AFLPScore version 1.4 Read-Me

Authors

Raj Whitlock, Helen Hipperson, Maria-Elena Mannarelli, Department of Animal and Plant Sciences, University of Sheffield, Sheffield S10 2TN
r.whitlock@sheffield.ac.uk

What does AFLPScore do?

AFLPScore implements methods of scoring data for dominant molecular markers (e.g. AFLPs, RAPDs, ISSRs) in an objective, flexible and repeatable fashion. More specifically the program interprets PCR-product fluorescence (peak height) data matrices created by software such as Applied Biosystems's GeneMapper and converts them into presence-absence (1-0) phenotypes tables. AFLPScore is an interactive scripting program, written in R, which operates within the R statistical software environment.

AFLPScore has three main functions:

Normalisation of peak-height data between different fingerprints or profiles to smooth differences in gross fingerprint fluorescence intensity or brightness
Determination of optimum scoring conditions by phenotype calling (estimating the presence or absence of bands or peaks) and error rate analysis, based on the normalised data
Phenotype calling from the normalised data to create a final phenotypes table, based on the optimised scoring conditions

Definitions

Fingerprint	Collection of PCR fragments or bands stemming from AFLP, RAPD or ISSR analysis of the DNA from a single individual
Locus	Denoting PCR fragment(s) that share a common fragment length class and that are assumed to result from PCR amplification at a unique genomic location
Peak-height	The maximum fluorescence intensity or "brightness" of a single PCR fragment or band within a fingerprint
Phenotype	Denoting either the presence (1) or absence (0) of a single PCR fragment or band.

Data normalisation

The first stage in AFLPScore analysis is the normalisation of the raw peak-height data. The total fluorescence intensity (or brightness), i , of the fingerprint of each individual is calculated as the sum of the height of every peak it possesses. The median value of fingerprint intensity is calculated across all the samples in the data table, m. A normalisation factor is calculated for each individual as $\frac{m}{i}$. After this the peak heights for

each individual are multiplied by their normalisation factor, to create a normalised peak height table. In time, we hope to implement different sorts of normalisation, such as normalisation to the maximum, or median signal in a sample.

Phenotype-calling

AFLPScore calls phenotypes (1s & 0s) by a two-step process, first of all determining which loci are suitable for inclusion in the data (based on the heights of the peaks that represent them) and then calling phenotypes at the retained loci (based on their peak heights).

The locus-selection threshold

The locus selection threshold determines which loci are to be included in the final phenotypes table. For each locus ("gel" position at which peaks are to be scored) the mean height of all peaks present is calculated (from the normalised data). Loci whose mean peak height is low may be more difficult to score reproducibly, as for some individuals drop-out may occur due to low intensity and non-detection of peaks. The locus selection threshold is simply a cut off point applied to the mean peak height data such that loci with values equal or greater to the threshold are retained for analysis. The remaining loci are rejected. AFLPScore keeps a record of which loci have been retained and removed by this threshold and outputs it as a text file.

The phenotype-calling thresholds

Phenotype-calling thresholds are for converting the peak height data at retained loci into a binary presence-absence (1-0) matrix of phenotypes. They define what constitutes a peak, and what does not. Within AFLPScore, you have a choice of applying either an absolute phenotype-calling threshold or a relative one. The absolute threshold compares each peak height to a single user-defined value. Peaks with a height equal to or greater than this value are scored as a "1" phenotype. All other peaks, or instances where no peak has been detected are scored as a "0" phenotype. The relative threshold establishes an individual scoring threshold for each locus. The threshold for each locus is a value that is related to the mean peak height for each locus via a user specified percentage value. For example, if a user specifies a relative threshold of 20%, the threshold for each locus will be a value that is 20% of each locus' mean peak height. Again, peaks within a locus whose height equals or exceeds the threshold value for that locus are scored as present ("1"). Other peaks or peak absences are scored as absent ("0").

