A command-line tool for data-driven fuzzy habitat modelling

A quick user's guide, developed by Christos Theodoropoulos

Currently updating … draft 15 Feb 2019

Contact Information

Hellenic Centre for Marine Research
Institute of Marine Biological Resources and Inland Waters
46.7 km Athens-Sounio ave.
19013
Anavyssos
Greece
Tel. +30 22910 76335
Fax. +30 22910 76419
Email. ctheodor@hcmr.gr
URL. http://www.hcmr.gr/en/

National Technical University of Athens
Department of Water Resources and Environmental Engineering
Iroon Polytechniou 5
15780
Athens
Greece
Tel. +30 210 7722809
Fax. +30 210 7722814
Email. stamou@central.ntua.gr
URL. https://www.hydro.ntua.gr/?set_language=en

To report errors, bugs, possible amendments or anything else you would like to indicate, please contact Mr. Christos Theodoropoulos at ctheodor@central.ntua.gr

TABLE OF CONTENTS

1. Overview

Let's suppose that you have a set of field observations, that is, microhabitat combinations of flow velocity (V), water depth (D) and substrate type (S). Let's also suppose that you have successfully associated each of these combinations with the microhabitat suitability (K), either using freshwater fish, benthic macroinvertebrates or any other biotic element of the aquatic ecosystem. You may now wonder which is the right habitat modelling method to enable you predict the habitat suitability of unknown microhabitat combinations based on your 'reference' dataset. Since it is very likely to conclude that fuzzy rule-based algorithms would be accurate enough for the job, this guide provides you with the technical help to implement one of these fuzzy models.

HABFUZZ is fully automated, you only need to prepare a single input file with your dataset to be used in the software, that's all. A small example dataset is provided to ease the understanding of the processes followed in HABFUZZ (Table 1).

> Note that to run HABFUZZ and get the output, there is no need to know its algorithms. Once you prepare the input file, all processes including fuzzy-rule development are internally applied by HABFUZZ

Table 1. Example dataset for this tutorial

Microhabitat	Flow velocity (m/s)	Water depth (m)	Substrate type
1	0.28	0.29	Boulders
2	0.05	0.08	Large stones
3	0.46	0.80	Small stones

1.1. The fuzzy logic algorithm

(You can just proceed to section 4 to run HABFUZZ)

As initially proposed by Zadeh (1965) and described in detail by Ross (2010), the process of determining the fuzzy rule-based habitat suitability, given the flow velocity, the water depth and the type of substrate, can be summarized in four steps (Fig. 1):

Step 1. *Fuzzification of the input variables*

In this step, the user defines categories (called membership functions or fuzzy sets) for each input variable and the input values of V, D and S are assigned to one or more membership functions. By this procedure, crisp numerical values of each input variable are converted to a fuzzy 'membership degree', ranging from 0 to 1 for each membership function. For example, a depth value of 14 cm may yield a membership degree of 0.7 for the 'shallow' membership function and 0.28 for the 'very shallow' membership function (Fig. 1). Then the process continues based on the fuzzy sets instead of the crisp numerical inputs.

Step 2. *Application of a fuzzy operator (AND or OR) in the antecedent (IF-THEN rules)*

According to the reference data for the target aquatic community, the AND (min) or OR (max) operator is applied to each combination of variables (membership functions since step 1) and the derived value is assigned to the membership function of the output variable (defined in step 1), in this case the habitat suitability. For example, if the user defines five membership functions for habitat suitability (bad, poor, moderate, good, high), then the application of the fuzzy operator would result in

$$f_4(K) = \min\left(f_2(D), f_3(V)\right)$$
$$f_3(K) = \min\left(f_3(D), f_2(V)\right)$$
$$f_4(K) = \min\left(f_2(D), f_4(V)\right)$$

where,
f_i denotes for the membership function of each input and output variable
V is the flow velocity
D is the water depth
K is the habitat suitability

etc., until all possible combinations of fuzzy inputs are assigned to an output membership function, based on the rationale that, for example, *IF flow velocity is f_3 (moderate) AND water depth is f_2 (shallow) THEN habitat suitability is f_4 (good).*

Step 3. *Aggregation of outputs*
In this step, the derived habitat suitability membership functions from each rule are combined into one fuzzy set. Usually, the OR (max) operator is applied to aggregate the same output fuzzy sets of the previous step. For example, the $f_4(K)$ is derived in the previous example two times by the IF-THEN rules. The final fuzzy set representing each habitat suitability class F_j would be

$$F_j = \max(f_i^1(K), f_i^2(K), \dots, f_i^v(K))$$

Step 4. *Defuzzification* - This final step is applied to derive one single habitat suitability value, by combining the membership degrees of all fuzzy habitat suitability classes. Among the various defuzzification methods, the 'centroid', 'maximum membership', 'weighted average' and 'mean-max membership methods' are implemented in HABFUZZ and described below.

