease_data** (0 or 1) 1 if there is a disease data file; not documentation yet

-**target_delta** The EM algorithm iterates and sums the frequency difference between iterations for each haplotype, stopping when this difference is less than this parameter

-max_iterations Stop the inference at this count of iterations even if target_delta hasn't been reached

-full_hap_call Infer haplotypes for the complete (ordered) set of loci in the input file

-subseq_hap_call Infer haplotypes for subsequences of the input loci. hapferret runs along the set of loci, inferring haplotypes first for all sequences of two (contiguous) loci, then for sequences of 3, 4, etc. loci, to the limit given by max_subblock

-max_subblock Maximum length of sequences of loci to infer in subseq_hap_call.

-calc_hap_call_entropy Calculated entropy—uncertainty—of the inference?

-n_bootstrap_reps Number of bootstrap replications, set to 0 for no bootstrapping—keep this setting for now.