Data filtering using the phenotype-calling threshold

When the data contain a high frequency of PCR noise peaks with low peak-height, it is possible that the locus-selection threshold will introduce bias in the selection of loci for further analysis. This is because the mean peak-height for loci with only a few "real" peaks will be dragged down by the low peak heights of the spurious peaks. These loci can then be excluded resulting in possible bias in allele frequency amongst the retained loci. To get round this potential problem, AFLPScore employs an optional (but recommended) filtering step to remove the noise peaks from the analysis *before* the locus-selection threshold is implemented. This filtering step works by using the phenotype-calling threshold as a cut-off below which peaks are classified as noise, and discounted. Locus selection can then proceed based on only the peaks that exceed the phenotype-calling threshold. In this way,

loci can be more sensitively phenotyped, with less bias in locus selection.

Choosing thresholds

After data normalisation, AFLPScore automatically generates a histogram of the mean peak heights of each locus, and reports a grand mean peak height (mean of the mean peak height at individual loci, across all loci). This output indicates the scale and range of peak heights at loci. Next histograms and density plots of peak heights are presented for each locus individually (Fig. 1). The locus name and the number of peaks that have contributed to each graph are shown on each plot. The plots can be used to assess whether PCR noise peaks may be present in the data and which threshold value might be appropriate to deal with them. A good starting point for choosing the locus selection threshold is a value around 20-25% of the grand mean peak height. An appropriate starting point for the allele calling threshold might be 3-5% of the grand mean peak height for an absolute threshold, or 10 -20% for a relative threshold. The locus selection threshold can be altered in the light of how many loci it excludes. However, ultimately, it is best to use error rate estimation to optimise both thresholds (see below). The optimum thresholds are those that minimise the error rate. Molecular marker data from tissue samples that have been subjected to repeated DNA extraction can be used to calculate the error rate. We advise using data from at least 20 individuals representative of the scope of the study you are carrying out, analysed in duplicate (n = 40 in total).

Error rate estimation

AFLPScore can calculate Bayesian (after Hadfield *et al*., 2006; Hadfield *et al*., in prep) and mismatch error rates for a range of user-specified locus-selection and phenotype-calling thresholds. Each possible combination of the specified thresholds is used to call phenotypes for the input file and then error rates are estimated from the duplicated samples within the created phenotypes table. The purpose of this option is to allow the easy identification of optimum values for thresholds.

The Bayesian error rate describes the error process at the allelic level. This may be important because error rates could be asymmetric between the two AFLP alleles “0” and “1” that underlie the observed phenotypes (Hadfield *et al*., in prep). For example, zero homozygotes (band absence) may be very hard to miss-score, whilst heterozygote band presence (1,0) may be more easy to miss-score than homozygote band presence (1,1). The Bayesian error rate quoted in AFLPScore is the probability of miss-scoring a 1 allele as a 0 allele.

The mismatch error rate is the percentage of differences in phenotype amongst the replicates of the duplicated sample. Imagine 10 individuals are fingerprinted twice (20 profiles) at 100 loci. There are 10 mismatches (10 instances where a locus shows a mismatch in phenotype within a replicate pair). The error rate is 10/(10*100)=1%.

