

USER MANUAL

APPENDIX A

in

“A neuro-fuzzy modeling tool to estimate fluvial
nutrient loads in watersheds under time-varying human
impact”

Rafael Marcé^{1*}, Marta Comerma¹, Juan Carlos García²,
and Joan Armengol¹

¹Department of Ecology, University of Barcelona, Diagonal 645, 08028 Barcelona, Spain

²Aigües Ter Llobregat, Sant Martí de l'Erm 30, 08970 Sant Joan Despí, Spain

*E-mail: rafamarce@ub.edu

April 2004

Contents

1. INTRODUCTION	3
2. MATLAB FILES MANAGEMENT	4
3. DATA PREPARATION.....	5
3.1 MAIN DATA FILE.....	5
3.2 CONTINUOUS INPUT FILE	6
4. STRUCTURE IDENTIFICATION	8
5. PARAMETER ESTIMATION.....	13
5.1 GENERATING TRAINING AND CHECKING FILES.....	13
5.2 CALCULATING LOADS	15
6. RESULTS FILES	18
6.1 LOADSANUAL.DAT	18
6.2 NONSENSEVALUES.DAT.....	18
6.3 FITANDRESIDUALS.DAT.....	19
6.4 PARAMETERSANFIS.DAT	22
6.5 OUTPUTSERIES.DAT	23
7. PERFORMANCE ANALYSIS.....	24
8. SPECIAL ERROR MESSAGE	26

1. INTRODUCTION

As stated in the *Introduction* section of the paper, fuzzy logic are little used in limnology, and almost completely ignored by classical statistics textbooks, and by standard statistical packages. The contents in Appendix A want to fill this gap, providing all tools needed to calculate loads with the method presented in the paper. The only pre-requisite is to have access to the MATLAB basic package plus the MATLAB Fuzzy Logic Toolbox, but no expertise with this software is required.

The complexity of the algorithms needed to apply ANFIS discouraged the authors from writing the codes in a free-access language, to give stand-alone executables. Instead, we wrote codes in MATLAB language, to take advantage of the ANFIS functions present in the MATLAB Fuzzy Logic Toolbox. Since MATLAB is not free software, use of the codes presented here limits to people having access to MATLAB. However, the authors will update this Appendix as soon as free access, powerful ANFIS functions appear (nowadays, only very limited ANFIS functions are available outside the MATLAB Fuzzy Logic Toolbox).

The following sections explain how to use the MATLAB m-files included in the Appendix to calculate constituent annual loads. But it should be stressed that not only loads, but also any regression problem can be analyzed with our procedure, even problems with more than two inputs (i.e. independent variables). However, only one output (i.e. dependent variable) is accepted by our method, and missing values are not allowed.

2. MATLAB FILES MANAGEMENT

Despite we wrote our codes to be used without any MATLAB training, some basic guidelines on MATLAB files management will be given here, because this MATLAB feature frequently disorients beginners.

To work with our m-files and data, we have to place all files in a folder included in the MATLAB working path. The working path is a collection of folders where MATLAB searches files when these are called. Any file located outside this path will be invisible to MATLAB.

The working path is accessible clicking the 'Path Browser' button in the MATLAB command window, or typing 'path' in the prompt. The most practical solution for beginners is to place files in the folder '.../MATLAB/Work', usually the default current directory (i.e. the folder where MATLAB preferentially loads or saves data). If after a run with our m-files the expected output files are not in '.../MATLAB/Work', it means that the current directory is not that folder (but we will find the output files in the folder defined as the current directory, of course!).

The MATLAB working path is easily edited with the 'Path Browser' tool. But we recommend beginners to work with the '.../MATLAB/Work' folder as the working directory, and only edit the MATLAB working path if either this folder is absent or it is not the current directory.

Remember that MATLAB is case sensitive. Thus, any data file or m-file should be named properly when requested (e.g. *Data.dat* and NOT *data.dat*).

3. DATA PREPARATION

Two data files are needed to run the m-files provided in this Appendix. **It is of great importance placing the input variables in the same column order in these two files. Failing to do so will cause nonsense results, or a general collapse during computation.**

A convenient format to work with MATLAB is ASCII. Our experience is that pasting data from a spreadsheet in the Windows Notepad and saving from this application has no conflicts with MATLAB. Remember that all data and m-files should be placed in a folder included in the MATLAB working path.

3.1 MAIN DATA FILE

We have to arrange our basic data in a file with three columns: day number, daily flow [$\text{m}^3 \cdot \text{s}^{-1}$], and mean daily nutrient concentration [$\text{mg constituent} \cdot \text{L}^{-1}$] (e.g. *Data.dat*). The day number must be assigned considering the January 1st of the first year in the database as the day one. Take account of leap years!!

