SHAZA User Manual

Shadow Zone Analysis

Version 2.00 July 2011

G.M. Macbeth[1]*, D Broderick[1],

J.R. Ovenden[1], R.C Buckworth[2]

[1]*Molecular Fisheries Laboratory, Queensland Primary Industries and Fisheries, Ritchie Building No. 64A, Research Road, University of Queensland, PO Box 6097, St Lucia, Queensland, 4072, Australia.*
[2]*Department of Regional Development, Primary Industry, Fisheries and Resources, PO Box 3000, Darwin, Northern Territory, 0801, Australia.*

***Corresponding author:**
Michael Macbeth
Ph: +61 7 33466518; E-mail: Michael.Macbeth@deedi.qld.gov.au
Current address: Centre for Applications in Natural Resource Mathematics, School of Mathematics and Physics, The University of Queensland, Queensland, 4072 Australia.

SHAZA User Manual

TABLE OF CONTENTS

1.0 QUICK START

1.1 Installation

Download and unzip a pre-compiled version of SHAZA for your computer platform from http://www.dpi.qld.gov.au/28_6899.htm

The quick start example assumes the **shaza** executable is in a Windows XP working directory called "c:\SHAZA". For more detailed instructions refer to section 3.1.

1.2 Data preparation

Most analysis will only require one input file which contains specimen identification and their genotypes. A GENEPOP file can be used. For more information see section '3.2.1 Multi-locus genotype data.

1.3 Run program

Open a command line window
eg. In Windows XP click '**start**', move cursor to the '**All programs**' tab, go to the '**Accessories**' folder and click on '**Command prompt**'. Within the command window go to the SHAZA directory eg:

cd c:\SHAZA <ENTER>

To analyse matching genotypes using model 1 you could use these command arguments:

shaza –1 FILE

Note there is a space before and after the '–1' argument.
'–1 FILE' defines the genotype data file *e.g.* FILE="your_base_pair_file.dat"
The command line would run model 1 with default settings. In this case shaza will run one or more iterations to find the best estimate of recapture numbers.

shaza.exe –1 your_base_pair_file. dat <ENTER>

1.4 Output files

Model 1 generates seven files:

SHAZA_log.txt	is a dump of screen output.
SHAZA_freq.txt	lists allele frequencies found in the input file
SHAZA_loci.txt	condensed allele frequency report suitable for input back into SHAZA
SHAZA_miss.txt	lists the frequency of missing loci combinations and a loci mismatch table
SHAZA_stat.txt	summary of convergence statistics in all iterations
SHAZA_pop.txt	lists corrected recapture estimates within and between populations
SHAZA_pair.txt	lists matching genotype pairs against cumulative number of Type I errors (false positive matches)

Perhaps the two files of most interest are SHAZA_pair.txt and SHAZA_pop.txt which list only the results for the recapture solution with the smallest standard error.

Table 1 in file SHAZA_pop.txt lists the estimate(s) of the number of corrected recaptures from the reference data defined from the '–1 FILE'. The following Table 2 in the same file lists the simulated population results using the recapture estimate from Table 1 as input. The corrected recaptures in Table 2 should be similar to those estimated in Table 1 with the standard deviation of the simulated recapture estimates used as an estimate of the standard error of corrected recaptures in the reference data.

GENERAL INFORMATION

2.1 System overview

SHAZA is a statistical package suitable for the analysis of matching genotypes. Version 2.00 replaces verions 1.00 with the new output describing mathematical terms using a similar notation to that published in (Macbeth *et. al*. 2011). Many other changes were made including an improved algorithm for estimating effective sample size. The program is based on ranking the likelihoods of individual matches against the cumulative estimation of false positives including (those identical matches occurring by chance called shadows). SHAZA was specifically designed to detect low frequency genotype matches from a large number of samples in an outbred population. In populations with related samples some bias may occur (Macbeth *et. al.* 2011).

The target users of this program would typically be researchers in wildlife studies who use genotype data (microsatellites or SNPs) to identify animals. SHAZA is suitable for estimating recaptures in a range of applications, including mark-release-recapture studies where animals are tagged using genotypes in:

(i) data that is low in statistical power *i.e.* when Type I (False positive) and Type II (False negative) errors are present.
(ii) data that is high in statistical power with genotype errors prevalent
(iii) data that has loci missing in many samples (*e.g.* through degraded or non-invasive sampling).

The unique ability of SHAZA to correct for both Type I and Type II errors leads to more accurate estimates of recaptures, which in turn improve the accuracy of population parameters from a mark-recapture study (*e.g.* Population size).

In forensic applications SHAZA is suitable for finding genotype matches and suspect matches from poor quality data containing missing loci and genotyping errors. This may be typical in crime scenes where DNA may be in small quantities and partially degraded.

The built in simulation options of SHAZA make it a powerful statistical tool for testing experimental design and robustness of model assumptions. For example, simulations demonstrated the robustness of SHAZA in estimating recapture numbers with data having low statistical power (e.g. too few loci and or too few alleles per loci) and with different error rates (Macbeth *et. al.* 2011). SHAZA could be used to test the relative cost of sample numbers versus cost per loci genotyped to optimise experimental objectives.

2.2 General theory

2.2.1 Background number of false positives

SHAZA determines matches using Log Likelihood Ratio's (*LLR*) with the mathematical derivations fully described in (Macbeth *et. al.* 2011). In this program we recognise that all genotype matching is based on probability. To find a match between two samples we must accept a certain level of Type I errors (false positives) that will occur by random chance (Fig. 1). If we accept an infinitely small number of Type I errors then we will have to reject all genotype comparisons as each pairwise comparison will not have sufficient power to be absolutely 100% certain that any match found will not be a false positive match. Assuming an outbred population with random genotypes, the relationship between *LLR* and the background level of Type I errors can be determined through simulation. Using this relationship the cumulative number of expected Type I errors for every pairwise match in field data (or simulated reference data) can be determined.

2.2.2 Corrected recaptures

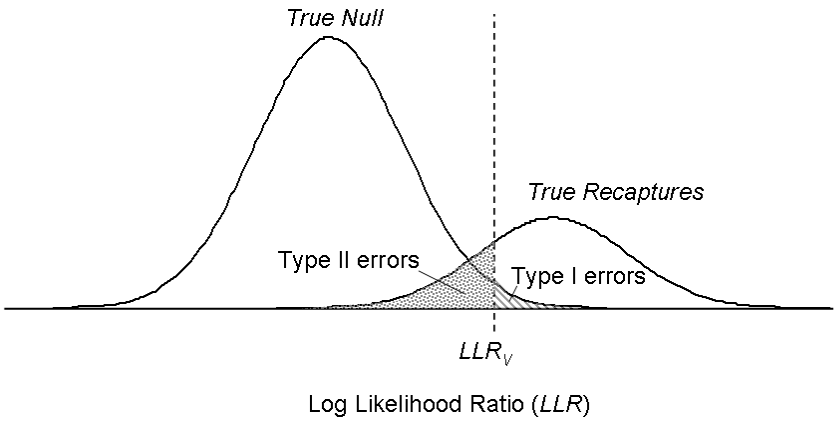
In some wildlife applications it may be desirable to determine the best estimate of total recaptures from a given dataset. Figure 1 illustrates that for a given background Log Likelihood Ratio (*LLR*) there is a certain proportion of true recaptures that are either accepted or rejected. SHAZA calculates different recapture estimates by implementing a multiple hypothesis testing model where the sum of Type I errors (*V*) are called false positives and the sum of Type II errors (*T*) are called false negatives. A log likelihood threshold (*LLRv*) is determined by simulation for a given number of *V* false positives (Figure 1). Simulations are then

used to determine an estimator of *V* which will find the corrected number of recaptures with the smallest standard error using the formulae:

Corrected recaptures = Matches + Type II errors – Type I errors

In applying this formulae we developed an intuitive way of estimating Type II errors which would otherwise be difficult to determine with precision (Macbeth *et. al.* 2011). A convergence algorithm is used to determine the estimator of corrected recaptures with the smallest standard error.

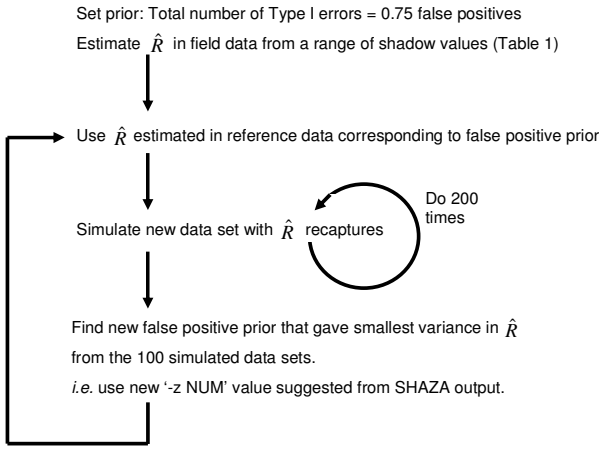
Figure 1. Hypothesis testing model showing the distribution of Type I and Type II errors in the presence of true recaptures and true null recaptures at a given Log Likelihood Ratio threshold.



2.2.3 Convergence

The convergence algorithm attempts to find the corrected number of recaptures ($\hat{R}$) with the smallest standard error. The convergence process is illustrated in Figure 2. Using a *V* prior $\hat{R}$ is determined from the reference data. Using $\hat{R}$ as input simulated populations are created which are used to find the standard error over a range of *V* values. The *V* value that gave the smallest standard error in the simulated population is used to find a new $\hat{R}$ from the reference data with the iteration process starting again. In most cases only two or three iteration runs may be required to achieve convergence with more iterations required in large data sets with low power (e.g. data having few loci with few alleles).

Figure 2. Convergence flowchart.



To estimate the total corrected recaptures ($\hat{R}$) in a dataset a prior starting value for *V* 'the total number of false positives' (Figure 1) is used which is defined by the '-z NUM' option on the command line where NUM is an integer chosen from Table 1. The default prior of *V* =0.75 false positives can be changed using this

option on the command line. For the defined number of false positives the best estimate of total recaptures may not be obtained and a message like:
"Possibly try another run with '-z 4' (V=0.250 False positives)" may be listed.

By default the process above is automated and the user is not required to enter '-z NUM' on the command line. By default SHAZA initiates a series of iteration runs to estimate the total number of recaptures in a population with the smallest standard error. Each iteration has by default 100 randomly generated genotype populations that emulate the real data set. This default can be increased using the '-s NUM' option where NUM is the number of simulated populations. It is possible to avoid these iterations by manually specifying the number of iterations to zero using '–c 0' on the command line and estimate recaptures for a given number of false positives given by '-z NUM' (Table 1). Setting these two options will reduce execution time but may not provide the best estimate of recaptures in the population and will not estimate the standard error of recapture estimates. To determine the standard error for a given number of false positives use: '-z NUM -c 1' on the command line.

Table 1. Expected number of false positives (*V*) associated with each '-z NUM' value.

NUM	False positives (*V*)	
0	0.001	'-z 0' *permitted with* '-*n* 1000' *option see: section* 6.0 *Example* 3.
1	0.010	
2	0.050	
3	0.125	
4	0.250	←
5	0.500	
6	0.750	default
7	1.000	
8	1.250	
9	1.500	
10	2.000	
11	2.500	
12	3.000	
13	3.500	
14	4.000	
15	4.500	
16	5.000	
.	.	} *0.500 increments*
.	.	
.	.	
48	21.000	
49	21.500	
50	22.000	

Note if convergence occurs at 0.010 false positives it may be best to re-run shaza with '–s 1000' and '–n 1000' options on the command line which may reduce the standard error of recapture estimates if the re-run converged at a lower level of ***V***=0.001 false positives.