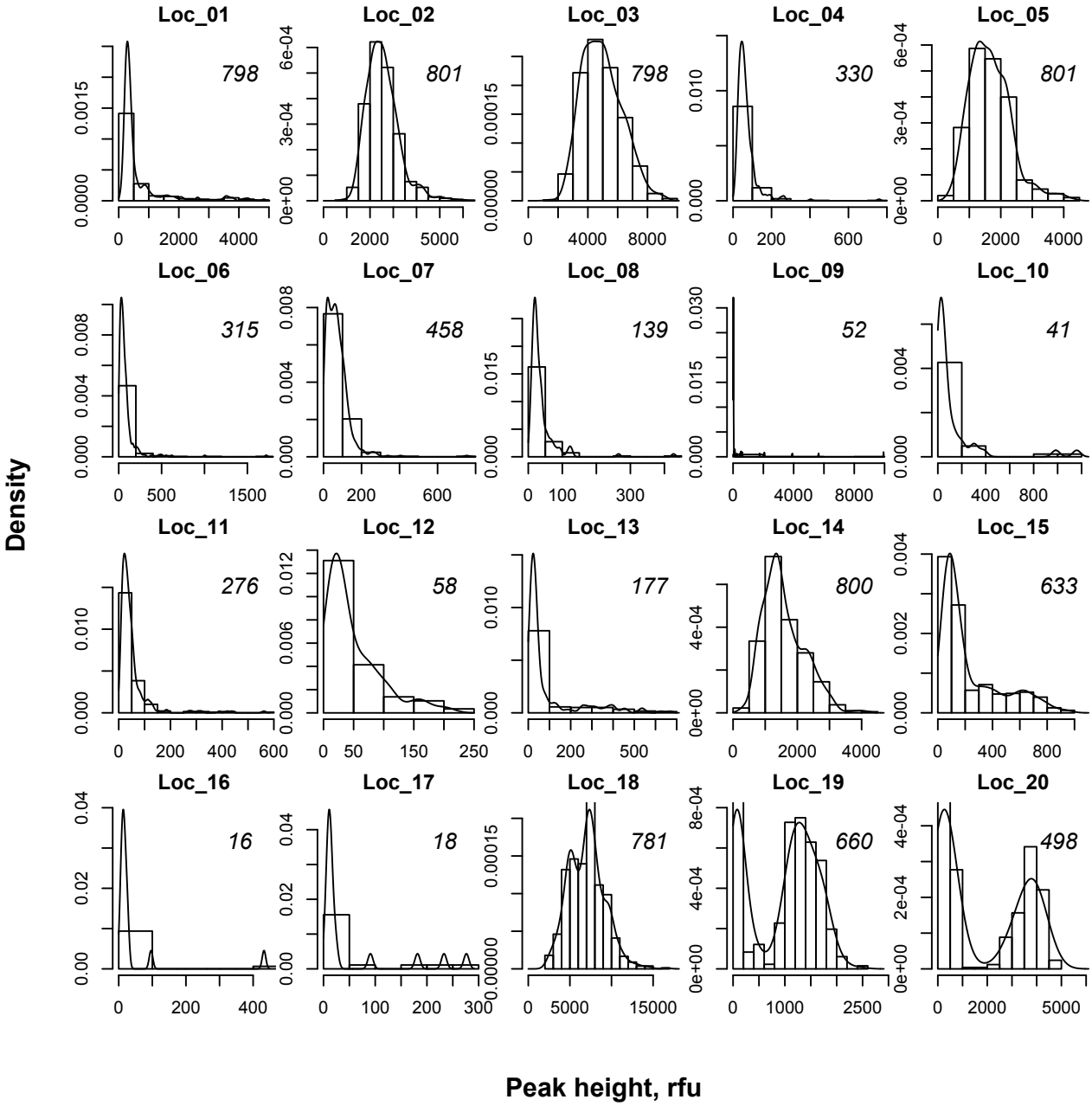
After error rate estimation for every combination of user-specified scoring thresholds, AFLPScore plots a graph of the error rates (Fig.2)

The error rate analyses assume that:

**The bayesian simulation of error rates works correctly (MCMC chain convergence etc)

**i.e. no model checking is currently carried out in this method

Fig. 1 *Histograms and density plots of peak heights of all peaks occurring at individual loci. Locus names are given at the head of each plot, and the number of peaks contributing to each plot is indicated at the top right of each plot.*