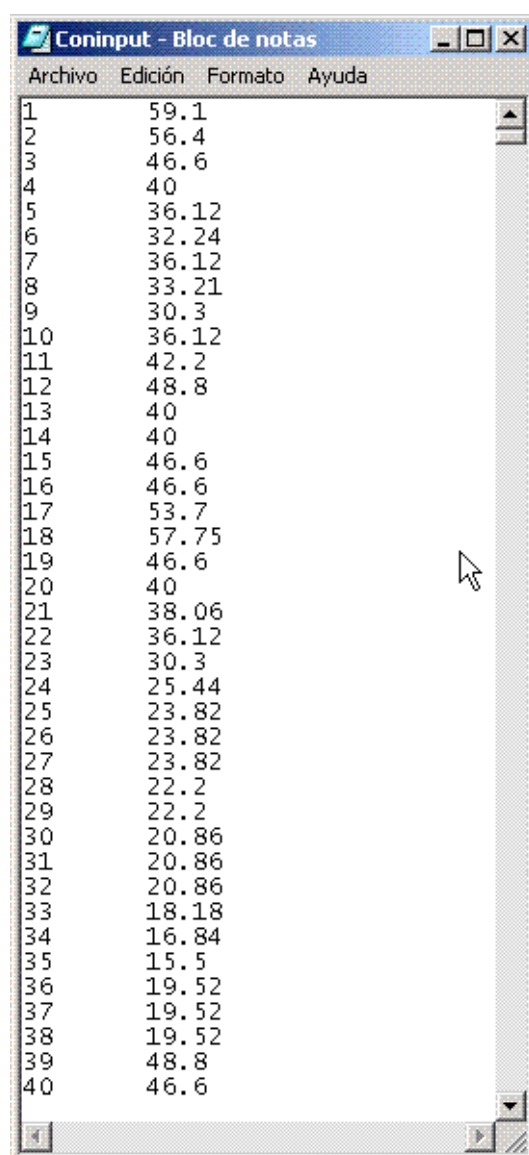
It is of great importance to place the output (nutrient concentration) in the last column, as MATLAB functions expect so. Units should be as above to maintain units coherence in the output files. Any missing value should be present.

9	36.12	0.117000
23	25.44	0.101000
37	19.52	0.107000
51	32.24	0.128000
65	28.68	0.137000
79	40.00	0.080000
100	44.40	0.117000
114	22.20	0.189000
128	78.15	0.084000
144	34.18	0.110000
170	64.50	0.091000
184	34.18	0.176000
200	34.18	0.108000
213	34.18	0.106000
233	16.84	0.127000
247	20.86	0.097000
261	38.06	0.134000
277	18.18	0.158000
291	20.19	0.183000
303	14.40	0.247000
326	15.50	0.273000
340	11.10	0.363000
352	8.00	0.198000
368	16.84	0.338000
382	11.10	0.458000
396	10.55	0.440000
416	6.50	0.633000
444	5.00	0.763000

3.2 CONTINUOUS INPUT FILE

We have shown in the paper that after a relationship (i.e. a FIS) is established between the inputs and the output, we use this fuzzy relationship to obtain a continuous output (i.e. daily constituent concentration) from a continuous input (i.e. daily flow and day number). Thus, we need a file to feed the method with such a continuous input. In the case of annual load calculations, the file must contain a column with daily flows, and a column with day numbers. The day number column will consist in a column covering from January 1st of the first year present in the database to December 31st of the last year in the database. For example, for a

database containing data from two non-leap years, the day number column will take values from 1 to 730. The flow column should consist in a continuous daily record corresponding to the days in the day number column. The file *Coninput.dat* is an example. For applications different than nutrient load calculations, this file could be the Main data file without the dependent variable, or other appropriate collection of inputs.



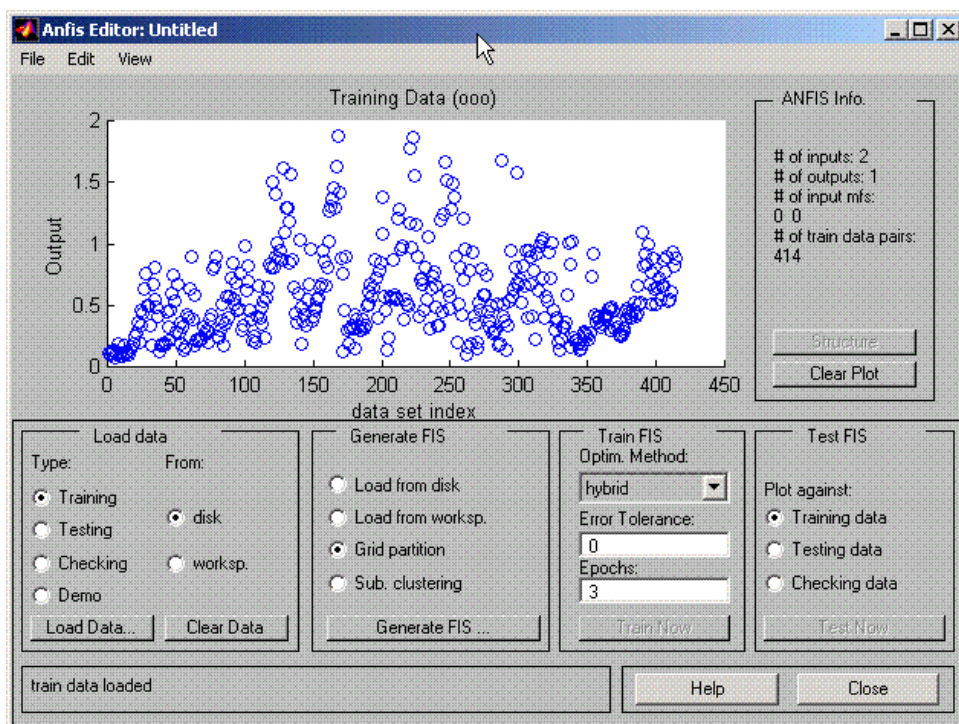
The screenshot shows a Notepad window titled "Coninput - Bloc de notas" with a menu bar containing "Archivo", "Edición", "Formato", and "Ayuda". The text area contains 40 rows of data, each with a day number in the first column and a numerical value in the second column. A mouse cursor is visible over the right side of the text area.

