Exploring the state sequence space for hidden Markov and semi-Markov chains

Yann Guédon

CIRAD, UMR AMAP and INRIA, Virtual Plants,
TA 40/PS2, 34398 Montpellier Cedex 5, France

Abstract

The knowledge of the state sequences that explain a given observed sequence for a known hidden Markovian model is the basis of various methods that may be divided into three categories: (i) enumeration of state sequences, (ii) summary of the possible state sequences in state profiles, (iii) computation of a global measure of the state sequence uncertainty. Concerning the first category, the generalized Viterbi algorithm for computing the top L most probable state sequences and the forward-backward algorithm for sampling state sequences are derived for hidden semi-Markov chains and hidden hybrid models combining Markovian and semi-Markovian states. Concerning the second category, a new type of state (and state change) profiles is proposed. The Viterbi forward-backward algorithm for computing these state profiles is derived for hidden semi-Markov chains and hidden hybrid models combining Markovian and semi-Markovian states. Concerning the third category, an algorithm for computing the entropy of the state sequence that explains an observed sequence is proposed. The complementarity and properties of these methods for exploring the state sequence space (including the classical state profiles computed by the forward-backward algorithm) are investigated and illustrated with examples.

Key words: Entropy; Generalized Viterbi algorithm; Forward-backward algorithm for sampling; Hidden Markov chain; Hidden semi-Markov chain; Plant structure analysis; Viterbi forward-backward algorithm.

1 Introduction

Our focus will be on hidden semi-Markov chains and hidden hybrid models combining Markovian and semi-Markovian states; see Guédon (2005) for this latter family of models. These statistical models generalize hidden Markov chains (see Ephraim

[1] E-mail address: guedon@cirad.fr

and Merhav (2002) for a tutorial about hidden Markovian models) and are particularly useful for analyzing homogeneous zones within sequences or detecting change points between zones. Hidden semi-Markov chains were first introduced in the context of speech recognition by Ferguson (1980) and have since been applied in such different contexts as for instance gene finding (Burge and Karlin, 1997; Lukashin and Borodovsky, 1998), protein secondary structure prediction (Schmidler *et al.*, 2000), the analysis of branching and flowering patterns in plants (Guédon *et al.*, 2001) and the analysis of rainfall data (Sansom and Thomson, 2001).

Once a hidden Markovian model has been estimated, it is generally of interest to understand the hidden state sequence structure underlying each observed sequence. The knowledge of solely the most probable state sequence computed by the Viterbi algorithm tells us nothing about the remainder of the state sequence space while the state profiles computed by the forward-backward algorithm do not highlight structural differences between possible state sequences. Questions of interest are (Foreman, 1993):

- Is the most probable state sequence most probable by a long way or are there other state sequences with near-optimal probability?
- Do these near-optimal sequences have state structures very similar to the most probable state sequence or do they differ greatly?

The knowledge of state sequences that explain a given observed sequence for a known hidden Markovian model is the objective of various methods that may be divided into three categories:

- enumeration of state sequences,
- state profiles i.e. state sequences summarized in a $J \times \tau$ array where J is the number of states and τ the length of the sequence,
- computation of a global measure of the state sequence uncertainty.

The first category includes two main methods, a deterministic method and a randomized method. In the case of hidden Markov chains, the top L most probable state sequences can be computed by the generalized Viterbi algorithm proposed by Foreman (1993) while state sequences can be simulated for a given observed sequence (Chib, 1996). This latter method was initially proposed as a building block of Bayesian estimation methods but has also been used for exploring the state sequence space (Cawley and Pachter, 2003). One outcome of this paper will be to derive a generalized Viterbi algorithm and an algorithm for sampling state sequences for hidden semi-Markov chains and hidden hybrid models combining Markovian and semi-Markovian states.

State profiles and state change profiles were first proposed as a way to explore the state sequence space by Churchill (1989). These state profiles are smoothed probabilities, computed by the forward-backward algorithm, as a function of the index parameter. In these state profiles, individual state sequences and in particular the most probable state sequence are not apparent. One of the main outcomes of

this paper is to propose a new type of state and state change profiles computed by the Viterbi forward-backward algorithm. This algorithm was initially proposed by Brushe *et al.* (1998) for estimating hidden Markov chains. In this paper, the Viterbi forward-backward algorithm is derived for hidden semi-Markov chains and hidden hybrid models combining Markovian and semi-Markovian states. We will show that these new state and state change profiles give two complementary synthetic points of view on the structural differences between state sequences and are thereby directly related to the outputs of the generalized Viterbi algorithm.

The entropy of the state sequence that explains an observed sequence for a known hidden Markov chain was proposed as a global measure of the state sequence uncertainty by Hernando *et al.* (2005). They derived an algorithm which can be transposed to hidden semi-Markov chains and hidden hybrid models. We here propose an alternative approach inspired by the statement of the EM algorithm for hidden semi-Markov chains (Guédon, 2003).

The remainder of this paper is organized as follows. Hidden hybrid Markov/semi-Markov chains are formally defined in Section 2. The Viterbi forward-backward algorithm is presented in Section 3 while the generalized Viterbi algorithm is presented in Section 4 and the forward-backward algorithm for sampling state sequences is presented in Section 5. An algorithm for computing the entropy of the state sequence that explains an observed sequence is derived in Section 6. All these algorithms are illustrated in Section 7 using two examples. Section 8 consists of concluding remarks.

2 Hidden hybrid Markov/semi-Markov chain definition

Let $\{S_t\}$ be a hybrid Markov/semi-Markov chain with finite state space $\{0, \ldots, J-1\}$; see Kulkarni (1995) for a general reference about Markov and semi-Markov models. This J-state hybrid Markov/semi-Markov chain is defined by the following parameters:

- initial probabilities $\pi_j = P(S_0 = j)$ with $\sum_j \pi_j = 1$,
- transition probabilities
  - semi-Markovian state j: for each $k \neq j, p_{jk} = P(S_{t+1} = k | S_{t+1} \neq j, S_t = j)$ with $\sum_{k \neq j} p_{jk} = 1$ and $p_{jj} = 0$,
  - Markovian state j: $\widetilde{p}_{jk} = P(S_{t+1} = k | S_t = j)$ with $\sum_k \widetilde{p}_{jk} = 1$.

It should be noted that absorbing states are Markovian by definition.

An explicit occupancy (or sojourn time) distribution is attached to each semi-Markovian state

$$d_j(u) = P(S_{t+u+1} \neq j, S_{t+u-v} = j, v = 0, \ldots, u-2 | S_{t+1} = j, S_t \neq j), \quad u = 1, \ldots, M_j,$$

where M_j denotes the upper bound to the time spent in state j. Hence, we assume that the state occupancy distributions are concentrated on finite sets of time points. For the particular case of the last visited state, we need to introduce the survivor function of the sojourn time in state j, $D_j(u) = \sum_{v \geq u} d_j(v)$.

If the process starts out at $t = 0$ in a given semi-Markovian state j, the following relation is verified

$$P(S_t \neq j, S_{t-v} = j, v = 1, \ldots, t) = d_j(t)\, \pi_j. \tag{1}$$

Relation (1) means that the process enters a 'new' state at time 0.

The implicit occupancy distribution of a nonabsorbing Markovian state j is the '1-shifted' geometric distribution with parameter $1 - \widetilde{p}_{jj}$

$$d_j(u) = (1 - \widetilde{p}_{jj})\, \widetilde{p}_{jj}^{u-1}, \qquad u = 1, 2, \ldots$$

Hybrid Markov/semi-Markov chains can be viewed as a sub-class of semi-Markov chains where the occupancy distributions of some nonabsorbing states are constrained to be geometric distributions.

A hidden hybrid Markov/semi-Markov chain can be viewed as a pair of stochastic processes $\{S_t, X_t\}$ where the output process $\{X_t\}$ is related to the state process $\{S_t\}$, which is a finite-state hybrid Markov/semi-Markov chain, by a probabilistic function or mapping denoted by f (hence $X_t = f(S_t)$). Since the mapping f is such that $f(j) = f(k)$ may be satisfied for some different j, k, that is a given output may be observed in different states, the state process $\{S_t\}$ is not observable directly but only indirectly through the output process $\{X_t\}$. To simplify the algorithm presentation, we consider a single discrete output process. This output process $\{X_t\}$ is related to the hybrid Markov/semi-Markov chain $\{S_t\}$ by the observation (or emission) probabilities

$$b_j(y) = P(X_t = y | S_t = j) \text{ with } \sum_y b_j(y) = 1.$$

The definition of the observation probabilities expresses the assumption that the output process at time t depends only on the underlying hybrid Markov/semi-Markov chain at time t. Note that X_t is considered univariate for convenience: the extension to the multivariate case is straightforward since, in this latter case, the elementary observed variables at time t are assumed to be conditionally independent given the state $S_t = s_t$.

In the sequel, $X_0^{\tau-1} = x_0^{\tau-1}$ is a shorthand for $X_0 = x_0, \ldots, X_{\tau-1} = x_{\tau-1}$ (this convention transposes to the state sequence $S_0^{\tau-1} = s_0^{\tau-1}$).

3 Viterbi forward-backward algorithm

Individual state sequences are not apparent in the usual state profiles computed by the forward-backward algorithm. The distinctive property of the state profiles computed by the Viterbi forward-backward algorithm is to select segments of individual state sequences and in particular the entire most probable state sequence. In this way, structural differences between sub-optimal state sequences and the most probable state sequence are highlighted.

For a semi-Markovian state j, the Viterbi forward-backward algorithm is based on the following decomposition of the probability of the observed sequence $x_0^{\tau-1}$ jointly with the most probable state sequence leaving state j at time t

$$\begin{aligned}
&\max_{s_0,\ldots,s_{t-1}} \max_{s_{t+1},\ldots,s_{\tau-1}} P\left(S_0^{t-1}=s_0^{t-1}, S_t=j, S_{t+1}\neq j, S_{t+1}^{\tau-1}=s_{t+1}^{\tau-1}, X_0^{\tau-1}=x_0^{\tau-1}\right)\\
&=\max_{s_{t+1},\ldots,s_{\tau-1}} P\left(X_{t+1}^{\tau-1}=x_{t+1}^{\tau-1}, S_{t+1}^{\tau-1}=s_{t+1}^{\tau-1}|S_{t+1}\neq j, S_t=j\right)\\
&\quad\times\max_{s_0,\ldots,s_{t-1}} P\left(S_{t+1}\neq j, S_t=j, S_0^{t-1}=s_0^{t-1}, X_0^t=x_0^t\right)\\
&=\beta_j(t)\,\alpha_j(t). \qquad (2)
\end{aligned}$$

For a Markovian state j, the Viterbi forward-backward algorithm is based on the following decomposition of the probability of the observed sequence $x_0^{\tau-1}$ jointly with the most probable state sequence being in state j at time t

$$\begin{aligned}
\gamma_j(t)&=\max_{s_0,\ldots,s_{t-1}} \max_{s_{t+1},\ldots,s_{\tau-1}} P\left(S_0^{t-1}=s_0^{t-1}, S_t=j, S_{t+1}^{\tau-1}=s_{t+1}^{\tau-1}, X_0^{\tau-1}=x_0^{\tau-1}\right)\\
&=\max_{s_{t+1},\ldots,s_{\tau-1}} P\left(X_{t+1}^{\tau-1}=x_{t+1}^{\tau-1}, S_{t+1}^{\tau-1}=s_{t+1}^{\tau-1}|S_t=j\right)\\
&\quad\times\max_{s_0,\ldots,s_{t-1}} P\left(S_t=j, S_0^{t-1}=s_0^{t-1}, X_0^t=x_0^t\right)\\
&=\widetilde{\beta}_j(t)\,\widetilde{\alpha}_j(t). \qquad (3)
\end{aligned}$$

The objective of the Viterbi forward-backward algorithm is to compute $\gamma_t(j)$ whatever the state type and this induces some supplementary difficulties for semi-Markovian states (for the same reasons, the computation of $P\left(S_t=j|\ X_0^{\tau-1}=x_0^{\tau-1}\right)$ for a semi-Markovian state j is less intuitive than the computation of $P\left(S_{t+1}\neq j,\ S_t=j|\ X_0^{\tau-1}=x_0^{\tau-1}\right)$ in the forward-backward algorithm; see Guédon (2003)). In the following, we will show that it is possible to design a backward recursion for semi-Markovian states whose complexities both in time and in space are similar to those of the forward recursion, that is $O\left(J\tau\left(J+\tau\right)\right)$-time in the worst case and $O\left(J\tau\right)$-space. This means that the computation of $\gamma_j(t)=\max_{s_0,\ldots,s_{t-1}}\max_{s_{t+1},\ldots,s_{\tau-1}} P\left(S_0^{t-1}=s_0^{t-1},\ S_t=j, S_{t+1}^{\tau-1}=s_{t+1}^{\tau-1}, X_0^{\tau-1}=x_0^{\tau-1}\right)$ instead of $\max_{s_0,\ldots,s_{t-1}}\max_{s_{t+1},\ldots,s_{\tau-1}} P\left(S_0^{t-1}=s_0^{t-1},\ S_t=j, S_{t+1}\neq j, S_{t+1}^{\tau-1}=s_{t+1}^{\tau-1}, X_0^{\tau-1}=x_0^{\tau-1}\right)$ for a

semi-Markovian state j does not entail a change in the order of magnitude of algorithm complexity.

