Markov chains computation for homogeneous and non-homogeneous data: MARCH 1.1 user's guide

André Berchtold

Université de Lausanne
Institut de Mathématiques Appliquées
SSP / BFSH2
CH-1015 LAUSANNE
Switzerland

Email: Andre@andreberchtold.com, Andre.Berchtold@imaa.unil.ch
Web: www.andreberchtold.com

March 27, 2001

Abstract

MARCH is a free software for the computation of different types of Markovian models including homogeneous Markov Chains, Hidden Markov Models (HMMs) and Double Chain Markov Models (DCMMs). The main characteristic of this software is the implementation of a powerful optimization method for HMMs and DCMMs combining a genetic algorithm with the standard Baum-Welch procedure. MARCH is distributed as a set of Matlab functions running under Matlab 5 or higher on any computing platform. A PC Windows version running independently from Matlab is also available.

Keywords: MARCH, Matlab, Independence model, Markov Chain (MC), Hidden Markov Model (HMM), Double Chain Markov Model (DCMM), Expectation-Maximization (EM) algorithm, Baum-Welch algorithm, Genetic Algorithm (GA).

Contents

1	Introduction	3
2	Theoretical background	3
2.1	Markovian models	3
2.2	Computational issues	5
2.3	Comparison of models	8
3	MARCH 1.1 user's guide	8
3.1	Availability	8
3.2	Copyright	8
3.3	Installation	8
3.4	Structure of the software	9
3.5	Data files	9
3.6	Choice of the maximal order	9
3.7	Models menu	10
3.8	Independence model	10
3.9	Homogeneous Markov chain	10
3.10	DCMM	11
3.11	Save and load results	15
3.12	Results file	15
3.13	Batch mode	15
3.14	Compiled version of MARCH for Windows	16
3.15	Tips and tricks	16
4	Examples	17
4.1	Theoretical example	17
4.2	Practical example	52
	References	76
A	The MARCH distribution	78
B	Variables	79
B.1	Independence model	79
B.2	Markov chain	79
B.3	DCMM	80
B.4	Other variables	80

List of Figures

1	A Double Chain Markov Model with first-order dependences.	5
2	Graphical display of hidden states (example 1).	46
3	Graphical display of hidden states (example 2).	74

1 Introduction

MARCH is a set of Matlab functions designed to compute various types of Markovian models for both homogeneous and non-homogenous data. This includes the independence model, homogeneous Markov Chains (MCs) of any order, Hidden Markov Models (HMMs) and Double Chain Markov Models (DCMMs). The standard version of MARCH requires Matlab 5 or higher to run, on any computing platform. In addition, a compiled version running independently from Matlab is available for the PC Windows platform.

This paper is organized as follows. Section 2 provides a short tutorial on Markovian models and on their computation. Section 3 is the user's guide of MARCH and Section 4 provides two examples. Finally, the Appendix gives more technical details about the functions and the variables used by the software.

2 Theoretical background

2.1 Markovian models

We deal here with Markovian models for categorical variables in discrete time. Basically, a Markov chain is a model in which the current value (time t) of a variable Y taking values in $\{1, \dots, K\}$ is fully explained by the knowledge of the value taken by the same variable at time $t-1$. This model is summarized in a *transition matrix* C_1 giving the probability distribution of Y_t given any possible value of Y_{t-1} :

$$C_1 = [P(Y_t = j | Y_{t-1} = i)] = [p_{ij}] = \begin{matrix} & & & t & & \\ & & & 1 & \dots & K \\ t-1 & 1 & \dots & K & & \\ & 1 & & & & \\ & \vdots & & & & \\ & K & & & & \end{matrix} \begin{vmatrix} p_{11} & \dots & p_{1K} \\ \vdots & \ddots & \vdots \\ p_{K1} & \dots & p_{KK} \end{vmatrix}$$

Each row of C_1 is a probability distribution summing to one. Since the current value is fully determined by the knowledge of only one past period, this model is said to be of order 1. This model is used in a lot of different fields including economy, chemistry, biology and meteorology.

More generally, a Markov chain of order f , $f \geq 0$, is a model in which the current value is explained by all lags up to $t - f$. The transition matrix is then of a larger size. For instance, for $f=2$ and $K=3$, we have

$$C_2 = \begin{matrix} & & & t & & & & & & & & \\ & & & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 \\ t-2 & t-1 & t-1 & 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 \\ & 1 & 1 & & & & & & & & & \\ & 2 & 1 & & & & & & & & & \\ & 3 & 1 & & & & & & & & & \\ & 1 & 2 & & & & & & & & & \\ & 2 & 2 & & & & & & & & & \\ & 3 & 2 & & & & & & & & & \\ & 1 & 3 & & & & & & & & & \\ & 2 & 3 & & & & & & & & & \\ & 3 & 3 & & & & & & & & & \end{matrix} \begin{vmatrix} p_{111} & 0 & 0 & p_{112} & 0 & 0 & p_{113} & 0 & 0 \\ p_{211} & 0 & 0 & p_{212} & 0 & 0 & p_{213} & 0 & 0 \\ p_{311} & 0 & 0 & p_{312} & 0 & 0 & p_{313} & 0 & 0 \\ 0 & p_{121} & 0 & 0 & p_{122} & 0 & 0 & p_{123} & 0 \\ 0 & p_{221} & 0 & 0 & p_{222} & 0 & 0 & p_{223} & 0 \\ 0 & p_{321} & 0 & 0 & p_{322} & 0 & 0 & p_{323} & 0 \\ 0 & 0 & p_{131} & 0 & 0 & p_{132} & 0 & 0 & p_{133} \\ 0 & 0 & p_{231} & 0 & 0 & p_{232} & 0 & 0 & p_{233} \\ 0 & 0 & p_{331} & 0 & 0 & p_{332} & 0 & 0 & p_{333} \end{vmatrix}$$

Given its particular form with a great number of zeros, this matrix can be rewritten in a reduced or compact form defined by Pegram (1980):

$$R_2 = \begin{array}{cc|ccc} & t-2 & t-1 & 1 & 2 & 3 \\ \hline 1 & 1 & 1 & p_{111} & p_{112} & p_{113} \\ 2 & 1 & 1 & p_{211} & p_{212} & p_{213} \\ 3 & 1 & 1 & p_{311} & p_{312} & p_{313} \\ 1 & 2 & 2 & p_{121} & p_{122} & p_{123} \\ 2 & 2 & 2 & p_{221} & p_{222} & p_{223} \\ 3 & 2 & 2 & p_{321} & p_{322} & p_{323} \\ 1 & 3 & 3 & p_{131} & p_{132} & p_{133} \\ 2 & 3 & 3 & p_{231} & p_{232} & p_{233} \\ 3 & 3 & 3 & p_{331} & p_{332} & p_{333} \end{array}$$

The special case $f = 0$ in which the current value is independent from the past is called the *independence* model. The transition matrix then becomes a single probability distribution:

$$C_0 = [P(Y_t = j)] = \begin{pmatrix} p_1 & \dots & p_K \end{pmatrix}$$

Consider now a more general situation. Let X be a variable taking values in the set $\{1, \dots, M\}$, and let Y be a variable taking values in the set $\{1, \dots, K\}$. We suppose that only the value taken by Y can be observed, the value taken by X at each period being hidden. Moreover, we suppose that the successive values taken by variables X and Y are not independent. We consider three different relations:

1. The value taken by X at time t (x_t) depends on the value taken by X at times $t-1$ to $t-\ell$, $\ell \geq 1$. X then follows an ℓ -th order homogeneous Markov chain.
2. The value of Y at time t depends on the values of Y at times $t-1$ to $t-f$, $f \geq 0$.
3. The value of Y at time t depends also on the value of X at time t .

When $f > 0$, the model describing the behavior of the observed variable Y is called a Double Chain Markov Model (DCMM). Figure 1 shows this model for the case $\ell=1$ and $f=1$.

A DCMM is fully described by three sets of probability distributions:

- A transition matrix A describing the relation between successive values of the hidden variable X .
- The initial distribution π of the hidden variable X . When A is of order $\ell=1$, π is a single probability distribution. When $\ell=2$, π is composed of the distribution of X at time 0 and of the conditional distribution of X_1 given X_0 , and so on.
- A set of transition matrices C_g , $g = 1, \dots, M$, describing the relation between successive outputs of the observed variable Y . When $X_t = g$, the matrix C_g is used.

At each time t , the hidden variable X takes one of the values $1, \dots, M$ according to A . The value of Y is then determined according to the transition matrix C_g corresponding to this hidden state.

The hidden states $1, \dots, M$ can be interpreted as a set of external conditions, or covariates, influencing the observed variable Y . This model is very good at the modeling of data such as animal behavior and wind speed.

Figure 1. A Double Chain Markov Model with first-order dependences. X denotes the hidden variable and Y the observed variable. At time t , X_t depends on X_{t-1} , and Y_t depends on both X_t and Y_{t-1} .

The well-known Hidden Markov Model (HMM) is a particular case of the DCMM in which the successive observations of Y are independent conditionally on X , that is $f = 0$. The set of transition matrices $\{C_1, \dots, C_M\}$ then reduces to a set of independence distributions. HMMs are widely used in speech recognition.

Finally, a homogeneous Markov chain can be viewed as a DCMM in which the external conditions represented by X never change. So, there is only one hidden state and one transition matrix C .

Since their introduction at the beginning of the 20th century by Russian mathematician Andrej Andreevic Markov, these models have generated a constantly growing literature. Homogeneous Markov chains are presented in detail e.g. in Dynkin (1965), Kemeny & Snell (1976), Doob (1990) and Berchtold (1998). Rabiner (1989) and MacDonald & Zucchini (1997) are good references about Hidden Markov Models. Finally, Paliwal (1993), Berchtold (1999a) and Berchtold (1999b) describe the more complete Double Chain Markov Model.

2.2 Computational issues

In the case of homogeneous Markov chains of order $f \geq 0$, the maximum likelihood estimator of the probability $p_{i_0|i_f, \dots, i_1}$ of observing $Y_t = i_0$ given $Y_{t-f} = i_f, \dots, Y_{t-1} = i_1$ is

$$\hat{p}_{i_0|i_f, \dots, i_1} = \frac{n_{i_f, \dots, i_1, i_0}}{n_{i_f, \dots, i_1, +}}$$

where $n_{i_f, \dots, i_1, i_0}$ is the number of sequences of the form

$$Y_{t-f} = i_f, \dots, Y_{t-1} = i_1, Y_t = i_0$$

in the data, and

$$n_{i_f, \dots, i_1, +} = \sum_{i_0=1}^K n_{i_f, \dots, i_1, i_0}$$

where K is the number of categories of the variable. The log-likelihood is then given by

$$LL = \sum_{i_0=1}^K \dots \sum_{i_f=1}^K n_{i_f, \dots, i_1, i_0} \log(\hat{p}_{i_0|i_f, \dots, i_1})$$

The computation of the more complex Double Chain Markov Model involves three different problems:

1. The estimation of the log-likelihood of the data given the model.
2. The estimation of π , A and $\{C_1, \dots, C_M\}$ given the data.
3. The estimation of the optimal sequence of hidden states given the model and the data.

Due to the structure of the DCMM, there is no direct formula to compute the log-likelihood. The problem is solved using an iterative procedure known as the forward-backward algorithm. The estimation of the model parameters is traditionally obtained by an Expectation-Maximization (EM) algorithm known in the speech recognition literature as the Baum-Welch algorithm. Finally, the optimal sequence of hidden states is computed using another iterative procedure called the Viterbi algorithm. Details about these procedures can be found e.g. in Rabiner (1989) for the forward-backward and the Baum-Welch algorithms, and in Forney (1973) for the Viterbi algorithm. In addition, McLachlan & Krishnan (1997) is a very good presentation of the EM method.

The estimation of the model parameters requires a very careful examination. Even if EM methods like the Baum-Welch algorithm are known to converge to an optimum of the solution space, nothing proves that this optimum is the global one. In fact, these algorithms tend to converge to the local optimum which is the closer to the starting values of the optimization procedure. The traditional remedy to this problem consists in running several times the EM algorithm, using different sets of random starting values, and to keep the best solution. More advanced versions of the EM algorithm (ECM, SEM, ...) can also be used, but they do not constitute a definitive answer to the local optima problem.

The use of a genetic algorithm (GA) is another possibility. This is an iterative procedure computing simultaneously several possible solutions (the population). At each iteration, a new population is created by combining the members of the current population using the principles of genetics, that is selection, crossover and mutation. This new population has a better probability to come close to the global optimum than the previous population had. This method presents the advantage to allow the exploration of a large part of the solution space, with a very high probability to find the global optimum. On the other hand, once the region of the solution space containing the global optimum has been determined, the robustness of GAs is balanced by the large number of iterations required to reach this optimum. See e.g. Holland (1975) and Coley (1999) for details about genetic algorithms and their implementation.