Other functions

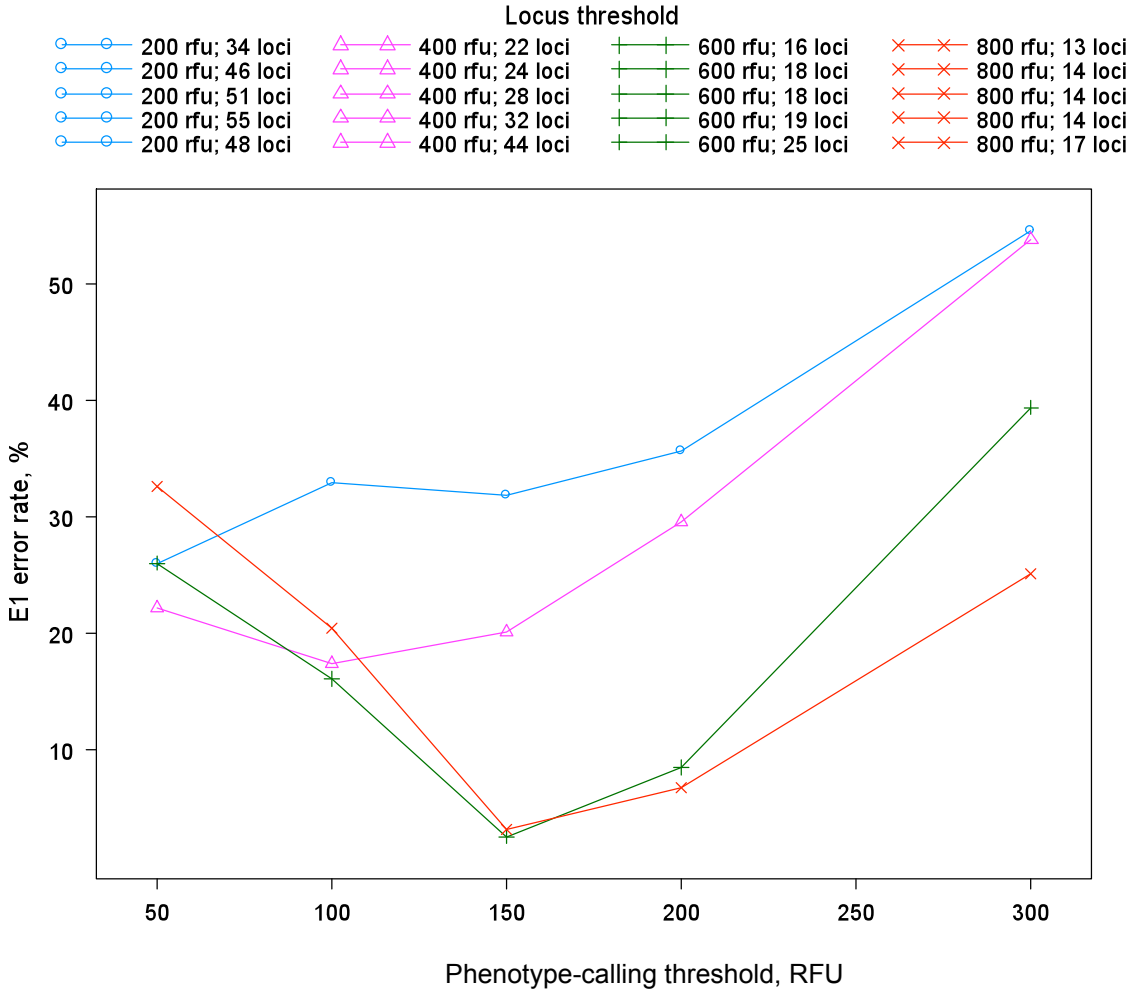
AFLPScore can normalise data to a user specified value. This function is of use where phenotype data have already been generated by the program at one point in time, but now the user wishes to analyse a second set of data using the same molecular marker or primer combination. The problem here is that the two sets of data might have different fingerprint intensities. Even using the same thresholds, this might result in differences in

scoring between the two data sets. User-specified normalisation allows the second dataset to be normalised to the same point as the first data set. Therefore the scoring should be comparable between the datasets for the same thresholds. An alternative solution is to combine the data in a single table and analyse them together.

AFLPScore can restrict analysis to a user-specified set of loci. This function is of use in ensuring the comparability of different datasets (e.g. different field seasons). The user supplies a list of loci to be analysed as a tab-delimited text file. All other loci are excluded. Further loci may be excluded by the locus selection threshold.

AFLPScore can override the locus selection threshold through the user setting it to zero. Overriding the locus selection threshold means that all loci will be retained in the project other than those specified for exclusion by the user. The main aim of this feature is to allow control over the loci for which you generate data. When you override the locus selection threshold and specify a set of loci for analysis, you can control exactly the loci for which you obtain data. However, such a course may lead to the generation of data of a lower quality than that obtained when using a locus selection threshold. This function may also be of use if you believe your marker data are of very even and high quality between loci within fingerprints (e.g. very even peak heights across an AFLP fingerprint).