For a semi-Markovian state j, the forward recursion is given by (Guédon, 2003, 2005),

$t = 0, \ldots, \tau - 2$:

$$\begin{aligned}\alpha_j(t) &= \max_{s_0,\ldots,s_{t-1}} P\left(S_{t+1} \neq j, S_t = j, S_0^{t-1} = s_0^{t-1}, X_0^t = x_0^t\right) \\ &= b_j(x_t) \max\left[\max_{1\leq u\leq t}\left[\left\{\prod_{v=1}^{u-1} b_j(x_{t-v})\right\} d_j(u) \max_{i\neq j}\{p_{ij}\alpha_i(t-u)\}\right],\right. \\ &\quad \left.\left\{\prod_{v=1}^{t} b_j(x_{t-v})\right\} d_j(t+1)\,\pi_j\right]. \qquad (4)\end{aligned}$$

The censoring at time $\tau - 1$ of the sojourn time in the last visited state distinguishes the case $t = \tau - 1$

$$\begin{aligned}&\alpha_j(\tau-1) \\ &= \max_{s_0,\ldots,s_{\tau-2}} P\left(S_{\tau-1} = j, S_0^{\tau-2} = s_0^{\tau-2}, X_0^{\tau-1} = x_0^{\tau-1}\right) \\ &= b_j(x_{\tau-1}) \max\left[\max_{1\leq u\leq \tau-1}\left[\left\{\prod_{v=1}^{u-1} b_j(x_{\tau-1-v})\right\} D_j(u) \max_{i\neq j}\{p_{ij}\alpha_i(\tau-1-u)\}\right],\right. \\ &\quad \left.\left\{\prod_{v=1}^{\tau-1} b_j(x_{\tau-1-v})\right\} D_j(\tau)\,\pi_j\right]. \qquad (5)\end{aligned}$$

For a Markovian state j, the forward recursion initialized for $t = 0$ by

$$\begin{aligned}\widetilde{\alpha}_j(0) &= P(S_0 = j, X_0 = x_0) \\ &= b_j(x_0)\,\pi_j,\end{aligned}$$

is given by,

$t = 1, \ldots, \tau - 1$:

$$\begin{aligned}\widetilde{\alpha}_j(t) &= \max_{s_0,\ldots,s_{t-1}} P\left(S_t = j, S_0^{t-1} = s_0^{t-1}, X_0^t = x_0^t\right) \\ &= b_j(x_t) \max_i\{\widetilde{p}_{ij}\widetilde{\alpha}_i(t-1)\}. \qquad (6)\end{aligned}$$

The probability of the observed sequence $x_0^{\tau-1}$ jointly with the most probable state sequence is $\max_j\{\alpha_j(\tau-1)\}$ (it should be noted that $\alpha_j(\tau-1) = \widetilde{\alpha}_j(\tau-1) =$

$\max_{s_0,\ldots,s_{\tau-2}} P\left(S_{\tau-1}=j,\ S_0^{\tau-2}=s_0^{\tau-2}, X_0^{\tau-1}=x_0^{\tau-1}\right)\big)$. The quantity $p_{ij}\alpha_i(t-u)$ in (4) and (5) should be replaced by $\widetilde{p}_{ij}\widetilde{\alpha}_i(t-u)$ if state i is Markovian and, conversely, the quantity $\widetilde{p}_{ij}\widetilde{\alpha}_i(t-1)$ in (6) should be replaced by $p_{ij}\alpha_i(t-1)$ if state i is semi-Markovian; see Guédon (2005). The resulting forward recursion computes in parallel $\alpha_j(t) = \max_{s_0,\ldots,s_{t-1}} P\left(S_{t+1}\neq j,\ S_t=j, S_0^{t-1}=s_0^{t-1}, X_0^t=x_0^t\right)$ for semi-Markovian states and $\widetilde{\alpha}_j(t) = \max_{s_0,\ldots,s_{t-1}} P\left(S_t=j,\ S_0^{t-1}=s_0^{t-1}, X_0^t=x_0^t\right)$ for Markovian states.

On the basis of decomposition (2) for semi-Markovian states and decomposition (3) for Markovian states, we can build the following backward recursion.

The backward recursion is initialized for $t=\tau-1$ by

$$\beta_j(\tau-1)=\widetilde{\beta}_j(\tau-1)=1,$$

where state j may be indifferently Markovian or semi-Markovian.

Hence,

$$\gamma_j(\tau-1)=\alpha_j(\tau-1)=\widetilde{\alpha}_j(\tau-1).$$

For a semi-Markovian state j, the backward recursion is given by,

$t=\tau-2,\ldots,0:$

$$\begin{aligned}
&\beta_j(t)\\
&=\max_{s_{t+1},\ldots,s_{\tau-1}} P\left(X_{t+1}^{\tau-1}=x_{t+1}^{\tau-1}, S_{t+1}^{\tau-1}=s_{t+1}^{\tau-1}|S_{t+1}\neq j, S_t=j\right)\\
&=\max_{k\neq j}\left[\max\left[\max_{1\leq u\leq\tau-2-t}\left\{\max_{s_{t+u+1},\ldots,s_{\tau-1}} P\left(X_{t+u+1}^{\tau-1}=x_{t+u+1}^{\tau-1}, S_{t+u+1}^{\tau-1}=s_{t+u+1}^{\tau-1}|\right.\right.\right.\right.\\
&\quad S_{t+u+1}\neq k, S_{t+u}=k\big) P\left(X_{t+1}^{t+u}=x_{t+1}^{t+u}|S_{t+u-v}=k, v=0,\ldots,u-1\right)\\
&\quad\times P\left(S_{t+u+1}\neq k, S_{t+u-v}=k, v=0,\ldots,u-2|S_{t+1}=k, S_t\neq k\right)\Big\},\\
&\quad P\left(X_{t+1}^{\tau-1}=x_{t+1}^{\tau-1}|S_{\tau-1-v}=k, v=0,\ldots,\tau-2-t\right)\\
&\quad\times P\left(S_{\tau-1-v}=k, v=0,\ldots,\tau-3-t|S_{t+1}=k, S_t\neq k\right)]\\
&\quad\left.\times P\left(S_{t+1}=k|S_{t+1}\neq j, S_t=j\right)\right]\\
&=\max_{k\neq j}\left[\max\left[\max_{1\leq u\leq\tau-2-t}\left[\beta_k(t+u)\left\{\prod_{v=0}^{u-1} b_k(x_{t+u-v})\right\} d_k(u)\right],\right.\right.\\
&\quad\left.\left.\left\{\prod_{v=0}^{\tau-2-t} b_k(x_{\tau-1-v})\right\} D_k(\tau-1-t)\right] p_{jk}\right]\\
&=\max_{k\neq j}\left\{\xi_k(t+1)\, p_{jk}\right\},
\end{aligned}\tag{7}$$

where

$$\xi_k(t+1) = \max_{s_{t+2},\ldots,s_{\tau-1}} P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1}, S_{t+2}^{\tau-1} = s_{t+2}^{\tau-1} | S_{t+1} = k, S_t \neq k\right).$$

For a Markovian state j, the backward recursion is given by (Brushe *et al.*, 1998), $t = \tau - 2, \ldots, 0$:

$$\begin{aligned}
\widetilde{\beta}_j(t) &= \max_{s_{t+1},\ldots,s_{\tau-1}} P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1}, S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1} | S_t = j\right) \\
&= \max_k \left\{ \max_{s_{t+2},\ldots,s_{\tau-1}} P\left(X_{t+2}^{\tau-1} = x_{t+2}^{\tau-1}, S_{t+2}^{\tau-1} = s_{t+2}^{\tau-1} | S_{t+1} = k\right) \right. \\
&\quad \left. \times P\left(X_{t+1} = x_{t+1} | S_{t+1} = k\right) P\left(S_{t+1} = k | S_t = j\right) \right\} \\
&= \max_k \left\{ \widetilde{\beta}_k(t+1)\, b_k(x_{t+1})\, \widetilde{p}_{jk} \right\} \\
&= \max_k \left\{ \widetilde{\xi}_k(t+1)\, \widetilde{p}_{jk} \right\}, \qquad (8)
\end{aligned}$$

where

$$\widetilde{\xi}_k(t+1) = \max_{s_{t+2},\ldots,s_{\tau-1}} P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1}, S_{t+2}^{\tau-1} = s_{t+2}^{\tau-1} | S_{t+1} = k\right).$$

Hence, $\xi_k(t+1)$ in (7) should be replaced by $\widetilde{\xi}_k(t+1)$ if state k is Markovian and, conversely, $\widetilde{\xi}_k(t+1)$ in (8) should be replaced by $\xi_k(t+1)$ if state k is semi-Markovian. The resulting backward recursion computes in parallel $\beta_j(t) = \max_{s_{t+1},\ldots,s_{\tau-1}} P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1},\ S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1} | S_{t+1} \neq j, S_t = j\right)$ for semi-Markovian states and $\widetilde{\beta}_j(t) = \max_{s_{t+1},\ldots,s_{\tau-1}} P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1},\ S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1} |\ S_t = j\right)$ for Markovian states.

For a semi-Markovian state j, the computation of $\gamma_j(t)$ requires the following additional maximization step. For each u (taken in decreasing order), the quantity

$$\begin{aligned}
&\max_{v \geq u} \max_{s_0,\ldots,s_t} \max_{s_{t+v+1},\ldots,s_{\tau-1}} P\left(S_0^t = s_0^t, S_t \neq j, S_{t+w} = j, w = 1, \ldots, v, S_{t+v+1} \neq j,\right. \\
&\left. S_{t+v+1}^{\tau-1} = s_{t+v+1}^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right)
\end{aligned}$$

should be compared with the current evaluation of

$$\begin{aligned}
\gamma_j(t+u) = \max_{s_0,\ldots,s_{t+u-1}} \max_{s_{t+u+1},\ldots,s_{\tau-1}} P\left(S_0^{t+u-1} = s_0^{t+u-1}, S_{t+u} = j, S_{t+u+1}^{\tau-1} = s_{t+u+1}^{\tau-1},\right. \\
\left. X_0^{\tau-1} = x_0^{\tau-1}\right),
\end{aligned}$$

where

$$
\begin{aligned}
&\max_{s_0,\ldots,s_t} \max_{s_{t+v+1},\ldots,s_{\tau-1}} P\left(S_0^t = s_0^t, S_t \neq j, S_{t+w} = j, w = 1,\ldots,v, S_{t+v+1} \neq j,\right.\\
&\left. S_{t+v+1}^{\tau-1} = s_{t+v+1}^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right)\\
=&\max_{s_{t+v+1},\ldots,s_{\tau-1}} P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1}, S_{t+v+1}^{\tau-1} = s_{t+v+1}^{\tau-1}, S_{t+v+1} \neq j,\right.\\
&\left. S_{t+v-w} = j, w = 0,\ldots,v-2 | S_{t+1} = j, S_t \neq j\right)\\
&\times \max_{s_0,\ldots,s_t} P\left(S_{t+1} = j, S_t \neq j, S_0^t = s_0^t, X_0^t = x_0^t\right). \qquad (9)
\end{aligned}
$$

The first term in (9) is directly extracted from the computation of $\xi_j(t+1)$ while the second term $\max_{s_0,\ldots,s_t} P(S_{t+1} = j,\ S_t \neq j, S_0^t = s_0^t, X_0^t = x_0^t) = \max_{i \neq j}\{p_{ij}\alpha_i(t)\}$ is computed and stored during the forward recursion; see Appendix A. The computation of $\xi_j(t+1)$ decomposes in two passes:

- a first pass for each u taken in increasing order justified by the recursive computation of $\prod_{v=0}^{u-1} b_j(x_{t+u-v})$,
- a second pass for each u taken in decreasing order for the two-step maximization described above.