a. Centroid defuzzification
Usually called the 'center of gravity' or 'center of area'. It can be defined by the algebraic expression

$$K = \frac{\int x f(x) dx}{\int f(x) dx}$$

which is numerically approximated in HABFUZZ by

$$K = \frac{\sum_{i=1}^{n} x_i (f(x_i))}{\sum_{i=1}^{n} (f(x_i))}$$

where,
$f(x_i)$ is the membership degree at value x_i

b. Maximum membership defuzzification

This is the maximum membership degree observed by the aggregation step:

$$K = \max(f(x))$$

c. Weighted average

This method can be used only for symmetrical output membership functions and is calculated by weighting each output membership function by its largest membership degree:

$$K = \frac{\sum_{i=1}^{n} \overline{x_i}(f(\overline{x_i}))}{\sum_{i=1}^{n}(f(\overline{x_i}))}$$

where,
$f(\overline{x_i})$ is the membership degree at the average value $\overline{x_i}$ of each membership function

d. Mean of maximum

This method resembles the 'maximum membership' method. However, the maximum membership degree may not be unique but a range of degrees, from which the mean value is derived:

$$K = \frac{x_a + x_b}{2}$$

where,
x_a is the first value reaching the highest membership degree of the class with the highest membership and
x_b is the last value with the highest membership degree of the class with the highest membership

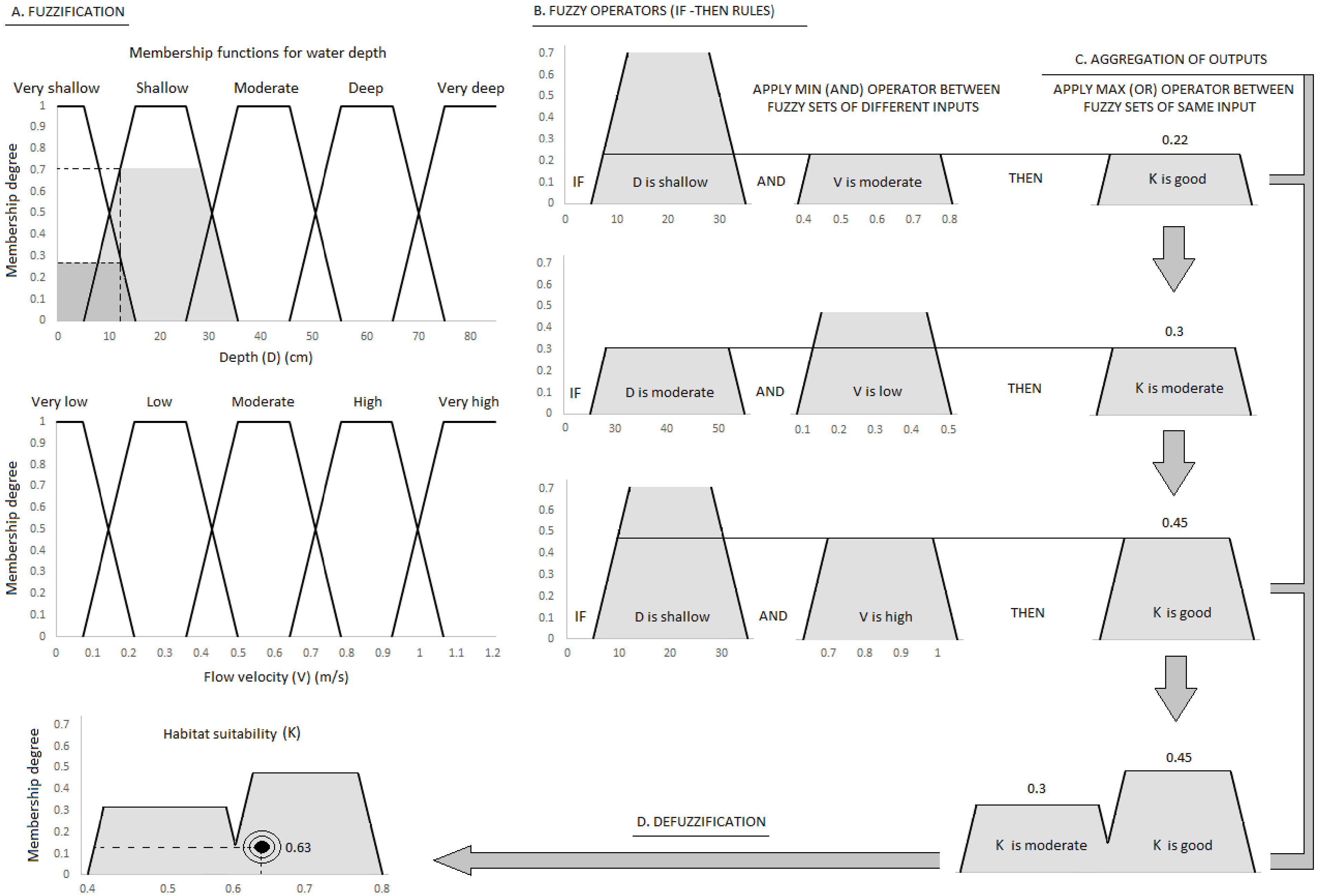


Fig. 1. The fuzzy logic algorithm

1.2. The fuzzy rule-based Bayesian algorithm

The fuzzy rule-based Bayesian algorithm utilizes the Bayesian joint probability and a classification system based on the 'expected utility' (described in Brookes et al., 2010) and can be summarized in three steps:

Step 1. *Fuzzification of the input variables*

This step is the same as in the fuzzy algorithm and results in the conversion of crisp numerical values to fuzzy 'degrees of membership', ranging from 0 to 1 for each membership function (fuzzy set).

