

qfingerprinting analysis.r¹ manual

by Alban Ramette

October 13, 2008

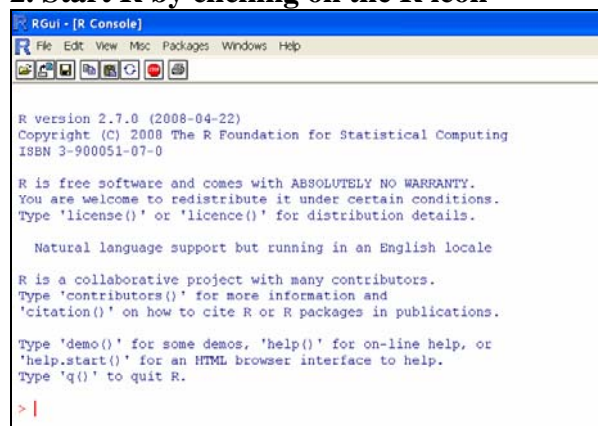
1. Preparation of the data table

See the Excel table provided as example ([D_for_qfingerprinting_analysis.xls](#)). Make sure while preparing the data that the row and column labels do not contain spaces.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	bins	01_cm_A	01_cm_A101_cm_A201_cm_A301_cm_A401_cm_A601_cm_B	01_cm_B101_cm_B201_cm_B301_cm_B401_cm_B501_cm_B601_cm_C	01_cm_C101_cm_C201_cm_C301_cm_C401_cm_C601_cm_D	01_cm_D101_cm_D201_cm_D301_cm_D401_cm_D601_cm_E	01_cm_E101_cm_E201_cm_E301_cm_E401_cm_E601_cm_F	01_cm_F101_cm_F201_cm_F301_cm_F401_cm_F601_cm_G	01_cm_G101_cm_G201_cm_G301_cm_G401_cm_G601_cm_H	01_cm_H101_cm_H201_cm_H301_cm_H401_cm_H601_cm_I	01_cm_I101_cm_I201_cm_I301_cm_I401_cm_I601_cm_J	01_cm_J101_cm_J201_cm_J301_cm_J401_cm_J601_cm_K	01_cm_K101_cm_K201_cm_K301_cm_K401_cm_K601_cm_L	01_cm_L101_cm_L201_cm_L301_cm_L401_cm_L601_cm_M	01_cm_M101_cm_M201_cm_M301_cm_M401_cm_M601_cm_N	01_cm_N101_cm_N201_cm_N301_cm_N401_cm_N601_cm_O	01_cm_O101_cm_O201_cm_O301_cm_O401_cm_O601_cm_P
2	101	0	0	0.23	0	0	0	0	0	0	0	0	0	0	0	0	0
3	103	0	0	0	0	2.56	9.56	0	0	0	2.28	0	0	0	0	0	0.26
4	105	0	0	0	0	0	0	0	0	0	0	0	0	0	0.81	0	0
435																	
436	991	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
437	993	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
438	995	0.19	0.45	0.75	0	0	0	0	0.55	0.53	0	0	0	0	0.21	0.29	0
439	997	0	0	0	0	0	0	0.19	0	0	0	0	0	0	0	0	0
440	Dilution	0	1	2	3	4	6	0	1	2	3	4	5	6	0	1	2
441	Sample	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
442																	

In this example, the first row contains the sample names, the first column contains the bin names and the last two rows contain the dilution level (0 to 6) and the sample identification number (1 to 9). The whole table (including the row and column labels) is then copied to a text file (e.g. using Notepad). An example is given in the [Dforqfingerprint.txt](#) file.

2. Start R by clicking on the R icon



3. Load the data into the R workspace

At the prompt (>), indicate in which directory you want to read and write the data (i.e. where you also put your .txt file. The directory should be created beforehand) and press enter. Note that quotes and \\ are used to indicate the path to the directory.

```
>setwd("c:\\R\\ARISA")
```

Then, load the data into the object D by writing (make sure to exactly type the dots, commas, and punctuation signs, as indicated and use " instead of "):

¹This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but **without any warranty**; without even the implied warranty of **merchantability** or **fitness for a particular purpose**. See the GNU General Public License for more details (Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA)

```
> D=read.table("Dforqfingerprint.txt",h=TRUE,row.names=1)
```

If you now type:

```
> D
```

You should see your data table appear in the R console (if it is a big table, you will not see the first rows, but just the end of the table):