In this manner, the mandatory maximization required for computing $\beta_j(t)$ is re-used in the computation of $\left(\gamma_j(t+u)\ ; u = 1,\ldots,\tau-1-t\right)$ with only a single supplementary maximization for each couple (t,u). For a semi-Markovian state j, $\gamma_j(t)$ is initialized at time t with $\beta_j(t)\,\alpha_j(t)$.

Hence, in the computation of

$$
\begin{aligned}
\gamma_j(t) =& \max_{s_0,\ldots,s_{t-1}} \max_{s_{t+1},\ldots,s_{\tau-1}} P\left(S_0^{t-1} = s_0^{t-1}, S_t = j, S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right)\\
=& \max_{u,v} \max_{s_0,\ldots,s_{t-u}} \max_{s_{t+v+1},\ldots,s_{\tau-1}} P\left(S_0^{t-u} = s_0^{t-u}, S_{t-u} \neq j,\right.\\
&\left. S_{t-u+w} = j, w = 1,\ldots,u+v, S_{t+v+1} \neq j, S_{t+v+1}^{\tau-1} = s_{t+v+1}^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right),
\end{aligned}
$$

the maximizations on v for each u are byproducts of the computation of $\left\{\beta_j(t-u)\,;\ u = 0,\ldots,t\right\}$ while the maximization on u requires a supplementary maximization for each u.

The computation of $\gamma_j(t)$ for semi-Markovian states requires some supplementary computations at time $t = 0$. It is thus necessary to compute

$$
\begin{aligned}
&\max_{s_1,\ldots,s_{\tau-1}} P\left(S_0=j,S_1^{\tau-1}=s_1^{\tau-1},X_0^{\tau-1}=x_0^{\tau-1}\right)\\
=&\max\left[\max_{1\leq u\leq\tau-1}\left[\beta_j\left(u-1\right)\left\{\prod_{v=1}^{u}b_j\left(x_{u-v}\right)\right\}d_j\left(u\right)\right],\right.\\
&\left.\left\{\prod_{v=1}^{\tau}b_j\left(x_{\tau-v}\right)\right\}D_j\left(\tau\right)\right]\pi_j\\
=&\,\xi_j\left(0\right)\pi_j.
\end{aligned}
$$

For each u (taken in decreasing order), the quantity

$$
\max_{v\geq u}\max_{s_v,\ldots,s_{\tau-1}} P\left(S_w=j,w=0,\ldots,v-1,S_v\neq j,S_v^{\tau-1}=s_v^{\tau-1},X_0^{\tau-1}=x_0^{\tau-1}\right)
$$

should then be compared with the current evaluation of

$$
\gamma_j\left(u-1\right)=\max_{s_0,\ldots,s_{u-2}}\max_{s_u,\ldots,s_{\tau-1}} P\left(S_0^{u-2}=s_0^{u-2},S_{u-1}=j,S_u^{\tau-1}=s_u^{\tau-1},X_0^{\tau-1}=x_0^{\tau-1}\right).
$$

It should be noted that the reestimation of state occupancy distributions (M-step of the EM algorithm) requires similar supplementary computations at time $t=0$; see Guédon (2003). In the case of a hidden semi-Markov chain, the complexity of this algorithm is $O\left(J\tau\left(J+\tau\right)\right)$-time in the worst case and $O\left(J\tau\right)$-space. An implementation of this algorithm is proposed in Appendix A in pseudo-code form where common computations between semi-Markovian states and Markovian states are highlighted.

In practice, the posterior probabilities should preferably be used

$$
\begin{aligned}
\overline{\gamma}_j\left(t\right)=&\max_{s_0,\ldots,s_{t-1}}\max_{s_{t+1},\ldots,s_{\tau-1}} P\left(S_0^{t-1}=s_0^{t-1},S_t=j,S_{t+1}^{\tau-1}=s_{t+1}^{\tau-1}|X_0^{\tau-1}=x_0^{\tau-1}\right)\\
=&\,\gamma_j\left(t\right)/P\left(X_0^{\tau-1}=x_0^{\tau-1}\right).
\end{aligned}
$$

where the normalizing constant $P\left(X_0^{\tau-1}=x_0^{\tau-1}\right)$ is computed by the forward recursion of the forward-backward algorithm presented in Guédon (2005); see also Section 5. Hence, the selected segments of state sequences are represented at the level of their posterior probability $\overline{\gamma}_j\left(t\right)$ in the state profiles; see the examples in Section 7.

We are now able to discuss the respective properties of the state profiles computed by the Viterbi forward-backward algorithm and the usual state profiles computed by the forward-backward algorithm. Since

$$
\begin{aligned}
L_j(t) &= P\left(S_t = j | X_0^{\tau-1} = x_0^{\tau-1}\right) \\
&= P\left(S_{t+1} \neq j, S_t = j | X_0^{\tau-1} = x_0^{\tau-1}\right) + P\left(S_{t+1} = j | X_0^{\tau-1} = x_0^{\tau-1}\right) \\
&\quad - P\left(S_{t+1} = j, S_t \neq j | X_0^{\tau-1} = x_0^{\tau-1}\right) \\
&= P\left(S_{t+1} \neq j, S_t = j | X_0^{\tau-1} = x_0^{\tau-1}\right) + L_j(t+1) \\
&\quad - P\left(S_{t+1} = j, S_t \neq j | X_0^{\tau-1} = x_0^{\tau-1}\right), \qquad (10)
\end{aligned}
$$

the smoothed probabilities $L_j(t)$ at time t are related to the other smoothed probabilities at times $t' \neq t$; these probabilities express all the possible state sequences for a given observed sequence. The quantities $\overline{\gamma}_j(t)$ may be unrelated to analog quantities computed at times $t' \neq t$ since these latter quantities may correspond to globally different state sequences. As a consequence, the variations of $L_j(t)$ as a function of t are generally smoother than the variations of $\overline{\gamma}_j(t)$ as a function of t.

The most probable state sequence can be directly deduced as

$$
\widetilde{s}_t = \arg\max_j \left\{ \max_{s_0,\ldots,s_{t-1}} \max_{s_{t+1},\ldots,s_{\tau-1}} P\left(S_0^{t-1} = s_0^{t-1}, S_t = j, S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1} | X_0^{\tau-1} = x_0^{\tau-1}\right) \right\}.
$$

The state sequence computed as

$$
\widetilde{s}_t = \arg\max_j P\left(S_t = j | X_0^{\tau-1} = x_0^{\tau-1}\right).
$$

may not be a valid state sequence.

Hence, the most probable state sequence is only apparent in the state profiles computed by the Viterbi forward-backward algorithm.

If the focus is on the state changes rather than on the states, the quantities $\max_{s_0,\ldots,s_{t-1}} \max_{s_{t+1},\ldots,s_{\tau-1}} P\left(S_0^{t-1} = s_0^{t-1},\ S_t = j, S_{t+1} \neq j, S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1} |\ X_0^{\tau-1} = x_0^{\tau-1}\right)$ or $\max_{s_0,\ldots,s_{t-1}} \max_{s_{t+1},\ldots,s_{\tau-1}} P\left(S_0^{t-1} = s_0^{t-1},\ S_{t-1} \neq j, S_t = j, S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1} |\ X_0^{\tau-1} = x_0^{\tau-1}\right)$ can be computed instead of $\overline{\gamma}_j(t)$ by close variants of the above-described algorithm (for semi-Markovian states, this in particular avoids a maximization step). It should be noted that the quantities $P\left(S_t = j \mid X_0^{\tau-1} = x_0^{\tau-1}\right), P\left(S_{t+1} \neq j,\ S_t = j |\ X_0^{\tau-1} = x_0^{\tau-1}\right)$ and $P\left(S_t = j,\ S_{t-1} \neq j |\ X_0^{\tau-1} = x_0^{\tau-1}\right)$ are linked (see (10)) while the quantities $\max_{s_0,\ldots,s_{t-1}} \max_{s_{t+1},\ldots,s_{\tau-1}} P\left(S_0^{t-1} = s_0^{t-1},\ S_t = j, S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1} |\ X_0^{\tau-1} = x_0^{\tau-1}\right), \max_{s_0,\ldots,s_{t-1}} \max_{s_{t+1},\ldots,s_{\tau-1}} P\left(S_0^{t-1} = s_0^{t-1},\ S_t = j, S_{t+1} \neq j, S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1} |\ X_0^{\tau-1} = x_0^{\tau-1}\right)$ and $\max_{s_0,\ldots,s_{t-1}} \max_{s_{t+1},\ldots,s_{\tau-1}} P\left(S_0^{t-1} = s_0^{t-1},\ S_{t-1} \neq j, S_t = j, S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1} |\ X_0^{\tau-1} = x_0^{\tau-1}\right)$ may correspond to globally different state sequences.

4 Generalized Viterbi algorithm

The objective of the generalized Viterbi algorithm is to compute the top L most probable state sequences for a given observed sequence (Foreman, 1993). In the case of a hidden Markov chain, the optimality of this algorithm relies on the fact that the top L partial most probable state sequences being in state j at time t (assuming that the number of possible partial state sequences at times $t-1$ and $t \geq L$) require at most computation of the top L partial most probable state sequences being in each state at time $t-1$, the extreme situation being the case where the top L partial most probable state sequences being in state j at time t are built from the top L partial most probable state sequences being in a given state i at time $t-1$. This principle directly transposes to the hidden semi-Markov chain case where the top L partial most probable state sequences leaving state j at time t require at most computation of the top L partial most probable state sequences leaving each state i at times $t-u$ for $u = 1, \ldots, t$. The main difference in the presentation of this algorithm compared with the simple Viterbi algorithm (see recursion (4), (5) and (6)) is the management of ranks of partial state sequences at times $t-u$.

Let L_t denote the number of partial state sequences at time t. The number of partial state sequences increases at each time step until it exceeds L, in which case it stays fixed at L.

For a semi-Markovian state j, the forward recursion is given by the following recursion. For each time t, the ranks of the partial state sequences $(r(t-u,j);$ $u = 1, \ldots, t; j = 0, \ldots, J-1)$ should be initialized at 1.

$n = 1, \ldots, L_t$:

$$\alpha_j^n(t) = b_j(x_t) \max \left[\max_{1 \leq u \leq t} \left[\left\{ \prod_{v=1}^{u-1} b_j(x_{t-v}) \right\} d_j(u) \max_{i \neq j} \left\{ p_{ij} \alpha_i^{r(t-u,i)}(t-u) \right\} \right], \right.$$
$$\left. \left\{ \prod_{v=1}^{t} b_j(x_{t-v}) \right\} d_j(t+1) \pi_j \right]. \tag{11}$$

The quantity $\alpha_j^n(t)$ is the probability of the partial observed sequence x_0^t jointly with the nth partial state sequence leaving state j at time t. The above maximization selects a duration u and a partial state sequence leaving state i at time $t-u$ with associated rank $r(t-u,i)$. This rank should then be incremented by one to prevent reselecting the same configuration. The particular case of being in the first state visited should be properly managed since this configuration can only be selected once (since $L_0 = 1$). More generally, the ranks for the values of $t-u$ such that $L_{t-u} < L$ should be carefully managed; see Appendix B.

The censoring at time $\tau - 1$ of the sojourn time in the last visited state distinguishes the case $t = \tau - 1$,

$n = 1, \ldots, L_t$:

$$\begin{aligned}
&\alpha_j^n(\tau-1) \\
&= b_j(x_{\tau-1}) \max\left[\max_{1\le u\le \tau-1}\left[\left\{\prod_{v=1}^{u-1} b_j(x_{\tau-1-v})\right\} D_j(u) \max_{i\neq j}\left\{p_{ij}\alpha_i^{r(\tau-1-u,i)}(\tau-1-u)\right\}\right],\right. \\
&\left.\left\{\prod_{v=1}^{\tau-1} b_j(x_{\tau-1-v})\right\} D_j(\tau)\,\pi_j\right]. \qquad (12)
\end{aligned}$$

In the case of a hidden semi-Markov chain, the complexity of this algorithm is $O(LJ\tau(J+\tau))$-time in the worst case and $O(LJ\tau)$-space. This space complexity may become a limitation in the case of long sequences.