Step 2. *Calculation of the Bayesian joint probability*

The joint probability for interdependent events is calculated as

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$$

where,

$P(A \cap B)$ is the probability of event A and event B occurring together

$P(A|B)$ is the conditional probability of event A occurring given the event B occurred

$P(B|A)$ is the conditional probability of event B occurring given the event A occurred

In this case, V, D and S are considered independent of each other and the joint probability is calculated by replacing P(A|B) with P(A). For example, the joint probability of the flow velocity being 0.5 m/s and the water depth being 0.2 m, given their probabilities P(V:0.5 = 0.8) and P(D:0.2 = 0.3) is 0.8 x 0.3 = 0.24. HABFUZZ uses the fuzzy sets (fuzzified input values) from step 1 as probabilities of occurrence of each input. For example, a crisp V value of 0.28 m/s corresponds to the moderate fuzzy set by 0.2 membership degrees and to the low fuzzy set by 0.8 membership degrees. These values are used as the probabilities of occurrence of the moderate and low V sets.

Step 3. *Classification of the outcome in habitat suitability classes*

Classification is applied by using the 'expected utility' equation, where in HABFUZZ, habitat suitability is expressed as a score (bad - 0.1, poor - 0.3, moderate - 0.5, good - 0.7, high - 0.9) and multiplied by the joint probability of occurrence of each habitat suitability class (sum of probability x score) as following:

$$EU\ (A) = \sum_{i=1}^{n} p(xi|A)U(xi)$$

where,

$EU(A)$ is the expected utility of action or event A

$P(x_i|A)$ is the probability of action or event A

$U(x_i)$ is a utility weight to convert a state to numerical values

2. Dependencies

It is advised to install the GNU FORTRAN Compiler (download at https://gcc.gnu.org/wiki/GFortranBinaries) to quickly compile HABFUZZ through the relevant Windows and OS X files (however, experienced users may also use their preferred compilers). For Mac users, Xcode (download at https://developer.apple.com/xcode/) with its relevant Command Line Tools should be installed to enable compiling through the GNU FORTRAN Compiler.

3. Installing

HABFUZZ has been tested on Windows 10 - 32 bit and 64 bit operating systems, Ubuntu 16.04 and OS X 10.11 El Capitan (with Xcode 7.3.1 and Xcode 7.3.1. Command Line Tools), using the GNU FORTRAN Compiler. Depending on your operating system, follow the relevant instructions to run HABFUZZ.

3.1. Windows users

If the user needs to modify the source code of HABFUZZ, re-compilation is necessary. Using the GNU FORTRAN Compiler, you can either run the *wcompile.bat* file, or open a command window, navigate to the 'HABFUZZ' subfolder and type the relevant commands:

gfortran -c fdeclarations.f95 ↵
gfortran -o habfuzz habfuzz.f95 fdeclarations.f95 classifier.f95 combinations.f95 ruler.f95 fuzzifier.f95 permutator.f95 rules2.f95 fuzzy.f95 fruler.f95 rules1.f95 centroid.f95 meanmax.f95 maxmem.f95 waver.f95 randomizer.f95 iterator.f95 iterator10.f95 tester.f95 ftester.f95 performance.f95 tencrossval.f95 ↵
*del *.o ↵*
*del *.mod ↵*

habfuzz.exe will then be replaced by the newly compiled one, being ready to run.

3.2. Linux users

Open the terminal and navigate to the HABFUZZ subfolder. If you don't have the GNU FORTRAN Compiler, you need to be a root user (administrator) and type

sudo apt-get install gfortran ↵

to install the compiler. Having gfortran installed, the commands necessary to compile are the following:

gfortran -c fdeclarations.f95 ↵
gfortran habfuzz.f95 fdeclarations.f95 classifier.f95 combinations.f95 ruler.f95 fuzzifier.f95 permutator.f95 rules2.f95 fuzzy.f95 fruler.f95 rules1.f95 centroid.f95 meanmax.f95 maxmem.f95 waver.f95 randomizer.f95 iterator.f95 iterator10.f95 tester.f95 ftester.f95 performance.f95 tencrossval.f95–o habfuzz ↵

Be careful to write exactly the abovementioned commands, arranging the source files in the order given above. Then you can run HABFUZZ by typing:

./habfuzz ↵

3.3. Mac OS X users

You need to have Xcode installed together with the GNU FORTRAN Compiler and be a root user to enable compilation. Open the terminal and navigate to the HABFUZZ subfolder. To compile, you can either run the *mcompile.sh* file (which automatically applies the compilation commands) by typing:

./mcompile.sh ↵

or manually type the commands:

gfortran -c fdeclarations.f95 ↵
gfortran -o habfuzz fdeclarations.f95 habfuzz.f95 classifier.f95 combinations.f95 ruler.f95 fuzzifier.f95 permutator.f95 rules2.f95 fuzzy.f95 fruler.f95 rules1.f95 centroid.f95 meanmax.f95 maxmem.f95 waver.f95 randomizer.f95 iterator.f95 tester.f95 ftester.f95 performance.f95 ↵