1	59.1
2	56.4
3	46.6
4	40
5	36.12
6	32.24
7	36.12
8	33.21
9	30.3
10	36.12
11	42.2
12	48.8
13	40
14	40
15	46.6
16	46.6
17	53.7
18	57.75
19	46.6
20	40
21	38.06
22	36.12
23	30.3
24	25.44
25	23.82
26	23.82
27	23.82
28	22.2
29	22.2
30	20.86
31	20.86
32	20.86
33	18.18
34	16.84
35	15.5
36	19.52
37	19.52
38	19.52
39	48.8
40	46.6

4. STRUCTURE IDENTIFICATION

In this section we explain how to use a graphical user interface included in the MATLAB Fuzzy Logic Toolbox to solve the structure identification. The basic purpose is to answer the question: *How many MFs are necessary for each input variable?*

- i) Launch MATLAB and type *anfisedit*. A graphical interface to work with ANFIS starts.
- ii) Click *Load data...* with the *Training* option active. Load the Main data file (e.g. *Data.dat*). The output variable appears plotted on the screen.



- iii) Click *Generate FIS...* with the *Grid partition* option active.

A new window appears. In the Input MF Type box choose

gaussmf, and in the Output MF Type box choose *constant*.

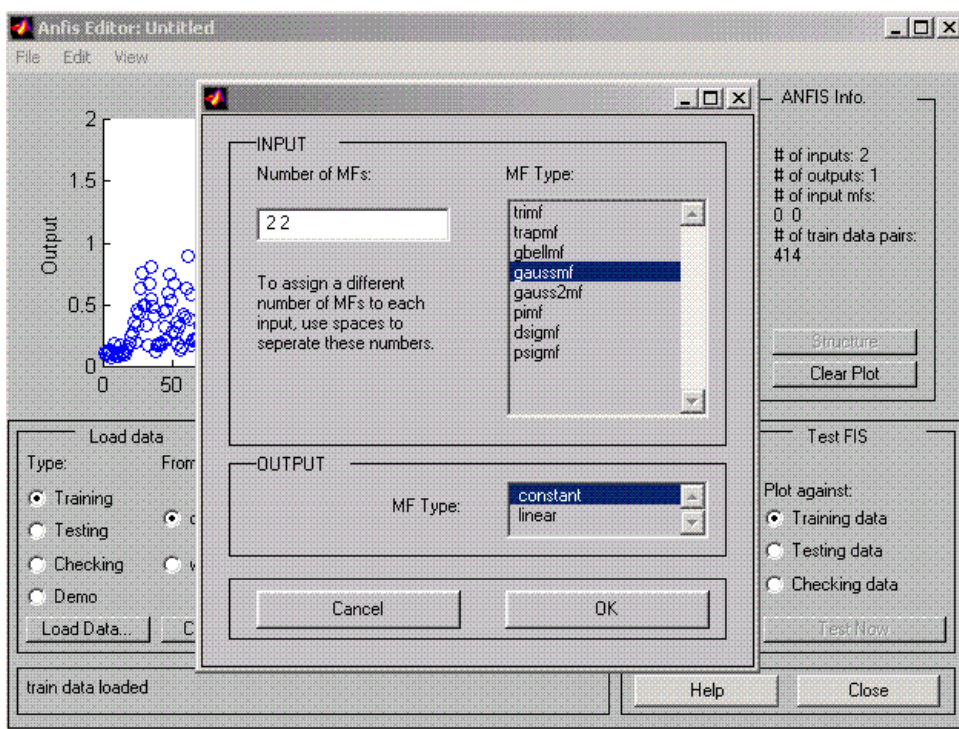
In the *Number of Input MFs* box type a number of membership functions for each input separated by a space