For a Markovian state j, the forward recursion is initialized for $t = 0$ by ($L_0 = 1$)

$$\widetilde{\alpha}_j^1(0) = b_j(x_0)\,\pi_j.$$

For each time t, the ranks of the partial state sequences $(r(t-1,j); j = 0, \ldots, J-1)$ should be initialized at 1. The forward recursion is given by (Foreman, 1993),

$n = 1, \ldots, L_t$:

$$\widetilde{\alpha}_j^n(t) = b_j(x_t) \max_i\left\{\widetilde{p}_{ij}\widetilde{\alpha}_i^{r(t-1,i)}(t-1)\right\}. \qquad (13)$$

The quantity $\widetilde{\alpha}_j^n(t)$ is the probability of the partial observed sequence x_0^t jointly with the nth partial state sequence being in state j at time t. The above maximization selects a partial state sequence being in state i at time $t-1$ with associated rank $r(t-1,i)$. This rank should then be incremented by one to prevent reselecting the same configuration. The rule described in Section 3 for mixing Markovian and semi-Markovian recursions directly transposes to recursions (11), (12) and (13).

The probability of the observed sequence $x_0^{\tau-1}$ jointly with the nth most probable state sequence is $\max_j\left\{\alpha_j^{r(j)}(\tau-1)\right\}$ $\left(\text{or } \max_j\left\{\widetilde{\alpha}_j^{r(j)}(\tau-1)\right\}\right)$ where the ranks of the state sequences $(r(j); j = 0, \ldots, J-1)$ are initialized at 1 and the rank of the selected state sequence is incremented by one to prevent reselecting the same state sequence.

To retrieve the top L most probable state sequences, the recursion described above should be complemented by a backtracking procedure. In this respect, our presentation differs from that of Foreman (1993) where the partial state sequences were built during the forward recursion for each time t, each state j and each rank n. In the case of semi-Markovian states, the backtracking procedure operates by jumps on the basis of three backpointers, the first giving the optimal preceding state, the second the associated rank and the third the optimal preceding time of transition

from this preceding state, while, in the case of Markovian states, the backtracking procedure operates step by step on the basis of two backpointers, the first giving the optimal preceding state and the second the associated rank. An implementation of this algorithm is proposed in Appendix B in pseudo-code form.

The Viterbi forward-backward algorithm and the generalized Viterbi algorithm give two complementary points of view on the state sequences that explain an observed sequence. While the top L most probable state sequences are enumerated by the generalized Viterbi algorithm, the output of the Viterbi forward-backward algorithm can be viewed as the superposition (from the less probable to the most probable) of all the state sequences in a $J \times \tau$ array. Hence, the computation of the quantities $\overline{\gamma}_j(t) = \max_{s_0,\ldots,s_{t-1}} \max_{s_{t+1},\ldots,s_{\tau-1}} P\left(S_0^{t-1} = s_0^{t-1},\ S_t = j, S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1} |\ X_0^{\tau-1} = x_0^{\tau-1}\right)$ may mask some state sequences of high probability (except the top two most probable state sequences). One drawback of the generalized Viterbi algorithm is that the number of state sequences should be a priori fixed without knowledge of the weight of these top L most probable state sequences with respect to all the state sequences.

The exact number of state sequences can be easily computed by a close variant of the forward recursion of the forward-backward algorithm presented in Guédon (2005) (see also Section 5) where the strictly positive parameters (initial, transition, state occupancy and observation probabilities) are replaced by one and the normalization step is removed. The total number of state sequences is then the sum of the forward quantities at time $\tau - 1$. One challenging problem would be to compute the number of state sequences whose cumulated posterior probability is $(1 - \epsilon)$ ($\epsilon = 10^{-2}$ or $\epsilon = 10^{-3}$ to fix the ideas).

5 Forward-backward algorithm for sampling state sequences

A different strategy for obtaining state sequences is to sample from the conditional distribution $P\left(S_0^{\tau-1} = s_0^{\tau-1} |\ X_0^{\tau-1} = x_0^{\tau-1}\right)$. Since

$$\begin{aligned}
&P\left(S_0^{\tau-1} = s_0^{\tau-1} | X_0^{\tau-1} = x_0^{\tau-1}\right) \\
=\ &P\left(S_v = s_0, v = 0, \ldots, u-2 | S_{u-1} = s_0, S_u \neq s_0, S_u^{\tau-1} = s_u^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right) \\
&\times P\left(S_{u-1} = s_{u-1} | S_{u-1} \neq s_u, S_u^{\tau-1} = s_u^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right) \\
&\times P\left(S_{u-1} \neq s_u, S_u^{\tau-1} = s_u^{\tau-1} | X_0^{\tau-1} = x_0^{\tau-1}\right),
\end{aligned}$$

the following conditional distributions should be used for sampling state sequences:

- *final state* (initialization)

$$P\left(S_{\tau-1} = s_{\tau-1} | X_0^{\tau-1} = x_0^{\tau-1}\right),$$

- *state occupancy* (semi-Markovian states)

$$
\begin{aligned}
&t = \tau - 1 : \\
&P\left(S_{\tau-1-u} \neq s_{\tau-1}, S_{\tau-1-v} = s_{\tau-1}, v = 1, \ldots, u-1 | S_{\tau-1} = s_{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right), \\
&t < \tau - 1 : \\
&P\left(S_{t-u} \neq s_t, S_{t-v} = s_t, v = 1, \ldots, u-1 | S_t = s_t, S_{t+1} \neq s_t, S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right),
\end{aligned}
$$

- *previous state*

$$
\begin{aligned}
&P\left(S_t = s_t | S_t \neq s_{t+1}, S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right), && \text{semi-Markovian state,} \\
&P\left(S_t = s_t | S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right), && \text{Markovian state.}
\end{aligned}
$$

The forward-backward algorithm for sampling state sequences decomposes into two passes, a forward pass which is the usual forward recursion of the forward-backward algorithm, and a backward pass for sampling state sequences.

For a semi-Markovian state j, the forward recursion is given by (Guédon, 2005),

$t = 0, \ldots, \tau - 2 :$

$$
\begin{aligned}
F_j(t) &= P\left(S_{t+1} \neq j, S_t = j | X_0^t = x_0^t\right) \\
&= \frac{b_j(x_t)}{N_t} \left[\sum_{u=1}^{t} \left\{ \prod_{v=1}^{u-1} \frac{b_j(x_{t-v})}{N_{t-v}} \right\} d_j(u) \sum_{i \neq j} p_{ij} F_i(t-u) \right. \\
&\quad \left. + \left\{ \prod_{v=1}^{t} \frac{b_j(x_{t-v})}{N_{t-v}} \right\} d_j(t+1) \pi_j \right], \qquad (14)
\end{aligned}
$$

where $N_t = P\left(X_t = x_t | X_0^{t-1} = x_0^{t-1}\right)$ is a normalizing factor.

The censoring at time $\tau - 1$ of the sojourn time in the last visited state distinguishes the case $t = \tau - 1$

$$
\begin{aligned}
F_j(\tau - 1) &= P\left(S_{\tau-1} = j | X_0^{\tau-1} = x_0^{\tau-1}\right) \\
&= \frac{b_j(x_{\tau-1})}{N_{\tau-1}} \left[\sum_{u=1}^{\tau-1} \left\{ \prod_{v=1}^{u-1} \frac{b_j(x_{\tau-1-v})}{N_{\tau-1-v}} \right\} D_j(u) \sum_{i \neq j} p_{ij} F_i(\tau - 1 - u) \right. \\
&\quad \left. + \left\{ \prod_{v=1}^{\tau-1} \frac{b_j(x_{\tau-1-v})}{N_{\tau-1-v}} \right\} D_j(\tau) \pi_j \right]. \qquad (15)
\end{aligned}
$$

For a Markovian state j, the forward recursion initialized for $t = 0$ by

$$\begin{aligned}\widetilde{F}_j(0) &= P(S_0 = j | X_0 = x_0) \\ &= \frac{b_j(x_0)}{N_0}\pi_j,\end{aligned}$$

is given by,

$t = 1, \ldots, \tau - 1$:

$$\begin{aligned}\widetilde{F}_j(t) &= P\left(S_t = j | X_0^t = x_0^t\right) \\ &= \frac{b_j(x_t)}{N_t}\sum_i \widetilde{p}_{ij}\widetilde{F}_i(t-1). \end{aligned} \tag{16}$$

The normalizing factor N_t is obtained directly during the forward recursion as follows

$$\begin{aligned}N_t &= P\left(X_t = x_t | X_0^{t-1} = x_0^{t-1}\right) \\ &= \sum_j P\left(S_t = j, X_t = x_t | X_0^{t-1} = x_0^{t-1}\right).\end{aligned}$$

with

$$P(S_0 = j, X_0 = x_0) = b_j(x_0)\pi_j,$$

and for a semi-Markovian state j,

$t = 1, \ldots, \tau - 1$:

$$\begin{aligned}&P\left(S_t = j, X_t = x_t | X_0^{t-1} = x_0^{t-1}\right) \\ &= P(X_t = x_t | S_t = j)\left\{P\left(S_t = j, S_{t-1} \neq j | X_0^{t-1} = x_0^{t-1}\right)\right. \\ &\quad \left. - P\left(S_t \neq j, S_{t-1} = j | X_0^{t-1} = x_0^{t-1}\right) + P\left(S_{t-1} = j | X_0^{t-1} = x_0^{t-1}\right)\right\} \\ &= b_j(x_t)\left\{\sum_{i \neq j} p_{ij}F_i(t-1) - F_j(t-1) + P\left(S_{t-1} = j | X_0^{t-1} = x_0^{t-1}\right)\right\},\end{aligned} \tag{17}$$

where $P\left(S_{t-1} = j | X_0^{t-1} = x_0^{t-1}\right) = P\left(S_{t-1} = j, X_{t-1} = x_{t-1} | X_0^{t-2} = x_0^{t-2}\right)/N_{t-1}$,

and for a Markovian state j,

$t = 1, \ldots, \tau - 1$:

$$P\left(S_t = j, X_t = x_t | X_0^{t-1} = x_0^{t-1}\right) = b_j\left(x_t\right) \sum_i \widetilde{p}_{ij} \widetilde{F}_i\left(t-1\right). \tag{18}$$

The quantity $p_{ij}F_i\left(t-u\right)$ in (14) and (15), and the quantity $p_{ij}F_i\left(t-1\right)$ in (17), should be replaced by $\widetilde{p}_{ij}\widetilde{F}_i\left(t-u\right)$ and $\widetilde{p}_{ij}\widetilde{F}_i\left(t-1\right)$ respectively if state i is Markovian and, conversely, the quantity $\widetilde{p}_{ij}\widetilde{F}_i\left(t-1\right)$ in (16) and (18) should be replaced by $p_{ij}F_i\left(t-1\right)$ if state i is semi-Markovian (Guédon, 2005). The resulting forward algorithm computes in parallel $F_j\left(t\right) = P\left(S_{t+1} \neq j,\ S_t = j |\ X_0^t = x_0^t\right)$ for semi-Markovian states and $\widetilde{F}_j\left(t\right) = P\left(S_t = j |\ X_0^t = x_0^t\right)$ for Markovian states. Note that $F_j\left(\tau-1\right) = \widetilde{F}_j\left(\tau-1\right) = P\left(S_{\tau-1} = j |\ X_0^{\tau-1} = x_0^{\tau-1}\right)$.

The backward pass can be seen as a stochastic backtracking procedure, in contrast to the optimal backtracking procedure of the (generalized) Viterbi algorithm (Cawley and Pachter, 2003).

The backward pass is initialized for $t = \tau - 1$ by,

$j = 0, \ldots, J-1$:

$$P\left(S_{\tau-1} = s_{\tau-1} | X_0^{\tau-1} = x_0^{\tau-1}\right) = \widetilde{F}_j\left(\tau-1\right) = F_j\left(\tau-1\right).$$

The final state $s_{\tau-1}$ is drawn from the smoothed probabilities

$$\left(P\left(S_{\tau-1} = j | X_0^{\tau-1} = x_0^{\tau-1}\right); j = 0, \ldots, J-1\right).$$

For a semi-Markovian state s_{t+1}, the backward pass relies for state change on,

$j = 0, \ldots, J-1$:

$$\begin{aligned}
&P\left(S_t = j | S_t \neq s_{t+1}, S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right) \\
&= \frac{P\left(S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1}, S_t \neq s_{t+1}, S_t = j, X_0^{\tau-1} = x_0^{\tau-1}\right)}{P\left(S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1}, S_t \neq s_{t+1}, X_0^{\tau-1} = x_0^{\tau-1}\right)} \\
&= \frac{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1}, S_{t+2}^{\tau-1} = s_{t+2}^{\tau-1} | S_{t+1} = s_{t+1}, S_t \neq s_{t+1}\right)}{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1}, S_{t+2}^{\tau-1} = s_{t+2}^{\tau-1} | S_{t+1} = s_{t+1}, S_t \neq s_{t+1}\right)} \\
&\quad \times \frac{P\left(S_{t+1} = s_{t+1} | S_{t+1} \neq j, S_t = j\right) P\left(S_{t+1} \neq j, S_t = j | X_0^t = x_0^t\right)}{P\left(S_{t+1} = s_{t+1}, S_t \neq s_{t+1} | X_0^t = x_0^t\right)} \\
&= \frac{p_{j s_{t+1}} F_j\left(t\right)}{G_{s_{t+1}}\left(t+1\right)},
\end{aligned}$$

where $F_j\left(t\right)$ and $G_{s_{t+1}}\left(t+1\right) = \sum_{i \neq s_{t+1}} p_{i s_{t+1}} F_i\left(t\right)$ are computed and stored during the forward recursion. If state j is Markovian, $p_{j s_{t+1}} F_j\left(t\right)$ should be replaced by

$\widetilde{p}_{js_{t+1}}\widetilde{F}_j(t)$ (Guédon, 2005).