A very interesting solution has been proposed by Brouard et al. (1998). The idea is to combine a genetic algorithm (to explore quickly the whole solution space) and a Baum-Welch algorithm (to find the exact parameter values of the optimum). The procedure used to compute DCMMs in MARCH is based on this approach. After each iteration of the genetic algorithm, the members of the population are improved through several iterations of a Baum-Welch algorithm. This method is summarized in Algorithm 1. *IterGA* is the number of iterations of the genetic algorithm, *IterBW* is the number of iterations of the Baum-Welch algorithm, and *PopSize* is the size of the population used by the genetic algorithm.

The computation of the new population using the genetic algorithm follows the steps given in Algorithm 2. Note that the parameters are recoded in binary form at the beginning of the procedure so that each member

Algorithm 1 Estimation of a DCMM.

Random initialization of the population.
Computation of the log-likelihood of each member of the population.
for $iterga=1:IterGA$ **do** {Iterations of the genetic algorithm.}
 Computation of the new population.
 Recomputation of the log-likelihood of each member of the population.
 for $iterbw=1:IterBW$ **do** {Iterations of the Baum-Welch algorithm.}
 for $pop=1:PopSize$ **do** {Loop on the population size.}
 Reestimation of π , A , and $\{C_1, \dots, C_M\}$ for the pop -th member of the population.
 Recomputation of the log-likelihood of the pop -th member of the population.
 end for
 end for
end for
Computation of the optimal sequence of hidden states (Viterbi algorithm).

of the population is described by a vector of zeros and ones. Members of the old population are selected to be part of the new population according to their log-likelihood. This implies that best members of the old population have a greater probability to be chosen. $PCross$ is the probability of crossover (exchange of a part of the parameter values) between two members of the population. $PMut$ is the probability of mutation, that is the probability with which a 0 is replaced by a 1, and vice versa, in the binary writing of a member of the new population.

Algorithm 2 Computation of the new population.

Binary coding of each member of the old (current) population.
for $pop=1:PopSize$ **do**
 Selection: Random selection of a member of the old population (based on the value of its log-likelihood), to become a member of the new population.
 if pop is even **then**
 Crossover: Exchange a part of the binary vectors describing the last two members included in the new population with probability $PCross$.
 end if
end for
for $pop=1:PopSize$ **do**
 Mutation: Change 0s to 1s and vice-versa in the binary vector describing the pop -th member of the new population with probability $PMut$.
end for
if the best member of the old population is not included in the new population **then**
 Elitism: Replace the first member of the new population by the best member of the old population.
end if
Decoding of the binary vectors into real parameters form.

The same algorithm is used to compute HMMs, the only difference being that the order of transition matrices C is set to zero. Note that this algorithm can also be used to compute homogeneous Markov chains, but this implies a lot of unnecessary computations, so particular procedures are implemented in MARCH.

2.3 Comparison of models

Several methods exist to compare the performance of different models. Among these methods generally based on the log-likelihood, the Akaike Information Criterion (AIC), proposed by Akaike (1974), and the Bayesian Information Criterion (BIC), proposed by Schwarz (1978), are now very popular. AIC is defined as

$$\text{AIC} = -2 LL + 2 p$$

where LL is the log-likelihood of the model and p is the number of independent parameters. BIC is defined as

$$\text{BIC} = -2 LL + p \log(n)$$

where n is the number of components in the log-likelihood. In both cases, the model with the smallest criterion value is chosen.

Katz (1981) showed that AIC is inconsistent and that it tends to overestimate the order of a Markov chain. Following Katz, we preconize the use of BIC rather than AIC even if, for the purpose of comparison, the value of AIC is also computed by MARCH. The full display of a DCMM is allowed for the models reaching the best log-likelihood or the best BIC only.

Note that, following the convention established by Bishop et al. (1975), parameters estimated to zero are not counted as parameters. Note also that, to insure a fair comparison among models, it is necessary to have exactly the same number of components in the log-likelihood of each model. The variable *MaxOrder* of MARCH can be used to insure that (see Section 3.6).

3 MARCH 1.1 user's guide

3.1 Availability

MARCH is a set of Matlab functions requiring Matlab 5 or higher on any computing platform. It can be freely downloaded from

<http://www.andreberchtold.com/software.html>

A fully compiled version of MARCH, running independently from Matlab, is also available for the PC Windows platform. See Section 3.14 for more details. Note that this compiled version runs in batch mode only.

3.2 Copyright

The rights of MARCH belong to André Berchtold. The software can be freely distributed and used, but it cannot be sold or modified without authorization.

3.3 Installation

To install the set of Matlab functions, download the *march-11.zip* archive and extract all files in a directory called MARCH. Then, include this directory into the Matlab path (see your Matlab's local guide for details). Refer to Section 3.14 for details concerning the compiled version of MARCH.

3.4 Structure of the software

MARCH is a text only software. It is run by typing the command *march* at the Matlab prompt. The following menu is then displayed:

```
Main menu
-----
1. Load data file
2. Maximal order
3. Models
4. Exit
Your choice:
```

Any choice leads either to another menu or to the modification of a parameter value.

3.5 Data files

The first option of the *Main menu* lets the user open a data file. This is required prior performing any other operation (except exiting the software). The data file must be in a directory in which Matlab can access (usually in a part of the Matlab path, or in the current working directory).

Data files are text files containing one or several independent sequences of observations of a categorical variable Y taking values in the finite set $\{1, \dots, K\}$, $K > 1$. Note that any value outside the range 1- K will be considered as missing by the software. Each sequence must be put into a different *column* of the data file. Moreover, each column must have the same number of rows, so shorter columns must be completed by any value outside the range 1- K .

To open a data file, the user must enter the name of the file

```
Name of the dataset:
```

and the number of categories (K) of the variable

```
Number of possible outputs (-1 to quit):
```

Each sequence will be truncated after the first occurrence of a value outside the range 1- K .

Note that the name of the data file must be of the form *name.ext*, where *name* is the name of the file and *.ext* is an extension of at least one character (*.dat* for instance). The extension is not optional!

3.6 Choice of the maximal order

MARCH can compute models with any order of dependence greater or equal to zero between successive observations. However, in order to allow a fair comparison between models having different orders, the log-likelihood of each model should have the same number of components. For that purpose, the second option of the *Main menu* lets the user set a maximal order (variable *MaxOrder*). Then, the log-likelihood of any model whose maximal order is lower or equal to *Maxorder* will have the same number of components.

In practice, each model is computed conditionally on the first *MaxOrder* observations of each sequence. For instance, if you have one sequence of length 100 and if *MaxOrder* is set to 10, then the independence model will be computed on data points 11 to 100, the first-order homogeneous Markov chain will be computed on data points 11 to 100, conditionally on observation 10, and so on. In this case, the log-likelihood of each model will have exactly 90 components.

3.7 Models menu

The *Models* option of the *Main menu* leads to the following menu:

```
Models menu
-----
1. Independence
2. Homogeneous Markov chain
3. DCMM
4. Back to Main menu
Your choice:
```

These options are described in the next three sections.

3.8 Independence model

The *Independence* option of the *Models menu* leads to the following menu:

```
Independence menu
-----
1. Computation
2. Save
3. Back to Models menu
Your choice:
```

1. **Computation:** This option computes the independence model and displays the main results including the number of independent data sequences, the number of possible outputs of the variable, the number of independent parameters, the number of components in the log-likelihood, AIC, BIC, the frequency distribution of the data and the corresponding probability distribution.
2. **Save:** This option saves the main results of the independence model into a file in the Matlab format. See Sections 3.11 and B.1 for more details.

3.9 Homogeneous Markov chain

The *Homogeneous Markov chain* option of the *Models menu* leads to the following menu:

```
Markov chains menu
-----
1. Type of output
2. Computation
3. Save
4. Back to Models menu
Your choice:
```

1. **Type of output:** This option leads to a menu giving the choice between two types of output. In the *short* mode, the transition matrix of the Markov chain and the corresponding crosstable are not displayed. This is useful when you are dealing with very large matrices. The default option is the *normal* mode in which all results are displayed.
2. **Computation:** When choosing this option, the software asks for the order of the Markov chain, and then computes the model and displays the main results including the number of independent data sequences, the number of possible outputs of the variable, the order of the model, the number of independent parameters, the number of components in the log-likelihood, the number of structural zeros (parameters estimated to zero), the log-likelihood, AIC, BIC, and (if the output mode is set to *normal*) the crosstable and the transition matrix, both in reduced form.
3. **Save:** This option saves the main results of the Markov chain into a file in the Matlab format. See Sections 3.11 and B.2 for more details.

3.10 DCMM

The *DCMM* option of the *Models menu* leads to the following menu:

```
DCMM menu
-----
1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu
Your choice:
```

Here you can define, compute, and display any Double Chain Markov Model, including Hidden Markov Models.

1. **Choice of a model:** This option leads to a menu giving access to the parameters of the DCMM:

```
Choice of a DCMM
-----
1. Number of hidden states (1)
2. Order of the hidden chain (1)
3. Constraint on A (1)
4. Order of the visible chain (1)
5. Back to DCMM menu
Your choice:
```

You can modify the number of hidden states (M) and the order of the hidden and of the visible chains. The third option lets you impose a constraint on the matrix of the hidden process. By choosing “diagonal”, the hidden state won’t change during a sequence (but it can be different for each sequence). This option is especially useful when you are dealing with several short independent sequences. In this case, it is likely that each sequence is homogeneous, but that these sequences are different each from another. By default, the matrix A has no constraint and hence is fully parameterized, allowing the hidden states to change during a sequence. Numbers in brackets give the current values of each parameter.

A HMM is defined by setting to zero the order of the visible chain.

2. **Optimization parameters:** This option leads to a menu giving access to the optimization parameters:

```
Optimization parameters
-----
1. Type of initialization (1)
2. Maximal number of iterations (genetic algorithm) (1)
3. Population size (1)
4. Crossover probability (0.5)
5. Mutation probability (0.01)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (1)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu
Your choice:
```

The type of initialization is to be chosen between “random” (for the beginning of a new optimization procedure) and “current” to resume an optimization procedure starting with the current values.

Options 2 to 6 are related to the genetic algorithm. You can set the number of iterations, the size of the population, the crossover probability, the mutation probability, and the number of digits used to code each parameter in binary form.

When the number of iterations of the genetic algorithm is set to zero, no optimization takes place and the current parameters are used to compute the log-likelihood and the optimal sequence of hidden states. When the population size is set to 1, the genetic part of the procedure is suppressed and the algorithm becomes a pure Baum-Welch algorithm. The crossover and mutation probabilities take values between 0 and 1, the default values (0.5 and 0.01) being good starting points. The number of digits used to code the parameters in binary form has a direct influence upon the quality of the optimization. The greater this number, the more precise the estimation. For instance, a 20 digits coding can represent $2^{20} = 1'048'576$ different values. For a parameter taking values between 0 and 1, the smaller difference between two values that can be handled by the software is then about $1/1'048'576 < 10^{-6}$ what is fairly accurate. On the other hand, a greater number of digits increases the computation time, so an arbitrage has to be done.

Options 7 and 8 are related to the Baum-Welch (B-W) part of the procedure. You can set the number of iterations and a stop criterion.

The Baum-Welch algorithm is very time-consuming, so the number of iterations should be as small as possible. When the B-W algorithm is combined with the genetic algorithm, a number of iterations lower than 10 is generally sufficient. On the other hand, when the procedure is a pure B-W algorithm (that is when the population size is set to 1), the number of iterations of the B-W algorithm should be larger (for instance 100 or 1'000). Do not forget that it is always possible to resume an optimization procedure by starting with the current values.

The stop criterion of the B-W algorithm is based on the minimum amount of change in the log-likelihood between two successive iterations. When this change is equal or lower than the criterion, the B-W algorithm stops. This stop criterion should be used with great care since it is known that the Baum-Welch algorithm, like any EM algorithm, can sometimes slows down to very small amount of change in the log-likelihood and then speeds up again. Note that when set to zero, the stop criterion is totally ignored by the software, even if two successive iterations lead exactly to the same log-likelihood. So, when the stop criterion is set to zero, the number of iterations of the B-W algorithm is always exactly equal to the maximal number of iterations.

3. **Type of output:** This option leads to a menu giving access to different types of output:

```
Type of output
-----
1. Short (no B-W iterations, no hidden states)
2. Normal (no hidden states)
3. Long
4. Back to DCMM menu
Your choice (2):
```

In the *short* mode, the best log-likelihood reached at the end of each iteration of the Baum-Welch algorithm is not displayed (even when no genetic algorithm iteration take place) and the optimal sequence of hidden states is not displayed. In the *normal* mode, the hidden states are not displayed, and everything is displayed in the *long* mode. The best log-likelihood reached at the end of each iteration of the genetic algorithm is always displayed. You must be aware that the display of the hidden states sequences can be very long in the case of large data files.

4. **Computation:** This option starts the optimization procedure. During the optimization phase, the best log-likelihood obtained after each iteration of the genetic algorithm and of the Baum-Welch algorithm is displayed according to the *Type of output* mode. Note that when the population size is greater than 1, the displayed log-likelihood does not always concern the same member of the population.