Fig. 2 *Plot of error rates calculated during AFLPScore error rate analysis. The legend gives locus-selection threshold and the numbers of retained loci*



Bugs fixed in AFLPScore version 1.3b

24/04/08 *AFLPScore 1.3b released. This version of AFLPScore corrected the following bugs:*

***The program hangs when the user chooses to normalise data to user-specified peak height value (now fixed)*
***Rows containing a sample name but no peaks cause the program to crash (these rows are now deleted (avoiding the crash) but their corresponding sample names are reported on screen and in the log)*

AFLPScore 1.3b differed from previous versions in several important respects:

***When the user chooses to calculate "mismatch" error rates, "Bayesian" error rates are not now calculated simultaneously, and do not appear in the error rate output table. This modification markedly speeds up error rate estimation when error rates are "mismatch".*
***Tidying up: AFLPScore tidies up the workspace (removes objects from the workspace) without creating error messages. N.B. if objects are present in the workspace before AFLPScore is run, then these will be deleted. AFLPScore retains the following objects in the R workspace (if the user has chosen to create them in the first place):*

gen (Binary phenotypes table)
AFLPScore (internal function from AFLPScore script)
AFLPError (internal function from AFLPScore script)
PlotError (internal function from AFLPScore script)

16/01/09 AFLPScore 1.4a released. This version of AFLPScore no longer depends on the DAAG or gdata packages. A bug that prevented the "restrict loci" option from working has been corrected. The script has been tested within the current Mac OS X version of R.

16/01/09 AFLPScore 2.0a UNDER DEVELOPMENT. This version of AFLPScore will (hopefully) improve the sensitivity (more loci, lower error rate) with which marker data can be scored.

Preparing to use the AFLPScore R-script

1. Install the R statistical computing software (www.r-project.org)
2. Install the following packages into R (packages listed below `MasterBayes` are usually installed automatically in Windows, upon installation of `MasterBayes`):

```
MasterBayes
coda
combinat
gtools
mvtnorm
genetics
```

3. Save the R-script file AFLPScore_vers1.3.R to a sensible location, possibly within the R program folder
4. Open R
5. Change the working directory of R to that containing the peak-height/fluorescence-intensity table to be analysed (File/Change dir). If you want to use a file to restrict the AFLP loci that you use, then this file must also be stored here. This folder will also eventually contain output from AFLPScore.

Format of the peak heights input file

- Input files should be in a text (tab delimited) format (.txt)
- The input file should be a table of non-normalised peak heights for all AFLP loci that were scored
- The first row should contain locus names
- The first column should contain the names of each individual for which an AFLP fingerprint has been observed
- Do not include any spaces in the row or column names for the input table (or anywhere else)
- An empty cell is acceptable as the top-left cell
- Band absence should be coded as an empty cell rather than "0". This is the default output from the “genotypes” table in ABI’s GeneMapper software
- Band presence should be coded as the intensity in relative fluorescence units (peak height) of a marker peak or band
- For locus names I advise using the following convention:
    - “Size of AFLP fragment in base pairs_Primer combination_Species”
    - e.g. 346_TCA_CCG_Ch (*Cirsium heterophyllum*, primer set TCA-CCG, 346 base- pairs)
    - Additional text could be added to denote the AFLP enzyme system in use
- If the data are to be used for error rate analysis as well as phenotype-calling, the input file must contain the duplicate data at the top of the data file, followed by the remaining unique profiles.
- The duplicated samples must occur in order such that pairs of profiles originating from the same individual occur next to each other.
- The names of duplicated samples originating from the same individual must share identical sample names
- An example file is supplied (peak_heights.txt).

Format of the file to specify analysis of a subset of loci

This file should be a text (tab-delimited) file that contains a single column of locus names, with no header row. R is case sensitive, so the locus names must match those given in the peak heights input file exactly. An example file is supplied (restrict.txt) that restricts loci in the file peak_heights.txt.

Interacting with AFLPScore at the command line in R

6. Change the working directory of R to the folder that contains the peak-heights table to be analysed
7. Source the AFLPScore script in R:

File/Source R Code…
Browse to the AFLPScore script and click OK

8. Follow instructions on the screen, and, with the exception of file names and threshold values, please respond to the questions in lower case and using a single character (e.g. “1”, “2”, “y”, “n”).