The state s_t is drawn from the conditional distribution

$$\left(P\left(S_t = j | S_t \neq s_{t+1}, S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right); j = 0, \ldots, J-1\right).$$

For a Markovian state s_{t+1}, the backward pass is given by (Chib, 1996),

$j = 0, \ldots, J-1$:

$$\begin{aligned}
&P\left(S_t = j | S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right) \\
&= \frac{P\left(S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1}, S_t = j, X_0^{\tau-1} = x_0^{\tau-1}\right)}{P\left(S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right)} \\
&= \frac{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1}, S_{t+2}^{\tau-1} = s_{t+2}^{\tau-1} | S_{t+1} = s_{t+1}\right) P\left(S_{t+1} = s_{t+1} | S_t = j\right)}{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1}, S_{t+2}^{\tau-1} = s_{t+2}^{\tau-1} | S_{t+1} = s_{t+1}\right) P\left(S_{t+1} = s_{t+1} | X_0^t = x_0^t\right)} \\
&\times P\left(S_t = j | X_0^t = x_0^t\right) \\
&= \frac{\widetilde{p}_{js_{t+1}}\widetilde{F}_j(t)}{\widetilde{G}_{s_{t+1}}(t+1)},
\end{aligned}$$

where the filtered probability $\widetilde{F}_j(t)$ and the predicted probability $\widetilde{G}_{s_{t+1}}(t+1) = \sum_i \widetilde{p}_{is_{t+1}}\widetilde{F}_i(t)$ are computed and stored during the forward recursion. If state j is semi-Markovian, $\widetilde{p}_{js_{t+1}}\widetilde{F}_j(t)$ should be replaced by $p_{js_{t+1}}F_j(t)$.

The state s_t is drawn from the conditional distribution

$$\left(P\left(S_t = j | S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right); j = 0, \ldots, J-1\right).$$

If the selected state $s_t = j$ is semi-Markovian, the backward pass relies for state occupancy on

$$
\begin{aligned}
&P\left(S_{t-u}\neq j,S_{t-v}=j,v=1,\ldots,u-1|S_t=j,S_{t+1}\neq j,S_{t+1}^{\tau-1}=s_{t+1}^{\tau-1},X_0^{\tau-1}=x_0^{\tau-1}\right)\\
=&\frac{P\left(S_{t+1}^{\tau-1}=s_{t+1}^{\tau-1},S_{t+1}\neq j,S_{t-v}=j,v=0,\ldots,u-1,S_{t-u}\neq j,X_0^{\tau-1}=x_0^{\tau-1}\right)}{P\left(S_{t+1}^{\tau-1}=s_{t+1}^{\tau-1},S_{t+1}\neq j,S_t=j,X_0^{\tau-1}=x_0^{\tau-1}\right)}\\
=&\frac{P\left(X_{t+1}^{\tau-1}=x_{t+1}^{\tau-1},S_{t+2}^{\tau-1}=s_{t+2}^{\tau-1}|S_{t+1}\neq j,S_t=j\right)}{P\left(X_{t+1}^{\tau-1}=x_{t+1}^{\tau-1},S_{t+2}^{\tau-1}=s_{t+2}^{\tau-1}|S_{t+1}\neq j,S_t=j\right)}\\
&\times\frac{P\left(X_{t-u+1}^{t}=x_{t-u+1}^{t}|S_{t-v}=j,v=0,\ldots,u-1\right)}{P\left(X_{t-u+1}^{t}=x_{t-u+1}^{t}|X_0^{t-u}=x_0^{t-u}\right)}\\
&\times\frac{P\left(S_{t+1}\neq j,S_{t-v}=j,v=0,\ldots,u-2|S_{t-u+1}=j,S_{t-u}\neq j\right)}{P\left(S_{t+1}\neq j,S_t=j|X_0^t=x_0^t\right)}\\
&\times P\left(S_{t-u+1}=j,S_{t-u}\neq j|X_0^{t-u}=x_0^{t-u}\right)\\
=&\left\{\prod_{v=0}^{u-1}\frac{b_j\left(x_{t-v}\right)}{N_{t-v}}\right\}\frac{d_j\left(u\right)G_j\left(t-u+1\right)}{F_j\left(t\right)},
\end{aligned}
\tag{19}
$$

for $u<t+1$ and

$$
\begin{aligned}
&P\left(S_{t-v}=j,v=1,\ldots,t|S_t=j,S_{t+1}\neq j,S_{t+1}^{\tau-1}=s_{t+1}^{\tau-1},X_0^{\tau-1}=x_0^{\tau-1}\right)\\
=&\left\{\prod_{v=0}^{t}\frac{b_j\left(x_{t-v}\right)}{N_{t-v}}\right\}\frac{d_j\left(t+1\right)\pi_j}{F_j\left(t\right)},
\end{aligned}
\tag{20}
$$

for $u=t+1$.

The conditional probabilities (19) (20) are simply the terms summed for each u in the computation of the forward probability $F_j\left(t\right)$ (14) divided by this forward probability.

If the state selected at time $\tau-1$ is semi-Markovian, the expressions (19) (20) should be replaced by

$$
\begin{aligned}
&P\left(S_{\tau-1-u}\neq j,S_{\tau-1-v}=j,v=1,\ldots,u-1|S_{\tau-1}=j,X_0^{\tau-1}=x_0^{\tau-1}\right)\\
=&\left\{\prod_{v=0}^{u-1}\frac{b_j\left(x_{\tau-1-v}\right)}{N_{\tau-1-v}}\right\}\frac{D_j\left(u\right)G_j\left(\tau-u\right)}{F_j\left(\tau-1\right)},
\end{aligned}
$$

for $u<\tau$ and

$$
\begin{aligned}
&P\left(S_{\tau-1-v}=j,v=1,\ldots,\tau-1|S_{\tau-1}=j,X_0^{\tau-1}=x_0^{\tau-1}\right)\\
=&\left\{\prod_{v=0}^{\tau-1}\frac{b_j\left(x_{\tau-1-v}\right)}{N_{\tau-1-v}}\right\}\frac{D_j\left(\tau\right)\pi_j}{F_j\left(\tau-1\right)},
\end{aligned}
$$

for $u = \tau$.

The time spent in state j is drawn from

$$\left(P\left(S_{t-u} \neq j, S_{t-v} = j, v = 1, \ldots, u-1 | S_t = j, S_{t+1} \neq j, S_{t+1}^{\tau-1} = s_{t+1}^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right);\right.$$
$$\left.u = 1, \ldots, \min\left(t+1, M_j\right)\right)$$

It should be noted that, in practice, the time complexity of the backward pass is relatively low since for a semi-Markovian state j, the conditional distribution both for the preceding states and the sojourn times in state j need only to be computed at state change times. In this respect, the backward pass is similar to the backtracking procedure of the Viterbi algorithm which operates by jumps (Guédon, 2003). Hence, it is possible to sample quite a large number of state sequences and to extract, among these state sequences, those which are structurally different with reference to the most probable state sequence (or with reference to already extracted state sequences) and whose probability is not negligible. This sampling algorithm can be used as a building block in estimation methods that rely either on stochastic versions of the EM algorithm (MCEM or SEM algorithm; see Tanner (1996) and McLachlan and Krishnan (1997)) or in Bayesian estimation methods (Scott, 2002). One interesting property of the sampling of state sequences lies in the fact that the number of state sequences has not to be a priori defined (unlike the number L of the top most probable state sequences computed by the generalized Viterbi algorithm which has to be a priori defined; see Section 4). Another interesting property is that the memory requirement does not depend on the number of sampled state sequences unlike the memory requirement of the generalized Viterbi algorithm which increases with L. The sampling of state sequences can therefore be recommended if the number of possible state sequences is very high.

6 Entropy as a global measure of the state sequence uncertainty

Following Hernando *et al.* (2005), we propose to use the entropy of the state sequence that explains an observed sequence as a global measure of the state sequence uncertainty. For a hidden Markovian model, this entropy is given by

$$\begin{aligned}
& H\left(S_0^{\tau-1} | X_0^{\tau-1} = x_0^{\tau-1}; \theta\right) \\
= & -\sum_{s_0, \ldots, s_{\tau-1}} P\left(S_0^{\tau-1} = s_0^{\tau-1} | X_0^{\tau-1} = x_0^{\tau-1}; \theta\right) \log P\left(S_0^{\tau-1} = s_0^{\tau-1} | X_0^{\tau-1} = x_0^{\tau-1}; \theta\right) \\
= & -\sum_{s_0, \ldots, s_{\tau-1}} P\left(S_0^{\tau-1} = s_0^{\tau-1} | X_0^{\tau-1} = x_0^{\tau-1}; \theta\right) \log P\left(S_0^{\tau-1} = s_0^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}; \theta\right) \\
& + \log P\left(X_0^{\tau-1} = x_0^{\tau-1}; \theta\right) \\
= & -E\left\{\log f\left(S_0^{\tau-1}, X_0^{\tau-1}; \theta\right) | X_0^{\tau-1} = x_0^{\tau-1}; \theta\right\} + \log P\left(X_0^{\tau-1} = x_0^{\tau-1}; \theta\right),
\end{aligned}$$

where θ designates the parameters of the model. The first term is the conditional expectation used in the statement of the EM algorithm (Dempster *et al.*, 1977; McLachlan and Krishnan, 1997) while the second term is the log-likelihood of the observed sequence. In the sequel, we will omit to note systematically θ. For a hidden semi-Markov chain, the entropy can be rewritten as

$$
\begin{aligned}
& H\left(S_0^{\tau-1}|X_0^{\tau-1}=x_0^{\tau-1}\right) \\
=& -\sum_j P\left(S_0=j|X_0^{\tau-1}=x_0^{\tau-1}\right)\log\pi_j \\
& -\sum_i\sum_{j\neq i}\sum_{t=0}^{\tau-2} P\left(S_{t+1}=j,S_t=i|X_0^{\tau-1}=x_0^{\tau-1}\right)\log p_{ij} \\
& -\sum_j\sum_u\left\{\sum_{t=0}^{\tau-2-u} P\left(S_{t+u+1}\neq j,S_{t+u-v}=j,v=0,\ldots,u-1,S_t\neq j|X_0^{\tau-1}=x_0^{\tau-1}\right)\right. \\
& \left.+P\left(S_u\neq j,S_{u-v}=j,v=1,\ldots,u|X_0^{\tau-1}=x_0^{\tau-1}\right)I\left(u\leq\tau-1\right)\right\}\log d_j\left(u\right) \\
& -\sum_j\sum_u\left\{P\left(S_{\tau-1-v}=j,v=0,\ldots,u-1,S_{\tau-1-u}\neq j|X_0^{\tau-1}=x_0^{\tau-1}\right)\right. \\
& \left.+P\left(S_{\tau-1-v}=j,v=0,\ldots,u-1|X_0^{\tau-1}=x_0^{\tau-1}\right)I\left(u=\tau\right)\right\}\log D_j\left(u\right) \\
& -\sum_j\sum_y\sum_{t=0}^{\tau-1} P\left(X_t=y,S_t=j|X_0^{\tau-1}=x_0^{\tau-1}\right)\log b_j\left(y\right) \\
& +\sum_{t=0}^{\tau-1}\log P\left(X_t=x_t|X_0^{t-1}=x_0^{t-1}\right). \qquad (21)
\end{aligned}
$$

For a Markovian state i, the terms corresponding to the state occupancy distribution and the transition probabilities ($p_{ij},\ j=1,...,J;j\neq i$) should be replaced by

$$
-\sum_j\sum_{t=0}^{\tau-2} P\left(S_{t+1}=j,S_t=i|X_0^{\tau-1}=x_0^{\tau-1}\right)\log\widetilde{p}_{ij}. \qquad (22)
$$