At the end of the optimization procedure, results are displayed for both the member of the population having the largest log-likelihood and the member of the population having the smallest BIC. Of course, when the same member of the population is best in both cases, it is displayed only once.

Results include the number of independent data sequences, the number of possible outputs of the variable, the number of hidden states, the order of the hidden chain, the order of the visible chain, the number of independent parameters, the number of components in the log-likelihood, the log-likelihood, AIC, BIC, the first hidden state distribution (π), the reduced form of the hidden transition matrix (A), and the reduced form of the visible transition matrices ($C_1, \dots, C_M$). The C_g matrices are only vectors in the case of a HMM. Moreover, when the *Type of output* is set to *long*, the optimal sequences of

hidden states corresponding to each sequence of observations are displayed as well. The value 0 denotes hidden states which cannot be computed.

5. **Display of results:** This option leads to the following menu:

```
Display of results
```

```
-----
```

1. Best model in term of log-likelihood (no hidden states)
2. Best model in term of log-likelihood (with text display of hidden states)
3. Best model in term of log-likelihood (with graphic display of hidden states)
4. Best model in term of log-likelihood (with text and graphic display of hidden states)
5. Best model in term of BIC (no hidden states)
6. Best model in term of BIC (with text display of hidden states)
7. Best model in term of BIC (with graphic display of hidden states)
8. Best model in term of BIC (with text and graphic display of hidden states)
9. Back to DCMM menu

```
Your choice:
```

This menu lets you display the best model in term of log-likelihood or in term of BIC, with or without the optimal sequences of hidden states. This is especially useful when you have loaded a file containing the results of a previous procedure of optimization.

You can choose between a text and/or a graphic display of the hidden states sequences. In text mode, the number of the hidden state corresponding to each observation is displayed. The value 0 denotes hidden states which cannot be computed. In graphic mode, a figure containing as many rows as data sequences is open, the top row corresponding to the first sequence, and each hidden state is represented as a different color. A particular color is used to represent the parts of the sequences for which no hidden state can be computed. This happens for the first elements of each sequence, and for the last elements of sequences shorter than the largest sequence in the data file. Note that the figure containing the graphical display of hidden states must be closed manually. See Section 4 for practical examples.

6. **Load:** This option lets you load a file generated by the *Save* option and containing the results of a previous optimization procedure. When a file is loaded, the *Type of initialization* parameter is automatically set to *Current values*. Note that files generated by the *Save* option of the *Independence* and *Markov chain* menus cannot be used here.
7. **Save:** This option leads to the following menu:

```
Save
```

```
----
```

1. Save all
2. Save best model in term of log-likelihood
3. Save best model in term of BIC
4. Back to DCMM menu

```
Your choice:
```

You can choose to save either the whole population or only the best model in term of log-likelihood or in term of BIC. The generated file is in the Matlab format. It can be reloaded into MARCH using the *Load* option, or it can be open at the Matlab prompt. See Sections 3.11 and B.3 for more details.

3.11 Save and load results

Results of each model can be saved in a file using the standard Matlab *.mat* format. Then, they can be open at the Matlab prompt for further analysis of the results. Moreover, results of a DCMM can also be reloaded into MARCH to display the results or to resume the optimization process. Another use of this option is to apply a model computed on a first data set onto another data set. Note then that the new data set must absolutely have the same number of categories than the data set from which the model was computed.

It is not necessary to type the extension *.mat* when giving the name of a file. The set of saved variables is different for each type of model, so only files created by the *Save* option of the *DCMM menu* can be reloaded into MARCH. The list of variables saved for each type of model is given in Appendix B.

3.12 Results file

In addition to the *Save* command, results are saved in another way. Each time the software is run, a text file called *march_res.txt* is created in the current directory, and everything displayed is copied into it. After exiting MARCH, this file can be open and edited using any text processor.

Note that the presence of several files named *march_res.txt* within the Matlab path can lead to an error. Note also that the current *march_res.txt* file is erased each time the MARCH software is run, so it is a good idea to rename the file after exiting MARCH if you want to keep it intact.

3.13 Batch mode

In addition to the interactive mode described in the previous sections, MARCH can also be run in batch mode. A batch file is a text file containing the list of commands to be executed by the software. It must have the following structure:

- One command per row.
- No space before the name of a file (data set or saving file).
- No blank row.
- Comments can be placed after each command, except file names. This is done by writing a % followed by the comment on the same row as the command.
- No comments only row.

For instance, the following batch file opens a data set named *elnino12.dat* containing a variable having two categories. Then, the maximal order is set to 2, a second-order Markov chain is computed and results are saved in a file named *mc2_res.mat*.

```
1 % Load a data file
elnino12.dat
2 % Number of possible outputs
2 % Set MaxOrder
2
3 % Go to Models menu
2 % Go to Markov chains menu
```

```

2 % Compute a Markov chain
2 % Order of the Markov chain
3 % Save results
mc2_res
4 % Go back to Models menu
4 % Go back to Main menu
4 % Exit MARCH

```

To run the batch file, type `march('batchname.bat')` at the Matlab prompt, where *batchname.bat* is the name of the batch file. Do not forget the quotes!

3.14 Compiled version of MARCH for Windows

In addition to the set of Matlab functions, there is also a fully compiled version of MARCH running independently from Matlab on the PC Windows platform (95 or higher).

The main advantage of the compiled version is its portability. Since it does not require Matlab, it can be used on any computer running under PC Windows 95 or higher. The compiled version of MARCH has the following limitations:

- It runs in batch mode only.
- The batch file must obligatory be named *march_bat.bat* and it must be placed in the same directory as the software.

To install the compiled distribution of MARCH 1.1, download the file *march-11-c-win.zip* from

<http://www.andreberchtold.com/software.html>

and extract the *zip* archive into a directory called *march_c*. Note that all files contained in the archive must be left in this directory, including the *bin* subdirectory. The executable file is named *march_c.exe*. To run the software, double-click on *march_c.exe*. This will automatically open and execute the batch file named *march_bat.bat*.

3.15 Tips and tricks

We provide here some hints about the use of MARCH.

- **Optimization of the best member of the population:** As noted before, the genetic algorithm used by MARCH can quickly find the region of the solution space containing the global optimum, but then it can failed for a long time to reach this optimum. So, a good strategy is to run the full procedure (genetic and Baum-Welch algorithms) for a time, and then to select the best model and to use it as the starting point of a Baum-Welch only procedure. This can be done in the following way: After the full procedure has stopped, save the best model (in term of log-likelihood or in term of BIC, depending on your interest) in a file. Then, load again this file containing a population of size 1, set the maximal number of iterations of the genetic algorithm to 1, and run a Baum-Welch algorithm with a large number of iterations. Note that it is a good idea to save as well in another file the whole population used by the genetic algorithm in order to be able to go back to that point if the B-W optimization of the best model is disappointing.

- **No genetic algorithm iterations:** When the number of iterations of the genetic algorithm is set to zero, the DCMM is not reestimated, but the log-likelihood and the optimal hidden states sequences are nevertheless computed. This is useful when you want to compute the same model on different data sets. To do that, save the best model in a file, go to the *Main menu*, load another data file, go back to the *DCMM menu*, reload the model previously saved, set the number of iterations of the genetic algorithm to zero, and compute the model. Results will include the log-likelihood and the optimal sequences of hidden states computed on the new data set using the model computed on the previous data set.

4 Examples

In this section, we provide two examples demonstrating the main features of MARCH. The data files and some other files used here can be downloaded from

<http://www.andreberchtold.com/software.html>

The archives are named *march-ex1.zip* for the example of section 4.1 and *march-ex2.zip* for the example of section 4.2.

4.1 Theoretical example

This example uses two data files named *ex_1a.dat* and *ex_1b.dat*. The first file contains three independent sequences of observations of a variable taking values 1 to 3. The first two sequences are of length 46. The last one is shorter with only 32 observations, so it had to be completed with the value 0, which is treated as a missing value by MARCH.

1	2	3
1	1	3
1	2	3
1	3	3
1	2	3
1	1	2
2	2	2
2	3	2
2	2	2
2	1	2
2	2	2
1	3	3
1	3	3
1	2	3
1	3	3
1	1	3
1	3	1
3	2	1
3	3	1
3	1	1
3	3	1
3	1	1
3	3	3

3	1	3
2	2	3
2	3	3
2	2	3
2	1	2
2	2	2
2	1	2
1	2	2
1	3	2
1	2	0
1	1	0
1	2	0
1	3	0
2	2	0
2	3	0
2	2	0
2	1	0
2	2	0
3	2	0
3	1	0
3	3	0
3	1	0
3	1	0

The second data file contains only one sequence of length 109 of the same variable. For the purpose of clarity, the file is here transposed and has to be read row by row, but the real file *ex_1b.dat* is truly a one *column* file.

1	1	1	2	1	1	1	2	1	1	1	2
3	3	2	3	2	3	2	3	2	3	3	2
3	1	1	1	2	1	1	1	2	1	1	1
2	3	3	2	3	2	3	1	2	3	3	2
2	3	3	2	3	2	1	1	1	2	1	1
1	3	1	1	1	2	1	1	1	2	3	2
3	2	2	2	2	2	1	1	1	1	1	2
1	2	2	2	2	2	1	1	1	2	2	2
1	1	1	1	2	1	2	2	1	1	1	1
2											

Using these two data files, we defined an analysis whose main points are the following:

1. The file named *ex_1a.dat* is loaded.
2. The maximal order is set to 2.
3. The independence model and homogeneous Markov chains of order 1 and 2 are computed. In the case of the second-order Markov chain, the transition matrix and the corresponding crosstable are not displayed. Results for each model are saved in a file in Matlab format.
4. A HMM with two hidden states and a first-order hidden transition matrix is defined.

5. The optimization parameters are modified in the following way:
 - 10 iterations of the genetic algorithm.
 - A population size of 20.
 - A mutation probability of 0.02.
 - 3 iterations of the Baum-Welch algorithm.
6. The output type is set to *long*.
7. The HMM is computed.
8. The whole population is saved in a file named *ex1_hmm.mat*.
9. The previous model is transformed into a DCMM by setting the order of the visible chain to 1.
10. The DCMM is computed with the same optimization parameters used for the HMM.
11. The whole population is saved in a file named *ex1_dcmm.mat*.
12. The best model in term of BIC is saved in a file named *ex1_dcmm_b.mat*.
13. The file *ex1_dcmm_b.mat* is loaded. By doing that, the population of size 20 used during the previous optimization procedure is replaced by the best solution in term of BIC and the type of initialization is automatically set to *current values*.
14. The optimization parameters are modified in the following way:
 - 1 iteration of the genetic algorithm.
 - 100 iterations of the Baum-Welch algorithm.
 - The stop criterion of the Baum-Welch algorithm is set to 0.000001.
15. The DCMM is computed. Formally, we try to improve the best model obtained through the previous optimization procedure by using a Baum-Welch only procedure.
16. The parameters of the final model are displayed. The hidden states are presented graphically.
17. The results are saved in a file named *ex1_dcmm_b2.mat*.
18. A new data file named *ex_1b.dat* is loaded.
19. The file *ex1_dcmm_b2.mat* containing the parameters of the best DCMM computed from the first data file is loaded.
20. The number of iterations of the genetic algorithm is set to zero.
21. The output type is set to *long*.
22. The DCMM is computed. Formally, since the number of iterations of the genetic algorithm is zero, MARCH just compute the log-likelihood and the optimal sequence of hidden states for the file *example2.dat*, using the best DCMM computed from *example1.dat*.

23. The software is stopped.

The following batch file named *march_ex1.bat* contains all the commands required to perform these tasks. Note that if you are using the compiled version of MARCH, you have to rename the batch file as *march_bat.bat*. The batch file contains the following commands and comments:

```
1 % Load a data file
ex_1a.dat
3 % Number of possible outputs
2 % Set MaxOrder
2
3 % Go to Models menu
1 % Independence model
1 % Computation
2 % Save results
ex1_indep
3 % Go back to Models menu
2 % Go to Markov chain menu
2 % Compute a Markov chain
1 % Order of the Markov chain
3 % Save results
ex1_mcl
1 % Type of output
1 % Short
3 % Go back to Markov chain menu
2 % Compute a Markov chain
2 % Order of the Markov chain
3 % Save results
ex1_mc2
4 % Go back to Models menu
3 % Go to DCMM menu
1 % Choice of a model
1 % Choice of the number of hidden states
2
4 % Choice of the order of the visible chain
0 % HMM
5 % Back to DCMM menu
2 % Optimization parameters
2 % Choice of the maximal number of iterations of the GA
10
3 % Choice of the population size
20
5 % Choice of the mutation probability
0.02
7 % Choice of the maximal number of iterations of the B-W algorithm
3
```

```

9 % Back to DCMM menu
3 % Type of output
3 % Long
4 % Back to DCMM menu
4 % Computation
7 % Save results
1 % All
ex1_hmm
4 % Back to DCMM menu
1 % Choice of a model
4 % Choice of the order of the visible chain
1 % DCMM
5 % Back to DCMM menu
4 % Computation
7 % Save results
1 % All
ex1_dcmm
3 % Best model in term of BIC
ex1_dcmm_b
4 % Back to DCMM menu
6 % Load
ex1_dcmm_b
2 % Optimization parameters
2 % Choice of the maximal number of iterations of the GA
1
7 % Choice of the maximal number of iterations of the B-W algorithm
100
8 % Choice of the stop criterion of the B-W algorithm
0.000001
9 % Back to DCMM menu
4 % Computation
5 % Display results
7 % Best model in term of BIC with graphic display of hidden states
9 % Back to DCMM menu
7 % Save results
3 % Best model in term of BIC
ex1_dcmm_b2
4 % Back to DCMM menu
8 % Back to Models menu
4 % Back to main menu
1 % Load a data file
ex1b.dat
3 % Number of possible outputs
3 % Go to Models menu
3 % Go to DCMM menu

```

```

6 % Load
ex1_dcmm_b2
2 % Optimization parameters
2 % Choice of the maximal number of iterations of the GA
0
9 % Back to DCMM menu
3 % Type of output
3 % Long
4 % Back to DCMM menu
4 % Computation
8 % Back to Models menu
4 % Go back to Main menu
4 % Exit MARCH

```

Note that the only purpose of this example is to demonstrate some of the capabilities of MARCH. It must not be regarded as a rigorous analysis of the data files! Moreover, since the parameters are randomly initialized, since a genetic algorithm makes great use of probabilities and random values, and since the numbers of iterations chosen here are sometimes too small to allow convergence, the results obtained will slightly differ each time the batch file is run. So, do not be surprised if your results are different.