HABFUZZ can then be executed from the command line by typing
./habfuzz ↵

4. Usage

4.1. Input and output data

To run HABFUZZ, you only need to prepare (i) a train-data file and (ii) a test-data file. These files should be named *traindata.txt* and *testdata.txt* respectively and should be located at the same folder with *habfuzz.exe.*

(i) traindata.txt

This file contains the data from which HABFUZZ will be trained to predict.
The file should have five columns where V, D, S, T (water temperature) and K are stored, respectively, for each microhabitat. Note that the first row should only contain one number describing the number of rows in the file (Fig. 2). If you don't need to include T in the calculations, you can just create a T column with the same T (randomly selected) for all microhabitats.

traindata.txt

```
10
0	0.19	0.05	11.9	0.274
0	0.18	0.04	21.1	0.286
1.576	0.27	0.05	11.8	0.292
0.485	0.87	0.03	12.2	0.295
1.08	0.26	0.05	11.1	0.297
0.3	0.34	0.05	12.1	0.348
0.14	0.16	0.03	11.9	0.35
0	0.06	0.07	16.8	0.361
0.19	0.64	0.05	18.8	0.369
0.75	0.19	0.04	22.1	0.369
```

Fig. 2. *traindata.txt* file example

(ii) testdata.txt

This file contains the data with unknown habitat suitability values, which are to be predicted by HABFUZZ based on the train data. This file should have the same format as the *traindata.txt* but without a K column (Fig. 3). Again, if you do not wish to use T, the T column should be there but with the same T value with the *traindata.txt* for all cells.

testdata.txt

```
20
0	0.16	0.04	11.6
0.1	0.35	0.07	14.6
0	0.19	0.03	14.8
0.11	0.42	0.07	17.9
0.08	0.24	0.07	22.3
0	0.07	0.022	12.2
0.05	0.14	0.024	16.8
0	0.76	0.04	21.6
0.802	0.33	0.04	11.8
0.09	0.48	0.026	12.2
0.88	0.34	0.03	23
0.1	0.52	0.04	14.9
0.006	0.24	0.024	12.5
0	0.21	0.07	11.6
0	0.03	0.022	14.8
1.444	0.53	0.05	12.5
1.42	0.42	0.05	12.3
0.15	0.39	0.03	11.9
0.95	0.51	0.07	13
0	0.19	0.05	11.9
```

Fig. 3. *testdata.txt* file example

The output of HABFUZZ is a file named *suitability.txt* containing a single column with all the predicted habitat suitabilities (ranging from 0 - unsuitable to 1 - suitable) calculated for each input element (node) in the same order as with the input file and a *log.txt* file with the internal parameters of the prediction process. Both files are placed by the program in the HABFUZZ subfolder.

During the fuzzy inference process, habitat suitability is initially a combination of fuzzy membership functions (five classes of habitat suitability - bad, poor, moderate, good, high) and through the defuzzification process it is converted into a crisp output ranging from 0 to 1. The inputs and the output suitability of HABFUZZ are depicted in Fig. 1. In the fuzzy Bayesian algorithm, habitat suitability is expressed using the same five classes. Each class is assigned with a utility score (bad - 0.1, poor - 0.3, moderate - 0.5, good - 0.7, high - 0.9) and multiplied by the joint probability of each combination observed.

Table 2. Manning's n for various substrate types used in HABFUZZ

Bed material	Size (diameter)	Manning's n
Boulders	>25 cm	0.070
Large stones	12-25 cm	0.050
Small stones	6-12 cm	0.040
Large gravel	2-6 cm	0.030
Medium gravel	0.6-2 cm	0.026
Fine gravel	0.2-0.6 cm	0.024
Sand	<0.2 cm	0.022
Silt	-	0.020

4.2. Running HABFUZZ

(using the example dataset of table 1)

After having the input files ready run the program. The command prompt opens and after a short welcome message the software asks for the inference process to be implemented (Fig. 4 and 5).

```
@@   @@   @@@@@   @@@@@   @@@@@@  @@    @@  @@@@@@  @@@@@@
@@   @@  @@   @@  @@  @@  @@      @@    @@     @@      @@
@@@@@@@  @@@@@@@  @@@@@   @@@@@   @@    @@    @@      @@
@@@@@@@  @@@@@@@  @@@@@   @@@@@   @@    @@   @@      @@
@@   @@  @@   @@  @@  @@  @@      @@    @@  @@      @@
@@   @@  @@   @@  @@@@@   @@       @@@@@@   @@@@@@  @@@@@@

Habfuzz 2.3

The open software for data-driven fuzzy
rule-based aquatic habitat suitability modelling

Fully automated with Monte Carlo and 10-fold cross-validation capability
Just provide your input data matrix and get the resulting suitability
If you need assistance contact us at ctheodor@hcmr.gr

Press ENTER to start
```