In summary-

Automated convergence (recommended) use:	shaza -1 base_pair-file.dat
No convergence with default V=0.75 false positives use:	shaza -1 base_pair-file.dat -c 0
Solution at given number of false positives (Table 1) use:	shaza -1 base_pair-file.dat –z NUM -c 1
If converged at **V**=0.010 false positives try:	shaza -1 base_pair_file.dat –s 1000 –n 1000 -z 0

2.3 Where to download SHAZA

SHAZA is freely available for use in non-commercial applications. SHAZA is easy to use and simple to install. The standard compiled version works with up to 10,000 genotype samples and is available for Windows-XP, Macintosh OS X (tested on Leopard 10.5.x) and Linux (tested on kernel 2.6.xx). The executable codes and example files are available through: http://www.dpi.qld.gov.au/28_6899.htm

2.4 Compilation

SHAZA is written in ANSI C and should compile with little modification in any platform with a C compiler. There are over ninety *.c source files including:

main.c	main program initiating command line interpreter and SHAZA engine
command.c	command line interpreter
command.h	include file for command line interpreter
shaza.c	SHAZA engine
shaza.h	include file for SHAZA engine

A **Makefile** is included to assist compilation. We have tested our compilation using the **gcc** compiler and the **make** command on Windows, Linux and Macintosh systems. On Linux and Macintosh systems the **make** command creates an executable file called **shaza** and on the Windows platform an executable file called **shaza.exe** is created. The installation of compilers is outside the scope of this reference manual.

2.5 System testing

SHAZA has been extensively tested under a wide range of simulated parameters including sample size, genotyping errors, presence of partial genotypes and number of loci in genotyping panel (Macbeth *et. al.* 2011).

2.6 Reporting bugs

If you think you have found a bug in SHAZA we would like to know. To assist us please describe how the bug could be demonstrated and provide an example input file(s), the SHAZA_log.txt file, the computer platform used as well as any other output files that demonstrate the error.

2.7 Future research and development

Possible areas of future research and development include:

- determine the relationship between *LLR* and Type II errors at given inbreeding rates
- allow different error rates to be defined for each locus and emulate allelic dropouts
- develop SHAZA methodology to estimate kinship matches *e.g.* parent – offspring
- develop graphic user interface

2.8 Random number generation

Many random numbers are called when simulating genotypes and as such it is important to avoid any pattern from pseudo-random number generation. SHAZA compiles a random number generator kindly provided by Takuji Nishimura and Makoto Matsumoto (Appendix 1).

2.9 Disclaimer

3.0 GETTING STARTED

3.1 Installation

SHAZA has been compiled on Windows XP, Linux (kernel 2.6) and Macintosh (OS-X). Only one of these three executable files will work with your computer. The executable name in Windows XP, Linux and Macintosh versions are called **shaza.exe, shaza** and **shaza** respectively.

There are three example files included in the distribution. These are '**fish_example.dat**' which contains microsatellite alleles from narrow-barred Spanish mackerel (*Scomberomorus commerson*) and two allele frequency data sets, '**frequency_one.dat**' and '**frequency_two.dat**'.

Download the zip file for your computer platform from http://www.dpi.qld.gov.au/28_6899.htm

Windows XP
Double click on the zip file and click on "Extract all files". Install in a directory like c:\shaza.
To be ready to run shaza.exe it is necessary to open up a command window. In XP click '**start**', move cursor to the '**All programs'** tab, go to the **'Accessories'** folder and click on '**Command prompt'**.
Go to the SHAZA directory e.g. **cd c:\shaza** <ENTER>
As an option you can edit the PATH environment variable so that **shaza** can run in any directory.

Linux and Macintosh
Open a terminal window and go to your home directory using: **cd ~** <ENTER>
Extract the zip files: **unzip shaza_*_100beta.zip** <ENTER>
and go into this directory using: **cd shaza** <ENTER>
As an option, if you have access to root privileges, login as root and copy the executable 'shaza' to a $PATH directory such as /usr/local/bin or /usr/bin to make the program run from any directory. The $PATH directories can be identified with the command: **echo $PATH** <ENTER>
Finally make sure the program is executable using: **chmod +x shaza** <ENTER>

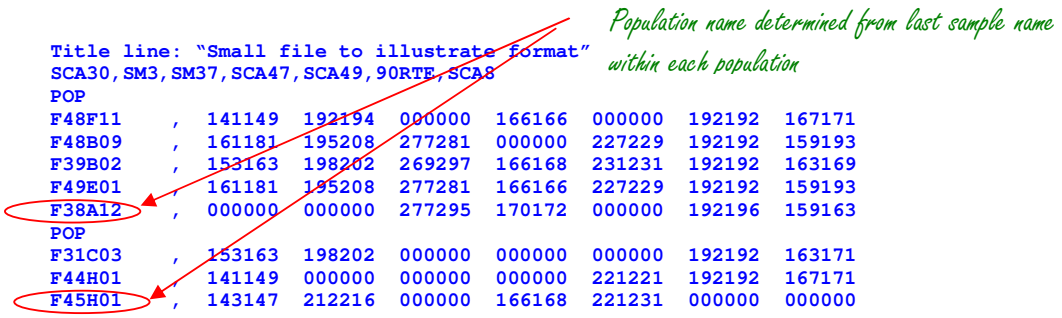
That's it! Installation is complete and you are ready to run your first shaza program.

3.2 Data input formats

3.2.1 Multi-locus genotype data

Alleles are read on the command line with the '-1 FILE' declaration where FILE is the name of the genotype data file such as "myBasePairData.txt". This file can be read in two different formats (i) GENEPOP and (ii) SHAZA spreadsheet format. The shaza program automatically detects which of the two file formats are used but they must conform to the format listed below:

(i) GENEPOP format is described in more detail in http://genepop.curtin.edu.au/help_input.html
SHAZA accepts the population separator "POP" by default and is case sensitive. Data can be subdivided into 1..100 populations by using "POP" in the first column. Missing alleles are denoted as '000'.



(ii) SHAZA csv format

This format is designed for easy conversion from data stored in spreadsheet columns written to comma separated (csv) or space separated data.

The following table shows data in Microsoft XL prior to converting to csv format:

Must have a word in this cell

Loci names for each allele pair must be identical as this is used to determine format type (i) *or* (ii)

Sample-ID	SCA30	SCA30	SM3	SM3	SM37	SM37	SCA47	SCA47	SCA49	SCA49	90RTE	90RTE	SCA8	SCA8
POP	GroupA													
F48F11	141	149	192	194			166	166			192	192	167	171
F48B09	161	181	195	208	277	281			227	229	192	192	159	193
F39B02	153	163	198	202	269	297	166	168	231	231	192	192	163	169
F49E01	161	181	195	208	277	281	166	166	227	229	192	192	159	193
F38A12					277	295	170	172			192	196	159	163
POP	GroupB													
F31C03	153	163	198	202							192	192	163	171
F44H01	141	149							221	221	192	192	167	171
F45H01	143	147	212	216			166	168	221	231				

In this format all strings within each cell contain no spaces between characters. The first line is the header line which indicates the sample identification column “Sample-ID” and the duplicate locus names of the other columns. The second line always has “POP” in the first column and an optional population name in the second column. The third line identifies the sample name (first string) followed by the alleles for each locus with this format repeated for each genotype sample. Missing alleles have no characters or can be defined as zero ”0”. To convert to csv format in Microsoft Excel, open the spreadsheet, click the ‘File’ tab, click the ‘Save as…’ tab , then at the bottom of the ‘Save As’ popup window find the ‘Save type as:’ option and select ‘CSV(Comma delimited)(*.csv)’ from the options given. You can then select a different file name before clicking the ‘Save’ button. The format is illustrated below in csv format:

```
Sample-ID,SCA30,SCA30,SM3,SM3,SM37,SM37,SCA47,SCA47,SCA49,SCA49,90RTE,90RTE,SCA8,SCA8
POP,GroupA,,,,,,,,,,,,,,
F48F11,141,149,192,194,,,166,166,,,192,192,167,171
F48B09,161,181,195,208,277,281,,,227,229,192,192,159,193
F39B02,153,163,198,202,269,297,166,168,231,231,192,192,163,169
F49E01,161,181,195,208,277,281,166,166,227,229,192,192,159,193
F38A12,,,,,277,295,170,172,,,192,196,159,163
POP,GroupB,,,,,,,,,,,,,,
F31C03,153,163,198,202,,,,,,,192,192,163,171
F44H01,141,149,,,,,,,221,221,192,192,167,171
F45H01,143,147,212,216,,,166,168,221,231,,,,
```

3.2.2 Allele frequencies

In some SHAZA models allele frequencies are read directly into SHAZA using the ‘-2 FILE’ command line argument where FILE is the name of the allele frequency file. For each locus there are two lines. The first line has the locus name (no spaces) in the first column followed by an integer for each allele (*e.g.* number of base pairs in each allele). The second line has the locus name (no spaces) in the first column followed by the frequencies of each allele. The format is repeated for multiple loci. An example with the first locus having four alleles and the second locus having five alleles is:

Allele 166 in LOCUS_1 has a frequency of 0.100

```
LOCUS_1     95     98     101     166
LOCUS_1     0.250  0.250  0.400   0.100
LOCUS_2     149    163    166     181     221
LOCUS_2     0.050  0.050  0.100   0.100   0.70
```

Alternatively a format without allele names can be read by SHAZA:

```
LOCUS_1     0.250  0.250  0.400   0.100
LOCUS_2     0.050  0.050  0.100   0.100   0.70
```

3.3 Command line options

Shaza is a command line program compared to a menu driven program. This makes it much easier to install on different operating systems. The code is less complicated and therefore easier to maintain and less prone to bugs leading to more time for rapid development.

For those not familiar with the older style command prompt it should be easy to learn. From the command prompt type the program name with a list of options (arguments) on the same line then hit the <ENTER> key. Section 1.3 gives instructions on how to get to the command line prompt from Windows XP.

A complete list of options with a short description of each is listed in Table 3 (page 10). Note all options can be listed in any order and all options are case sensitive.

3.3.1 Basic options

A knowledge of these basic options is all that is needed to run the most common SHAZA analysis. The '-1 FILE' and the '-2 FILE' options denote which file type (section 3.2 Data input formats) is on the command line. There is always a space between the argument identifier and the associated string or value (*e.g.* space between '-1' and 'FILE').

The '-e NUM' option defines the error rate used to estimate Log Likelihood Ratio's (LLR's). The NUM value in '-e NUM' defines the error rate per locus ε defined in equation 2 of Macbeth *et al* (2011). A default value of –e 0.010 defines a 1% error per locus. If unknown the default value of '-e NUM' could be used provided it is thought to be a conservatively high estimate (Macbeth *et al* 2011) - see also section 6.0 Example 3.

3.3.2 Output file names

SHAZA runs produce up to eight different files depending on the model used (Table 2). The ten models are described in detail in section 3.4 with a short description of each file provided in Table 3. Models 7 to10 are used to create simulated genotypes and no pair matching is performed. The composite genotype list in "SHAZA_join.txt" is optional in models 1 to 6 as defined by the '-j' option. A simulated reference data file "SHAZA_simm.dat" is produced in models 3 to 10.