The results file generated from this batch file looks like the output named *march_res_ex1.txt* which is reproduced and commented below.

1. The first data file is loaded.

```

*****
*                MARCH    1.1                *
* Copyright Andre Berchtold, 2000--2001 *
*****

Main menu
-----
1. Load data file
2. Maximal order
3. Models
4. Exit
Your choice: 1

Name of the dataset: ex_1a.dat

Number of possible outputs (-1 to quit): 3

*** Data file successfully loaded ***

```

2. The maximal order is set to 2.

```

Main menu

```

```

-----
1. Load data file
2. Maximal order
3. Models
4. Exit
Your choice: 2

Maximal order (0) (-1 to quit): 2

*** Maximal order successfully set to 2 ***

```

3. The independence model and homogeneous Markov chains of order 1 and 2 are computed. Among these three models, the second-order Markov chain achieves the best log-likelihood (-94.0466), but despite a much smaller log-likelihood, the first-order Markov chain is preferred in term of BIC (261.679) because of its parsimony (6 independent parameters against 18).

```

Main menu
-----
1. Load data file
2. Maximal order
3. Models
4. Exit
Your choice: 3

Models menu
-----
1. Independence
2. Homogeneous Markov chain
3. DCMM
4. Back to Main menu
Your choice: 1

```

```

Independence menu
-----
1. Computation
2. Save
3. Back to Models menu
Your choice: 1

```

```

*****
* Independence model *
*****

```

Number of independent sequences: 3

Number of possible outputs: 3

Number of independent parameters: 2

Number of components in the log-likelihood: 118

Log-likelihood: -129.12

AIC: 262.241

BIC: 267.782

Distribution of data:

35	44	39
----	----	----

Probability distribution of data:

0.2966	0.3729	0.3305
--------	--------	--------

Independence menu

1. Computation

2. Save

3. Back to Models menu

Your choice: 2

Name of the saving file (without extension): exl_indep

*** Results successfully saved ***

Independence menu

1. Computation

2. Save

3. Back to Models menu

Your choice: 3

Models menu

1. Independence

2. Homogeneous Markov chain

3. DCMM

4. Back to Main menu

Your choice: 2

Markov chains menu

1. Type of output

2. Computation

3. Save

4. Back to Models menu

Your choice: 2

Order of the Markov chain (-1 to quit): 1

*** Markov chain of order 1 successfully specified ***


```

*****
* Markov chain *
*****

Number of independent sequences: 3

Number of possible outputs: 3

Order of the model: 1

Number of independent parameters: 6

Number of components in the log-likelihood: 118

Number of parameters estimated to zero: 0

Log-likelihood: -116.528

AIC: 245.055

BIC: 261.679

Crosstable (reduced form):
      20      10      6
      9      23     11
      6      11     22

Transition matrix (reduced form):
0.5556  0.2778  0.1667
0.2093  0.5349  0.2558
0.1538  0.2821  0.5641

Markov chains menu
-----
1. Type of output
2. Computation
3. Save
4. Back to Models menu
Your choice: 3

Name of the saving file (without extension): ex1_mcl

*** Results successfully saved ***

Markov chains menu
-----
1. Type of output
2. Computation
3. Save
4. Back to Models menu
Your choice: 1

```

Type of output

1. Short (no display of the contingency table and the transition matrix)
2. Normal
3. Back to Markov chains menu

Your choice (2): 1

Type of output

1. Short (no display of the contingency table and the transition matrix)
2. Normal
3. Back to Markov chains menu

Your choice (1): 3

Markov chains menu

1. Type of output
2. Computation
3. Save
4. Back to Models menu

Your choice: 2

Order of the Markov chain (-1 to quit): 2

*** Markov chain of order 2 successfully specified ***

* Markov chain *

Number of independent sequences: 3

Number of possible outputs: 3

Order of the model: 2

Number of independent parameters: 18

Number of components in the log-likelihood: 118

Number of parameters estimated to zero: 0

Log-likelihood: -94.0466

AIC: 224.093

BIC: 273.965

Markov chains menu

1. Type of output
2. Computation
3. Save
4. Back to Models menu

Your choice: 3

Name of the saving file (without extension): ex1_mc2

*** Results successfully saved ***

Markov chains menu

1. Type of output
2. Computation
3. Save
4. Back to Models menu

Your choice: 4

4. A first-order Hidden Markov Model with two hidden states is defined.

Models menu

1. Independence
2. Homogeneous Markov chain
3. DCMM
4. Back to Main menu

Your choice: 3

DCMM menu

1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu

Your choice: 1

Choice of a DCMM

1. Number of hidden states (1)
2. Order of the hidden chain (1)
3. Constraint on A (1)
4. Order of the visible chain (1)
5. Back to DCMM menu

Your choice: 1

Number of hidden states (1) (-1 to quit): 2

Choice of a DCMM

1. Number of hidden states (2)
2. Order of the hidden chain (1)
3. Constraint on A (1)
4. Order of the visible chain (1)
5. Back to DCMM menu

Your choice: 4

Order of the visible chain (1) (0 for HMM, -1 to quit): 0

Choice of a DCMM

1. Number of hidden states (2)
2. Order of the hidden chain (1)
3. Constraint on A (1)
4. Order of the visible chain (0)
5. Back to DCMM menu

Your choice: 5

*** HMM successfully specified ***

5. New optimization parameters are set.

DCMM menu

1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu

Your choice: 2

Optimization parameters

1. Type of initialization (1)
2. Maximal number of iterations (genetic algorithm) (1)
3. Population size (1)
4. Crossover probability (0.5)
5. Mutation probability (0.01)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (1)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu

Your choice: 2

Maximal number of iterations (1) (-1 to quit): 10

Optimization parameters

1. Type of initialization (1)
2. Maximal number of iterations (genetic algorithm) (10)
3. Population size (1)
4. Crossover probability (0.5)
5. Mutation probability (0.01)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (1)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu

Your choice: 3

Population size (1) (-1 to quit): 20

Optimization parameters

1. Type of initialization (1)
2. Maximal number of iterations (genetic algorithm) (10)
3. Population size (20)
4. Crossover probability (0.5)
5. Mutation probability (0.01)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (1)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu

Your choice: 5

Mutation probability (0.01) (-1 to quit): 0.02

Optimization parameters

1. Type of initialization (1)
2. Maximal number of iterations (genetic algorithm) (10)
3. Population size (20)
4. Crossover probability (0.5)
5. Mutation probability (0.02)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (1)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu

Your choice: 7

Maximal number of iterations (1) (-1 to quit): 3

Optimization parameters

1. Type of initialization (1)
2. Maximal number of iterations (genetic algorithm) (10)

```

3. Population size (20)
4. Crossover probability (0.5)
5. Mutation probability (0.02)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (3)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu
Your choice: 9

```

*** Optimization parameters successfully specified ***

6. The output type is set to *long* to display everything including hidden states.

```

DCMM menu
-----
1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu
Your choice: 3

```

```

Type of output
-----
1. Short (no B-W iterations, no hidden states)
2. Normal (no hidden states)
3. Long
4. Back to DCMM menu
Your choice (2): 3

```

```

Type of output
-----
1. Short (no B-W iterations, no hidden states)
2. Normal (no hidden states)
3. Long
4. Back to DCMM menu
Your choice (3): 4

```

7. The HMM is computed. In this case, the best value achieved after each iteration of the genetic algorithm is most of the time equal to the previous best value. This can be somewhat disappointing, but do not forget that 20 models are evaluated at each iteration, so that even if this is not displayed, a large part of the solution space is explored. From that point of view, the genetic algorithm helps to insure that the solution displayed is a good approximation of the *global* optimum.

Here, the optimization procedure does not converge (the difference in log-likelihood between the last two iterations is not exactly zero). So, in practice, we should run again an optimization procedure before using this model.

The same model is best in term of both log-likelihood and BIC. The two hidden states are clearly different: In the first one, values 2 and 3 have both high probabilities (0.4643 and 0.4112), when value 1 is highly preferred in state 2 with probability 0.9987. The two zeros at the beginning of each hidden states sequence correspond to the two data on which each model has been conditioned by choosing a maximal order equal to 2.

DCMM menu

1. Choice of a model
 2. Optimization parameters
 3. Type of output
 4. Computation
 5. Display of results
 6. Load
 7. Save
 8. Back to Model menu
- Your choice: 4

Initialization

-127.106

Iteration 1 (GA)

-127.106

Iteration 1 (B-W)

-124.586

Iteration 2 (B-W)

-123.445

Iteration 3 (B-W)

-122.796

Iteration 2 (GA)

-122.796

Iteration 1 (B-W)

-122.282

Iteration 2 (B-W)

-121.469

Iteration 3 (B-W)

-121.174

Iteration 3 (GA)

-121.174

Iteration 1 (B-W)

-120.781

Iteration 2 (B-W)

-120.541

Iteration 3 (B-W)

-120.312

Iteration 4 (GA)

-120.312

Iteration 1 (B-W)

-120.066
Iteration 2 (B-W)
-119.793
Iteration 3 (B-W)
-119.496

Iteration 5 (GA)
-119.496
Iteration 1 (B-W)
-119.105
Iteration 2 (B-W)
-118.806
Iteration 3 (B-W)
-118.556

Iteration 6 (GA)
-118.515
Iteration 1 (B-W)
-118.329
Iteration 2 (B-W)
-118.195
Iteration 3 (B-W)
-118.101

Iteration 7 (GA)
-118.101
Iteration 1 (B-W)
-118.035
Iteration 2 (B-W)
-117.952
Iteration 3 (B-W)
-117.888

Iteration 8 (GA)
-117.888
Iteration 1 (B-W)
-117.87
Iteration 2 (B-W)
-117.862
Iteration 3 (B-W)
-117.858

Iteration 9 (GA)
-117.858
Iteration 1 (B-W)
-117.854
Iteration 2 (B-W)
-117.852
Iteration 3 (B-W)
-117.85


```

Iteration 10 (GA)
-117.85
Iteration 1 (B-W)
-117.848
Iteration 2 (B-W)
-117.847
Iteration 3 (B-W)
-117.846

*** Model successfully computed ***

*****
* HMM *
*****

*** Best model in term of BIC and log-likelihood ***

Number of independent sequences: 3

Number of possible outputs: 3

Number of hidden states: 2

Order of the hidden chain: 1

Order of the visible chain: 0

Number of independent parameters: 7

Number of components in the log-likelihood: 118

Log-likelihood: -117.846

AIC: 249.692

BIC: 269.087

First hidden state distribution:
0.6743    0.3257

Transition matrix between hidden states (reduced form):
0.9468  0.0532
0.2288  0.7712

Transition matrices between outputs (reduced form):

State 1
0.1245  0.4643  0.4112

State 2
0.9987  0.0000  0.0013

```

Hidden states:

Sequence 1:

0	0	2	2	2	2	1	1	1	1	1	2
2	2	2	2	2	1	1	1	1	1	1	1
1	1	1	1	1	1	2	2	2	2	2	2
1	1	1	1	1	1	1	1	1	1		

Sequence 2:

0	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	2	2		

Sequence 3:

0	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	2	2	2	2	2	2	1	1
1	1	1	1	1	1	1	1				

8. The whole population is saved in a file named *ex1_hmm.mat*.

DCMM menu

1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu

Your choice: 7

Save

1. Save all
2. Save best model in term of log-likelihood
3. Save best model in term of BIC
4. Back to DCMM menu

Your choice: 1

Name of the saving file (without extension): ex1_hmm

*** All results successfully saved ***

Save

1. Save all
2. Save best model in term of log-likelihood
3. Save best model in term of BIC

4. Back to DCMM menu
Your choice: 4

9. A DCMM with two hidden states and first-order transition matrices for both the hidden and the visible chains is defined.

```
DCMM menu
-----
1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu
Your choice: 1
```

```
Choice of a DCMM
-----
1. Number of hidden states (2)
2. Order of the hidden chain (1)
3. Constraint on A (1)
4. Order of the visible chain (0)
5. Back to DCMM menu
Your choice: 4
```

Order of the visible chain (0) (0 for HMM, -1 to quit): 1

```
Choice of a DCMM
-----
1. Number of hidden states (2)
2. Order of the hidden chain (1)
3. Constraint on A (1)
4. Order of the visible chain (1)
5. Back to DCMM menu
Your choice: 5
```

*** DCMM successfully specified ***

10. The DCMM is computed. During the last iterations, the log-likelihood does not change (-80.681), indicating that an optimum has been found. With 13 independent parameters, this model achieves a BIC a lot smaller than the BICs obtained by the previous models. The same model is best in term of both log-likelihood and BIC.