Fig. 4. The welcome-screen of HABFUZZ

```
@@   @@   @@@@@   @@@@@   @@@@@@  @@     @@  @@@@@@  @@@@@@
@@   @@  @@   @@  @@  @@  @@      @@     @@     @@      @@
@@@@@@@  @@@@@@@  @@@@@   @@@@@   @@     @@    @@      @@
@@@@@@@  @@@@@@@  @@@@@   @@@@@   @@     @@   @@      @@
@@   @@  @@   @@  @@  @@  @@      @@     @@  @@      @@
@@   @@  @@   @@  @@@@@   @@       @@@@@@    @@@@@@  @@@@@@

Habfuzz 2.3

The open software for data-driven fuzzy
rule-based aquatic habitat suitability modelling

Fully automated with Monte Carlo and 10-fold cross-validation capability
Just provide your input data matrix and get the resulting suitability
If you need assistance contact us at ctheodor@hcmr.gr

Press ENTER to start

Select modelling method
1: Fuzzy logic
2: Fuzzy Bayesian inference
```

Fig. 5. Selection of modelling algorithm in HABFUZZ

After selecting the modelling algorithm, the user is asked to select the cross validation scheme (either Monte Carlo or ten-fold) (Fig. 6). In ten-fold cross validation, the initial dataset is randomly partitioned in ten equal-sized subsamples. Nine subsamples are used as the training dataset and the remaining subsample is used for model validation. This process is repeated ten times (folds), using a different subsample for validation at each iteration. The Monte Carlo scheme also includes ten iterations but at each iteration the initial dataset is randomly partitioned in two subsamples. The first subsample contains 90% of the initial data and is used for calibration, and the second subsample contains the remaining 10% and is used for validation. At each iteration, the initial data is again randomly partitioned and thus the same data may be randomly included in each subsample more than once, in contrast to the ten-fold cross validation scheme. The performance of each model is evaluated as the average percentage of the correctly classified instances (CCI) between each iteration of the ten-fold cross-validation process

```
@@@@@@@  @@@@@@@  @@@@@   @@@@@   @@     @@    @@      @@
@@@@@@@  @@@@@@@  @@@@@   @@@@@   @@     @@    @@      @@
@@   @@  @@   @@  @@  @@  @@      @@     @@   @@      @@
@@   @@  @@   @@  @@@@@   @@       @@@@@@   @@@@@@  @@@@@@

Habfuzz 2.3

The open software for data-driven fuzzy
rule-based aquatic habitat suitability modelling

Fully automated with Monte Carlo and 10-fold cross-validation capability
Just provide your input data matrix and get the resulting suitability
If you need assistance contact us at ctheodor@hcmr.gr

Press ENTER to start

Select modelling method
1: Fuzzy logic
2: Fuzzy Bayesian inference
1

Select cross-validation scheme
1: Monte Carlo
2: Ten fold
```

Fig. 6. Selection of the cross-validation method

4.2.1. The fuzzy algorithm

If the fuzzy algorithm is selected, the program, after selecting the cross-validation scheme, prompts the user to select the desired management scenario to implement (Fig. 7). There are three available scenarios based on the method used for deriving the outcome of each IF-THEN rule from the reference conditions of the program, (i) the average scenario, where the different suitability values for the same combinations of flow velocity, water depth and substrate type are averaged to derive the final habitat suitability, (ii) the worst scenario, where the final suitability is derived from the minimum observed suitability and (iii) the optimum scenario where the final suitability is derived by the maximum observed suitability. A default scenario is also present (the moderate scenario). Note that if a specific combination in the observed data does not match a combination in the reference data, the program returns a value of '-1' for the habitat suitability.

After selecting the desired scenario, the user is asked to select the defuzzification method (see section 1) (Fig. 8). A default method (centroid) is available. After selecting the defuzzification method, HABFUZZ calls the relevant subroutines to perform the tasks selected. The program informs the user when the process is completed and indicates the *suitability.txt* file created where the suitability values are stored and the *log.txt* file with the fuzzy membership degrees for each class. Both files are located in the HABFUZZ subfolder.

```
The open software for data-driven fuzzy
rule-based aquatic habitat suitability modelling

Fully automated with Monte Carlo and 10-fold cross-validation capability
Just provide your input data matrix and get the resulting suitability
If you need assistance contact us at ctheodor@hcmr.gr

Press ENTER to start

Select modelling method
1: Fuzzy logic
2: Fuzzy Bayesian inference
1

Select cross-validation scheme
1: Monte Carlo
2: Ten fold
1

Select the preferred scenario to implement
1: Average
2: Worst
3: Optimum
4: Default
```

Figure 7. Selection of the desired scenario

```
Select modelling method
1: Fuzzy logic
2: Fuzzy Bayesian inference
1

Select cross-validation scheme
1: Monte Carlo
2: Ten fold
1

Select the preferred scenario to implement
1: Average
2: Worst
3: Optimum
4: Default
1

Select the preferred defuzzification method
1: Centroid
2: Max membership
3: Weighted average
4: Mean-max membership
5: Default
```

Figure 8. Selection of the defuzzification method

Using the example data of table 1, the crisp input values for V and D, are fuzzified as depicted in table 3.

Table 3. Fuzzification results for the example dataset. Different microhabitat are shaded differently.