923	0.00	0.00	0.00
925	0.00	0.00	0.00
927	0.00	0.00	0.00
929	0.00	0.00	0.00
931	0.10	0.00	0.44
933	0.00	0.00	0.00
935	0.00	0.00	0.00
937	0.00	0.00	0.00
939	0.00	0.00	0.00
941	0.24	0.00	0.27
943	0.00	0.00	0.00
945	0.00	0.00	0.00
947	0.00	0.00	0.00
949	0.00	0.00	0.00
951	0.00	0.00	0.00
953	0.00	0.00	0.00
955	0.00	0.00	0.00
957	0.00	0.00	0.00
963	0.00	0.00	0.00
965	0.00	0.00	0.00
967	0.00	0.00	0.00
969	0.00	0.00	0.00
971	0.00	0.00	0.00
973	0.00	0.00	0.00
975	0.00	0.00	0.00
977	0.00	0.00	0.00
979	0.00	0.00	0.00
983	0.00	0.00	0.00
985	0.00	0.00	0.00
987	0.00	0.00	0.00
989	0.00	0.00	0.00
991	0.00	0.00	0.00
993	0.00	0.00	0.00
995	0.00	0.00	0.00
997	0.00	0.00	0.00
Dilution	0.00	1.00	2.00
sample	0.00	0.00	0.00

In our example, the table was produced in Excel in a OTUs-by-samples format (because Excel is limited to 256 columns). We need to transpose the data to obtain a samples-by-OTUs table and save it under the same name D:

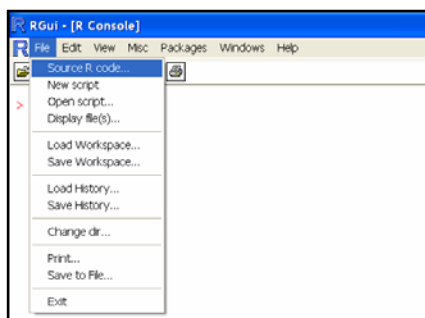
```
> D=t(D)
```

If the table was initially in the right format, there is no need to transpose the table. An error message may appear if the sample direction in the table is not correct. It is therefore always recommended to double-check the table orientation to avoid wrong calculations.

We are now ready to run the R qfingerprinting script on the data.

4. Running the qfingerprinting script

In the menu bar, go to Source R code...



```
> D=read.table("Dforqfingerprint.txt",h=TRUE,row.names=1)
> source("C:\\R\\ARISA\\qfingerprinting.r")
> |
```

And indicate the location of your saved version of the **qfingerprinting.r** script.

Apply the function to the data **D** and store the results in the object **Dres**:

```
> Dres= qfingerprinting(D)
```

The script starts by indicating some basic information about the version, expected data format and ask if you want to proceed. Type “y” (without the quotes), if the data table corresponds to the description provided, otherwise type “n” and see point 3) above.

```
> Dres= qfingerprinting(D)
qfingerprinting function v.1.3. by A. Ramette
D contains the samples-by-OTUs table with the last 2 columns corresponding
to the Dilution_order (e.g. 1..6), Sample_order (e.g. 1..n).
D must be a complete table containing all replicates and all dilution levels
(i.e. missing replicates or dilution levels have to be present as a series of 0.
Continue? (y/n)..... |
```

After few seconds, the following message should appear in the R console, indicating that the calculations are finished:

```

RGui - [R Console]
File Edit View Misc Packages Windows Help

> Dres= qfingerprinting(D)
qfingerprinting function v.1.3. by A. Ramette
D contains the samples-by-OTUs table with the last 2 columns corresponding
to the Dilution_order (e.g. 1..6), Sample_order (e.g. 1..n).
D must be a complete table containing all replicates and all dilution levels
(i.e. missing replicates or dilution levels have to be present as a series of 0.
Continue? (y/n)..... y
Done.
> |

```

By typing:

>Dres

You will get the whole table converted into \log_{10} abundance values (see reference for detailed explanation about how the consensus and continuity rules are applied to convert peak presence into a \log_{10} abundance table).