The two hidden states have well defined roles. State 1 appears in sequences 1 and 3, when state 2 appears in sequence 2. State 1 is defined by high probabilities for the visible variable to keep the same value from period to period (values 0.8261, 0.8462 and 0.8400 on the main diagonal of the matrix). On the other hand, state 2 is defined by high probabilities to switch from one value to another (low

probabilities on the main diagonal). This behavior was expected in regard of the data file analyzed. Note that the hidden transition matrix has been estimated as the identity matrix, even if no constraint has been put on this matrix. This explain why the hidden states do not change during a sequence. The two zeros at the beginning of each hidden states sequence correspond to the two data on which each model has been conditioned by choosing a maximal order equal to 2.

DCMM menu

1. Choice of a model
 2. Optimization parameters
 3. Type of output
 4. Computation
 5. Display of results
 6. Load
 7. Save
 8. Back to Model menu
- Your choice: 4

Initialization

-118.767

Iteration 1 (GA)

-118.445

Iteration 1 (B-W)

-115.039

Iteration 2 (B-W)

-113.791

Iteration 3 (B-W)

-111.48

Iteration 2 (GA)

-111.48

Iteration 1 (B-W)

-97.0619

Iteration 2 (B-W)

-84.6848

Iteration 3 (B-W)

-81.425

Iteration 3 (GA)

-81.425

Iteration 1 (B-W)

-80.9483

Iteration 2 (B-W)

-80.7727

Iteration 3 (B-W)

-80.7115

Iteration 4 (GA)

-80.7115

Iteration 1 (B-W)

-80.6904
Iteration 2 (B-W)
-80.684
Iteration 3 (B-W)
-80.6819

Iteration 5 (GA)
-80.6819
Iteration 1 (B-W)
-80.6813
Iteration 2 (B-W)
-80.6811
Iteration 3 (B-W)
-80.681

Iteration 6 (GA)
-80.681
Iteration 1 (B-W)
-80.681
Iteration 2 (B-W)
-80.681
Iteration 3 (B-W)
-80.681

Iteration 7 (GA)
-80.681
Iteration 1 (B-W)
-80.681
Iteration 2 (B-W)
-80.681
Iteration 3 (B-W)
-80.681

Iteration 8 (GA)
-80.681
Iteration 1 (B-W)
-80.681
Iteration 2 (B-W)
-80.681
Iteration 3 (B-W)
-80.681

Iteration 9 (GA)
-80.681
Iteration 1 (B-W)
-80.681
Iteration 2 (B-W)
-80.681
Iteration 3 (B-W)
-80.681

```

Iteration 10 (GA)
-80.681
Iteration 1 (B-W)
-80.681
Iteration 2 (B-W)
-80.681
Iteration 3 (B-W)
-80.681

*** Model successfully computed ***

*****
* DCMM *
*****

*** Best model in term of BIC and log-likelihood ***

Number of independent sequences: 3

Number of possible outputs: 3

Number of hidden states: 2

Order of the hidden chain: 1

Order of the visible chain: 1

Number of independent parameters: 13

Number of components in the log-likelihood: 118

Log-likelihood: -80.681

AIC: 187.362

BIC: 223.381

First hidden state distribution:
0.6667    0.3333

Transition matrix between hidden states (reduced form):
1.0000  0.0000
0.0000  1.0000

Transition matrices between outputs (reduced form):

State 1
0.8261  0.0870  0.0870
0.0769  0.8462  0.0769
0.0400  0.1200  0.8400

```

```

State 2
0.0769  0.6154  0.3077
0.4118  0.0588  0.5294
0.3571  0.5714  0.0714

```

Hidden states:

Sequence 1:

```

0    0    1    1    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    1    1    1    1    1

```

Sequence 2:

```

0    0    2    2    2    2    2    2    2    2    2    2
2    2    2    2    2    2    2    2    2    2    2    2
2    2    2    2    2    2    2    2    2    2    2    2
2    2    2    2    2    2    2    2    2    2    2    2

```

Sequence 3:

```

0    0    1    1    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    1    1    1    1    1

```

11. The whole population is saved in a file named *ex1_dcmm.mat*.

DCMM menu

```

1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu
Your choice: 7

```

Save

```

1. Save all
2. Save best model in term of log-likelihood
3. Save best model in term of BIC
4. Back to DCMM menu
Your choice: 1

```

Name of the saving file (without extension): *ex1_dcmm*

*** All results successfully saved ***

12. The best model in term of BIC is saved in a file named *ex1_dcmm_b.mat*.

```

Save
----
1. Save all
2. Save best model in term of log-likelihood
3. Save best model in term of BIC
4. Back to DCMM menu
Your choice: 3

Name of the saving file (without extension): ex1_dcmm_b

*** Best model successfully saved ***

```

```

Save
----
1. Save all
2. Save best model in term of log-likelihood
3. Save best model in term of BIC
4. Back to DCMM menu
Your choice: 4

```

13. The file named *ex1_dcmm_b.mat* is loaded.

```

DCMM menu
-----
1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu
Your choice: 6

Name of the loading file (without extension): ex1_dcmm_b

*** File successfully loaded ***

```

14. New optimization parameters are set in order to perform a Baum-Welch only optimization procedure.

```

DCMM menu
-----
1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save

```


8. Back to Model menu

Your choice: 2

Optimization parameters

1. Type of initialization (2)
2. Maximal number of iterations (genetic algorithm) (10)
3. Population size (1)
4. Crossover probability (0.5)
5. Mutation probability (0.02)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (3)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu

Your choice: 2

Maximal number of iterations (10) (-1 to quit): 1

Optimization parameters

1. Type of initialization (2)
2. Maximal number of iterations (genetic algorithm) (1)
3. Population size (1)
4. Crossover probability (0.5)
5. Mutation probability (0.02)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (3)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu

Your choice: 7

Maximal number of iterations (3) (-1 to quit): 100

Optimization parameters

1. Type of initialization (2)
2. Maximal number of iterations (genetic algorithm) (1)
3. Population size (1)
4. Crossover probability (0.5)
5. Mutation probability (0.02)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (100)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu

Your choice: 8

Stop criterion (0) (0 to inactivate criterion, -1 to quit): 1e-006

Optimization parameters

1. Type of initialization (2)

```

2. Maximal number of iterations (genetic algorithm) (1)
3. Population size (1)
4. Crossover probability (0.5)
5. Mutation probability (0.02)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (100)
8. Stop criterion (B-W algorithm) (1e-006)
9. Back to DCMM menu
Your choice: 9

```

```

*** Optimization parameters successfully specified ***

```

15. The optimization of the DCMM resumes, starting from the best model obtained through the previous optimization procedure. It appears that the previous model cannot be improved. The log-likelihood does not change and the procedure ends after one iteration only.

```

DCMM menu
-----

```

```

1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu
Your choice: 4

```

```

Initialization

```

```

-80.681

```

```

Iteration 1 (B-W)

```

```

-80.681

```

```

*** Model successfully computed ***

```

```

*****

```

```

* DCMM *

```

```

*****

```

```

*** Best model in term of BIC and log-likelihood ***

```

```

Number of independent sequences: 3

```

```

Number of possible outputs: 3

```

```

Number of hidden states: 2

```

```

Order of the hidden chain: 1

```

```

Order of the visible chain: 1

```

```

Number of independent parameters: 13

Number of components in the log-likelihood: 118

Log-likelihood: -80.681

AIC: 187.362

BIC: 223.381

First hidden state distribution:
0.6667    0.3333

Transition matrix between hidden states (reduced form):
1.0000  0.0000
0.0000  1.0000

Transition matrices between outputs (reduced form):

State 1
0.8261  0.0870  0.0870
0.0769  0.8462  0.0769
0.0400  0.1200  0.8400

State 2
0.0769  0.6154  0.3077
0.4118  0.0588  0.5294
0.3571  0.5714  0.0714

Hidden states:
Sequence 1:
  0    0    1    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1

Sequence 2:
  0    0    2    2    2    2    2    2    2    2    2    2
  2    2    2    2    2    2    2    2    2    2    2    2
  2    2    2    2    2    2    2    2    2    2    2    2
  2    2    2    2    2    2    2    2    2    2    2    2

Sequence 3:
  0    0    1    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1
  1    1    1    1    1    1    1    1    1    1    1    1

```

16. The best model is displayed and the hidden states are represented graphically (see Figure 2). Note that in practice, the figure with the hidden states appears in a separate window.

DCMM menu

1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu

Your choice: 5

Display of results

1. Best model in term of log-likelihood (no hidden states)
2. Best model in term of log-likelihood (with text display of hidden states)
3. Best model in term of log-likelihood (with graphic display of hidden states)
4. Best model in term of log-likelihood (with text and graphic display of hidden states)
5. Best model in term of BIC (no hidden states)
6. Best model in term of BIC (with text display of hidden states)
7. Best model in term of BIC (with graphic display of hidden states)
8. Best model in term of BIC (with text and graphic display of hidden states)
9. Back to DCMM menu

Your choice: 7

* DCMM *

*** Display of the best model in term of BIC ***

Number of independent sequences: 3

Number of possible outputs: 3

Number of hidden states: 2

Order of the hidden chain: 1

Order of the visible chain: 1

Number of independent parameters: 13

Number of components in the log-likelihood: 118

Log-likelihood: -80.681

AIC: 187.362

BIC: 223.381

First hidden state distribution:

0.6667 0.3333

Transition matrix between hidden states (reduced form):

1.0000 0.0000
0.0000 1.0000

Transition matrices between outputs (reduced form):

State 1

0.8261 0.0870 0.0870
0.0769 0.8462 0.0769
0.0400 0.1200 0.8400

State 2

0.0769 0.6154 0.3077
0.4118 0.0588 0.5294
0.3571 0.5714 0.0714

Display of results

1. Best model in term of log-likelihood (no hidden states)
2. Best model in term of log-likelihood (with text display of hidden states)
3. Best model in term of log-likelihood (with graphic display of hidden states)
4. Best model in term of log-likelihood (with text and graphic display of hidden states)
5. Best model in term of BIC (no hidden states)
6. Best model in term of BIC (with text display of hidden states)
7. Best model in term of BIC (with graphic display of hidden states)
8. Best model in term of BIC (with text and graphic display of hidden states)
9. Back to DCMM menu

Your choice: 9

17. The model is saved in a file named *ex1_dcmm_b2.mat*.

DCMM menu

1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu

Your choice: 7

Save

1. Save all
2. Save best model in term of log-likelihood

Figure 2. Graphical display of the hidden states corresponding to the best DCMM computed from the *ex_1a.dat* data file. Each sequence appears on a different row, and each state is associated with a different color. A particular color is also associated with the value 0 corresponding to situations where no hidden state can be computed (here, the first 2 observations of each sequence, and the last 14 values of sequence 3 which were not observed). In this example, state 1 occurs during the totality of sequences 1 and 3, and state 2 occurs during sequence 2.

```

3. Save best model in term of BIC
4. Back to DCMM menu
Your choice: 3

Name of the saving file (without extension): ex1_dcmm_b2

*** Best model successfully saved ***

Save
----
1. Save all
2. Save best model in term of log-likelihood
3. Save best model in term of BIC
4. Back to DCMM menu
Your choice: 4

```

18. A new data file named *ex_1b.dat* is loaded.

DCMM menu

1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu

Your choice: 8

Models menu

1. Independence
2. Homogeneous Markov chain
3. DCMM
4. Back to Main menu

Your choice: 4

Main menu

1. Load data file
2. Maximal order
3. Models
4. Exit

Your choice: 1

Name of the dataset: ex_1b.dat

Number of possible outputs (-1 to quit): 3

*** Data file successfully loaded ***

19. The file *ex1_dcmm_b2.mat* containing the best DCMM model computed from the first data file is loaded.

Main menu

1. Load data file
2. Maximal order
3. Models
4. Exit

Your choice: 3

Models menu

1. Independence
2. Homogeneous Markov chain
3. DCMM

4. Back to Main menu

Your choice: 3

DCMM menu

1. Choice of a model

2. Optimization parameters

3. Type of output

4. Computation

5. Display of results

6. Load

7. Save

8. Back to Model menu

Your choice: 6

Name of the loading file (without extension): ex1_dcmm_b2

*** File successfully loaded ***

20. The number of iterations of the genetic algorithm is set to zero. By doing that, the whole optimization procedure is bypassed and only the log-likelihood and the optimal sequence of hidden states are computed.