	Fuzzy membership classes						
Crisp inputs	V Very Low	V Low	V Moderate	D Very Shallow	D Shallow	D Moderate	D Very Deep
V = 0.28	0	0.8	0.2				
V = 0.05	1	0	0				
V = 0.46	0	0	1				
D = 0.29				0	0.333	0.667	0
D = 0.08				0.333	0.667	0	0
D = 0.80				0	0	0	1

V: Flow velocity, D: Water depth

HABFUZZ then checks each combination of inputs (their corresponding fuzzy sets) and assigns a membership degree using the AND (min) operator to the relevant habitat suitability class they belong according to the IF-THEN rules calculated by the program itself, based on the training dataset, and depending on the selected management scenario. Let's assume that the user has chosen the moderate scenario. The membership degree of each combination to the suitability classes (including the substrate type) is depicted in table 4.

Table 4. Checking the relevant IF-THEN rules and assigning membership degrees to the suitability class by applying the AND (min) operator.

V	D	S	K
Moderate (0.2)	Moderate (0.667)	Boulders (1)	-
Moderate (0.2)	Shallow (0.333)	Boulders (1)	High (0.2)
Low (0.8)	Moderate (0.667)	Boulders (1)	Moderate (0.667)
Low (0.8)	Shallow (0.333)	Boulders (1)	Good (0.333)
Very Low (1)	Very Shallow (0.667)	Large stones (1)	Good (0.667)
Very Low (1)	Shallow (0.333)	Large stones (1)	Good (0.333)
Moderate (1)	Very Deep (1)	Small stones (1)	-

V: Flow velocity, D: Water depth, S: Substrate type, K: Habitat suitability

HABFUZZ then combines the same habitat suitability classes observed (aggregation step) using the OR (max) operator and the different membership degrees of all classes observed are defuzzified using one of the methods described in section 1. The results of the aggregation and defuzzification processes (in this case we have chosen the centroid defuzzification method) are depicted in table 5.

Table 5. Aggregation of outputs using the OR (max) operator. It can be seen that microhabitat 3 is not referred in the IF-THEN rules and a value of -1 is returned by HABFUZZ.

Microhabitat	V	D	S	K
1	0.28	0.29	Boulders	0.622
2	0.05	0.08	Large stones	0.700
3	0.46	0.80	Small stones	-1

V: Flow velocity, D: Water depth, S: Substrate type, K: Habitat suitability

4.2.2. The fuzzy Bayesian inference process

If the fuzzy Bayesian process is selected, the program immediately calculates the habitat suitability according to the steps described previously and outputs two *.txt* files, the *suitability.txt* and the *log.txt* with the same contents as in the fuzzy inference process.

Again, using the example data of table 1, the crisp input values for V and D, are fuzzified as depicted in table 3. The process then treats the fuzzified membership degrees as the probability of each observation occurring, suggesting for example that *'the probability of habitat suitability being high is the joint probability that V is moderate, D is shallow and S is boulders'*. In the example dataset, this concept is depicted for each microhabitat in tables 6, 7 and 8.

Table 6. (A) The joint probability table for the fuzzified inputs of microhabitat 1 (S=Boulders, not shown but included). (B) Joint probability after including the probability of the habitat suitability (not shown) class for each combination.

(A) Microhabitat 1	D (P)		
V (P)	Shallow (0.333)	Moderate (0.667)	JP = P(V) x P(D) x P(S) Substrate's P is always 1 since S is not fuzzified.
Low (0.8)	0.2664	0.5336	
Moderate (0.2)	0.0666	0.1334	

(B) Microhabitat 1	D (P)						
V (P)	Shallow (0.333)				Moderate (0.667)		JP = P(V) x P(D) x P(S) x P(K)
Low (0.8)	0.1455	0.0729	0.0239	0.0239	0.2668	0.2668	
Moderate (0.2)	0.0667				-		

V: Flow velocity, D: Water depth, S: Substrate type, JP: Joint probability, K: Habitat suitability; Blue colour: High K, Green colour: Good K; Yellow colour: Moderate K, Red: Bad K

Table 7. (A) The joint probability table for the fuzzified inputs of microhabitat 2 (S=Large stones, not shown but included). (B) Joint probability after including the probability of the habitat suitability class for each combination. JP: Joint probability.

(A) Microhabitat 2	D (P)		
V (P)	Very Shallow (0.667)	Shallow (0.333)	JP = P(V) x P(D) x P(S)
Low (1)	0.667	0.333	

(B) Microhabitat 2	D (P)							
V (P)	Very Shallow (0.667)			Shallow (0.333)				Joint Probability = P(V) x P(D) x P(S) x P(K)
Very Low (1)	0.1667	0.3335	0.1667	0.0639	0.1412	0.1022	0.0256	

V: Flow velocity, D: Water depth, S: Substrate type, JP: Joint probability, K: Habitat suitability, Blue colour: High suitability, Green colour: Good suitability, Yellow: Moderate suitability, Orange: Poor suitability

Table 8. The joint probability table for the fuzzified inputs of microhabitat 2 (S=Small stones, not shown but included). Since the specific combination is nor present in the *rules.f95* file, no further calculations are applied.