In addition to these files there is one temporary file created "SHAZA_temp.txt" which is deleted during normal execution of SHAZA. The prefix 'SHAZA" can be changed using the '-p STR' command as described in section 3.3.3 'File specific options'.

Table 2. Output files generated by different SHAZA models.

	SHAZA model									
Output files	1	2	3	4	5	6	7	8	9	10
SHAZA_log.txt	y	y	y	y	y	y	y	y	y	y
SHAZA_freq.txt	y	y	y	y	y	y	y	y	y	y
SHAZA_loci.txt	y		y		y		y		y	
SHAZA_miss.txt	y	y	y	y	y	y	y	y	y	y
SHAZA_pair.txt	y	y	y	y	y	y	y	y	y	y
SHAZA_stat.txt	y	y	y	y	y	y			y	y
SHAZA_pop.txt	y	y	y	y	y	y				
SHAZA_simm.txt			y	y	y	y	y	y	y	y
SHAZA_join.txt	o	o	o	o	o	o				

y-yes, o-optional

3.3.3 File specific options

The '–d STR' option can change the "SHAZA" prefix in all output file names listed in Table 2. This feature may be useful when the output from different data sets needs to be kept and uniquely identified. This option could also be used to change the output directory using a path referenced from the current directory position.

The '-p' option actions SHAZA to write a simulated genotype file without analysing the data.

The '-j' option actions SHAZA to form composite genotypes by joining groups of matches together and re-naming them as one individual. A file 'SHAZA_join.dat' is created when the '-j' option is defined. The names of one or more matching pairs is replaced by GP_'n' where 'n' is the number of the matching group from 1 to the number_of_matching_groups_found *e.g.* GP_1 for the first group. A list of the samples in each group can be found in file 'SHAZA_pair.txt' listed in the 'Clustered groups of pairwise matches' table.

Three simple algorithm steps are used to form the composite genotypes: (i) When there are no different alleles between a matching pair all alleles then they are combined with any missing loci in each sample overwritten by alleles where available. (ii) When there are different alleles between a matching pair and the difference is due to a potential allelic dropout then the heterozygote is used when forming the composite. (iii) When there are different alleles between a matching pair all alleles and the difference cannot be explained as a potential allelic dropout then the first genotype in the group is used. The user should attempt to resolve differences in matches to avoid step (iii) as this results in a creation of potentially erroneous genotypes. Depending on the application it may be best to define small number of false positives using the '–z NUM' option to improve the accuracy of the formed composite genotypes.

3.3.4 Simulated population options

These options are used to generate a simulated reference population which is written in GENEPOP format to file 'SHAZA_simm.txt'.

The '-M NUM' option sets the size of the first simulated population. If recaptures between two populations are required then the additional population with '-H NUM' samples can be defined. Next the number of recaptures within the M population (or between the M and H) populations is defined with the '-R NUM' option.

The missing loci combinations can be determined from those within an allele file '-1 FILE' or by setting the proportion of loci randomly genotyped with the '-Y NUM' option. For example setting '–Y 0.90' will simulate genotypes with 100%-90%=10% missing loci.

The '-E NUM' option defines the typing error rate per locus simulated in the population(s). This is a related statistic to the '-e NUM' option which defines the error rate per locus assumed in the population when analysing the data. Loci errors in SHAZA were drawn for every loci within every genotype sample by randomly sampling a number between zero and one. If the random number is below the proportion given by '–E NUM', then an error is created for that sample at that locus. When creating an error within a locus a haplotype is randomly chosen with its allele replaced from a randomly drawn allele from those in the sample population '-1 FILE' or from the genotype frequency file '-2 FILE' if defined.

The allele frequencies used when simulating populations come from either individual alleles defined by the '-1 FILE' or if defined on the command line an allele frequency file '-2 FILE'.

3.3.5 Other options

The '-b NUM' option defines the way in which missing loci are simulated and can only be used when analysing allele data (Models 1 and 2). The missing loci combinations can be sampled in three ways: (i) '-b 0' (that's -b with a space then a zero) Missing loci combinations sampled in the same order as those found in the '-1 FILE', (ii) '-b 1' Missing loci combinations sampled by bootstrapping those combinations across multiple populations, and (iii) '-b 2' Missing loci combinations sampled by bootstrapping those

combinations within each population. If there is only one population defined, *i.e.* one POP in the '-1 FILE', then options '-b 1' and '-b 2' are equivalent.

Table 3. Summary of SHAZA options.

Argument	Default	Example	Range	Comment
Basic options				
-1 FILE		-1 ./basepair.dat -2 /frequency.dat		Input file 1: containing base pair data with directory position relative to shaza.exe file
-2 FILE				Input file 2: containing allele frequency data with directory position relative to shaza.exe file
-e NUM	0.01	-e 0.04	0.0001 to 0.20	Typing error per locus (in percent *100) used for *LLR* analysis (e.g. for 2% error use 200)
Output file names				
	SHAZA_log.txt			All screen output logged to this file
	SHAZA_freq.txt			Allele frequency report
	SHAZA_miss.txt			Missing allele combination report
	SHAZA_pair.txt			Matching pair report
	SHAZA_pop.txt			Within and between population analysis (s.e. of ecapture numbers shown if –s option used)
	SHAZA_stat.txt			Summary of solutions at a different number of false positives (also lists convergence runs)
	SHAZA_simm.dat			Simulated genotype file
	SHAZA_join.dat			Genotypes file with composite and non-matching genotypes set by [-j] option
	SHAZA_loci.dat			Allele frequency file generated from data for potential use as -2 FILE
File specific options				
-j		-j		Joining matches to form composite genotype file SHAZA_join.dat
-p		-p		Create simulated genotype file with partial genotypes and write to SHAZA_simm.dat
-d STR	./SHAZA	-d ../mydir/run1		"SHAZA" prefix in all output files replaced with this prefix
				Can include directory position relative to current directory
				In example shown SHAZA_log.txt is replaced with ../mydir/run1_log.txt
Simulated population options				
-M NUM		-M 200	50 to10000	Number of genotypes created when simulating a reference data set
-H NUM	0	-H 100	>20	Number of genotypes simulated in a second group (M+H must also be < 10000)
-R NUM		-R 20	1..M/2	If no second group '-H 0' this option defines the number of matches (recaptures) simulated within M samples
			1..H	OR if second group this option defines number of matches (recaptures) simulated between M and H samples
-Y NUM	1.00	-T 0.80	0.50 to1.00	Simulated proportion of loci randomly genotyped (was –T in version 1.00)
-E NUM	-e value	-E 0.020	0.0001 to 0.20	Typing error per locus (percent * 100) simulated in (M+H) genotypes (e.g. for 2% error use 200)
Other options				
-b NUM	1	-b 2	0,1,2	When estimating s.e. of recapture numbers missing loci combinations can be sampled in three ways:
				0 replicate missing loci combinations in found in each sample
				1 random bootstrap missing loci combinations across alll populations
				2 random bootstrap missing loci combinations within populations
-c	10	-c 5	0..10	Convergence iterations to find corrected recaptures with smallest standard error estimate
-k STR	POP	-k BLOCK		Separator in [-1 FILE] used to partition populations
-m NUM	10	-m 20	≥3	Minimum number of records per population used in analysis
-n NUM	100	-n 200	100..1000	Number of simulated data sets used to determine relationship between LLR and false positives
-r NUM		-r 2468		Initialise random number seed to this value
-s NUM	100	-s 200	10..10000	Number of simulated data sets used to determine standard error of recapture estimates
-w		-w		Do not list potential matches with LLR below threshold in SHAZA_pair.txt
-z NUM	6	-z 12	0..50	Initialise prior number of false positives (V). Default corresponds to V=0.75 (Table 1)

NUM = an integer associated with preceding flag
FILE = a file name with no blank spaces associated with preceding flag
STR = a character string with no blank spaces associated with preceding flag

The '-c NUM' option defines the maximum number of iterations used when estimating recapture numbers.

The '-k STR' option defines the separator between populations (section: 3.2.1 Multi-locus genotype data). The default is 'POP'. If your data file uses something else then the separator can be changed using this option. The separator between populations in output files SHAZA_simm.dat and SHAZA_join.dat can also be changed using the '-K STR' option.

The '-n NUM' option defines the number of simulated data sets used to determine the relationship between *LLR*'s and the number of false positives. The simulated data sets emulate genotype errors and missing loci combinations with no recaptures. The default NUM=100 iterations can be changed to a maximum NUM=1000 iterations. With 1000 iterations the smallest increment of a false positive that can be detected is 1/1000=0.001 false positive.

The '-m NUM' option can be used to discard individual populations if the number of samples with them is less than the NUM value defined in this option.

The '-r NUM' option can be used to define a random number seed. With this option defined the same output will be achieved with the same data files and options defined within each computer platform (e.g. Windows XP). This option will not normally be used except when demonstrating an example.

The '-s NUM' option defines the number of simulated populations used to estimate the standard errors of recapture numbers. This option will have no effect when combined with '–c 0'.

The 'w' option can be used to stop printing potential matches. That is, the 'Additional pairs that missed out on match criteria' will not be listed in file SHAZA_pair.txt

The '-z NUM' option defines the maximum number of false positives allowed in determining matches and is discussed in detail in section '2.2.3 Convergence'.

3.4 SHAZA models

There are ten analytical models automatically defined by the combination of commend line arguments. For example Model 1 is a straightforward analysis of genotype matches from a single file containing alleles. This model can be executed from the general syntax: "shaza.exe -1 FILE" where "-1" tells SHAZA that the following string is the name of the file containing alleles and "FILE" is the generic file name to be substituted with the name of an existing data file. For example in Windows XP to run the narrow-barred Spanish Mackerel data, which is in SHAZA_csv format containing base pairs, enter these commands:

```
C:\> cd shaza                                <ENTER>
C:\shaza> shaza.exe  -1  fish_example.dat  <ENTER>
```

An example of model 1 is described in more detail in Section 4.0.

While model 1 will be suitable for most genotype match analysis, SHAZA is also an extensive simulation package capable of testing statistical power of match analyses under different scenarios. There are three classes of analysis (i) genotype match analysis, (ii) simulation modelling and (iii) genotype file creation through simulation. In total there are 10 models defined in SHAZA (Table 4).

The options in the command line column of Table 4 are used to automatically determine the model number. Each model has a number of options permitted with the syntax of each listed in Table 3.

Table 4 summarises which file is used to determine the allele frequencies used in each SHAZA model. As listed the allele frequencies used in the analysis are determined from within the allele data file '-1 FILE' or determined directly from an allele frequency file '-2 FILE'. When both the '-1 FILE' and '-2 FILE' occur on the same command line, allele frequencies are read from the '-2 FILE'.

Table 4 summarises how missing loci combinations are simulated in each SHAZA model. Missing loci combinations can be simulated from the '-Y NUM' option by defining the proportion of null loci and

implemented in SHAZA through random sampling. Missing loci combinations can also be determined from the allele file using the '-b NUM' option described in section 3.3.5.

Table 4 lists the number of population cohorts that can be analysed with each model. The compiled shaza.exe file accepts up to 100 population cohorts when analysing allele data from field collection. Each population can be a spatial or temporal group of data and is defined by the word "POP" at the beginning of each group of genotypes.