The entropy can thus be directly extracted as a byproduct of the forward-backward algorithm; see Guédon (2003, 2005) for the practical computation of the terms involved in (21) and (22). The entropy is upper bounded by log(number of possible state sequences). The exact number of state sequences can be computed by a close variant of the forward recursion of the forward-backward algorithm presented in Guédon (2005); see Section 5. The entropy is also upper bounded by (Cover and Thomas, 1991; Theorem 2.6.6)

$$
H\left(S_0^{\tau-1}|X_0^{\tau-1}=x_0^{\tau-1}\right)\leq\sum_{t=0}^{\tau-1}H\left(S_t|X_0^{\tau-1}=x_0^{\tau-1}\right),
$$

with

$$\sum_{t=0}^{\tau-1} H\left(S_t|X_0^{\tau-1}=x_0^{\tau-1}\right) = -\sum_{t=0}^{\tau-1}\sum_j P\left(S_t=j|X_0^{\tau-1}=x_0^{\tau-1}\right)\log P\left(S_t=j|X_0^{\tau-1}=x_0^{\tau-1}\right)$$
$$= -\sum_{t=0}^{\tau-1}\sum_j L_j(t)\log L_j(t).$$

A specific algorithm was proposed by Hernando *et al.* (2005) for computing the entropy of the state sequence that explains an observed sequence in the case of a hidden Markov chain. This algorithm includes the classical forward recursion as a building block and requires a forward recursion on the entropies of partial state sequences being in state j at time t

$$\begin{aligned}
&H\left(S_0^{t-1}|S_t=j,X_0^t=x_0^t\right)\\
=&-\sum_{s_0,\ldots,s_{t-1}} P\left(S_0^{t-1}=s_0^{t-1}|S_t=j,X_0^t=x_0^t\right)\log P\left(S_0^{t-1}=s_0^{t-1}|S_t=j,X_0^t=x_0^t\right)\\
=&\sum_i P\left(S_{t-1}=i|S_t=j,X_0^{t-1}=x_0^{t-1}\right)\left\{H\left(S_0^{t-2}|S_{t-1}=i,X_0^{t-1}=x_0^{t-1}\right)\right.\\
&\left.-\log P\left(S_{t-1}=i|S_t=j,X_0^{t-1}=x_0^{t-1}\right)\right\},
\end{aligned}$$

with

$$P\left(S_{t-1}=i|S_t=j,X_0^{t-1}=x_0^{t-1}\right)=\frac{\widetilde{p}_{ij}\widetilde{F}_i(t-1)}{\widetilde{G}_j(t)}.$$

The termination step is given by

$$\begin{aligned}
&H\left(S_0^{\tau-1}|X_0^{\tau-1}=x_0^{\tau-1}\right)\\
=&-\sum_{s_0,\ldots,s_{\tau-1}} P\left(S_0^{\tau-1}=s_0^{\tau-1}|X_0^{\tau-1}=x_0^{\tau-1}\right)\log P\left(S_0^{\tau-1}=s_0^{\tau-1}|X_0^{\tau-1}=x_0^{\tau-1}\right)\\
=&\sum_j \widetilde{F}_j(\tau-1)\left\{H\left(S_0^{\tau-2}|S_{\tau-1}=j,X_0^{\tau-1}=x_0^{\tau-1}\right)-\log\widetilde{F}_j(\tau-1)\right\}.
\end{aligned}$$

This algorithm can be transposed to hidden semi-Markov chains. In this case, the algorithm requires a forward recursion on the entropies of partial state sequences leaving state j at time t

$$\begin{aligned}
&H\left(S_0^{t-1}|S_t=j,S_{t+1}\neq j,X_0^t=x_0^t\right)\\
=&-\sum_{s_0,\ldots,s_{t-1}} P\left(S_0^{t-1}=s_0^{t-1}|S_t=j,S_{t+1}\neq j,X_0^t=x_0^t\right)\\
&\times\log P\left(S_0^{t-1}=s_0^{t-1}|S_t=j,S_{t+1}\neq j,X_0^t=x_0^t\right).
\end{aligned}$$

Whatever the approach retained, the complexity of the algorithm for computing the entropy of the state sequence that explains an observed sequence is the complexity of the forward-backward algorithm, that is $O(J\tau(J+\tau))$-time in the worst case and $O(J\tau)$-space. The entropy may be normalized by the sequence length τ, or unormalized depending on the application context. For instance, if the number of state changes is fixed irrespective of the sequence length (this is the case for the Corsican pine sequences presented in Section 7.1), the unormalized entropy should be used.

7 Application to the analysis of plant structure

7.1 Growth and branching Corsican pin trunks

The data set comprised four sub-samples of Corsican pines (*Pinus nigra* Arn. *ssp. laricio* Poir., *Pinaceae*) planted in a forest stand in Centre region (France): 31 six-year-old trees, 29 twelve-year-old trees (first year not measured), 31 eighteen-year-old trees (first year not measured) and 13 twenty–three-year-old trees (two first years not measured). Trees of the first sub-sample (six years old) remained for two years in the nursery before transplantation while trees of the three other sub-samples remained for three years in the nursery before transplantation. Plantation density was 1800 trees/ha for the first sub-sample (six year old) and 2200 trees/ha for the three other sub-samples. Tree trunks were described by annual shoot from the base to the top where two variables were recorded for each annual shoot: length (in cm) and number of branches per tier. The very first non-measured annual shoots were always very short. The annual shoot is defined as the segment of stem established within a year. In the Corsican pine case, branches of roughly equivalent size are located at the top of the shoot just below the shoot limit and thus form a tier of branches.

The observed growth is the result of the modulation of the endogenous growth component by climatic factors. For the endogenous growth component, two assumptions can be made:

- the endogenous growth component takes the form of a trend,
- the endogenous growth component is structured as a succession of phases separated by marked transitions.

The objective of this study was to investigate this latter assumption.

From these four sub-samples (different climatic years are thus mixed for a given stage of development), a three-state hidden semi-Markov chain composed of two successive transient states followed by a final absorbing state was estimated.

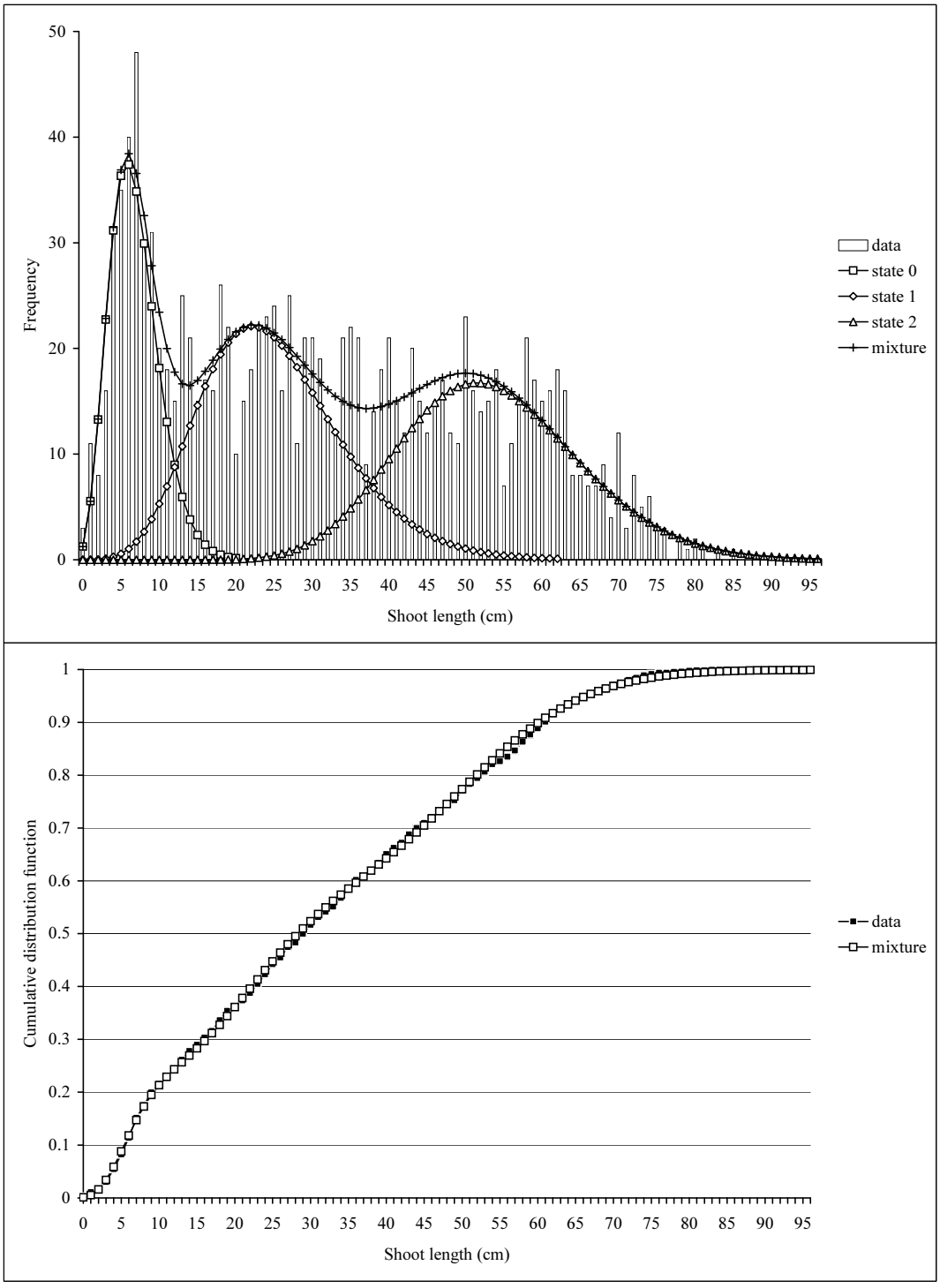

Fig. 1. Corsican pine: mixture of observation distributions for the variable "length of the annual shoot".

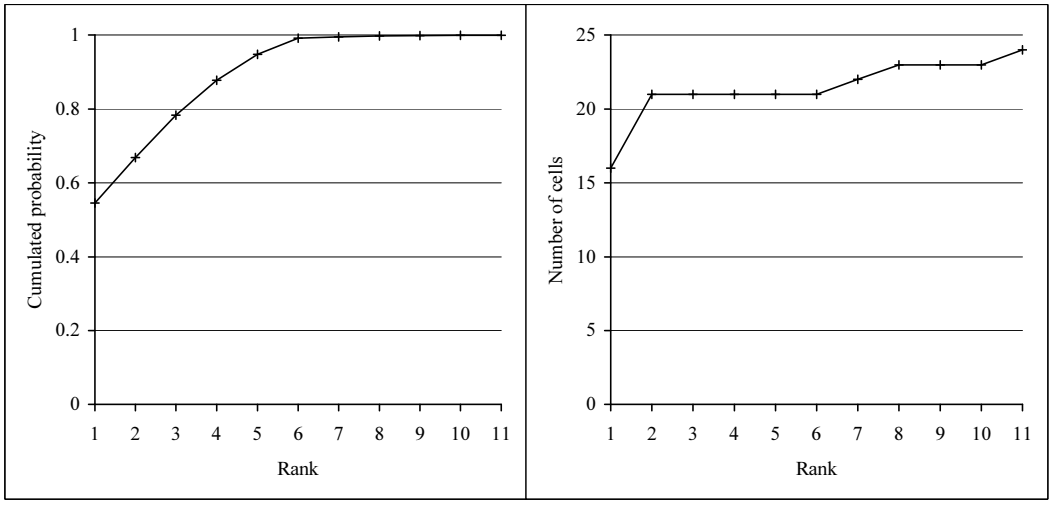

Fig. 2. Corsican pine: characteristics of state sequences.