Microhabitat 3	D (P)	
V (P)	Very Deep (1)	Joint Probability = P(V) x P(D) x P(S)
Moderate (1)	1	

V: Flow velocity, D: Water depth, S: Substrate type, JP: Joint probability

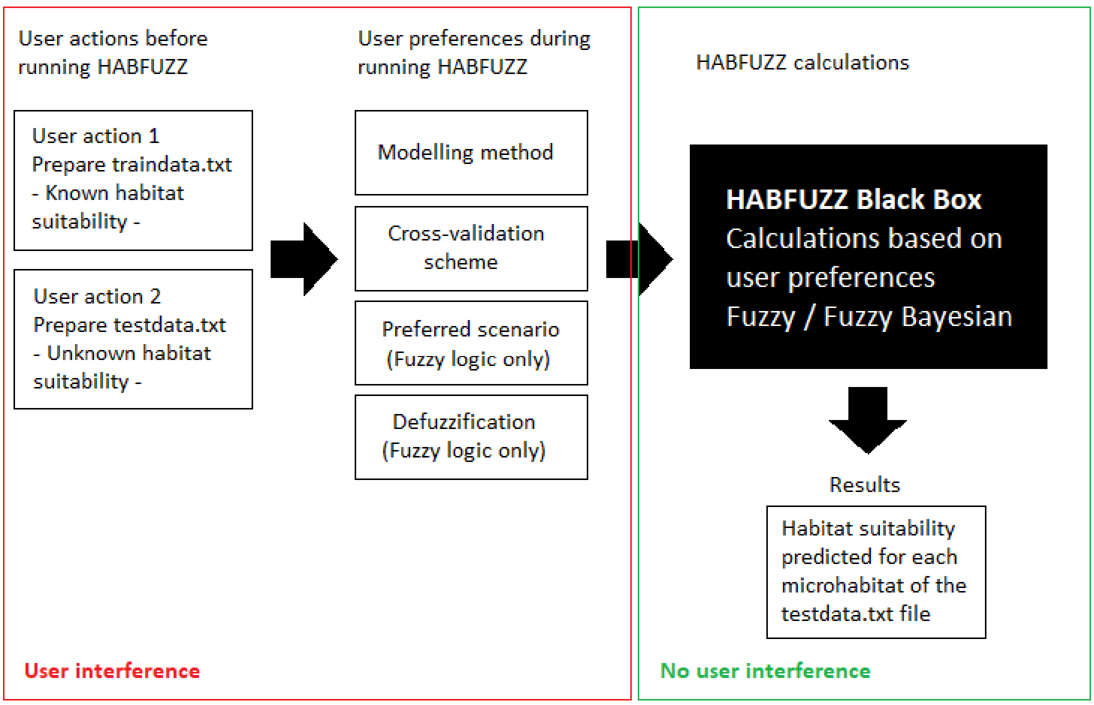
HABFUZZ then assigns a score at each habitat suitability class to calculate the final suitability output (bad - 0.1, poor - 0.3, moderate - 0.5, good - 0.7, high - 0.9), using the 'expected utility' equation. Each probability from tables 6B and 7B is multiplied by the score of each relevant suitability class and all products are summed (for each microhabitat) to derive the final habitat suitability. The results of the fuzzy Bayesian inference process are presented in table 9.

Table 9. The fuzzy Bayesian calculation of habitat suitability using the 'expected utility (EU)' equation

Microhabitats	Joints probability combinations							EU
1 -->	0.1455 x 0.9	0.0729 x 0.7	0.0239 x 0.5	0.0239 x 0.1	0.2668 x 0.7	0.2668 x 0.5	0.0667 x 0.9	0.577
2 -->	0.1667 x 0.9	0.3335 x 0.7	0.1667 x 0.5	0.0639 x 0.9	0.1412 x 0.7	0.1022 x 0.5	0.0256 x 0.3	0.677

4.3. Modifying the code according to the user preferences

The software has pre-defined categories for V, D, S and T stored in the *fdeclarations.f95* file. The user may wish to change these categories in order to adapt the program in specific cases. For every change at each file of HABFUZZ, re-compilation is necessary based on the aforementioned description. The calculation of the IF-THEN rules is then adapted to the new categories defined by the user. Experienced users may also wish to modify various parts of the open code.



References

Brookes C.J., Kumar V. and Lane S.N. 2010. A comparison of Fuzzy, Bayesian and Weighted Average formulations of an in-stream habitat suitability model. Proceedings of the International Congress on Environmental Modelling and Software, 5-8 Jul 2010, Ottawa, Canada. Available at http://www.iemss.org/iemss2010/papers/S20/S.20.07.Model%20selection%20and%20uncertainty%20A%20comparison%20of%20Fuzzy,%20Bayesian%20and%20Weighted%20Average%20formulations%20of%20an%20instream%20habitat%20suitability%20model%20-%20CHRISTOPHER%20BROOKES.pdf

Mamdani E.H. and Assilian S. 1975. An experiment in linguistic synthesis with a fuzzy logic Controller. International Journal of Man-Machine Studies 7: 1-13.

Ross T.J. 2010. Fuzzy logic with engineering applications. Third Edition, John Wiley and Sons, UK.

Zadeh L.A. 1965. Fuzzy sets. Information and Control 8: 338–353.