Table 4. Summary of SHAZA models

Model	Class	Command line	Options permitted	Allele frequencies	Missing loci from	Population cohorts
1	Analyse	-1 FILE	bcdejkmnrswz	-1 FILE	-1 FILE [-b NUM]	<=100
2	Analyse	-1 FILE -2 FILE	bcdejkmnrswz	-2 FILE	-1 FILE [-b NUM]	<=100
3	Simulate	-1 FILE -M NUM -Y NUM	cdejknrswzEHMRY	-1 FILE	-Y NUM	1 or 2
4	Simulate	-2 FILE -M NUM -Y NUM	cdejknrswzEHMRY	-2 FILE	-Y NUM	1 or 2
5	Simulate	-1 FILE -M NUM	cdejknrswzEHMR	-1 FILE	-1 FILE	1 or 2
6	Simulate	-1 FILE -2 FILE -M NUM	cdejknrswzEHMR	-2 FILE	-1 FILE	1 or 2
7	Create	-1 FILE -M NUM -Y NUM -p	dkprEHMRY	-1 FILE	-Y NUM	1 or 2
8	Create	-2 FILE -M NUM -Y NUM -p	dkprEHMRY	-2 FILE	-Y NUM	1 or 2
9	Create	-1 FILE -M NUM -p	dkprEHMR	-1 FILE	-1 FILE	1 or 2
10	Create	-1 FILE -2 FILE -M NUM -p	dkprEHMR	-2 FILE	-1 FILE	1 or 2

NUM = an integer argument associated with preceding argument
FILE = a file name with no blank spaces

A few examples of how to run each model are shown in Table 5. The '**your_base_pair_file.dat**' is a file containing genotypes in GENEPOP or SHAZA_csv. The file '**allele_frequency.dat**' contains allele frequencies (see**:** Section 3.2, Data input formats). There are two examples for model 1. Each example illustrates how different option arguments can be listed in the commend line.

Table 5. Example command line for different models.

Model No:		
1	shaza -1 **your_base_pair_file.dat**	
1	shaza -1 **your_base_pair_file.dat**	-b 2 -d run2 -e 300
2	shaza -1 **your_base_pair_file.dat**	-2 **allele_frequency.dat**
3	shaza -1 **your_base_pair_file.dat**	-M 200 -R 10 -Y 80
4	shaza -2 **allele_frequency.dat**	-M 200 -R 10 -Y 80
5	shaza -1 **your_base_pair_file.dat**	-M 200 -R 10
6	shaza -1 **your_base_pair_file.dat**	-2 **allele_frequency.dat** -M 200 -R 10
7	shaza -1 **your_base_pair_file.dat**	-M 200 -R 10 -p -Y 80
8	shaza -2 **allele_frequency.dat**	-M 200 -R 10 -p -Y 80
9	shaza -1 **your_base_pair_file.dat**	-M 200 -R 10 -p
10	shaza -1 **your_base_pair_file.dat**	-2 **allele_frequency.dat** -M 200 -R 10 -p

3.5 Program limits

In addition to the range of integer variables defined in Table 3 there are a number of other constants built into the compiled version of SHAZA including string lengths of input data (Table 6). These constants are thought to provide sufficient scope for most analysis to proceed without the need for source code recompilation. There are many other settings defined in shaza which can be viewed in the shaza.h and command.h source code files.

Table 6 SHAZA constraints defined during computer compilation.

	Maximum character length (no spaces)	Maximum integer size	#define name
SHAZA options			
-1 FILE	510		
-2 FILE	510		
-K STR	100		
Internal data limits			
Sample name	20**		SOA
Loci name	20		SOL
Line within -1 FILE	400		STR
Line within -2 FILE	400		STR
Name of population	80		BLOCKSIZ
Number of alelles		80	NOA
Number of loci		32	NOL
Number of shadow pairs		50	RANK_SFZ
Combinations of missing loci		5000	PERM
Number of populations		100	BLOCK
Matching pairs		6000	PAIRS
Sample size		30000	ANIM

** spaces allowed in sample name when using GENEPOP format

4.0 Example 1
– Finding matches in one population

This example demonstrates an analysis from field data using Model 1. An example data file with narrow-barred Spanish mackerel (*Scomberomorus commerson*) genotypes is provided in file: **fish_example.dat**. This example with 233 genotypes has seven loci and is used to demonstrate the operation and features of SHAZA. Copy **fish_example.dat** to the new working directory created during installation (*e.g* c:\SHAZA).

Open a command line window
e.g. In XP click '**start**', move cursor to the '**All programs'** tab, go to the **'Accessories'** folder and click on '**Command prompt'**.
Within the command window type:
cd c:\SHAZA <ENTER>

To analyse matching genotypes type:
shaza.exe –1 fish_example.dat –r 246 <ENTER>

The '-r NUM' option initialises the random number seed to give exactly the same output each time for a given operating system platform. Using **'–r 246'** the results from a Windows OS run will be similar to that listed here.

4.1 SHAZA_log.txt

The standard output stream from the above example (blue) is also recorded in 'SHAZA_log.txt' and should display:

```
Tues Jun 28 09:48:00 2011

Executable file: SHAZA Version 2.00


_____________________________________________________________
Model No: 1   Convergence iteration 1 with 0.750 false positives
_____________________________________________________________

STAGE  1: Listing command line options and defaults

          ____________________________________________________________________________

          Table 1  Options list
                   Status equals one if set by command line
          ____________________________________________________________________________

          Option   Status   Value
             -b    0        1
             -c    0        10
             -d    0        "SHAZA"
             -e    0        0.0100
             -E    0        0.0100
             -f    1
             -j    0
             -H    0        0
             -i1   1        "fish_example.dat"
             -i2   0
             -k    0        "POP"
             -m    0        10
             -M    0        0
             -n    0        100
             -o1   0        "SHAZA_freq.txt"
             -o2   0        "SHAZA_miss.txt"
             -o3   0        "SHAZA_pair.txt"
             -o4   0        "SHAZA_pop.txt"
             -o5   0        "SHAZA_stat.txt"
             -o6   0        "SHAZA_simm.txt"
             -o7   0        "SHAZA_join.txt"
```

Default error rate per locus

```
-o8    0        "SHAZA_loci.txt"
-o9    0        "SHAZA_log.txt"
-p     0
-r     1        246
-R     0        0
-s     0        100

-x     0
-w     0
-Y     0        1.00
-z     0        6
```

Random number seed set for demonstration purposes

```
STAGE  2: Reading allele base pairs for match analysis from:          FILE: "fish_example.dat"
          SHAZA spreadsheet format assumed for reading alleles
          A total of 233 genotype samples were read from 7 loci.


STAGE  3: Create locus/allele frequency report.                       FILE: "SHAZA_freq.txt"
STAGE  4: Report missing loci combinations in reference data.         FILE: "SHAZA_miss.txt"
STAGE  5: Create locus/allele frequency file for SHAZA input.         FILE: "SHAZA_loci.txt"

STAGE  6: Finding false positive distribution using 100 simulated data sets.
STAGE  7: Append resample estimates from reference data.              FILE: "SHAZA_stat.txt"

STAGE  8: Report pairwise matches to:                                 FILE: "SHAZA_pair.txt"
STAGE  9: Report resample percentage within sampling cohorts.         FILE: "SHAZA_pop.txt"

STAGE 11: Finding s.e. of resample estimate using 100 simulation runs.
```

Simulated R_hat recaptures using whole numbers

R_hat determined from reference data

```
          Convergence_iteration=1      run= 1/100  (Simulating  11 recaptures) 10.54
          Convergence_iteration=1      run= 2/100  (Simulating  11 recaptures) 10.54
          Convergence_iteration=1      run= 3/100  (Simulating  11 recaptures) 10.54
          Convergence_iteration=1      run= 4/100  (Simulating  11 recaptures) 10.54
               .       .       .       .       .
               .       .       .       .       .
               .       .       .       .       .
          Convergence_iteration=4      run=100/100  (Simulating  10 recaptures) 9.47

NOTE:   Convergence found with '-z  5' (V=0.500 false positives)


Convergence summary of total corrected recaptures (R_hat)
                    Reference       Simulated
                    data            data
 -z     False       R_hat           R_hat    s.e.
        positives
  3       0.125      9.493           9.370   1.049
  4       0.250     10.207          10.089   1.019
  5       0.500      9.826           9.665   0.937 (min s.e.)
  6       0.750     10.541          10.525   1.194
```

The solution with the smallest standard error (s.e.) is provided with a history of other solutions searched.

```
******Finished  shaza.exe ******
Finish time: Tue Jun 28 09:48:40 2011
Elapse time:  39.4 seconds
```

As indicated by the standard output stream six files were generated: SHAZA_pair.txt, SHAZA_freq.txt, SHAZA_miss.txt, SHAZA_loci.txt, SHAZA_stat.txt and SHAZA_pop.txt.

4.2 SHAZA_pair.txt

SHAZA_pair.txt — List of pairwise match's found.

As listed in the SHAZA_log.txt file there were 4 convergence iterations with *V*<=0.50 false positives yielding the smallest standard error. At this level there were 10 pairwise matches identified. There were two match's determined with only three loci (6 allele match). The additional pairs listed may contain match's of potential interest which missed out due to lack of statistical power and may be re-examined. A list of clustered groups of pairwise matches is also listed.

```
Tue Jun 28 09:48:40 2011

Executable file: SHAZA Version 2.00
Input data file: fish_example.dat
Output     file: SHAZA_pair.txt
```

Number of mismatching alleles between two samples.

Mismatches that are a potential allelic dropout

```
Convergence iteration 3 with 0.500 false positives

List of pairwise match(s) having cumulative number of false positives <=0.50
```

Rank	Sample_pairs	Populations		Alleles Total	Mismatch	Dropout	Match log likelihood ratio (LLR)	Cumulative number of false positives
1	F49E01 : F48B09	1	1	12	0	0	24.70	0.000
2	F71D11 ? F72E10	1	1	14	1	1	20.91	0.000
3	Sc2185 : F87E09	1	1	10	0	0	18.30	0.000
4	F44G12 : F45H01	1	1	8	0	0	17.16	0.000
5	Sc2132 : F87E09	1	1	8	0	0	14.24	0.010
6	F66F02 : F66F03	1	1	8	0	0	11.74	0.015
7	F67D07 ? F67H01	1	1	12	2	0	10.31	0.035
8	F32F06 : F34A11	1	1	6	0	0	9.85	0.055
9	Sc1290 : F97E06	1	1	8	0	0	9.66	0.070
10	F44H01 : F48F11	1	1	6	0	0	7.81	0.235
	Additional pairs that missed out on match criteria:							
11	F31C03 ? F39B02	1	1	8	1	0	6.79	0.545
12	F30F09 ? F38C12	1	1	8	2	1	5.07	1.825
13	Sc2185 : Sc2132	1	1	4	0	0	4.37	3.185
14	F97E06 : F47D09	1	1	4	0	0	3.81	4.680
15	F35D07 ? F38D05	1	1	8	2	0	3.76	4.875
16	F42A12 ? F38E08	1	1	8	2	1	2.84	8.840
17	F46H01 ? F31C03	1	1	6	1	0	2.63	9.550
18	F45H01 ? F39A01	1	1	4	2	0	1.87	14.240
19	F30D07 : F47H10	1	1	2	0	0	1.86	14.445
	. .	.	.	.				
	. .	.	.	.				
34	F45C06 ? F43G12	1	1	10	3	1	1.12	20.200
35	F43B11 ? F39H01	1	1	6	1	1	1.01	21.310

'?' indicates that the match does not have identical genotypes

```
Clustered groups of pairwise matches

Group

GP1      F49E01 F48B09
GP2      F71D11 F72E10
GP3      Sc2185 F87E09 Sc2132
GP4      F44G12 F45H01
GP5      F66F02 F66F03
GP6      F67D07 F67H01
GP7      F32F06 F34A11
GP8      Sc1290 F97E06
GP9      F44H01 F48F11
```

Samples within each group are assumed to be from the same individual. This list contains an expectation of 0.25 false positives. If greater certainty is required remove those matches with the highest cumulative number of false positives.