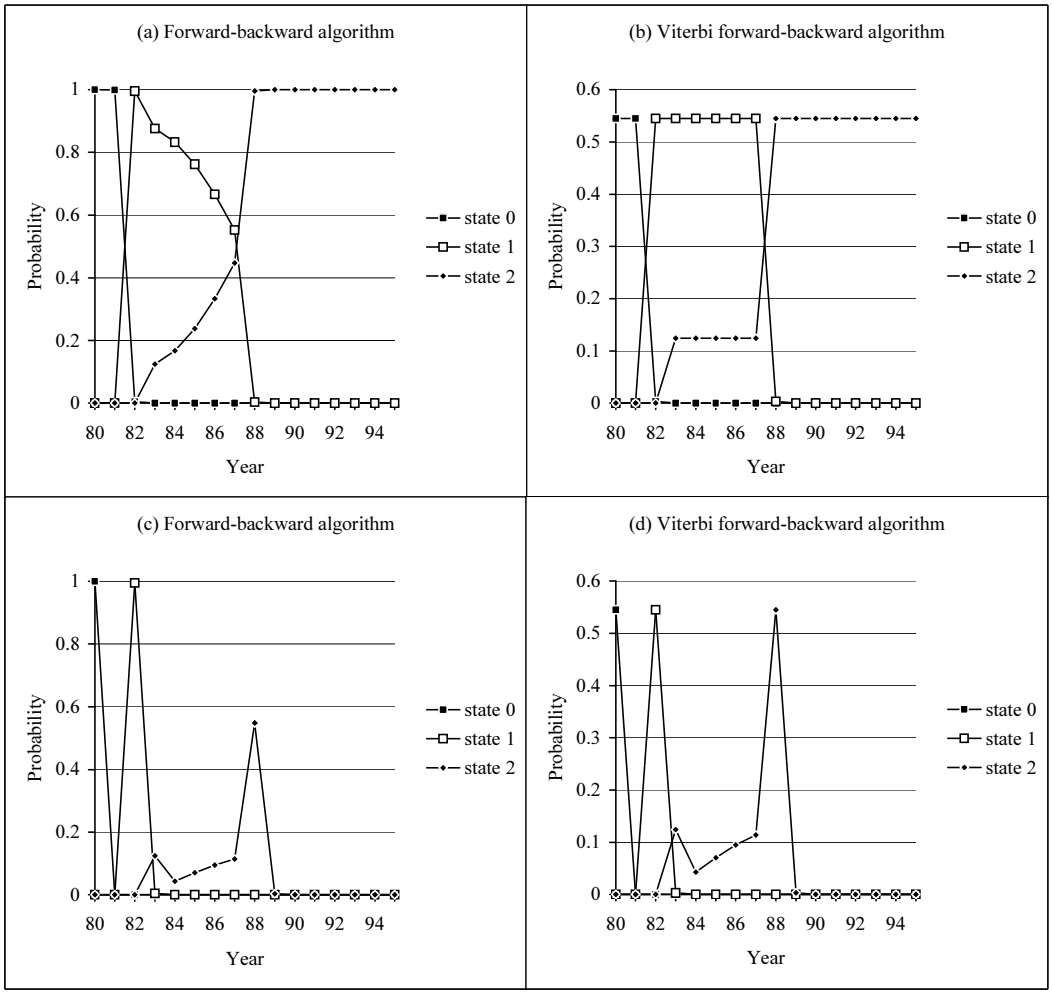

Fig. 3. Corsican pine: state and state change profiles.

This hidden semi-Markov chain represents the succession of growth phases. The EM algorithm was initialized with a three-state "left-right" model with the three possible initial states and possible transitions from each state to the following states (i.e. from state j to state k with $k > j$ and in particular from state 0 to state 2). The convergence of the EM algorithm required 17 iterations. In the estimated model, state 0 is the only possible initial state and state 1 cannot be skipped. This deterministic succession of states supports the assumption of a succession of growth phases. For the two observed variables (length of the annual shoot, number of branches per tier), the observation distributions are parametric discrete distributions chosen among binomial, Poisson and negative binomial distributions with an additional shift parameter. The fit of the empirical marginal distribution for the variable "length of the annual shoot" by the mixture of observation distributions is shown in Fig. 1. Hence, the three states correspond clearly to different ranges of values for the variable "length of the annual shoot".

We selected two eighteen-year-old individuals which correspond to some of the higher values for the entropy. The two individuals are among the 10 eighteen-year-old individuals above 1.42 for the state sequence entropy. For 12 eighteen-year-old individuals, the state sequence entropy is below 1.15.

There are 34 possible state sequences for the first individual but the cumulated posterior probability of the top 11 most probable state sequences exceeds 0.999 (Fig. 2). The observed bivariate sequence is:

```
5 4 20 39 36 34 36 31 54 51 34 41 68 60 43 75  (length of the annual shoot in cm)
2 3  5  8  6  7  6  6  8 11  5  7  8  8  5  8  (number of branches per tier)
```

The top 5 most probable state sequences are:

```
0 0 1 1 1 1 1 1 2 2 2 2 2 2 2 2  (0.55  0.55  16)
0 0 1 2 2 2 2 2 2 2 2 2 2 2 2 2  (0.12  0.67  21)
0 0 1 1 1 1 1 2 2 2 2 2 2 2 2 2  (0.11  0.78  21)
0 0 1 1 1 1 2 2 2 2 2 2 2 2 2 2  (0.1   0.88  21)
0 0 1 1 1 2 2 2 2 2 2 2 2 2 2 2  (0.07  0.95  21)
```

where the first indicator in parentheses is the posterior probability of the state sequence, the second indicator is the cumulated posterior probability of the top n most probable state sequences and the last indicator is the number of cells used by the top n most probable state sequences in the $J \times \tau$ array. The main structural difference between the sub-optimal state sequences and the most probable state sequence is the difference in year for the transition from state 1 to state 2. It should be noted that the 3rd, 4th and 5th state sequences are masked in the state profiles (Fig. 3b), while they are apparent in the state change profiles (Fig. 3d) both computed by the Viterbi forward-backward algorithm.

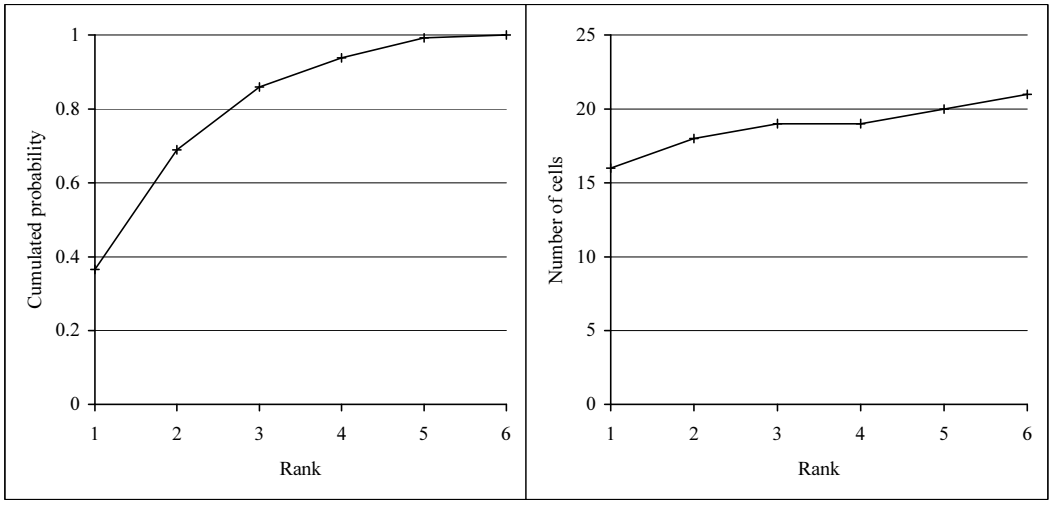

Fig. 4. Corsican pine: characteristics of state sequences.

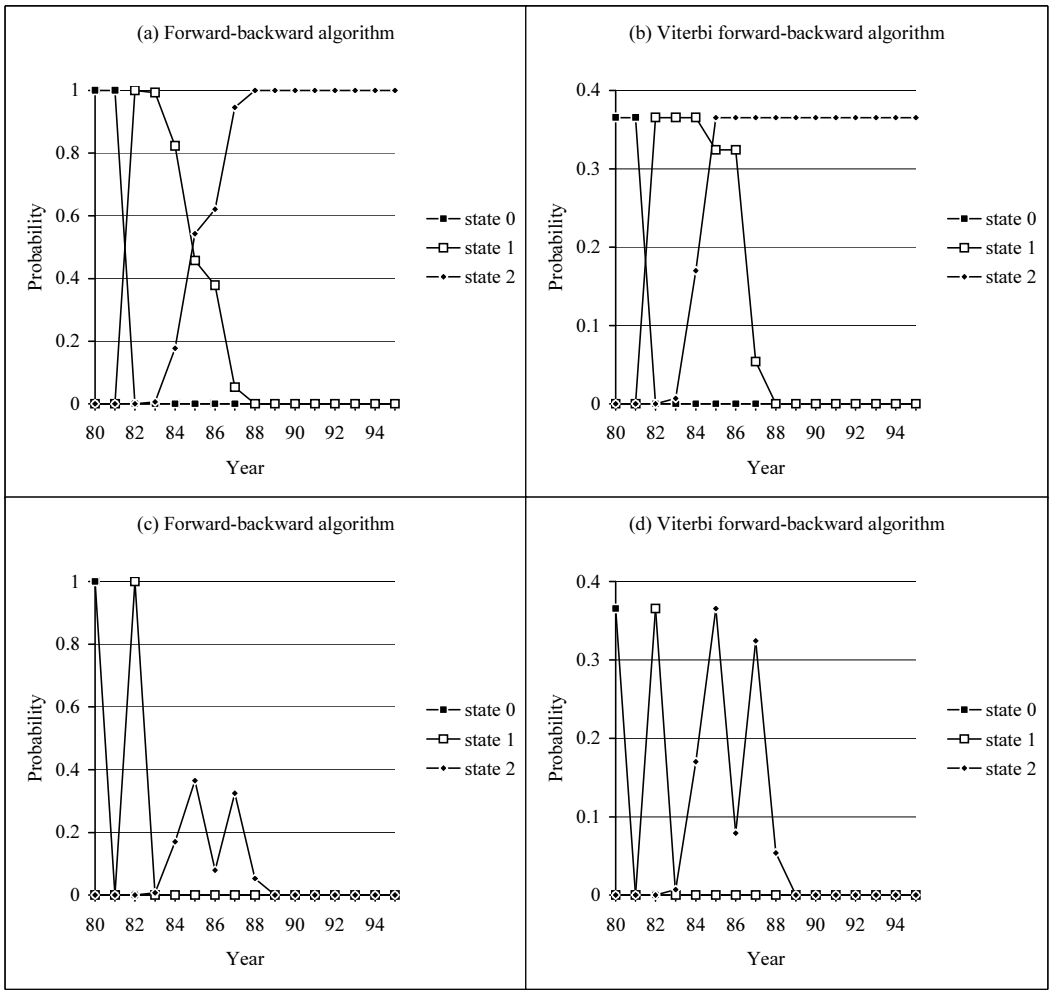

Fig. 5. Corsican pine: state and state change profiles.

This example illustrates the fact that the state profiles and state change profiles computed by the forward-backward algorithm (Fig. 3ac) give two different points of view on the same information while the state profiles and the state change profiles computed by the Viterbi forward-backward algorithm (Fig. 3bd) may highlight different properties of the state sequences; see Section 3 for a rationale for this behavior.

There are 24 possible state sequences for the second individual but the cumulated posterior probability of the top 6 most probable state sequences exceeds 0.999 (Fig. 4). The observed bivariate sequence is:

```
4 4 29 30 35 41 36 46 63 57 61 53 47 65 58 74  (length of the annual shoot in cm)
1 2  3  5  6  7  3  5  6  3  5  5  5  5  5  7  (number of branches per tier)
```

The top 5 most probable state sequences are:

```
0 0 1 1 1 2 2 2 2 2 2 2 2 2 2 2  (0.37  0.37  16)
0 0 1 1 1 1 1 2 2 2 2 2 2 2 2 2  (0.32  0.69  18)
0 0 1 1 2 2 2 2 2 2 2 2 2 2 2 2  (0.17  0.86  19)
0 0 1 1 1 1 2 2 2 2 2 2 2 2 2 2  (0.08  0.94  19)
0 0 1 1 1 1 1 1 2 2 2 2 2 2 2 2  (0.05  0.99  20)
```

It should be noted that, among the top 5 most probable state sequences, only the 4th state sequence is masked in the state profiles (Fig. 5b) while this state sequence is apparent in the state change profiles (Fig. 5d) both computed by the Viterbi forward-backward algorithm. As for the first example, the main structural difference between the sub-optimal state sequences and the most probable state sequence is the difference in year for the transition from state 1 to state 2. While the state profiles computed by the forward-backward algorithm are quite similar (Figs. 3a and 5a), there are in fact marked differences between the sub-optimal state sequences. For the first example, the relative weight of the second most probable state sequence is only 0.23 but with a transition from state 1 to state 2 five years before with reference to the most probable state sequence (Fig. 3b), while for the second example, the relative weight of the second most probable state sequence is 0.89 but with a transition from state 1 to state 2 two years before (Fig. 5b). The number of cells can be interpreted as the beam size of the top n most probable state sequences. Hence, large differences in the number of cells (Figs. 2 and 4) highlight structurally different state sequences and may be used to detect such sequences automatically.