(e.g. start with 2 2). Here the inputs are ordered as in our

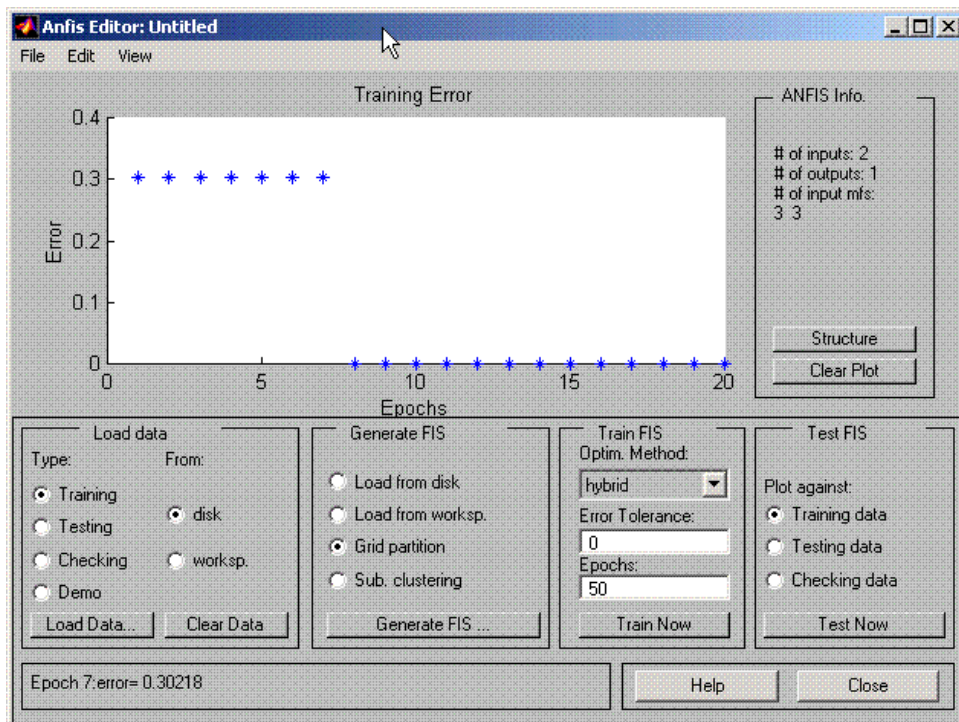
files. Click *OK*. A graphical diagram of the generated *zero-*

order Sugeno type model is available clicking the button

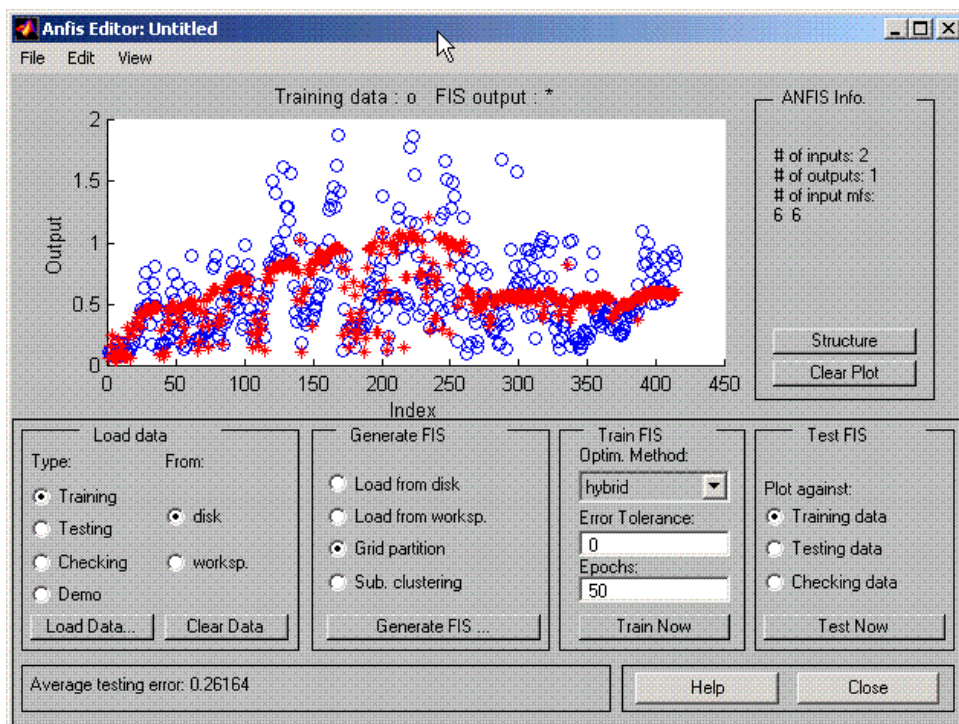
Structure.



- iv) Set the *Optim. Method* to *hybrid*, and the error tolerance to 0. Write a number of *epochs* for the training process (e.g. 50). Click *Training Now*. A picture of the evolution of the Mean Square Error between modeled and observed values is displayed. If evidence exist that more epochs will significantly decrease the Error, click the *Train Now* button again.



- v) When a reasonably stable Error value is achieved, click *Test Now* with the *Training data* option active. Record the Average Testing Mean Square Error displayed in the box at the bottom of the window, and also the total number of epochs used to achieve a stable Error value.



- vi) Repeat steps iii) to v) with different number of Input MFs, until a decision can be taken. This step includes some subjectivity, because in addition to the Mean Square Error, the total number of parameters should also be considered as a criterion. Each gaussian MF has two parameters, and the total output parameters are the product of the number of MF

in each input. Total number of parameters should not exceed $1/6$ the number of cases present in the Main data file.