Inputting your peak heights table

9. Enter the file name for the input table at the prompt

Control of data normalisation

10. Decide whether you want to normalise the peak height data to a specified value. If you chose “y” then provide the value for normalisation

Subsetting your input peak-heights table

11. Decide whether you want to restrict analysis to a subset of loci. If you chose “y” then provide the name of the file that specifies the loci you want to select

Interpreting the peak-height histograms and summary statistics

Assess the peak-height histograms and summary statistics to determine a range of locus-selection and phenotype-calling thresholds for testing under error rate analysis or for final phenotype calling (see above for guidance). Any of the graphs produced in R can be saved during analysis by clicking on them to bring them to the front and then right-clicking them and choosing to save them. At this stage the median fingerprint intensity is also reported. This value is the one to which you might want to normalise future marker data for the same primer-combination.

Error rate analysis

12. Decide whether or not to carry out the error rate analysis or to proceed immediately to final phenotype-calling
13. Choose either absolute or relative methods for phenotype calling
14. Decide whether or not you want to apply the filtering option
15. Enter the number of duplicate fingerprint pairs within the input file
16. Enter a list of locus-selection thresholds. The list should not contain an “and” or an “&”, but may or may not contain spaces e.g.:

    Either “200, 400, 600, 800” or “200,400,600,800” are acceptable here
    The list of thresholds must not end with a comma or a comma followed by spaces.

17. Enter a list of phenotype-calling thresholds. The list should not contain an “and” or an “&”, but may or may not contain spaces e.g.:

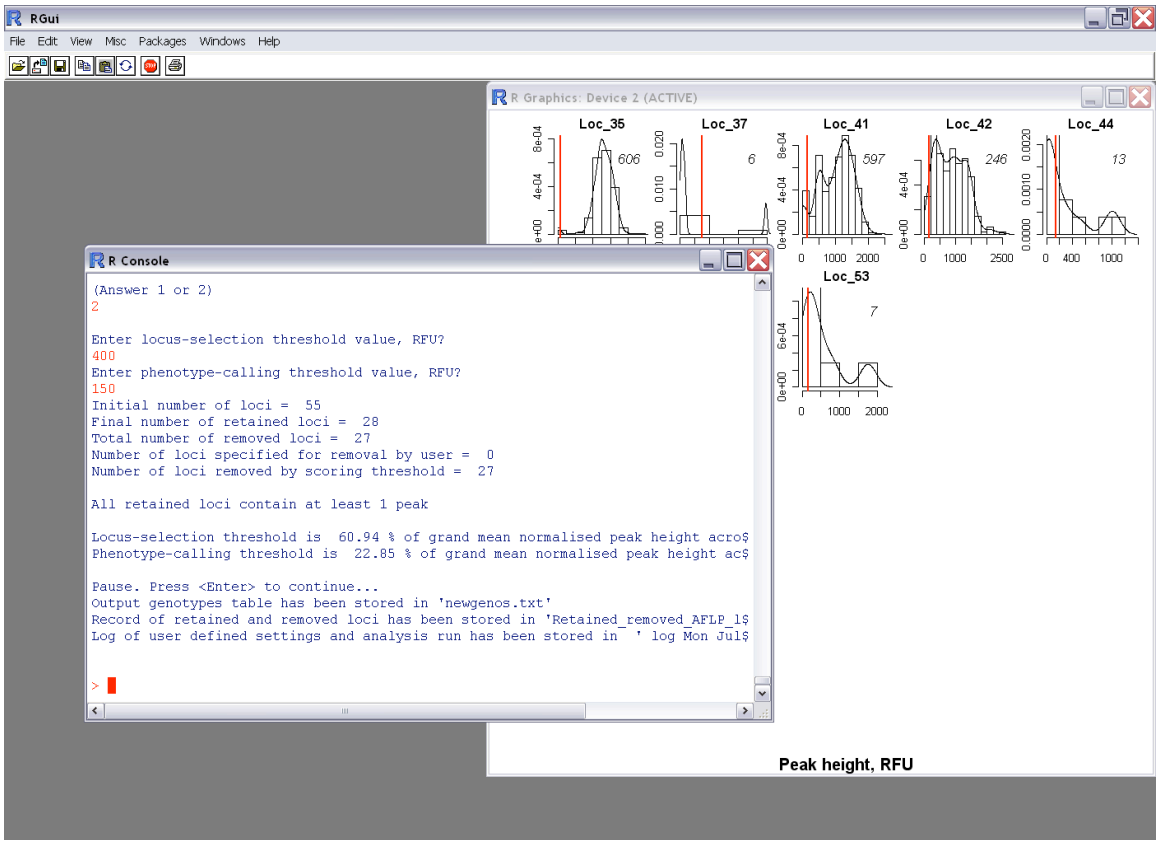
    “50, 100, 200, 300, 400” is acceptable for an absolute threshold
    “1,2,5,10,15,20,50,100” is acceptable for a relative threshold (corresponding to 1%, 2%, 5% etc.)
    Neither of these lists of thresholds may end with a comma or a comma followed by spaces.