The screenshot shows the RGui console displaying a large table of \log_{10} abundance values. The table has 9 rows and 100 columns. The first row contains values from 5 to 743. The second row contains values from 6 to 743. The third row contains values from 7 to 743. The fourth row contains values from 8 to 743. The fifth row contains values from 9 to 743. The sixth row contains values from 1 to 743. The seventh row contains values from 2 to 743. The eighth row contains values from 3 to 743. The ninth row contains values from 4 to 743. The tenth row contains values from 5 to 743. The eleventh row contains values from 6 to 743. The twelfth row contains values from 7 to 743. The thirteenth row contains values from 8 to 743. The fourteenth row contains values from 9 to 743. The fifteenth row contains values from 1 to 743. The sixteenth row contains values from 2 to 743. The seventeenth row contains values from 3 to 743. The eighteenth row contains values from 4 to 743. The nineteenth row contains values from 5 to 743. The twentieth row contains values from 6 to 743. The twenty-first row contains values from 7 to 743. The twenty-second row contains values from 8 to 743. The twenty-third row contains values from 9 to 743. The twenty-fourth row contains values from 1 to 743. The twenty-fifth row contains values from 2 to 743. The twenty-sixth row contains values from 3 to 743. The twenty-seventh row contains values from 4 to 743. The twenty-eighth row contains values from 5 to 743. The twenty-ninth row contains values from 6 to 743. The thirtieth row contains values from 7 to 743. The thirty-first row contains values from 8 to 743. The thirty-second row contains values from 9 to 743. The thirty-third row contains values from 1 to 743. The thirty-fourth row contains values from 2 to 743. The thirty-fifth row contains values from 3 to 743. The thirty-sixth row contains values from 4 to 743. The thirty-seventh row contains values from 5 to 743. The thirty-eighth row contains values from 6 to 743. The thirty-ninth row contains values from 7 to 743. The fortieth row contains values from 8 to 743. The forty-first row contains values from 9 to 743. The forty-second row contains values from 1 to 743. The forty-third row contains values from 2 to 743. The forty-fourth row contains values from 3 to 743. The forty-fifth row contains values from 4 to 743. The forty-sixth row contains values from 5 to 743. The forty-seventh row contains values from 6 to 743. The forty-eighth row contains values from 7 to 743. The forty-ninth row contains values from 8 to 743. The fiftieth row contains values from 9 to 743. The fifty-first row contains values from 1 to 743. The fifty-second row contains values from 2 to 743. The fifty-third row contains values from 3 to 743. The fifty-fourth row contains values from 4 to 743. The fifty-fifth row contains values from 5 to 743. The fifty-sixth row contains values from 6 to 743. The fifty-seventh row contains values from 7 to 743. The fifty-eighth row contains values from 8 to 743. The fifty-ninth row contains values from 9 to 743. The sixtieth row contains values from 1 to 743. The sixty-first row contains values from 2 to 743. The sixty-second row contains values from 3 to 743. The sixty-third row contains values from 4 to 743. The sixty-fourth row contains values from 5 to 743. The sixty-fifth row contains values from 6 to 743. The sixty-sixth row contains values from 7 to 743. The sixty-seventh row contains values from 8 to 743. The sixty-eighth row contains values from 9 to 743. The sixty-ninth row contains values from 1 to 743. The seventieth row contains values from 2 to 743. The seventy-first row contains values from 3 to 743. The seventy-second row contains values from 4 to 743. The seventy-third row contains values from 5 to 743. The seventy-fourth row contains values from 6 to 743. The seventy-fifth row contains values from 7 to 743. The seventy-sixth row contains values from 8 to 743. The seventy-seventh row contains values from 9 to 743. The seventy-eighth row contains values from 1 to 743. The seventy-ninth row contains values from 2 to 743. The eightieth row contains values from 3 to 743. The eighty-first row contains values from 4 to 743. The eighty-second row contains values from 5 to 743. The eighty-third row contains values from 6 to 743. The eighty-fourth row contains values from 7 to 743. The eighty-fifth row contains values from 8 to 743. The eighty-sixth row contains values from 9 to 743. The eighty-seventh row contains values from 1 to 743. The eighty-eighth row contains values from 2 to 743. The eighty-ninth row contains values from 3 to 743. The ninetieth row contains values from 4 to 743. The ninety-first row contains values from 5 to 743. The ninety-second row contains values from 6 to 743. The ninety-third row contains values from 7 to 743. The ninety-fourth row contains values from 8 to 743. The ninety-fifth row contains values from 9 to 743. The ninety-sixth row contains values from 1 to 743. The ninety-seventh row contains values from 2 to 743. The ninety-eighth row contains values from 3 to 743. The ninety-ninth row contains values from 4 to 743. The one hundred row contains values from 5 to 743.

The script automatically saves and exports the result table to [Out.txt](#) file of the working directory (e.g. to C:\R\ARISA).