This procedure works with gaussian input MFs and constant output MFs. We purposely omitted a discussion about the many options included in the ANFIS package, because this is beyond the scope of the Appendix. Although we encourage researchers to change these options if enhanced performance is expected, alternative ANFIS configurations usually gave poor results or an unacceptable amount of parameters. Researchers should consult the MATLAB Fuzzy Logic Toolbox documentation if they want to use alternative configurations.

5. PARAMETER ESTIMATION

When the structure identification problem is solved, the next step is to assign values to input and output MFs parameters. Once such values are assigned, we can use the generated FIS to calculate loads. This section explains how to obtain nutrient loads with the Monte-Carlo analysis explained in the paper.

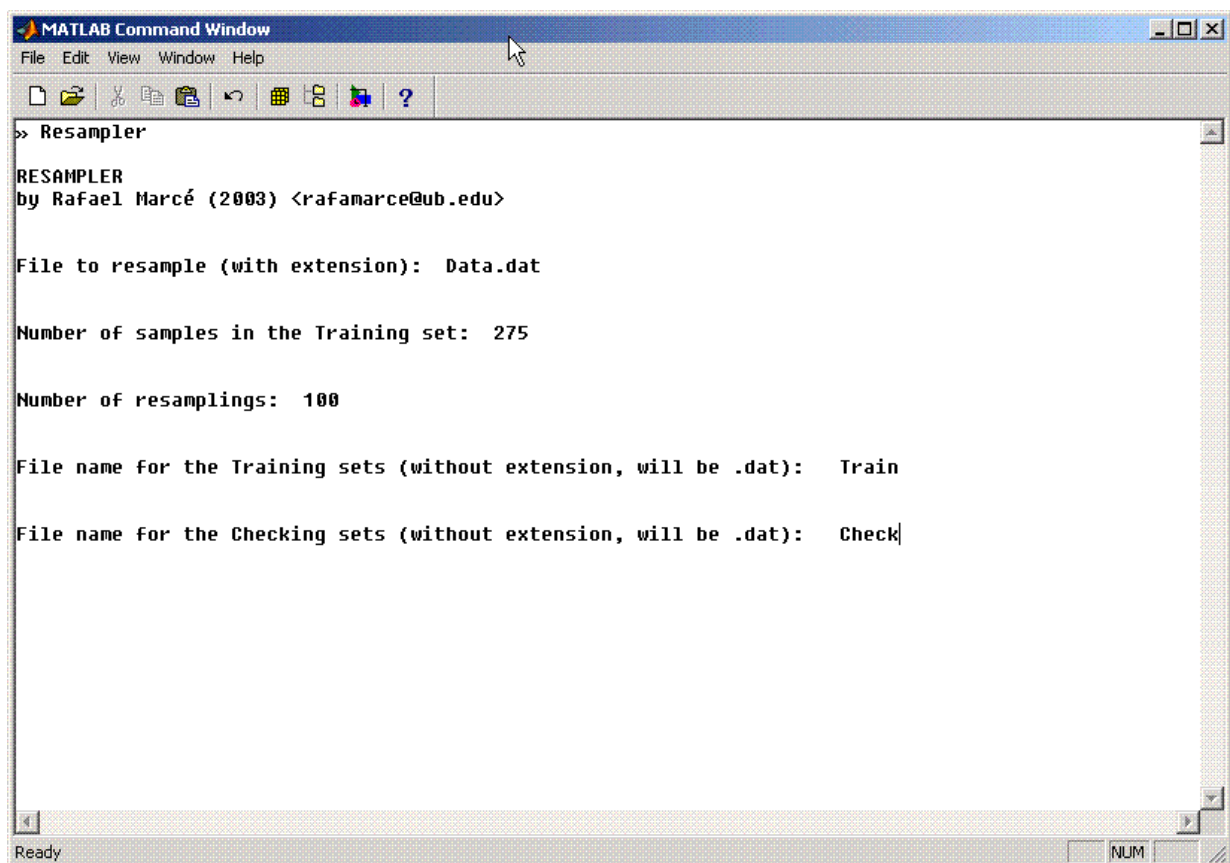
5.1 GENERATING TRAINING AND CHECKING FILES

For computational convenience, we first randomly generate the different training and checking data files, which will be used during the Monte-Carlo ANFIS estimation. The m-file *Resampler.m* is the code that performs this action. Launch MATLAB and type *Resampler*. The application request some information:

- i) First we have to type the name of the Main data file to be resampled (e.g. *Data.dat*).
- ii) Then, the amount of samples desired in the Training set should be specified. We assigned 2/3 of the cases in the Main data file. All cases not included in the Training set go to the Checking set.
- iii) The number of resamplings refers to the amount of Training and corresponding Checking data files we want to generate. We used 1000 in the paper (this

generates 2000 files). We recommend generating few more sets than we need (see Section 8).