18. Enter the method of error rate estimation to be included in the error rates plot (if you choose “mismatch” only mismatch error rates will be calculated (quicker); if you choose “bayes” both error rates are calculated (slower))
19. Decide whether you wish to save the results of the error rate analysis to an output text file
20. If you chose to save the results, enter the file name to which the results are to be saved
21. If you chose “mismatch” above, then Bayesian error rates will not appear in this table

Final phenotype calling from the input file

Decide whether you wish to carry out the final phenotype calling step
Choose either absolute or relative methods for phenotype calling
Decide whether or not you want to apply the filtering option
Enter a single locus-selection threshold
Enter a single phenotype-calling threshold
The script will now complete the analysis of the peak-heights table you provided.

Fig 3 *Screen-shot showing AFLPScore in use within the R software*



In this final stage of the analysis, graphs of the peak-height distribution of the retained loci are given along with the position of the selected phenotype-calling threshold as a red vertical bar. Various statistics are also reported on-screen, such as the number of loci that were retained in the analysis, and the value of the thresholds relative to the grand mean peak height. A warning is given if some loci included in the phenotypes table do not

contain any peaks, and these loci are reported. A warning is given if some data rows didn't contain any peaks at all (these rows are deleted from the data and their sample names are reported). All the information reported on screen is also output as a text log file (with the exception of the peak-heights histogram and other graphs). A record of the loci retained or excluded at the various stages of the analysis is also provided as a text file.

Remember to rename and store the output phenotypes file and locus record file so that it doesn't get overwritten next time. The name of the log file will always be unique, and so will never be overwritten.

The R object called "gen" created by this script (which is the phenotypes table output to a text file), is in the correct format for downstream manipulation by the collection of AFLP data processing functions for R called AFLPdat.

Strategy for carrying out error rate analysis

I strongly recommend using the "filtering" option that can improve the number of loci retained for similar or lower error rates
Start with mismatch error rates (for a faster analysis) and try a wide range of scoring parameters
The range of specified locus-selection thresholds should initially span the interval 0 to twice the mean normalised peak height of the data
The range of specified absolute phenotype calling thresholds (if chosen) should initially span the range 0 to the mean normalised peak height of the data
The range of a relative phenotype calling threshold should initially span the range 0 to 100%
If it is possible to identify an area of threshold values that may contain a minimum desired error rate then the error rate analysis can be repeated, choosing new threshold values that will identify the minimum value more clearly
Repeat the last step as desired
Once the location of the minimum error rate has been identified, repeat the error rate analysis to compute Bayesian error rates, if these are needed

Disclaimer

We take no responsibility for erroneous data generated as a result of using the AFLPScore R-script, whether through bugs in the script itself or through its misapplication. It is always wise to make some checks of called phenotypes against the original chromatograms to make sure that the two tally to an acceptable degree. We also note that a critical step in obtaining good phenotype data is to have a robust and carefully checked bin-set for calling the peak height data that AFLPScore uses as an input.

Citing AFLPScore

AFLPScore may be cited as:

Whitlock R, Hipperson H, Mannarelli, M, Butlin, RK & Burke, T (2008) An objective, rapid and reproducible method for scoring AFLP peak-height data that minimizes genotyping error. *Molecular Ecology Resources*, **8**, 725-735.

Bugs, extensions and updates

Please report any bugs or errant behaviour of AFLPScore to Raj Whitlock. Please also contact Raj if you have suggestions on modifications to AFLPScore that would be useful to you, or if you wish to collaborate on extending the development of AFLPScore. New versions of AFLPScore may be developed from time to time, please see the website or contact Raj for details.