4.3 SHAZA_freq.txt

SHAZA_freq.txt Locus/allele frequency report determined from allele data.
The number and proportion of each allele within each locus are indicated together with the frequency missing within each loci.

```
Tue Jun 28 09:48:00 2011

Executable file: SHAZA Version 2.00
Input data file: fish_example.dat
Output     file: SHAZA_freq.txt
```

Missing allele found in loci SCA30

Number of samples with missing alleles in loci SCA30

Allele frequencies of 233 records read from file <fish_example.dat>

loci name	loci number	allele name	allele number	allele count	allele frequency	cumulative frequency
SCA30	1	M	0	34	–	–
SCA30	1	133	1	19	0.04398	0.04398
SCA30	1	135	2	12	0.02778	0.07176
SCA30	1	137	3	3	0.00694	0.07870
SCA30	1	139	4	12	0.02778	0.10648
SCA30	1	141	5	85	0.19676	0.30324
.	.	.	.	.	.	.
.	.	.	.	.	.	.
SCA30	1	181	24	5	0.01157	0.98611
SCA30	1	187	25	4	0.00926	0.99537
SCA30	1	189	26	2	0.00463	1.00000

loci name	loci number	allele name	allele number	allele count	allele frequency	cumulative frequency
SM3	2	M	0	52	–	–
SM3	2	182	1	1	0.00242	0.00242
SM3	2	190	2	8	0.01932	0.02174
SM3	2	192	3	57	0.13768	0.15942
SM3	2	194	4	2	0.00483	0.16425
SM3	2	195	5	16	0.03865	0.20290
SM3	2	196	6	70	0.16908	0.37198
SM3	2	198	7	69	0.16667	0.53865
.	.	.	.	.	.	.
.	.	.	.	.	.	.
SCA8	7	163	5	78	0.18571	0.37619
SCA8	7	165	6	51	0.12143	0.49762
SCA8	7	167	7	61	0.14524	0.64286
SCA8	7	169	8	3	0.00714	0.65000
SCA8	7	171	9	22	0.05238	0.70238
SCA8	7	173	10	15	0.03571	0.73810
SCA8	7	175	11	27	0.06429	0.80238
SCA8	7	177	12	7	0.01667	0.81905
SCA8	7	179	13	15	0.03571	0.85476
SCA8	7	181	14	6	0.01429	0.86905
SCA8	7	183	15	16	0.03810	0.90714
SCA8	7	185	16	9	0.02143	0.92857
SCA8	7	187	17	2	0.00476	0.93333
SCA8	7	189	18	8	0.01905	0.95238
SCA8	7	191	19	2	0.00476	0.95714
SCA8	7	193	20	3	0.00714	0.96429
SCA8	7	195	21	1	0.00238	0.96667
SCA8	7	197	22	2	0.00476	0.97143
SCA8	7	199	23	1	0.00238	0.97381
SCA8	7	201	24	6	0.01429	0.98810
SCA8	7	203	25	2	0.00476	0.99286
SCA8	7	205	26	2	0.00476	0.99762
SCA8	7	207	27	1	0.00238	1.00000

4.4 SHAZA_miss.txt

SHAZA_miss.txt List of missing loci combinations found in genotype data.
The 'M' shows which alleles were missing within each locus combination. The frequency of each missing loci combination is also reported.

```
Tue Jun 28 09:48:00 2011

Executable file: SHAZA Version 2.00
Input data file: fish_example.dat
Output     file: SHAZA_miss.txt

Missing allele report

_____________________________________________________

              Locus                          Frequency
              1   2   3   4   5   6   7
_____________________________________________________
Combination
      1       .   .   M   .   M   .   .          3
      2       .   .   .   M   .   .   .         15
      3       .   .   .   .   .   .   .         91
      4       M   M   .   .   M   .   .          1
      5       .   .   M   M   .   .   M          2
      6       M   M   M   .   .   .   .          4
      7       .   .   M   .   M   .   M          1
      8       .   .   M   M   M   .   .         12
      9       M   .   .   M   .   .   M          2
     10       M   .   M   M   .   .   .          1
     11       M   M   .   .   .   .   M          1
     12       M   .   M   .   M   .   .          1
     13       .   .   M   M   .   .   .          7
     14       .   .   .   M   M   .   .         15
     15       .   .   .   .   M   .   M          1
     16       .   .   M   .   .   .   M          2
     17       .   .   .   M   .   .   M          3
     18       M   M   .   .   .   .   .          5
     19       .   .   M   .   .   .   .         40
     20       .   .   .   .   .   .   M          2
     21       .   M   M   M   .   .   .          4
     22       .   .   M   .   .   M   M          2
     23       .   .   .   .   .   M   M          5
     24       .   M   M   .   .   M   .          1
     25       .   M   M   .   M   .   .          2
     26       .   M   .   .   .   M   M          1
     27       M   M   .   M   .   .   .          1
     28       .   .   .   M   M   .   M          1
     29       .   M   .   M   .   .   .          1
     30       .   M   .   .   M   .   .          1
     31       .   M   M   .   .   .   .          1
     32       .   M   .   .   .   .   .          3
     33       M   .   .   .   .   .   .          1
_____________________________________________________
                                              233
_____________________________________________________

 Pairwise mismatch analysis of loci

_____________________________________
 Number      Comparable     Mismatch
_____________________________________
   0                0           16
   1               44          239
   2              721         1502
   3             2014         4300
   4             7655         7339
   5             6059         6258
   6             6440         5188
   7             4095         2186
_____________________________________
```

Missing loci combination 1 has alleles missing at loci 3 and 5.

The names corresponding to each loci number are listed in SHAZA_freq.txt

There were 91 samples with no missing loci.

There were 239 pairs that mismatched by 1 loci

There were 2014 pairs that had only 3 loci shared between them

4.5 SHAZA_loci.txt

SHAZA_loci.txt Locus/allele frequencies for SHAZA input.

The format is the same as that described in section "Data input formats (Section 3.2.2 allele frequencies)" above. The file can be used as an input into SHAZA using the -2 FILE command.

Line defining locus names in base pairs

Line defining allele frequencies

```
SCA30          133      135      137      139      141      143      145      147      149    . . .
SCA30      0.04398  0.02778  0.00694  0.02778  0.19676  0.01389  0.04167  0.05093  0.11574    . . .
SM3            182      190      192      194      195      196      198      200      202    . . .
SM3        0.00242  0.01932  0.13768  0.00483  0.03865  0.16908  0.16667  0.00966  0.04589    . . .
SM37           263      269      271      273      275      277      279      281      283    . . .
SM37       0.00333  0.15667  0.04333  0.03333  0.00667  0.11333  0.04667  0.09667  0.07333    . . .
SCA47          158      162      164      166      168      170      172      174      176    . . .
SCA47      0.00296  0.05621  0.00888  0.51775  0.16568  0.15385  0.07396  0.00592  0.00592    . . .
SCA49          221      223      225      227      229      231      233      235      237    . . .
SCA49      0.14615  0.02308  0.02051  0.10000  0.26923  0.15385  0.14872  0.03590  0.01795    . . .
90RTE          184      190      192      194      196      198      200      202      206    . . .
90RTE      0.00223  0.00223  0.74554  0.02009  0.10268  0.03125  0.06920  0.00893  0.01116    . . .
SCA8           149      157      159      161      163      165      167      169      171    . . .
SCA8       0.00238  0.00476  0.17381  0.00952  0.18571  0.12143  0.14524  0.00714  0.05238    . . .
```

4.6 SHAZA_stat.txt

SHAZA_stat.txt List of match statistics across different levels.

There are two tables in this output file. Table 1 is the statistical analysis of the reference data at different threshold levels (*V*). Within the range of *V* values listed the corrected number of recaptures does vary and at this stage we do not know which is the best estimate. The standard errors (s.e.) of corrected recapture estimates are given in Table 2 of this file. As the default '-z 6' was initially used with10.54 recaptures simulated to estimate s.e. (from Table 1 with z=6). The lowest s.e.=0.9665 occurred when estimating corrected recaptures at z=5 (i.e. a threshold of ≤0. 50 false positives).

```
Tue Jun 28 09:48:00 2011

Executable file: SHAZA Version 2.00
Input data file: fish_example.dat
Output     file: SHAZA_stat.txt
```

Prior recapture estimate with 0.75 false positives is 10.54

Table 1 Cumulative matches in reference data when increasing false positives (-z NUM)
Determined from pooled data over all POP=1 populations

z	LLR threshold (LLRv)	False positives (V)	Matches (M)	Recaps. (M-V)	False negatives (T)	Effective size (E)	Corrected recaptures (M-V+T)	Percentage recaptures
1	12.20	0.010	5	4.99	1.04	211.84	6.03	2.66
2	9.90	0.050	7	6.95	0.71	221.88	7.66	3.40
3	8.72	0.125	9	8.88	0.62	225.22	9.49	4.25
4	7.75	0.25	10	9.75	0.46	227.67	10.21	4.58
5	6.92	0.50	10	9.50	0.33	229.06	9.83	4.40
6	6.29	0.75	11	10.25	0.29	229.72	10.54	4.74
7	5.89	1.00	11	10.00	0.26	230.04	10.26	4.60
8	5.59	1.25	11	9.75	0.22	230.34	9.97	4.47
9	5.33	1.50	11	9.50	0.20	230.53	9.70	4.34
10	4.92	2.00	12	10.00	0.18	230.92	10.18	4.57
11	4.69	2.50	12	9.50	0.15	231.13	9.65	4.32
12	4.43	3.00	12	9.00	0.12	231.43	9.12	4.07
.	.	.	.	.				
.	.	.	.	.				

44	1.28	19.00	28	9.00	0.03	232.65	9.03	4.03
45	1.20	19.50	29	9.50	0.03	232.66	9.53	4.26
46	1.15	20.00	33	13.00	0.04	232.67	13.04	5.93
47	1.09	20.50	34	13.50	0.04	232.67	13.54	6.17
48	1.04	21.00	34	13.00	0.04	232.68	13.04	5.93
49	1.01	21.50	35	13.50	0.04	232.68	13.54	6.17
50	0.97	22.00	35	13.00	0.03	232.69	13.03	5.93

Convergence iteration 1 with 0.750 false positives

. . . . .
. . . . .
. . . . .