- iv) Next, the desired name for the Training and Checking data files is requested. These are a prefix, because the complete file name is this prefix plus an automatically added number and extension. For example, if we answer *Train* and *Check*, the generated files will be *Train1.dat*, *Train2.dat*, ..., and *Check1.dat*, *Check2.dat*,...

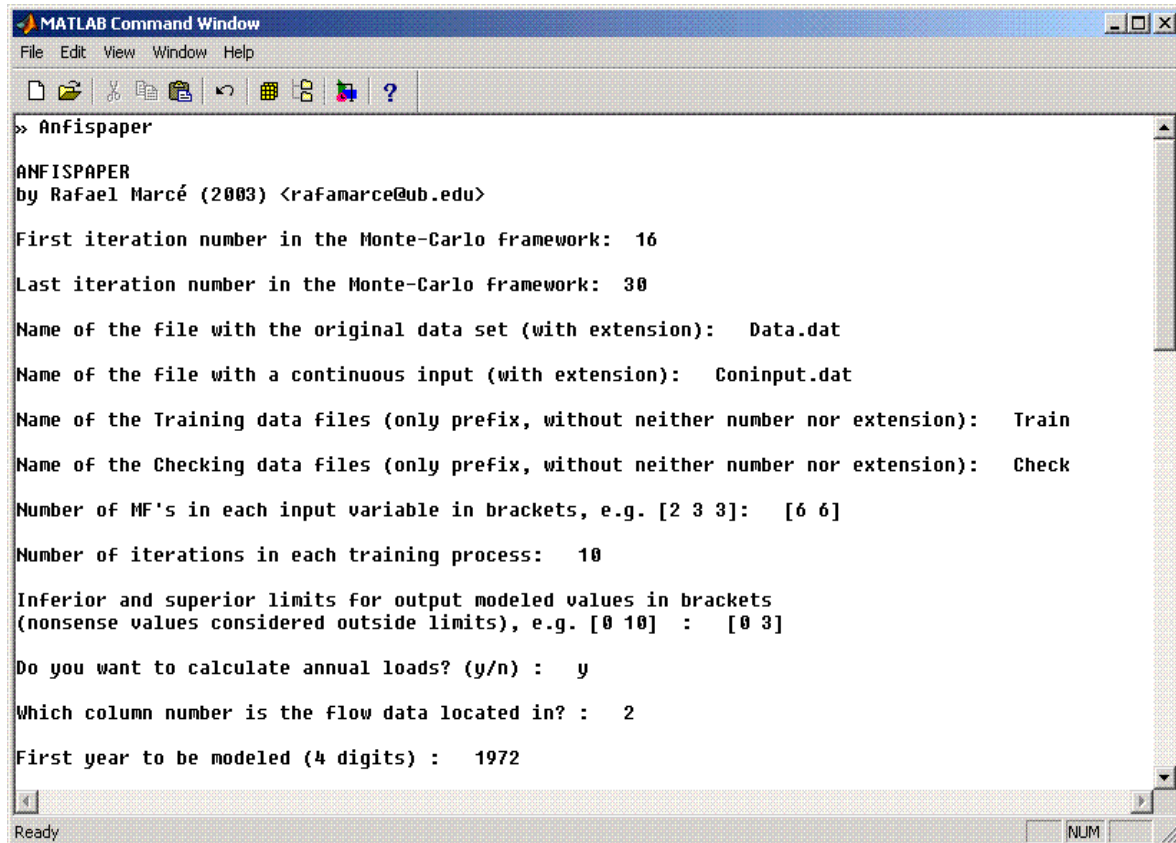


5.2 CALCULATING LOADS

When Training and Checking data files have been generated, the last step to obtain the desired annual loads is to run *Anfispaper.m*. Type *Anfispaper* in the MATLAB prompt. As above, a number of questions should be answered:

- i) First and last iteration number in the Monte-Carlo framework. This is to specify the amount of Monte-Carlo iterations. These numbers are related to the index in the training and checking files generated through *Resampler.m*, so we cannot go beyond the number of resamplings generated in section 5.1. We can start with a small number of iterations (e.g. answering 1-100), and repeat calculations with different training and checking files (e.g. answering 101-200, 201-300,...) until loads and variances are stable (the program does NOT calculate this, the researcher should collect results after each run). This is very useful, because there is no a priori method to know how many iterations are needed to attain stable results. Typing 1000 or 10000 iterations will result in a very time-consuming computation (several hours), whereas 100 iterations could take only few minutes.

- ii) The four next questions refer to the names of the different working data files (Main data file, Continuous data file, Training and Checking files).



```
» Anfispaper

ANFISPAPER
by Rafael Marcé (2003) <rafamarce@ub.edu>

First iteration number in the Monte-Carlo framework: 16
Last iteration number in the Monte-Carlo framework: 30
Name of the file with the original data set (with extension): Data.dat
Name of the file with a continuous input (with extension): Coninput.dat
Name of the Training data files (only prefix, without neither number nor extension): Train
Name of the Checking data files (only prefix, without neither number nor extension): Check
Number of MF's in each input variable in brackets, e.g. [2 3 3]: [6 6]
Number of iterations in each training process: 10
Inferior and superior limits for output modeled values in brackets
(nonsense values considered outside limits), e.g. [0 10] : [0 3]
Do you want to calculate annual loads? (y/n) : y
Which column number is the flow data located in? : 2
First year to be modeled (4 digits) : 1972
```

- iii) The information we collected during the Structure Identification *trial-and-error* procedure is incorporated through the seventh and eighth questions. First we have to type the number of MFs we have considered optimal for each input. Then, the number of training epochs needed during the ANFIS training. This number is actually an approximation, and it is better to place this limit beyond the minimum number of epochs needed during the *trial-and-error* step.