APPENDIX

Manning's n for channels (Chow, 1959)

Type of Channel and Description	Minimum	Normal	Maximum
Natural streams - minor streams (top width at floodstage < 100 ft)			
1. Main Channels			
a. clean, straight, full stage, no rifts or deep pools	0.025	0.030	0.033
b. same as above, but more stones and weeds	0.030	0.035	0.040
c. clean, winding, some pools and shoals	0.033	0.040	0.045
d. same as above, but some weeds and stones	0.035	0.045	0.050
e. same as above, lower stages, more ineffective slopes and sections	0.040	0.048	0.055
f. same as "d" with more stones	0.045	0.050	0.060
g. sluggish reaches, weedy, deep pools	0.050	0.070	0.080
h. very weedy reaches, deep pools, or floodways with heavy stand of timber and underbrush	0.075	0.100	0.150
2. Mountain streams, no vegetation in channel, banks usually steep, trees and brush along banks submerged at high stages			
a. bottom: gravels, cobbles, and few boulders	0.030	0.040	0.050
b. bottom: cobbles with large boulders	0.040	0.050	0.070
3. Floodplains			
a. Pasture, no brush			
1.short grass	0.025	0.030	0.035
2. high grass	0.030	0.035	0.050
b. Cultivated areas			
1. no crop	0.020	0.030	0.040
2. mature row crops	0.025	0.035	0.045
3. mature field crops	0.030	0.040	0.050
c. Brush			
1. scattered brush, heavy weeds	0.035	0.050	0.070
2. light brush and trees, in winter	0.035	0.050	0.060
3. light brush and trees, in summer	0.040	0.060	0.080
4. medium to dense brush, in winter	0.045	0.070	0.110
5. medium to dense brush, in summer	0.070	0.100	0.160
d. Trees			
1. dense willows, summer, straight	0.110	0.150	0.200

2. cleared land with tree stumps, no sprouts	0.030	0.040	0.050
3. same as above, but with heavy growth of sprouts	0.050	0.060	0.080
4. heavy stand of timber, a few down trees, little undergrowth, flood stage below branches	0.080	0.100	0.120
5. same as 4. with flood stage reaching branches	0.100	0.120	0.160
4. Excavated or Dredged Channels			
a. Earth, straight, and uniform			
1. clean, recently completed	0.016	0.018	0.020
2. clean, after weathering	0.018	0.022	0.025
3. gravel, uniform section, clean	0.022	0.025	0.030
4. with short grass, few weeds	0.022	0.027	0.033
b. Earth winding and sluggish			
1. no vegetation	0.023	0.025	0.030
2. grass, some weeds	0.025	0.030	0.033
3. dense weeds or aquatic plants in deep channels	0.030	0.035	0.040
4. earth bottom and rubble sides	0.028	0.030	0.035
5. stony bottom and weedy banks	0.025	0.035	0.040
6. cobble bottom and clean sides	0.030	0.040	0.050
c. Dragline-excavated or dredged			
1. no vegetation	0.025	0.028	0.033
2. light brush on banks	0.035	0.050	0.060
d. Rock cuts			
1. smooth and uniform	0.025	0.035	0.040
2. jagged and irregular	0.035	0.040	0.050
e. Channels not maintained, weeds and brush uncut			
1. dense weeds, high as flow depth	0.050	0.080	0.120
2. clean bottom, brush on sides	0.040	0.050	0.080
3. same as above, highest stage of flow	0.045	0.070	0.110
4. dense brush, high stage	0.080	0.100	0.140
5. Lined or Constructed Channels			
a. Cement			
1. neat surface	0.010	0.011	0.013
2. mortar	0.011	0.013	0.015
b. Wood			
1. planed, untreated	0.010	0.012	0.014
2. planed, creosoted	0.011	0.012	0.015
3. unplanned	0.011	0.013	0.015
4. plank with battens	0.012	0.015	0.018
5. lined with roofing paper	0.010	0.014	0.017

c. Concrete			
1. trowel finish	0.011	0.013	0.015
2. float finish	0.013	0.015	0.016
3. finished, with gravel on bottom	0.015	0.017	0.020
4. unfinished	0.014	0.017	0.020
5. gunite, good section	0.016	0.019	0.023
6. gunite, wavy section	0.018	0.022	0.025
7. on good excavated rock	0.017	0.020	
8. on irregular excavated rock	0.022	0.027	
d. Concrete bottom float finish with sides of:			
1. dressed stone in mortar	0.015	0.017	0.020
2. random stone in mortar	0.017	0.020	0.024
3. cement rubble masonry, plastered	0.016	0.020	0.024
4. cement rubble masonry	0.020	0.025	0.030
5. dry rubble or riprap	0.020	0.030	0.035
e. Gravel bottom with sides of:			
1. formed concrete	0.017	0.020	0.025
2. random stone mortar	0.020	0.023	0.026
3. dry rubble or riprap	0.023	0.033	0.036
f. Brick			
1. glazed	0.011	0.013	0.015
2. in cement mortar	0.012	0.015	0.018
g. Masonry			
1. cemented rubble	0.017	0.025	0.030
2. dry rubble	0.023	0.032	0.035
h. Dressed ashlar/stone paving	0.013	0.015	0.017
i. Asphalt			
1. smooth	0.013	0.013	
2. rough	0.016	0.016	
j. Vegetal lining	0.030		0.500