Convergence iteration 4 with 0.125 false positives

With V=0.125 *false positives there were* 9.49 *recaptures in the reference data (Table 1) which are used in this simulation.*

Table 2 Recaptures found when simulating 9.49 pairwise comparisons with given false positives.
Standard error (s.e.) estimated from 100 simulation runs
'*' indicates the estimate with minimum variance
'#' indicates the z value used to estimate recaptures from reference data

z	False positives (V)	Matches (M)	Recaps. (M-V)	False negatives (T)	Effective size (E)	Corrected recaptures	Percentage recaptures
1	0.010	7.59 (1.29)	7.58	1.61	211.35 (2.89)	9.19 (1.56)	4.11 (0.73)
2	0.050	8.51 (1.15)	8.46	0.87	221.69 (1.82)	9.33 (1.25)	4.18 (0.58)
3#	0.125	8.88 (0.99)	8.76	0.61	225.14 (1.37)	9.37 (1.05)	4.19 (0.49)
4	0.25	9.16 (0.87)	8.91	0.45	227.31 (1.00)	9.36 (0.92)*	4.19 (0.43)
5	0.50	9.47 (1.10)	8.97	0.34	228.72 (0.77)	9.31 (1.14)	4.16 (0.53)
6	0.75	9.88 (1.12)	9.13	0.27	229.54 (0.65)	9.40 (1.16)	4.21 (0.54)
7	1.00	10.15 (1.25)	9.15	0.24	230.01 (0.59)	9.39 (1.28)	4.20 (0.60)
8	1.25	10.39 (1.34)	9.14	0.21	230.31 (0.53)	9.35 (1.37)	4.19 (0.64)
9	1.50	10.62 (1.30)	9.12	0.19	230.55 (0.50)	9.31 (1.33)	4.17 (0.62)
10	2.00	11.32 (1.58)	9.32	0.16	230.97 (0.43)	9.48 (1.61)	4.25 (0.76)
11	2.50	11.78 (1.76)	9.28	0.14	231.21 (0.38)	9.42 (1.79)	4.22 (0.84)
12	3.00	12.38 (1.93)	9.38	0.13	231.38 (0.35)	9.51 (1.96)	4.26 (0.92)
13	3.50	12.78 (2.09)	9.28	0.12	231.55 (0.31)	9.40 (2.12)	4.21 (1.00)
14	4.00	13.30 (2.23)	9.30	0.11	231.69 (0.29)	9.41 (2.26)	4.22 (1.06)
15	4.50	13.95 (2.61)	9.45	0.10	231.81 (0.26)	9.55 (2.64)	4.29 (1.24)
.	.	.	.	.			
.	.	.	.	.			
.	.	.	.	.			
46	20.00	29.06 (6.74)	9.20	0.03	232.68 (0.08)	9.23 (6.53)	4.21 (3.10)
47	20.50	29.57 (6.87)	9.25	0.03	232.69 (0.07)	9.28 (6.60)	4.24 (3.13)
48	21.00	30.05 (6.97)	9.24	0.03	232.69 (0.07)	9.27 (6.69)	4.24 (3.17)
49	21.50	30.59 (7.13)	9.30	0.03	232.70 (0.07)	9.33 (6.82)	4.27 (3.25)
50	22.00	30.98 (7.33)	9.22	0.02	232.71 (0.07)	9.24 (6.98)	4.23 (3.33)

NOTE: Convergence found with '-z 5' (V=0.500 false positives)

Convergence summary of total corrected recaptures (R_hat)

		Reference data	Simulated data		
-z	False positives	R_hat	R_hat	s.e.	
3	0.125	9.493	9.370	1.049	
4	0.250	10.207	10.089	1.019	
5	0.500	9.826	9.665	0.937	(min s.e.)
6	0.750	10.541	10.525	1.194	

Summary of all convergence iterations showing the difference in the standard error (s.e.) of total recapture estimates at different levels.

4.7 SHAZA_pop.txt

SHAZA_pop.txt List of match's found between different populations.
This file has two tables. Table 1 lists the corrected number of matches from the reference data. Table 2 lists recaptures from replicated simulated data which is used to estimate the standard error (s.e.) of recaptures from the reference data. Recapture estimates with s.e. are determined within and between each population when there is more than one population defined by POP.

```
Tue Jun 28 09:48:28 2011

Executable file: SHAZA Version 2.00
Input data file: fish_example.dat
Output     file: SHAZA_pop.txt


_____________________________________________________
Convergence iteration 3 with 0.500 false positives
_____________________________________________________
```

Table 1 Within population recapture estimate from reference data

Population number n1	Population size s1	Effective sample size E	False positives V	Matches found M	Recaptures Corrected R_hat	(%) P
1	233	229.06	0.500	10	9.83	4.40

Table 2 Within population recapture estimate from 100 simulation runs

Population number n1	Population size s1	Effective sample size E	False positives V	Matches found M	Recaptures Corrected R_sim	s.e.
1	233	228.89	0.500	9.83	9.67	0.89

5.0 Example 2
- Simulating matches in two populations

This example demonstrates 10 simulated matches between two populations when providing allele frequencies (Model 4).

5.1 Generate example data and find matches

Multi-allelic frequencies of 0.25, 0.25, 0.2, 0.15, 0.05, 0.05, 0.02, 0.01, 0.01, 0.005 and 0.005 were used to generate eight loci. These were stored in file "frequency_one.dat" as shown below:

Allele number line

Allele frequency line

```
L1    1     2     3     4     5     6     7     8     9     10    11
L1    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L2    1     2     3     4     5     6     7     8     9     10    11
L2    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L3    1     2     3     4     5     6     7     8     9     10    11
L3    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L4    1     2     3     4     5     6     7     8     9     10    11
L4    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L5    1     2     3     4     5     6     7     8     9     10    11
L5    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L6    1     2     3     4     5     6     7     8     9     10    11
L6    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L7    1     2     3     4     5     6     7     8     9     10    11
L7    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L8    1     2     3     4     5     6     7     8     9     10    11
L8    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
```

Locus name

The following command line runs model 4 which generates an example genotype file with two populations and finds matches between them:

shaza.exe –2 frequency_one.dat –M 200 –H 100 –R 10 –Y 0.85 –r 123 <ENTER>

Options within the command line are

-M 200	Generates 200 samples in population 1,
-H 100	Generates 100 samples in population 2,
-R 10	Generates 10 recaptures between population 1 and 2,
-Y 0.85	Simulates data with 85 percent of loci genotyped,
-r 123	Set random number seed to replicate results for demonstration

5.2 SHAZA_log.txt

The standard output from this run was:

```
Thu Jun 30 11:55:13 2011
Executable file: SHAZA Version 2.00

_____________________________________________________________
Model No: 4   Convergence iteration 1 with 0.750 false positives
_____________________________________________________________

STAGE  1: Listing command line options and defaults
          ____________________________________________________________________________________

          Table 1  Options list
                   Status equals one if set by command line
          ____________________________________________________________________________________

           Option   Status   Value
             -b     0        1
             -c     1        10
             -d     0        "SHAZA"
             -e     0        0.0100
             -E     0        0.0100
```

```
-f    0
-j    0
-H    1        100
-l    0
-2    1        "frequency_one.dat"
-k    0        "POP"
-m    0        10
-M    1        200
-n    0        200
-o1   0        "SHAZA_freq.txt"
-o2   0        "SHAZA_miss.txt"
-o3   0        "SHAZA_pair.txt"
-o4   0        "SHAZA_pop.txt"
-o5   0        "SHAZA_stat.txt"
-o6   0        "SHAZA_simm.txt"
-o7   0        "SHAZA_join.txt"
-o8   0        "SHAZA_loci.txt"
-o9   0        "SHAZA_log.txt"
-p    0
-r    1        123
-R    1        10
-s    1        100
-x    0
-w    0
-Y    1        0.85
-z    0        6
```

Input/output files

```
STAGE  2: Reading allele frequencies from:                    FILE: "frequency_one.dat"
STAGE  3: Writing simulated reference genotypes to:           FILE: "SHAZA_simm.txt"
          This reference data has 300 samples containing 10 recaptures in 2 population(s).
          Missing loci combinations randomly sampled using Y=0.85
```

i.e. 15% of loci missing

```
          Simulated relationships:
          ____________________________________
          List of pairwise matches
          Sample_1              Sample_2
          ____________________________________
          291                   1
          292                   2
          293                   3
          294                   4
          295                   5
          296                   6
          297                   7
          298                   8
          299                   9
          300                   10
          ____________________________________
```

The two samples may not have identical genotypes as some loci may be missing in each and there may be simulated genotype errors within each sample.

Output files

```
STAGE  4: Create locus/allele frequency report.               FILE: "SHAZA_freq.txt"
STAGE  5: Report missing loci combinations in reference data. FILE: "SHAZA_miss.txt"

STAGE  6: Finding  zones using 100 simulated data sets.
STAGE  7: Append resample estimates from reference data.      FILE: "SHAZA_stat.txt"

STAGE  8: Report Pairwise matches to:                         FILE: "SHAZA_pair.txt"
STAGE  9: Report resample percentage within sampling cohorts. FILE: "SHAZA_pop.txt"

STAGE 10: Finding s.e. of resample estimates within each population using 100 simulation runs.

          Convergence_iteration=1      run= 1/100  (Simulating  10 recaptures)  9.84
          Convergence_iteration=1      run= 2/100  (Simulating  10 recaptures)  9.84
          Convergence_iteration=1      run= 3/100  (Simulating  10 recaptures)  9.84
          Convergence_iteration=1      run= 4/100  (Simulating  10 recaptures)  9.84
          Convergence_iteration=1      run= 5/100  (Simulating  10 recaptures)  9.84
          Convergence_iteration=1      run= 6/100  (Simulating   9 recaptures)  9.84
          Convergence_iteration=1      run= 7/100  (Simulating  10 recaptures)  9.84
. . . .
. . . .
```

```
          Convergence_iteration=1      run=98/100  (Simulating  10 recaptures)  9.84
          Convergence_iteration=1      run=99/100  (Simulating  10 recaptures)  9.84
          Convergence_iteration=1      run=100/100  (Simulating  10 recaptures) 9.84

. . . .
. . . .

          Convergence_iteration=2      run=100/100  (Simulating 10 recaptures) 10.33

NOTE:   Convergence found with  '-z  4' (V=0.250 false positives)
```

Convergence summary of total corrected recaptures (R_hat)

-z	False positives	Reference data R_hat	Simulated data R_hat	s.e.	
4	0.250	10.164	10.147	0.697	(min s.e.)
6	0.750	9.433	9.958	0.947	

Estimated from data pooled across all populations. For between population estimates see file: SHAZA_pop.txt

```
******Finished  shaza.exe ******
Finish time: Thu Jun 30 11:56:27 2011
Elapse time: 1 minute 13.9 seconds
```

5.3 SHAZA_stat.txt

This output file contains information from five convergence iterations.

```
Thu Jun 30 11:55:13 2011

Executable file: SHAZA Version 2.00
Input data file: frequency_one.dat
Output     file: SHAZA_stat.txt
```

These corrected recapture estimates are estimated by pooling data in one population (For within and between population estimates see SHAZA_pop.txt)

```
Table 1  Cumulative matches in reference data when increasing false positives (-z NUM)
         Determined from pooled data over all POP=2 populations

  z   LLR          False      Matches  Recaps.    False      Effective   Corrected    Percentage
      threshold    positives                      negatives  size        recaptures   recaptures
      (LLRv)       (V)        (M)      (M-V)      (T)        (E)         (M-V+T)
-------------------------------------------------------------------------------------------------
  1     11.72        0.010      10        9.99       1.61     278.25       11.60        4.02
  2     10.59        0.050      10        9.95       1.06     285.03       11.01        3.81
  3      9.14        0.125      10        9.88       0.58     291.50       10.45        3.61
  4      8.26        0.25       10        9.75       0.41     293.78       10.16        3.51
  5      7.36        0.50       10        9.50       0.26     295.99        9.76        3.36
  6      6.85        0.75       10        9.25       0.18     297.05        9.43        3.25
  7      6.38        1.00       10        9.00       0.15     297.55        9.15        3.15
  8      5.98        1.25       11        9.75       0.14     297.87        9.89        3.41
  9      5.71        1.50       11        9.50       0.12     298.15        9.62        3.31
 10      5.26        2.00       11        9.00       0.09     298.46        9.09        3.13
. . . .
. . . .
. . . .
 48      1.16       21.00       28        7.00       0.01     299.80        7.01        2.39
 49      1.12       21.50       30        8.50       0.01     299.81        8.51        2.92
 50      1.12       22.00       30        8.00       0.01     299.81        8.01        2.74
```