- iv) Inferior and superior limits for output modeled values. All values beyond these limits will be considered as nonsense values, as explained in the paper. Researcher expertise and/or information in the Main database could help answering this question.
- v) If annual loads are considered, two more questions should be answered. First we have to type the column number where the flow input is placed in the Main and Continuous data files. The last question is the first year sampled in the Main data set. This number will be used to deal with leap years, so it is NOT an option.

After answering these questions, the computer will calculate during several minutes, depending on the number of Monte-Carlo iterations and the velocity of the processor. When calculations end, the time elapsed (in seconds) is showed on the screen. Some warning messages could also be present on the prompt. These messages can be completely ignored. Results are stored in ASCII files located in the MATLAB working folder, which can be opened by standard spreadsheets and statistical packages. Bear in mind that *Anfispaper* will overwrite results files generated during previous runs. Place valuable results outside the working folder.

6. RESULTS FILES

6.1 LOADSANUAL.DAT

Annual load data in Kg constituent per year. There is one row for each Monte-Carlo iteration, and one column for each year modeled (the first column is the first year modeled). From this data we can calculate mean annual loads and variances, as well as test the normality of distributions. This file will be missing if annual loads are not considered during calculations.

6.2 NONSENSEVALUES.DAT

This file includes all the information about nonsense values detected during daily nutrient concentration prediction. This file contains a row for each value modeled during the evaluation of the Continuous input file, and a column for each Monte-Carlo iteration (it could be a very large file!!). Each cell in this file contains a number:

0. The value was not considered as a nonsense value. No action was taken.

1. Nonsense value considered. The modeled value was smaller than the user-defined inferior limit. The value was replaced by the preceding modeled output.

2. Nonsense value considered. The modeled value was higher than the user-defined superior limit. The

value was replaced by the preceding modeled output.

3. Nonsense value considered. One or more input values were beyond the limits of the collection of inputs present in the Main data file. The ANFIS function in the MATLAB Fuzzy Logic Toolbox does not model such values (i.e. it do not extrapolate, simply assigns an statistical value to this output). The value was replaced by the preceding modeled output.

6.3 FITANDRESIDUALS.DAT

This file includes information about several diagnostic analysis implemented in *Anfispaper*. There is a row for each Monte-Carlo iteration. The number of columns will vary depending on the number of inputs (34 columns in a two input problem):

Column 1. Training Mean Square Error between observed and modeled concentration values plus Checking Mean Square Error.

Column 2. Coefficient of determination of the regression between modeled and observed concentration values. Only Training data set considered.

Column 3. Coefficient of determination of the regression between modeled and observed concentration values. Only Checking data set considered.

Column 4. P-value of the Kolmogorov-Smirnov test for normality of the residuals. If p-value > 0.05 residuals are normal.

Column 5. Mean value of the residuals.

Column 6. Variance of the residuals.

Next columns (2+number of inputs). Residuals Lag-1 autocorrelation. The first column refers to the model residuals. Then, there is a column for residual series sorted by each input, and a final column for residuals sorted by modeled values.

Next columns (2+number of inputs). P-value of the Kendall rank coefficient test for presence of trends in the residuals. If p-value < 0.05 a trend is present. The first column refers to the model residuals. Then, there is a column for residual series sorted by each input, and a final column for residuals sorted by modeled values.

Next columns (2+number of inputs). P-value of the coefficient of determination (r^2) of the regression between residuals and an independent variable. If p-value < 0.05 the slope of the regression is not zero.

Thus, a linear trend is present in the series. The first column refers to the model residuals. Then, there is one column for a regression with each input, and a final column for a regression with modeled values.

Next columns (2+number of inputs). Runs test T statistic . If T statistic is < 1.96 no trend is present (no difference between observed and expected runs).

The first column refers to the model residuals. Then, there is a column for residual series sorted by each input, and a final column for residuals sorted by modeled values.

Next columns (2+number of inputs). Expected runs if residuals were random.

Next columns (2+number of inputs). Observed runs. The first column refers to the model residuals. Then, there is a column for residual series sorted by each input, and a final column for residuals sorted by modeled values.

Next columns (2+number of inputs). Runs interval. If $\text{observed runs} > (\text{expected runs} + \text{interval})$ there is a cyclic trend. If $\text{observed runs} < (\text{expected runs} + \text{interval})$ there is a monotonic trend.

6.4 PARAMETERSANFIS.DAT

This file includes information about the parameters fitted during ANFIS estimation. There is a row for each Monte-Carlo iteration. The columns will vary depending on the number of inputs and MFs:

Column 1. Standard deviation of the **first** gaussian curve (MF) of the input 1 (the input order is the order in the data files).