DCMM menu

1. Choice of a model

2. Optimization parameters

3. Type of output

4. Computation

5. Display of results

6. Load

7. Save

8. Back to Model menu

Your choice: 2

Optimization parameters

1. Type of initialization (2)

2. Maximal number of iterations (genetic algorithm) (1)

3. Population size (1)

4. Crossover probability (0.5)

5. Mutation probability (0.01)

6. Number of digits per parameter (20)

7. Maximal number of iterations (B-W algorithm) (1)

8. Stop criterion (B-W algorithm) (0)

9. Back to DCMM menu

Your choice: 2

Maximal number of iterations (1) (-1 to quit): 0

Optimization parameters

1. Type of initialization (2)
2. Maximal number of iterations (genetic algorithm) (0)
3. Population size (1)
4. Crossover probability (0.5)
5. Mutation probability (0.01)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (1)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu

Your choice: 9

*** Optimization parameters successfully specified ***

21. The output type is set to *long* to display everything including hidden states.

DCMM menu

1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu

Your choice: 3

Type of output

1. Short (no B-W iterations, no hidden states)
2. Normal (no hidden states)
3. Long
4. Back to DCMM menu

Your choice (2): 3

Type of output

1. Short (no B-W iterations, no hidden states)
2. Normal (no hidden states)
3. Long
4. Back to DCMM menu

Your choice (3): 4

22. The log-likelihood and the optimal sequence of hidden states are computed for the second data file, based on the model computed from the first data file. It appears that the first hidden state is used throughout the sequence. This is a consequence of the fact that the transition matrix between hidden states is here the identity matrix, what implies that the hidden state cannot change during the sequence.

Since state 1 is better for the main part of the sequence, is is chosen, even if state 2 could be better at some point.

DCMM menu

1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu

Your choice: 4

*** Model successfully computed ***

* DCMM *

*** Best model in term of BIC and log-likelihood ***

Number of independent sequences: 1

Number of possible outputs: 3

Number of hidden states: 2

Order of the hidden chain: 1

Order of the visible chain: 1

Number of independent parameters: 13

Number of components in the log-likelihood: 107

Log-likelihood: -155.078

AIC: 336.156

BIC: 370.903

First hidden state distribution:

0.6667 0.3333

Transition matrix between hidden states (reduced form):

1.0000 0.0000

0.0000 1.0000

Transition matrices between outputs (reduced form):

```

State 1
0.8261  0.0870  0.0870
0.0769  0.8462  0.0769
0.0400  0.1200  0.8400

```

```

State 2
0.0769  0.6154  0.3077
0.4118  0.0588  0.5294
0.3571  0.5714  0.0714

```

Hidden states:

Sequence 1:

```

0      0      1      1      1      1      1      1      1      1      1      1
1      1      1      1      1      1      1      1      1      1      1      1
1      1      1      1      1      1      1      1      1      1      1      1
1      1      1      1      1      1      1      1      1      1      1      1
1      1      1      1      1      1      1      1      1      1      1      1
1      1      1      1      1      1      1      1      1      1      1      1
1      1      1      1      1      1      1      1      1      1      1      1
1      1      1      1      1      1      1      1      1      1      1      1
1      1      1      1      1      1      1      1      1      1      1      1
1

```

23. The software is stopped.

DCMM menu

1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu

Your choice: 8

Models menu

1. Independence
2. Homogeneous Markov chain
3. DCMM
4. Back to Main menu

Your choice: 4

Main menu

1. Load data file
2. Maximal order

```

3. Models
4. Exit
Your choice: 4

```

4.2 Practical example

This example concerns the behavior of a young monkey (*macaca nemestrina*)¹. The subject was observed during 17 periods of 5 minutes each. After recoding, each observation represent the main behavior during a 5-seconds period. In this example, there are 3 possible behaviors: Passivity, Exploration, and Fear. Previous experiences (Berchtold, 1999b, Berchtold & Sackett, 2000) showed that a DCMM with 2 hidden states and order 1 relations for both the hidden and the visible Markov chains fits well this type of data.

First of all, the data file (*m46.dat*) is loaded and the maximal order is set to 3.

```

*****
*                MARCH  1.1                *
* Copyright Andre Berchtold, 2000--2001 *
*****

Main menu
-----
1. Load data file
2. Maximal order
3. Models
4. Exit
Your choice: 1

Name of the dataset: m46.dat

Number of possible outputs (-1 to quit): 3

*** Data file successfully loaded ***

Main menu
-----
1. Load data file
2. Maximal order
3. Models
4. Exit
Your choice: 2

Maximal order (0) (-1 to quit): 3

*** Maximal order successfully set to 3 ***

```

Then, the independence model and homogeneous Markov chains of order 1 to 3 are computed.

¹Thanks to Professor Gene Sackett, University of Washington, for providing the data.

Main menu

1. Load data file
2. Maximal order
3. Models
4. Exit

Your choice: 3

Models menu

1. Independence
2. Homogeneous Markov chain
3. DCMM
4. Back to Main menu

Your choice: 1

Independence menu

1. Computation
2. Save
3. Back to Models menu

Your choice: 1

```
*****
* Independence model *
*****
```

Number of independent sequences: 17

Number of possible outputs: 3

Number of independent parameters: 2

Number of components in the log-likelihood: 969

Log-likelihood: -1045.85

AIC: 2095.7

BIC: 2105.45

Distribution of data:

342	388	239
-----	-----	-----

Probability distribution of data:

0.3529	0.4004	0.2466
--------	--------	--------

Independence menu

1. Computation
2. Save

3. Back to Models menu

Your choice: 3

Models menu

1. Independence

2. Homogeneous Markov chain

3. DCMM

4. Back to Main menu

Your choice: 2

Markov chains menu

1. Type of output

2. Computation

3. Save

4. Back to Models menu

Your choice: 2

Order of the Markov chain (-1 to quit): 1

*** Markov chain of order 1 successfully specified ***

* Markov chain *

Number of independent sequences: 17

Number of possible outputs: 3

Order of the model: 1

Number of independent parameters: 6

Number of components in the log-likelihood: 969

Number of parameters estimated to zero: 0

Log-likelihood: -727.504

AIC: 1467.01

BIC: 1496.27

Crosstable (reduced form):

229	82	31
93	277	21
20	29	187

Transition matrix (reduced form):

0.6696	0.2398	0.0906
0.2379	0.7084	0.0537
0.0847	0.1229	0.7924

Markov chains menu

1. Type of output
2. Computation
3. Save
4. Back to Models menu

Your choice: 2

Order of the Markov chain (-1 to quit): 2

*** Markov chain of order 2 successfully specified ***

* Markov chain *

Number of independent sequences: 17

Number of possible outputs: 3

Order of the model: 2

Number of independent parameters: 18

Number of components in the log-likelihood: 969

Number of parameters estimated to zero: 0

Log-likelihood: -693.722

AIC: 1423.44

BIC: 1511.22

Crosstable (reduced form):

175	41	11
49	33	13
5	8	7
33	49	3
52	213	12
8	15	6
4	4	22
4	3	14
12	22	151

Transition matrix (reduced form):

0.7709	0.1806	0.0485
--------	--------	--------

0.5158	0.3474	0.1368
0.2500	0.4000	0.3500
0.3882	0.5765	0.0353
0.1877	0.7690	0.0433
0.2759	0.5172	0.2069
0.1333	0.1333	0.7333
0.1905	0.1429	0.6667
0.0649	0.1189	0.8162

Markov chains menu

1. Type of output
2. Computation
3. Save
4. Back to Models menu

Your choice: 2

Order of the Markov chain (-1 to quit): 3

*** Markov chain of order 3 successfully specified ***

* Markov chain *

Number of independent sequences: 17

Number of possible outputs: 3

Order of the model: 3

Number of independent parameters: 46

Number of components in the log-likelihood: 969

Number of parameters estimated to zero: 8

Log-likelihood: -668.838

AIC: 1429.68

BIC: 1653.98

Crosstable (reduced form):

140	27	6
32	13	4
3	1	1
19	12	3
26	20	7
4	1	3
1	3	0

0	3	1
4	2	6
15	26	1
15	19	1
3	4	1
10	37	1
36	168	9
6	8	2
2	2	0
0	3	0
6	10	6
2	2	6
2	1	10
0	1	6
0	0	3
3	2	7
1	1	4
4	4	13
1	1	12
7	17	126

Transition matrix (reduced form):

0.8092	0.1561	0.0347
0.6531	0.2653	0.0816
0.6000	0.2000	0.2000
0.5588	0.3529	0.0882
0.4906	0.3774	0.1321
0.5000	0.1250	0.3750
0.2500	0.7500	0
0	0.7500	0.2500
0.3333	0.1667	0.5000
0.3571	0.6190	0.0238
0.4286	0.5429	0.0286
0.3750	0.5000	0.1250
0.2083	0.7708	0.0208
0.1690	0.7887	0.0423
0.3750	0.5000	0.1250
0.5000	0.5000	0
0	1.0000	0
0.2727	0.4545	0.2727
0.2000	0.2000	0.6000
0.1538	0.0769	0.7692
0	0.1429	0.8571
0	0	1.0000
0.2500	0.1667	0.5833
0.1667	0.1667	0.6667
0.1905	0.1905	0.6190
0.0714	0.0714	0.8571
0.0467	0.1133	0.8400

Markov chains menu

```

-----
1. Type of output
2. Computation
3. Save
4. Back to Models menu
Your choice: 4

```

The comparison of the BIC values obtained for each model show that the best order of dependence is the first one (BIC=1496.27), the independence model being clearly disqualified. Moreover, we observe that the crosstable corresponding to the third order model contains a lot of very small values, indicating that the number of data points (969) is maybe too small to insure fiable results. On the other hand, this problem does not occur in the first order case.

Since the data exhibit a clear relation of dependence between successive observations, we tried to fit a DCMM with 2 hidden states and first-order matrices for both the hidden and the visible Markov chains. To obtain an accurate estimation, we used a 2-step strategy: First, we used a combination of a Genetic Algorithm and a Baum-Welch algorithm to explore quickly the whole solution space. Then, we improved the solution through a Baum-Welch only procedure.

The first step uses a population of size 100 and 50 iterations of a Genetic Algorithm. The crossover probability is 0.5 and the mutation probability is set to 0.005. Moreover, after each iteration of the GA, each member of the population is improved through one Baum-Welch iteration.

```

Models menu
-----
1. Independence
2. Homogeneous Markov chain
3. DCMM
4. Back to Main menu
Your choice: 3

```

```

DCMM menu
-----
1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu
Your choice: 1

```

```

Choice of a DCMM
-----
1. Number of hidden states (1)
2. Order of the hidden chain (1)
3. Constraint on A (1)
4. Order of the visible chain (1)
5. Back to DCMM menu
Your choice: 1

```

Number of hidden states (1) (-1 to quit): 2

Choice of a DCMM

1. Number of hidden states (2)
2. Order of the hidden chain (1)
3. Constraint on A (1)
4. Order of the visible chain (1)
5. Back to DCMM menu

Your choice: 5

*** DCMM successfully specified ***

DCMM menu

1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu

Your choice: 2

Optimization parameters

1. Type of initialization (1)
2. Maximal number of iterations (genetic algorithm) (1)
3. Population size (1)
4. Crossover probability (0.5)
5. Mutation probability (0.01)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (1)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu

Your choice: 2

Maximal number of iterations (1) (-1 to quit): 50

Optimization parameters

1. Type of initialization (1)
2. Maximal number of iterations (genetic algorithm) (50)
3. Population size (1)
4. Crossover probability (0.5)
5. Mutation probability (0.01)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (1)
8. Stop criterion (B-W algorithm) (0)

9. Back to DCMM menu

Your choice: 3

Population size (1) (-1 to quit): 100

Optimization parameters

1. Type of initialization (1)
2. Maximal number of iterations (genetic algorithm) (50)
3. Population size (100)
4. Crossover probability (0.5)
5. Mutation probability (0.01)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (1)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu

Your choice: 5

Mutation probability (0.01) (-1 to quit): .005

Optimization parameters

1. Type of initialization (1)
2. Maximal number of iterations (genetic algorithm) (50)
3. Population size (100)
4. Crossover probability (0.5)
5. Mutation probability (0.005)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (1)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu

Your choice: 9

*** Optimization parameters successfully specified ***

DCMM menu

1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu

Your choice: 4

Initialization

-909.6055

Iteration 1 (GA)

```
-909.6052
Iteration 1 (B-W)
-721.6868

Iteration 2 (GA)
-721.6867
Iteration 1 (B-W)
-715.3176

Iteration 3 (GA)
-715.3175
Iteration 1 (B-W)
-711.1236

Iteration 4 (GA)
-711.1235
Iteration 1 (B-W)
-707.2004

Iteration 5 (GA)
-707.2003
Iteration 1 (B-W)
-703.8308

Iteration 6 (GA)
-703.8188
Iteration 1 (B-W)
-701.3014

Iteration 7 (GA)
-701.3013
Iteration 1 (B-W)
-699.6043

Iteration 8 (GA)
-699.6035
Iteration 1 (B-W)
-698.4899

[...]

Iteration 49 (GA)
-692.2661
Iteration 1 (B-W)
-692.2255

Iteration 50 (GA)
-692.2254
Iteration 1 (B-W)
-692.1939
```

```

*** Model successfully computed ***

*****
* DCMM *
*****

*** Best model in term of log-likelihood ***

Number of independent sequences: 17

Number of possible outputs: 3

Number of hidden states: 2

Order of the hidden chain: 1

Order of the visible chain: 1

Number of independent parameters: 15

Number of components in the log-likelihood: 969

Log-likelihood: -692.194

AIC: 1414.39

BIC: 1487.53

First hidden state distribution:
    0.6596    0.3404

Transition matrix between hidden states (reduced form):
    0.9044    0.0956
    0.1281    0.8719

Transition matrices between outputs (reduced form):

State 1
    0.8729    0.1271    0.0000
    0.1136    0.8863    0.0001
    0.0071    0.0074    0.9855

State 2
    0.3084    0.4399    0.2516
    0.4296    0.4341    0.1363
    0.1459    0.2137    0.6404

*** Best model in term of BIC ***

Number of independent sequences: 17

```

```

Number of possible outputs: 3

Number of hidden states: 2

Order of the hidden chain: 1

Order of the visible chain: 1

Number of independent parameters: 14

Number of components in the log-likelihood: 969

Log-likelihood: -694.986

AIC: 1417.97

BIC: 1486.24

First hidden state distribution:
    0.7285    0.2715

Transition matrix between hidden states (reduced form):
    0.8493    0.1507
    0.2082    0.7918

Transition matrices between outputs (reduced form):

State 1
    0.8858    0.1142         0
    0.0836    0.9062    0.0102
    0.0010    0.0202    0.9788

State 2
    0.3512    0.4247    0.2241
    0.4685    0.4127    0.1188
    0.1870    0.2482    0.5648

```

In this case, the best models in term of log-likelihood and in term of BIC are slightly different at this point of the optimization process. Since our main criterion is BIC, we continued the analysis with the best model in term of BIC. To replace the whole population by the best model only, we saved this model in a file named *dcmm_a1_bic.mat*, and then we reloaded this file.

```

DCMM menu
-----
1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load

```

```
7. Save
8. Back to Model menu
Your choice: 7
```

Save

```
1. Save all
2. Save best model in term of log-likelihood
3. Save best model in term of BIC
4. Back to DCMM menu
Your choice: 3
```

Name of the saving file (without extension): dcmm_a1_bic

*** Best model successfully saved ***

Save

```
1. Save all
2. Save best model in term of log-likelihood
3. Save best model in term of BIC
4. Back to DCMM menu
Your choice: 4
```

DCMM menu

```
1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu
Your choice: 6
```

Name of the loading file (without extension): dcmm_a1_bic

*** File successfully loaded***

Then, we defined a new optimization procedure using 200 iterations of a Baum-Welch algorithm, starting with the current parameter values. Moreover, we decided to stop the optimization in the case of a difference in log-likelihood between two successive iterations lower than 0.00001.

DCMM menu

```
1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
```


5. Display of results
6. Load
7. Save
8. Back to Model menu

Your choice: 2

Optimization parameters

1. Type of initialization (2)
2. Maximal number of iterations (genetic algorithm) (50)
3. Population size (1)
4. Crossover probability (0.5)
5. Mutation probability (0.005)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (1)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu

Your choice: 2

Maximal number of iterations (50) (-1 to quit): 1

Optimization parameters

1. Type of initialization (2)
2. Maximal number of iterations (genetic algorithm) (1)
3. Population size (1)
4. Crossover probability (0.5)
5. Mutation probability (0.005)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (1)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu

Your choice: 7

Maximal number of iterations (1) (-1 to quit): 200

Optimization parameters

1. Type of initialization (2)
2. Maximal number of iterations (genetic algorithm) (1)
3. Population size (1)
4. Crossover probability (0.5)
5. Mutation probability (0.005)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (200)
8. Stop criterion (B-W algorithm) (0)
9. Back to DCMM menu

Your choice: 8

Stop criterion (0) (0 to inactivate criterion, -1 to quit): .00001

Optimization parameters

1. Type of initialization (2)
2. Maximal number of iterations (genetic algorithm) (1)
3. Population size (1)
4. Crossover probability (0.5)
5. Mutation probability (0.005)
6. Number of digits per parameter (20)
7. Maximal number of iterations (B-W algorithm) (200)
8. Stop criterion (B-W algorithm) (1e-005)
9. Back to DCMM menu

Your choice: 9

*** Optimization parameters successfully specified ***

DCMM menu

1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu

Your choice: 4

Initialization

-694.9864

Iteration 1 (B-W)

-694.1105

Iteration 2 (B-W)

-693.7155

Iteration 3 (B-W)

-693.5127

Iteration 4 (B-W)

-693.3812

Iteration 5 (B-W)

-693.2777

[...]

Iteration 135 (B-W)

-691.8744

Iteration 136 (B-W)

-691.8744

Iteration 137 (B-W)

-691.8744

Iteration 138 (B-W)

-691.8744

Iteration 139 (B-W)

```

-691.8744

*** Model successfully computed ***

*****
* DCMM *
*****

*** Best model in term of BIC and log-likelihood ***

Number of independent sequences: 17

Number of possible outputs: 3

Number of hidden states: 2

Order of the hidden chain: 1

Order of the visible chain: 1

Number of independent parameters: 14

Number of components in the log-likelihood: 969

Log-likelihood: -691.874

AIC: 1411.75

BIC: 1480.02

First hidden state distribution:
    0.6330    0.3670

Transition matrix between hidden states (reduced form):
    0.9240    0.0760
    0.0907    0.9093

Transition matrices between outputs (reduced form):

State 1
    0.8654    0.1346         0
    0.1279    0.8721    0.0000
    0.0142    0.0000    0.9858

State 2
    0.3176    0.4288    0.2535
    0.3915    0.4797    0.1288
    0.1263    0.1952    0.6786

```

The procedure stopped after 139 iterations. In this example, states 1 and 2 define totally different

situations. State 1 corresponds to a situation where a same behavior (Passivity, Exploration, or Fear) is generally repeated several times successively before switching to another behavior (the probabilities on the main diagonal of the corresponding transition matrix are all very high: 0.8654, 0.8721, and 0.9858). On the other hand, state 2 allows more quick changes from one behavior to another (the transition matrix contains only non-zero probabilities, and the probabilities on the main diagonal are smaller than the same probabilities in the state 1 matrix). Finally, the same state can be active during several observations before the other state becomes active (high probabilities on the main diagonal of the hidden transition matrix A), but it is also possible to switch quickly from one state to the another during a 5-minutes period.

The hidden states are then displayed, both in text and graphic modes.

DCMM menu

1. Choice of a model
2. Optimization parameters
3. Type of output
4. Computation
5. Display of results
6. Load
7. Save
8. Back to Model menu

Your choice: 5

Display of results

1. Best model in term of log-likelihood (no hidden states)
2. Best model in term of log-likelihood (with text display of hidden states)
3. Best model in term of log-likelihood (with graphic display of hidden states)
4. Best model in term of log-likelihood (with text and graphic display of hidden states)
5. Best model in term of BIC (no hidden states)
6. Best model in term of BIC (with text display of hidden states)
7. Best model in term of BIC (with graphic display of hidden states)
8. Best model in term of BIC (with text and graphic display of hidden states)
9. Back to DCMM menu

Your choice: 8

* DCMM *

*** Display of the best model in term of BIC ***

Number of independent sequences: 17

Number of possible outputs: 3

Number of hidden states: 2

Order of the hidden chain: 1

Order of the visible chain: 1

Number of independent parameters: 14

Number of components in the log-likelihood: 969

Log-likelihood: -691.874

AIC: 1411.75

BIC: 1480.02

First hidden state distribution:

0.6330 0.3670

Transition matrix between hidden states (reduced form):

0.9240 0.0760
0.0907 0.9093

Transition matrices between outputs (reduced form):

State 1

0.8654 0.1346 0
0.1279 0.8721 0.0000
0.0142 0.0000 0.9858

State 2

0.3176 0.4288 0.2535
0.3915 0.4797 0.1288
0.1263 0.1952 0.6785

Hidden states:

Sequence 1:

Columns 1 through 12

0 0 0 2 2 2 2 2 2 2 2 2

Columns 13 through 24

2 2 2 2 2 2 2 2 2 2 1 1

Columns 25 through 36

1 1 1 1 1 1 1 1 1 1 1 1

Columns 37 through 48

1 1 1 1 1 1 1 1 1 1 1 1

Columns 49 through 60

1 1 1 1 1 1 1 1 1 1 1 1

Sequence 2:

Columns 1 through 12

0 0 0 1 1 1 1 1 1 1 1 1

Columns 13 through 24

1 1 1 1 1 1 1 2 2 2 2 1

Columns 25 through 36

1 1 1 1 1 1 1 1 1 1 1 1

Columns 37 through 48												
1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 49 through 60												
1	1	1	1	1	1	1	1	1	1	1	1	1

Sequence 3:

Columns 1 through 12												
0	0	0	1	1	1	1	1	1	1	1	1	1
Columns 13 through 24												
1	2	2	2	2	2	2	2	2	2	2	2	2
Columns 25 through 36												
2	2	2	2	2	2	2	2	2	2	2	2	2
Columns 37 through 48												
2	2	2	2	2	2	2	2	2	2	2	2	2
Columns 49 through 60												
2	2	2	2	2	2	2	2	1	1	1	1	1

Sequence 4:

Columns 1 through 12												
0	0	0	1	1	1	1	1	1	1	1	1	1
Columns 13 through 24												
1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 25 through 36												
1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 37 through 48												
1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 49 through 60												
1	1	1	2	2	2	2	2	2	2	2	2	2

Sequence 5:

Columns 1 through 12												
0	0	0	1	1	1	1	1	1	1	1	1	1
Columns 13 through 24												
1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 25 through 36												
1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 37 through 48												
1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 49 through 60												
1	1	1	1	1	1	2	2	2	2	2	2	2

Sequence 6:

Columns 1 through 12												
0	0	0	1	1	1	1	1	1	1	1	1	1
Columns 13 through 24												
2	2	2	2	2	2	2	2	2	1	1	1	1
Columns 25 through 36												
1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 37 through 48												
1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 49 through 60												

1	1	2	2	2	2	2	2	2	2	2	2	2
---	---	---	---	---	---	---	---	---	---	---	---	---

Sequence 7:

Columns 1 through 12	0	0	0	2	2	2	2	2	2	2	2	2
Columns 13 through 24	2	2	2	2	2	2	2	2	2	2	2	2
Columns 25 through 36	2	2	2	2	2	2	2	1	1	1	1	1
Columns 37 through 48	1	1	1	1	1	1	1	1	1	1	1	1
Columns 49 through 60	1	1	1	2	2	2	2	2	2	2	2	2

Sequence 8:

Columns 1 through 12	0	0	0	1	1	1	1	1	1	1	1	1
Columns 13 through 24	1	1	1	1	1	1	2	2	2	2	2	2
Columns 25 through 36	2	2	2	2	2	2	2	2	2	2	2	2
Columns 37 through 48	2	2	1	1	1	1	1	1	1	1	1	1
Columns 49 through 60	1	1	1	1	1	1	1	1	1	1	1	1

Sequence 9:

Columns 1 through 12	0	0	0	1	1	1	1	1	1	1	1	1
Columns 13 through 24	1	1	1	1	1	1	1	2	2	2	2	1
Columns 25 through 36	1	1	1	1	1	1	1	1	1	1	1	1
Columns 37 through 48	1	1	1	1	1	1	1	1	1	1	1	1
Columns 49 through 60	1	1	1	1	1	1	1	1	2	2	2	2

Sequence 10:

Columns 1 through 12	0	0	0	1	1	1	2	1	1	1	1	1
Columns 13 through 24	1	1	1	1	1	1	1	1	1	1	1	1
Columns 25 through 36	1	1	1	1	1	1	1	1	1	1	1	1
Columns 37 through 48	1	1	1	1	1	1	1	1	1	1	1	1
Columns 49 through 60	1	1	1	1	1	1	1	1	1	1	1	2

Sequence 11:

Columns 1 through 12												
0	0	0	2	2	2	2	2	2	2	2	2	2
Columns 13 through 24												
2	2	2	2	2	2	2	2	2	2	2	2	1
Columns 25 through 36												
1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 37 through 48												
1	1	2	2	1	1	1	1	1	1	1	1	1
Columns 49 through 60												
1	1	1	1	1	1	1	1	1	1	1	1	1

Sequence 12:

Columns 1 through 12												
0	0	0	2	2	2	2	2	2	2	2	2	2
Columns 13 through 24												
2	2	1	1	1	1	1	1	1	1	1	1	1
Columns 25 through 36												
1	1	1	1	2	2	2	2	2	2	2	2	2
Columns 37 through 48												
2	2	2	2	1	1	1	1	1	1	1	1	1
Columns 49 through 60												
1	1	1	1	1	1	1	1	1	1	1	1	1

Sequence 13:

Columns 1 through 12												
0	0	0	2	2	2	2	2	2	2	2	2	2
Columns 13 through 24												
2	2	2	2	2	2	2	2	2	2	2	2	2
Columns 25 through 36												
2	2	2	2	2	2	2	2	2	2	2	2	2
Columns 37 through 48												
2	2	2	2	2	2	2	2	2	2	2	2	2
Columns 49 through 60												
2	2	2	2	2	2	2	2	2	2	2	2	2

Sequence 14:

Columns 1 through 12												
0	0	0	2	2	2	2	2	2	2	2	2	2
Columns 13 through 24												
2	2	2	2	2	2	2	2	2	2	2	2	2
Columns 25 through 36												
2	2	2	2	2	2	2	2	2	2	2	2	2
Columns 37 through 48												
2	2	2	2	2	2	2	2	2	2	2	2	2
Columns 49 through 60												
2	2	2	2	2	2	2	2	2	2	2	2	2

Sequence 15:

Columns 1 through 12												
0	0	0	1	1	1	1	1	1	1	1	1	1
Columns 13 through 24												

	2	2	1	1	1	1	1	1	1	1	1	1
Columns 25 through 36	1	1	1	2	2	2	2	2	2	2	2	2
Columns 37 through 48	2	2	2	2	2	2	2	2	2	2	2	2
Columns 49 through 60	2	2	2	2	2	2	2	2	2	2	2	2

Sequence 16:

Columns 1 through 12	0	0	0	1	1	1	1	1	1	1	1	1
Columns 13 through 24	1	1	1	1	1	1	1	1	1	1	1	1
Columns 25 through 36	1	1	1	1	2	2	2	2	2	2	2	2
Columns 37 through 48	2	2	2	2	1	1	1	1	1	1	1	1
Columns 49 through 60	1	1	1	1	1	1	1	1	1	1	1	1

Sequence 17:

Columns 1 through 12	0	0	0	2	2	2	2	2	2	2	2	2
Columns 13 through 24	2	2	2	2	2	2	2	2	2	2	2	2
Columns 25 through 36	2	2	2	2	2	2	2	2	2	2	2	2
Columns 37 through 48	2	2	1	1	1	1	1	1	1	1	1	1
Columns 49 through 60	1	1	1	2	2	2	2	2	2	2	2	2

Figure 3 gives a graphic representation of the hidden states. Obviously, the subject stays generally for some time in one of the two main types of behavior represented by states 1 and 2, and he never switches more than 3 times from one state to the another during a same 5-minutes period.

The following final sequence of commands is used to quit the software.

Display of results

```

1. Best model in term of log-likelihood (no hidden states)
2. Best model in term of log-likelihood (with text display of hidden states)
3. Best model in term of log-likelihood (with graphic display of hidden states)
4. Best model in term of log-likelihood (with text and graphic display of hidden states)
5. Best model in term of BIC (no hidden states)
6. Best model in term of BIC (with text display of hidden states)
7. Best model in term of BIC (with graphic display of hidden states)
8. Best model in term of BIC (with text and graphic display of hidden states)
9. Back to DCMM menu
Your choice: 9

```

Figure 3. Graphical display of the hidden states corresponding to the best DCMM computed from the *m46.dat* data file. Each of the 17 5-minutes periods appears on a different row, and each state is associated with a different color. A particular color is also associated with the value 0 corresponding to situations where no hidden state can be computed (here, the first 3 observations of each sequence).

DCMM menu

1. Choice of a model
 2. Optimization parameters
 3. Type of output
 4. Computation
 5. Display of results
 6. Load
 7. Save
 8. Back to Model menu
- Your choice: 8

Models menu

1. Independence
 2. Homogeneous Markov chain
 3. DCMM
 4. Back to Main menu
- Your choice: 4

Main menu

```
-----  
1. Load data file  
2. Maximal order  
3. Models  
4. Exit  
Your choice: 4
```

References

- AKAIKE, H. (1974) A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19, 716-723.
- BERCHTOLD, A. (1998) *Chaînes de Markov et Modèles de Transition : Applications aux Sciences Sociales*. Editions HERMES, Paris.
- BERCHTOLD, A. (1999a) The Double Chain Markov Model. *Communications in Statistics: Theory and Methods*, 28 (11), 2569-2589.
- BERCHTOLD, A. (1999b) High-Order Extensions of the Double Chain Markov Model. Technical Report 356, Department of Statistics, University of Washington. Forthcoming in *Stochastic Models*.
- BERCHTOLD, A., G.P. SACKETT (2000) Sequential Analysis of Developmental Change in Behavior Patterns Using Markov Modeling of Nonstationary Data. Universities of Lausanne and Washington.
- BISHOP, Y.M.M., S.E. FIENBERG, P.W. HOLLAND (1975) *Discrete Multivariate Analysis*. MIT Press, Cambridge.
- BROUARD, T., M. SLIMANE, G. VENTURINI, J.-P. ASSELIN DE BEAUVILLE (1998) *Apprentissage génétique hybride de chaînes de Markov cachées*. Ritschard, Berchtold, Duc & Zighed editors, Editions Hermès, Paris.
- COLEY, D.A. (1999) *An Introduction to Genetic Algorithms for Scientists and Engineers*. World Scientific, Singapore.
- DOOB, J.L. (1990) *Stochastic Processes*. John Wiley & Sons, New York.
- DYNKIN, E.B. (1965) *Markov Processes, Vol. I & II*. Springer-Verlag, Berlin.
- FORNEY, G.D. (1973) The Viterbi Algorithm. *Proceedings of the IEEE*, 61, 268-278.
- HOLLAND, J.H. (1975) *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.
- KATZ, R.W. (1981) On some criteria for estimating the order of a Markov chain. *Technometrics*, 23, 243-249.
- KEMENY, J.G., J.L. SNELL (1976) *Finite Markov Chains*. Springer-Verlag, New York.
- MACDONALD, I.L., W. ZUCCHINI (1997) *Hidden Markov and Other Models for Discrete-valued Time Series*. Chapman & Hall, London.
- MCLACHLAN, G.J., T. KRISHNAN (1997) *The EM Algorithm and Extensions*. Wiley, New York.
- PALIWAL, K.K. (1993) Use of temporal correlation between successive frames in a hidden Markov model based speech recognizer. *Proceedings ICASSP*, Vol. 2, 215-218.
- PEGRAM, G.G.S. (1980) An Autoregressive Model for Multilag Markov Chains. *Journal of Applied Probability*, 17, 350-362.

RABINER, L.R. (1989) A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, Vol. 77, No 2, 257-286.

SCHWARZ, G. (1978) Estimating the dimension of a model. *Annals of Statistics*, 6, 414–464.

A The MARCH distribution

The full Matlab functions distribution of MARCH 1.1 contains the following files:

- `march.m`: Main file (Main menu).
- `march_abic.m`: Computation of AIC and BIC.
- `march_dcmm.m`: DCMM menu.
- `march_dcmm_bp.m`: Backward procedure used by the Baum-Welch algorithm.
- `march_dcmm_bw.m`: Baum-Welch reestimation of parameters.
- `march_dcmm_c.m`: Computing of a DCMM.
- `march_dcmm_com.m`: Choice of the DCMM parameters.
- `march_dcmm_dis.m`: Display of the results of a DCMM.
- `march_dcmm_eps.m`: Computation of *Epsilon* (auxiliary matrix used by the Baum-Welch algorithm).
- `march_dcmm_fp.m`: Forward procedure used by the Baum-Welch algorithm (also used to compute the log-likelihood).
- `march_dcmm_gam.m`: Computation of *Gamma* (auxiliary matrix used by the Baum-Welch algorithm).
- `march_dcmm_i.m`: Random initialization of a DCMM.
- `march_dcmm_op.m`: Choice of the DCMM optimization parameters.
- `march_dcmm_va.m`: Viterbi algorithm.
- `march_global_parameters.m`: Setting of the *MaxOrder* variable.
- `march_indep.m`: Independence menu.
- `march_indep_c.m`: Computation of the independence model.
- `march_load_data.m`: Loading of a data file.
- `march_mc.m`: Markov chains menu.
- `march_mc_c.m`: Computation of a Markov chain.
- `march_models.m`: Models menu.
- `bin2gray.m`: Conversion of a standard binary string to the corresponding Gray encoding.
- `gray2bin.m`: Conversion of a Gray encoding to the corresponding standard binary string.

Details about the variables used by each function can be obtained by typing `help function_name` at the Matlab prompt.

B Variables

Sections B.1 to B.3 describe the variables saved with each type of model. Section B.4 describes other important variables used by the software.

B.1 Independence model

When the *Save* option of the *Independence menu* is used, a Matlab file containing the following variables is created:

- K : Number of possible outputs (or categories) of the variable.
- $Nbparam$: Number of independent parameters.
- $NbLLdata$: Number of components in the log-likelihood.
- LL : Log-likelihood.
- AIC : Akaike Information Criterion.
- BIC : Bayesian Information Criterion.
- Ind_c : Frequency distribution of the observations ($1 \times K$).
- Ind_p : Probability distribution of the observations ($1 \times K$).

B.2 Markov chain

When the *Save* option of the *Markov chains menu* is used, a Matlab file containing the following variables is created:

- K : Number of possible outputs (or categories) of the variable.
- $Order$: Order of the Markov chain.
- $Nbparam$: Number of independent parameters.
- $NbLLdata$: Number of components in the log-likelihood.
- $Nbzeros$: Number of parameters estimated to zero.
- LL : Log-likelihood.
- AIC : Akaike Information Criterion.
- BIC : Bayesian Information Criterion.
- RT : Reduced form of the crosstable corresponding to the transition matrix ($K^{Order} \times K$).
- R : Reduced form of the transition matrix ($K^{Order} \times K$).
- C : Transition matrix ($K^{Order} \times K^{Order}$).

B.3 DCMM

When the *Save* option of the *DCMM menu* is used, a Matlab file containing the following variables is created. Depending on the option chosen, the file contains either the whole population or only one member of it. In the latter case, *PopSize* is equal to 1.

- *K*: Number of possible outputs (or categories) of the variable.
- *M*: Number of hidden states.
- *PopSize*: Size of the population used by the genetic algorithm.
- *OrderHC*: Order of the hidden chain.
- *OrderVC*: Order of the visible chain.
- *Nbparam*: Number of independent parameters.
- *NbLLdata*: Number of components in the log-likelihood.
- *LL*: Log-likelihood.
- *AIC*: Akaike Information Criterion.
- *BIC*: Bayesian Information Criterion.
- *pi*: Distribution of the first *OrderHC* hidden states ($M^{OrderHC} \times M \times OrderHC \times PopSize$).
- *A*: Transition matrix between hidden states ($M^{OrderHC} \times M^{OrderHC} \times PopSize$).
- *RA*: Reduced form of the transition matrix between hidden states ($M^{OrderHC} \times M \times PopSize$).
- *AConst*: Constraint on *A* (1: no constraint, 2: *A* “diagonal”).
- *C*: *M* transition matrices between successive outputs of the visible variable ($K^{OrderVC} \times K^{OrderVC} \times M \times PopSize$).
- *RC*: Reduced form of the *M* transition matrices between successive outputs of the visible variable ($K^{OrderVC} \times K \times M \times PopSize$).
- *X*: Matrix of hidden states sequences ($N \times \max(T) \times PopSize$).

B.4 Other variables

Other important variables used by MARCH include:

- *Y*: Matrix of observed data ($N \times \max(T)$). Each row is an independent sequence.
- *N*: Number of independent sequences of data.
- *T*: Vector giving the length of each independent sequence of data, not including missing values (1 x *N*).

- *MaxOrder*: Maximal order of a model.
- *Dbatch*: 0 when MARCH runs in interactive mode, 1 when MARCH runs in batch mode.
- *InitPar*: Type of parameters initialization (1: random initialization, 2: current values).
- *IterGA*: Maximal number of iterations of the genetic algorithm.
- *PCross*: Probability of crossover.
- *PMut*: Probability of mutation.
- *Nod*: Number of digits used to code each parameter in binary form.
- *IterBW*: Maximal number of iterations of the Baum-Welch algorithm.
- *StopBW*: Stop criterion for the Baum-Welch algorithm.
- *Epsilon*: Auxiliary matrix used by the Baum-Welch algorithm ($M^{OrderHC} \times M \times \max(T) - 1 \times N$). Joint probability of $OrderHC + 1$ successive hidden states.
- *Gamma*: Auxiliary matrix used by the Baum-Welch algorithm ($M^{OrderHC} \times M \times \max(T) \times N$). Joint probability of $OrderHC$ successive hidden states.