Recapture simulations are based on the sum of recaptures estimated within and between populations

```
Convergence iteration 1 with 0.750 false positives

Table 2  Recaptures found when simulating 9.84 pairwise comparisons at given  levels.
         For recapture estimates within populations see: SHAZA_pop.txt
         Standard error (s.e.) estimated from 100 simulation runs
         '*' indicates the estimate with minimum variance
         '#' indicates the z value used to estimate recaptures from reference data

  z    False      Matches          Recaps.    False      Effective          Corrected          Percentage
       positives                                negatives  size               recaptures         recaptures
       (S)        (M)              (M-V)      (T)        (E)                (M-V+T)
-------------------------------------------------------------------------------------------------------------
  1    0.010       8.57 ( 1.12)     8.56       1.34   278.72 ( 2.50)      9.90 ( 1.24)      3.42 ( 0.46)
  2    0.050       9.12 ( 0.97)     9.07       0.95   285.35 ( 1.88)     10.02 ( 1.01)      3.46 ( 0.38)
  3    0.125       9.55 ( 0.80)     9.43       0.53   291.83 ( 1.13)      9.96 ( 0.78)      3.43 ( 0.30)
  4    0.25        9.72 ( 0.77)     9.47       0.38   294.09 ( 0.86)      9.85 ( 0.73)*     3.40 ( 0.28)
  5    0.50       10.11 ( 0.80)     9.61       0.26   296.01 ( 0.59)      9.87 ( 0.77)      3.40 ( 0.29)
  6#   0.75       10.51 ( 0.97)     9.76       0.20   296.99 ( 0.43)      9.96 ( 0.95)      3.43 ( 0.36)
  7    1.00       10.84 ( 1.17)     9.84       0.16   297.59 ( 0.35)     10.00 ( 1.16)      3.45 ( 0.43)
  8    1.25       11.21 ( 1.39)     9.96       0.14   297.91 ( 0.32)     10.10 ( 1.39)      3.49 ( 0.51)
  9    1.50       11.42 ( 1.45)     9.92       0.12   298.16 ( 0.29)     10.04 ( 1.44)      3.47 ( 0.53)
 10    2.00       11.94 ( 1.43)     9.94       0.10   298.48 ( 0.25)     10.04 ( 1.42)      3.47 ( 0.52)
. . . .
. . . .
. . . .
 48   21.00       29.19 ( 5.27)     8.28       0.01   299.81 ( 0.03)      8.29 ( 5.10)      2.87 ( 1.82)
 49   21.50       30.13 ( 5.47)     8.74       0.01   299.82 ( 0.03)      8.76 ( 5.25)      3.04 ( 1.88)
 50   22.00       30.13 ( 5.47)     8.27       0.01   299.82 ( 0.03)      8.28 ( 5.20)      2.87 ( 1.86)

NOTE:   Above run initiated with      '-z  6' (V=0.750 False positives)
        Trying another run with       '-z  4' (V=0.250 False positives)
```

Convergence iteration 2 with 0.250 false positives

Table 2 Recaptures found when simulating 10.33 pairwise comparisons with given false positives.
For recapture estimates within populations see: SHAZA_pop.txt
Standard error (s.e.) estimated from 100 simulation runs
'*' indicates the estimate with minimum variance
'#' indicates the z value used to estimate recaptures from reference data

z	False positives (V)	Matches (M)	Recaps. (M-V)	False negatives (H)	Effective size (E)	Corrected recaptures (M-V+H)	Percentage recaptures
1	0.010	8.97 (1.21)	8.96	1.43	278.40 (2.57)	10.39 (1.33)	3.59 (0.50)
2	0.050	9.41 (0.96)	9.36	0.99	285.21 (1.91)	10.35 (0.96)	3.57 (0.38)
3	0.125	9.87 (0.90)	9.74	0.56	291.72 (1.18)	10.30 (0.83)	3.56 (0.34)
4#	0.25	10.00 (0.80)	9.75	0.40	294.03 (0.91)	10.15 (0.70)*	3.50 (0.30)
5	0.50	10.45 (0.97)	9.95	0.27	295.96 (0.63)	10.22 (0.88)	3.53 (0.36)
6	0.75	10.83 (1.03)	10.08	0.21	296.95 (0.46)	10.29 (0.94)	3.55 (0.38)
7	1.00	11.03 (1.15)	10.03	0.16	297.56 (0.36)	10.19 (1.07)	3.52 (0.42)
8	1.25	11.35 (1.30)	10.10	0.14	297.88 (0.33)	10.24 (1.23)	3.54 (0.47)
9	1.50	11.61 (1.39)	10.11	0.13	298.13 (0.30)	10.24 (1.33)	3.54 (0.51)
10	2.00	12.07 (1.60)	10.07	0.10	298.46 (0.26)	10.17 (1.55)	3.51 (0.58)
. . . .							
. . . .							
. . . .							
48	21.00	29.97 (5.78)	9.05	0.01	299.81 (0.03)	9.06 (5.62)	3.15 (2.03)
49	21.50	30.90 (5.99)	9.47	0.01	299.82 (0.03)	9.48 (5.86)	3.31 (2.12)
50	22.00	30.90 (5.99)	8.99	0.01	299.82 (0.03)	9.00 (5.82)	3.14 (2.10)

NOTE: Convergence found with '-z 4' (V=0.250 false positives)

Convergence summary of total corrected recaptures (R_hat)
(estimated from data pooled across all populations)

		Reference data	Simulated data	
-z	False positives	R_hat	R_hat	s.e.
4	0.250	10.164	10.147	0.697 (min s.e.)
6	0.750	9.433	9.958	0.947

5.4 SHAZA_pair.txt

The file "SHAZA_pair.txt" produced from the above SHAZA run is listed below. The first 10 genotypes were identified correctly with the 10th match below the threshold value of V=0.25 false positives. Although this was a simulation the annotated comments would indicate our thoughts if this were a real set of data.

```
Thu Jun 30 11:56:27 2011.

Executable file: SHAZA Version 2.00
Input data file: frequency_one.dat
Output     file: SHAZA_pair.txt


          Simulated relationships:
          ____________________________________
          List of pairwise matches
          Sample_1              Sample_2
          ____________________________________
          291                   1
          292                   2
          293                   3
          294                   4
          295                   5
          296                   6
          297                   7
          298                   8
          299                   9
          300                   10
          ____________________________________

_____________________________________________
Convergence iteration 2 with 0.250 false positives
_____________________________________________
```

If we were analysing field data we would try to resolve any genotyping errors e.g. re-examine electropherogram to help explain allelic dropouts or mismatching alleles.

```
_____________________________________________________________________________________________
List of pairwise match(s) having cumulative number of false positives <=0.25

_____________________________________________________________________________________________
Rank      Sample_pairs     Populations      Alleles                          Match log    Cumulative
                                            Total Mismatch Dropout           likelihood   number
                                                                             ratio (LLR)  of false
                                                                                          positives
_____________________________________________________________________________________________
  1       297 : 7              2    1        16      0       0              25.33         0.000
  2       293 : 3              2    1        16      0       0              25.05         0.000
  3       300 : 10             2    1        12      0       0              24.99         0.000
  4       294 : 4              2    1        16      0       0              21.87         0.000
  5       299 : 9              2    1        10      0       0              18.90         0.000
  6       292 : 2              2    1        12      0       0              18.66         0.000
  7       298 : 8              2    1        12      0       0              16.87         0.000
  8       295 : 5              2    1        12      0       0              16.42         0.000
  9       291 ? 1              2    1        14      1       1              13.97         0.005
 10       296 ? 6              2    1        16      1       0              12.61         0.005
_____________________________________________
```

As the two samples in this match had no missmatching alleles the electropherogram could be re-examined or samples re-genotyped.

```
Additional pairs that missed out on match criteria:
_____________________________________________
 11       165 ? 44        e    1    1        10      2       1              6.37          1.010
 12       144 : 106       e    1    1         4      0       0              4.57          3.315
 13       203 ? 138       e    2    1        12      3       0              4.11          4.170
 14       230 ? 214       e    2    2         8      1       0              3.47          5.850
 15       248 ? 145       e    2    1         6      1       1              3.46          5.955
 16       208 : 71        e    2    1         2      0       0              2.28         12.495
 17       155 : 12        e    1    1         2      0       0              2.06         13.665
 18       144 ? 62        e    1    1         6      2       0              2.04         14.445
 19       279 ? 169       e    2    1         6      1       0              1.85         15.225
 20       279 ? 227       e    2    2         6      1       0              1.85         15.225
 21       212 ? 107       e    2    1         6      2       0              1.81         15.620
 22       214 ? 71        e    2    1         6      1       0              1.63         16.400
```

23	279 ? 143	e	2	1	6	2	0	1.57	16.985
24	216 ? 128	e	2	1	6	2	1	1.57	17.180
25	269 ? 119	e	2	1	6	1	0	1.41	18.350
26	222 ? 202	e	2	2	6	2	1	1.41	18.350
27	119 ? 107	e	1	1	6	1	0	1.23	20.305
28	134 ? 133	e	1	1	6	2	0	1.16	20.890
29	299 ? 142	e	2	1	10	3	0	1.16	21.180
30	225 ? 65	e	2	1	10	2	0	1.16	21.180

'e' in table above denotes false positive match

Clustered groups of pairwise matches

Group	
GP1	297 7
GP2	293 3
GP3	300 10
GP4	294 4
GP5	299 9
GP6	292 2
GP7	298 8
GP8	295 5
GP9	291 1
GP10	296 6

5.5 SHAZA_pop.txt

File "SHAZA_pop.txt" produced from the above run is listed below. From the simulated reference data a corrected recapture estimate of 10.33 was determined from the 10 matches found (Table 1 output listed below). *P* is the percentage recapture rate determined from the average number of pairwise comparisons between populations n1=2 and n2=1. From Table 2 output listed below the standard error (s.e.) of recaptures between populations n1=2 and n2=1 was 0.64. Within the 100 iterations of simulated data there were, by chance, recaptures found within the n1 and n2 populations. These both had corrected simulated recapture estimates below their measured standard error.

```
Thu Jun 30 11:55:58 2011

Executable file: SHAZA Version 2.00
Input data file: frequency_one.dat
Output     file: SHAZA_pop.txt
```

Total number of false positives from all blocks should approximate total accepted in convergence iteration

```
Convergence iteration 2 with 0.250 false positives
```

Estimate from a single simulated data set.
If models 1 and 2 were used this output would be from real field data.

```
Table 1  Within and between population recapture estimates from data

  Population    Population    Effective     Effective      False      Matches   Recaptures
  number        size          block size    sample size    positives  found     Corrected   (%)
  n1   n2       s1    s2       b             e              V          M         R_hat       P

   2 : 1       100 : 200       141.42        138.26         0.120      10        10.33       7.30
```

$b = (s1.s2)^{0.5} = (100 \times 200)^{0.5} = 141.42$

Estimate from 100 simulated data sets.

```
Table 2  Within and between population recapture estimates from 100 simulation runs

  Population    Population    Effective     Effective      False      Matches   Recaptures
  number        size          block size    sample size    positives  found     Corrected
  n1   n2       s1    s2       b             e              V          M         R_sim       s.e.

   1 : 1       200 : 200       200.00        195.95         0.105      0.05      0.05        0.25
   2 : 1       100 : 200       141.42        138.58         0.112      9.91      10.20       0.64
   2 : 2       100 : 100       100.00         97.69         0.033      0.04      0.04        0.20
```

Between population effective sample size (e) *determined from the square root of the number of 'enabled' pairwise comparisons.*

Average matches within populations found by random chance during the 100 simulations.