Column 2. Mean of the **first** gaussian curve (MF) of the input 1.

Column 3. Standard deviation of the **second** gaussian curve (MF) of the input 1.

Column 4. Mean of the **second** gaussian curve (MF) of the input 1.

(...)

MF parameters of Input 2 are placed next, and so on.

Output MFs parameters follow the input parameters. Remember that output MFs are constants. Thus, there is only one parameter for each output MF. To understand how output constants are sorted in this file we have to bear in mind that each output parameter is the consequent of an *if-and-then* rule. Indeed, the output constants are sorted following an *if-and-then* rule order. The order of the output parameters are:

Column 1 after input parameters:

[If *Input 1* is MF 1 and *Input 2* is MF 1 then *Output* is] **Constant 1**

Column 2 after input parameters:

[If *Input 1* is MF 1 and *Input 2* is MF 2 then *Output* is] **Constant 2**

Etc...

Remember that we can draw the gaussian MFs from the parameters in this file. A gaussian curve is defined by the equation:

$$f(x) = e^{\frac{-(x-c)^2}{2\sigma^2}}$$

where c is the mean parameter and σ the standard deviation. x is the value of the input variable (flow or time).

6.5 OUTPUTSERIES.DAT

This file includes all modeled values during simulation (i.e. daily nutrient concentration). Here, the nonsense values are already processed. This file contains a row for each value modeled from the evaluation of the Continuous input file, and a column for each Monte-Carlo iteration. Take in mind that this file can be very large.

7. PERFORMANCE ANALYSIS

The m-file *Performance.m* implements the performance analysis explained in the paper. If annual loads are not considered, the comparisons between modeled and observed values are between sums of outputs instead of between sums of loads. Other possibilities can be easily programmed. Contact the [authors](#) for assistance.

- i) Launch MATLAB and type *Performance*. A number of questions should be answered.
- ii) ‘First subsampling frequency’: the performance analysis is repeated for training and checking sets of different sizes (always maintaining the 2/3-1/3 ratio). The frequency refers to the number of samples in the first training+checking sets. Then, calculations are repeated with $2*frequency$ cases in these sets, then with $3*frequency$ cases, and so on (until the total number of cases in the Main data file are reached).
- iii) ‘Number of calculations in each sampled frequency’: Inside each frequency, calculation will be repeated several times (i.e. a Monte-Carlo framework for each frequency). This number is also the index to call training and checking files (e.g. if we answer 100, training and checking files numbered from 1 to 100 will be used to built new training and checking sets according to the defined frequency, as well as the evaluation sets).

- iv) Following questions are equivalent to that present in *Anfispaper*. See above.

After answering these questions, the computer will calculate during several minutes, depending on the number of Monte-Carlo iterations, number of different frequencies, and the velocity of the processor. The output consists in two files for EACH frequency:

- i) **PerfLoadfreq.dat**. This file contains the loads (or other result if annual loads are not considered) calculated through evaluation of inputs present in the different Evaluation sets, and the actual load calculated from the outputs present in these sets (nonsense values excluded, see explanations in the paper). There is a row for each Monte-Carlo iteration. The first column is the observed load, the second is the modeled load. From these files we can calculate mean bias and variance, and we can also test the significance of this bias.
- ii) **PerfNonSensefreq.dat**. This file is equivalent to the *Nonsensevalues.dat* file above. We calculated the mean number and variance of nonsense values in each frequency from these files.

8. SPECIAL ERROR MESSAGE

In very rare occasions, an *Anfis* or a *Performance* run crash, displaying an error message of the type:

‘ANFIS FUNCTION CANNOT WORK WITH THE CHECKING FILE
NUMBER 134. THIS FILE SHOULD BE MODIFIED.’ (*Anfis* run)

‘ANFIS FUNCTION CANNOT WORK WITH THE CHECKING FILE
NUMBER 456 IN FREQUENCY 300. THIS FILE SHOULD BE
MODIFIED’ (*Performance* run)

This is due to a ‘bug’ in the *anfis* MATLAB Fuzzy Logic Toolbox function. Usually, if the *anfis* function finds an input space in the Checking data beyond the limits of the input space in the corresponding Training set, this Checking case is not modeled, and a statistical output is assigned. But if a Checking input value is really far beyond the limits defined in the Training set, *anfis* could fail evaluating rules including this input, and the run crashes.

In practice, we never found this error during *Anfis* runs. Only in one occasion a *Performance* run crashed. However, we included this error message to help researches in other situations. When an error of this type appears, the checking file named in the error message and its corresponding training set should be eliminated, and replaced by other combination of training and checking sets. This is way we recommended

to generate more training and checking files than needed in section 5.1. For example, if we want to work with 1000 Training and Checking sets, we generate more (e.g. 1010). Then, if any error of this type appears, we can discard the problematic files and replace it with one of the pairs not in use (i.e. we have to rename the files). Remember that the collection of Training and Checking files should be numbered without gaps. Otherwise, the applications will crash.