$R\_hat = [b(1-(1-pir)^{b}] = [141.42(1-(1-pir)^{141.42}] = 10.33$

where $pir = 1-\exp(\ln(1-(M-S)/e)/e)$

$= 1-\exp(\ln(1-(10-0.120)/138.26)/138.26) = 0.000536$

Assuming no multiple recaptures the average between population percentage recapture estimate $100.R\_hat/(s1 \times s2)^{0.5} = 7.30$

Note: Can also estimate the percentage of between population recaptures expressed as a percentage in

population 1 as: $100.R\_hat/s1 = 100*9.97/100 = 10.33$

population 2 as: $100.R\_hat/s2 = 100*9.97/200 = 5.17$

6.0 Example 3 - Matches with genotype errors

One common problem encountered is that by increasing the number of loci the chance of genotype errors increases. This example demonstrates how easy it is to find matching genotypes in data that is full of genotype errors provided there is sufficient statistical power in the data.

6.1 Generate example data

First we generate an example dataset with sufficient power for error detection. Multi-allelic frequencies of 0.25, 0.25, 0.2, 0.15, 0.05, 0.05, 0.02, 0.01, 0.01, 0.005 and 0.005 were used to generate twenty loci. These are stored in file "frequency_two.dat" as shown below:

```
L1    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L2    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L3    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L4    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L5    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L6    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L7    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L8    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L9    0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L10   0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L11   0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
L12   0.25  0.25  0.20  0.15  0.05  0.05  0.02  0.01  0.01  0.005 0.005
```

Run model 8 to generate a data set with commands:
Set 200 samples in population 1 '-M 200',
Set 10 recaptures within population 1 '-R 10',
Set 0.95 proportion of loci genotyped '-Y 0.95
Set 20 percent error per locus '-E 0.20' (set extremely high to demonstrate the power of SHAZA)
Set random number seed starting value '-r 111'
Create simulated data file '-p'

shaza.exe -2 frequency_two.dat –M 200 –R 20 –Y 0.90 -E 0.20 –r 111 -p <ENTER>

As can be seen from the standard output (and also in SHAZA_log.txt) the names of the pairwise recaptures that were simulated are listed (e.g. sample 181 was a recapture of sample 1, sample 182 was a recapture of sample 2, etc)

```
Simulated relationships:
_____________________________________________
List of pairwise matches
Sample_1              Sample_2
_____________________________________________
181                   1
182                   2
183                   3
184                   4
185                   5
186                   6
187                   7
188                   8
189                   9
190                   10
191                   11
192                   12
193                   13
194                   14
195                   15
196                   16
197                   17
198                   18
199                   19
200                   20
_____________________________________________
```

The example file generated from the above run is: **SHAZA_simm.txt**
This file has genotypes with on average 95% of loci missing and on average have an extreme 20% erroneous genotypes per loci with the erroneous genotype sampled randomly from allele frequencies in the population.

6.2 Find matches in example data

Using the example file generated in section 6.1 we use Model 1 to estimate the number of matches (recaptures) in the data with commands:
Set locus error rate used in analysis '-e 0.20' (set at least as high as the true error rate in the data)
Set random number seed starting value '-r 111'

shaza.exe -1 SHAZA_simm.txt -e 0.20 –r 111 <ENTER>

6.3 SHAZA_stat.txt

The summary table at the bottom of SHAZA_stat.txt shows the results from three iterations used to find convergence. The solution for the number of recaptures with the smallest standard error is 20.34.

Convergence summary of total corrected recaptures (R_hat)

-z	False positives	Reference data R_hat	Simulated data R_hat	s.e.
2	0.050	20.343	20.364	0.699 (min s.e.)
6	0.750	19.346	19.479	1.194

Convergence iteration 2 with 0.050 false positives

List of pairwise match(s) having cumulative number of false positives <=0.05

Rank	Sample_pairs	Populations	Alleles Total	Mismatch	Dropout	Match log likelihood ratio (LLR)	Cumulative number of false positives
1	Rec182 ? Rec2	1 1	34	5	1	31.73	0.000
2	Rec181 ? Rec1	1 1	30	2	0	31.53	0.000
3	Rec183 ? Rec3	1 1	32	3	1	30.52	0.000
4	Rec186 ? Rec6	1 1	32	3	2	30.48	0.000
5	Rec184 ? Rec4	1 1	32	3	1	28.52	0.000
6	Rec189 ? Rec9	1 1	30	3	2	26.06	0.000
7	Rec195 ? Rec15	1 1	26	3	1	23.95	0.000
8	Rec198 ? Rec18	1 1	28	3	2	23.33	0.000
9	Rec193 ? Rec13	1 1	36	7	1	21.44	0.000
10	Rec187 ? Rec7	1 1	34	7	3	21.14	0.000
11	Rec199 ? Rec19	1 1	38	6	2	20.19	0.000
12	Rec190 ? Rec10	1 1	30	6	3	20.18	0.000
13	Rec194 ? Rec14	1 1	24	4	1	19.92	0.000

6.4 SHAZA_pair.txt

It appears that there was adequate statistical power in this data set to determine all matches. On average we would expect only 0.05 false positives. In this example true match pairs were identified with up to 9 alleles mismatching between identified matching pairs.

14	Rec197	?	Rec17	1	1	32	5	0	19.80	0.000
15	Rec196	?	Rec16	1	1	34	6	3	19.78	0.000
16	Rec192	?	Rec12	1	1	32	6	1	18.86	0.000
17	Rec200	?	Rec20	1	1	32	6	4	18.64	0.000
18	Rec188	?	Rec8	1	1	28	5	1	12.47	0.005
19	Rec185	?	Rec5	1	1	26	5	1	11.49	0.005
20	Rec191	?	Rec11	1	1	32	9	5	9.48	0.035

Additional pairs that missed out on match criteria:

21	Rec153	?	Rec147	1	1	32	17	3	4.16	1.355
22	Rec165	?	Rec15	1	1	36	17	3	3.22	2.655
23	Rec198	?	Rec185	1	1	32	15	5	2.27	5.290
24	Rec60	?	Rec18	1	1	28	13	5	2.23	5.430
25	Rec172	?	Rec100	1	1	30	15	5	2.18	5.605
26	Rec96	?	Rec71	1	1	30	13	4	2.14	5.770
27	Rec109	?	Rec87	1	1	32	15	5	1.99	6.515
28	Rec41	?	Rec12	1	1	26	14	4	1.75	7.640
29	Rec105	?	Rec89	1	1	24	10	3	1.63	8.220
30	Rec196	?	Rec81	1	1	36	19	5	1.62	8.320
31	Rec156	?	Rec5	1	1	26	11	3	1.52	8.790
32	Rec103	?	Rec38	1	1	28	12	2	1.30	10.340
33	Rec174	?	Rec114	1	1	32	18	5	1.23	10.785
34	Rec198	?	Rec60	1	1	36	17	5	1.00	12.695
35	Rec149	?	Rec116	1	1	32	15	8	0.84	14.135
36	Rec122	?	Rec86	1	1	26	13	4	0.69	15.510
37	Rec86	?	Rec35	1	1	38	20	8	0.54	17.220
38	Rec128	?	Rec18	1	1	22	10	2	0.51	17.625
39	Rec79	?	Rec24	1	1	30	14	3	0.39	18.930
40	Rec186	?	Rec115	1	1	26	14	4	0.37	19.240
41	Rec185	?	Rec89	1	1	28	14	6	0.35	19.455
42	Rec154	?	Rec83	1	1	34	16	4	0.24	20.740
43	Rec143	?	Rec58	1	1	32	15	5	0.18	21.795

Clustered groups of pairwise matches

Group		
GP1	Rec182	Rec2
GP2	Rec181	Rec1
GP3	Rec183	Rec3
GP4	Rec186	Rec6
GP5	Rec184	Rec4
GP6	Rec189	Rec9
GP7	Rec195	Rec15
GP8	Rec198	Rec18
GP9	Rec193	Rec13
GP10	Rec187	Rec7
GP11	Rec199	Rec19
GP12	Rec190	Rec10
GP13	Rec194	Rec14
GP14	Rec197	Rec17
GP15	Rec196	Rec16
GP16	Rec192	Rec12
GP17	Rec200	Rec20
GP18	Rec188	Rec8
GP19	Rec185	Rec5
GP20	Rec191	Rec11

7.0 Reference

Macbeth GM, Broderick D, Ovenden JR, Buckworth RC (2011) **Likelihood-based genetic mark-recapture estimates when genotype samples are incomplete and contain typing errors.** Theoretical Population Biology doi:10.1016/j.tpb.2011.06.006

APPENDIX 1

Random number copyright notice

Appendix 2

Frequently asked questions

How do I cite the program?

Please use the reference listed in section 7.0 of this document.

How do I estimate recapture numbers between populations 1 and 2 when there are matches within populations?

Each analysis is unique but in most cases it may be best to form composite genotypes within populations prior to determining matches between populations.

How do I estimate recapture numbers in inbred populations?

SHAZA currently assumes outbred populations with inbreeding expected to increase the proportion of false positives in the data. We hope to address this issue in the future.

Can I analyse more than 10,000 samples?

The compiled version is restricted to 10,000 samples. The code can be simply recompiled to run up to 92,682 samples in most computers (i.e. those with a maximum *unsigned int* value of 4,294,967,295). Substantially more samples are possible by recoding using *unsigned long long int*.

How do I correct for increased number of false positives with a larger sample size?

SHAZA automatically adjusts for false positives with the increased sample size in your data.

Can I split my data up into small populations for separate analysis?

We recommend not to split up your data into small groups for separate analysis as this will lead too many matches occurring by chance across all the multiple analysis.

I have four genotypes with six pairwise matches between them. Will this affect the corrected number of matches?

SHAZA was designed for low numbers of recaptures. Care must be taken to interpret triplets or higher number of match pairs. It may be sufficient to form composite matches between them e.g. Wolf scats within a temporal group (population).

Can I use SHAZA to help with experimental design?

The built in simulation options of SHAZA make it a powerful tool for this purpose.

Can SHAZA analyse more than 32 loci?

SHAZA does not make extensive use of dynamically allocated memory. We attempted to balance the needs for most users while not excessively utilising too much computer memory. This means that a fixed size in a number of fields was used (Table 6). Greater than 32 loci will also require some programming as it was dependent on 32bit architecture.

Would it be better to use the largest values permitted values of –s and –n to improve estimates?

The default values are thought to balance speed and accuracy particularly in large files such as 8000 genotypes which may take many hours to run. We give users the option of experimenting by varying these parameters when analysing their own data.

Can SHAZA determine pedigree relationship in a population?

This is currently under development and follows that same methodology as recaptures which are effectively monozygotic twins.

I have found SHAZA to be an excellent tool but I would like it to also implement …?

If you have any constructive suggestions that can be used to improve SHAZA please contact the corresponding author.

This page intentionally left blank.