7.2 Branching structure of red oak growth units

This second example focuses on the branching structure of growth units observed on the main axis of four-year-old red oaks (Quercus rubra L.). These trees were observed in the Nabas forest (southwest of France). Red oak is characterized by a polycyclic growth which means that an annual shoot may be composed of several successive growth units (i.e. portion of the axis built up between two resting phases).

If growth occurs in two stages within a year, the annual shoot is said to be bicyclic, and comprises a spring growth unit followed by a summer growth unit; for more detailed information, see Heuret et al. (2003). Summer growth units were described node by node from the top to the base. For each node, the type of axillary production chosen among latent bud (0), one-year-delayed monocyclic offspring shoot (1) and one-year-delayed polycyclic offspring shoot (2), was recorded.

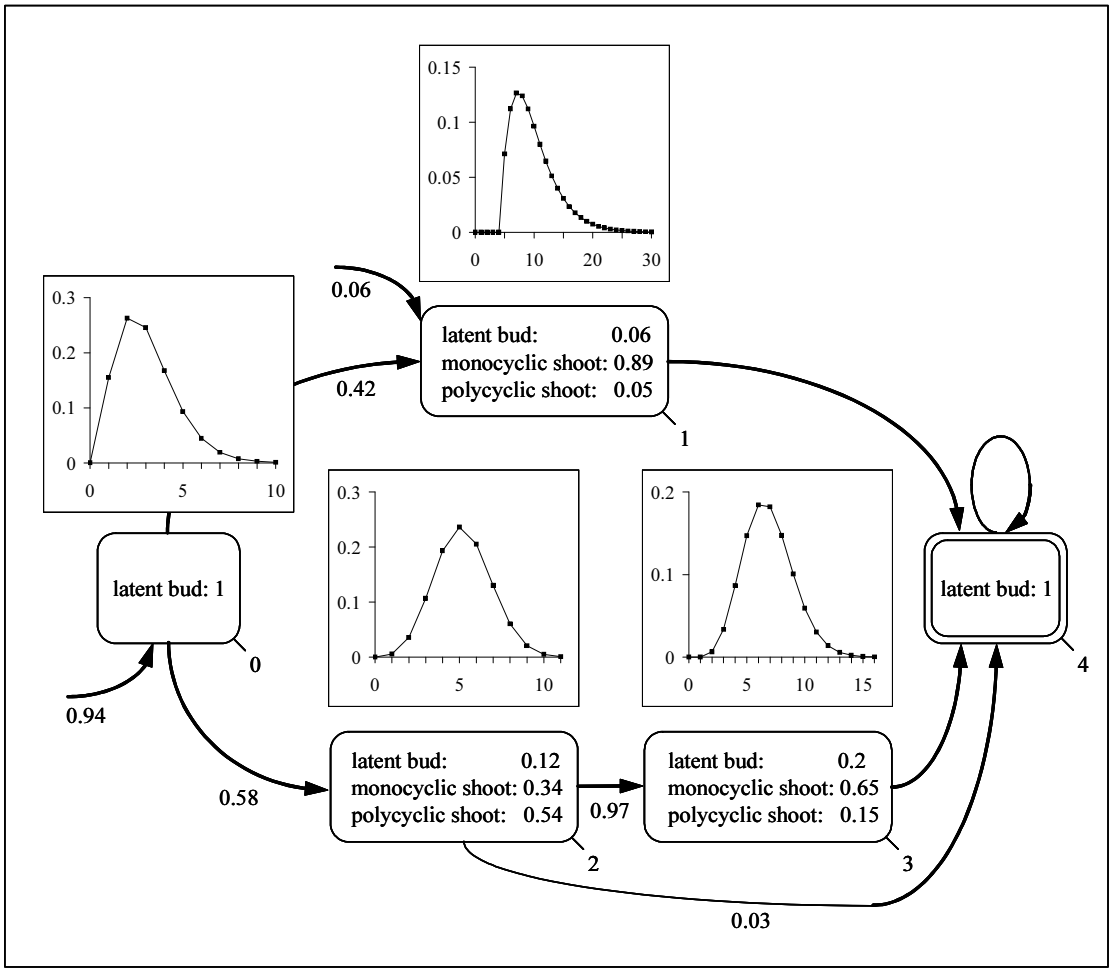

Fig. 6. Red oak: Estimated hidden semi-Markov chain.

The hidden semi-Markov chain estimated from 55 branching sequences is composed of five states (Fig. 6). The initial state is followed by either state 1 or by states 2 and 3. This corresponds to two categories of growth unit: (i) those that possess only one-year-delayed monocyclic offspring shoots (state 1) and (ii) those that possess one-year-delayed polycyclic offspring shoots towards the top and one-year-delayed monocyclic offspring shoots lower down (succession of states 2 and 3); see Heuret et al. (2003) for a more detailed biological discussion. Most individuals are unambiguous regarding the choice between state 1 and states 2 and 3.

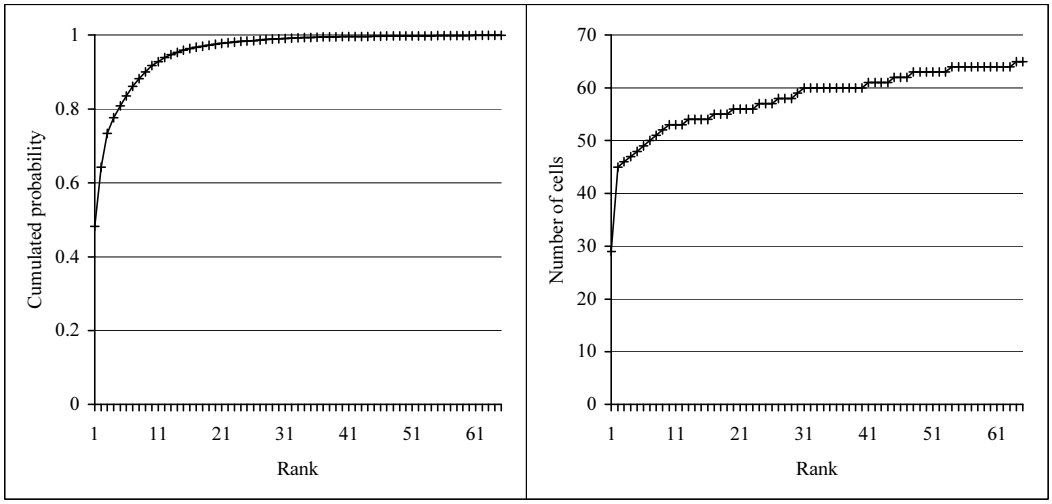

Fig. 7. Red oak: Characteristics of state sequences.

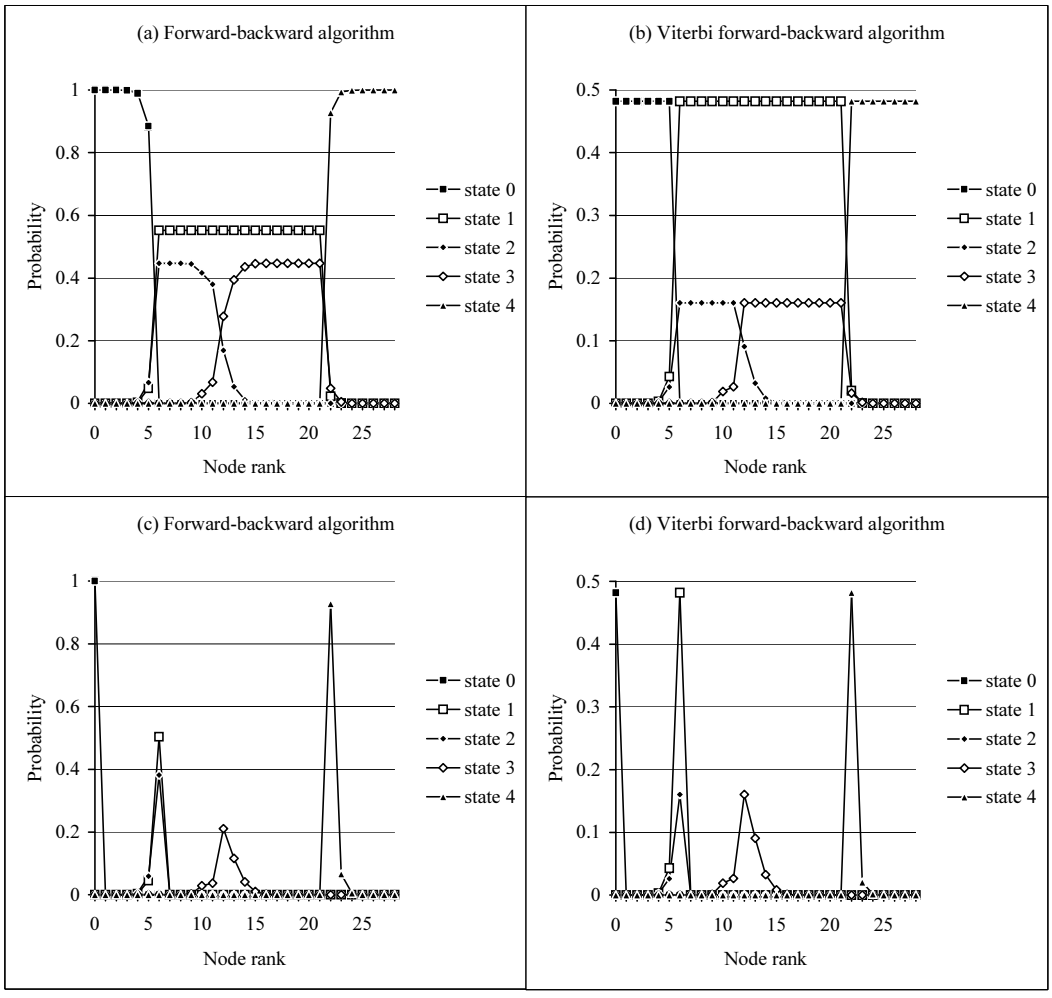

Fig. 8. Red oak: State and state change profiles.

We chose one of the ambiguous individuals. There are 840 possible state sequences for the selected individual but the cumulated posterior probability of the top 65 most probable state sequences exceeds 0.999 (Fig. 7). The observed sequence is:

```
0 0 0 0 0 0 1 1 1 2 1 2 1 1 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0
```

The top 10 most probable state sequences are:

```
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 4 4 4 4 4 4  (0.482  0.482  29)
0 0 0 0 0 0 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4  (0.16   0.642  45)
0 0 0 0 0 0 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4  (0.091  0.733  46)
0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 4 4 4 4 4 4  (0.043  0.776  47)
0 0 0 0 0 0 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4  (0.033  0.809  48)
0 0 0 0 0 0 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4  (0.026  0.835  49)
0 0 0 0 0 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4  (0.026  0.861  50)
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 4 4 4 4 4 4  (0.021  0.882  51)
0 0 0 0 0 0 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4  (0.019  0.901  52)
0 0 0 0 0 0 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4  (0.017  0.918  53)
```

To illustrate the behaviour of the forward-backward algorithm for sampling state sequences, we generated 1000 state sequences. Among these 1000 state sequences, we obtained 497 times the most probable state sequence, 74 other state sequences visiting state 1 and 429 state sequences visiting states 2 and 3. While the state profiles computed by the forward-backward algorithm focus on the relative frequencies of these two categories of state sequences (Fig. 8 ac), the state profiles computed by the Viterbi forward-backward algorihm focus on the second most probable state sequence visiting state 2 and 3 as a structurally different alternative to the most probable state sequence visiting state 1 (Fig. 8 bd). While the marginal probabilities for the two main paths (state 1 versus states 2 and 3) within the model are quite close (Fig. 8 a), the posterior probability of the most probable state sequence visiting state 1 is about three times the posterior probability of the second most probable state sequence visiting states 2 and 3 (Fig. 8 b).

8 Concluding remarks

The Corsican pine and the red oak examples illustrate two contrasted situations. While in the Corsican pine example, state sequence uncertainty reduces to uncertainty concerning state change times in a fixed succession of states, in the red oak example, the uncertainty concerns both the state change times and the succession of states. We chose simple examples (simple non-ergodic models estimated from samples of short sequences) in order to illustrate the complementarity between the proposed methods, including the generalized Viterbi algorithm for computing the top L most probable state sequences. One should be aware that in other contexts where ergodic models are estimated from long sequences, only state (or state change) profiles can be used for exploring the state sequence space structure.

Exploring the state sequence space may consist in extracting state sequences, whose probabilities are not negligible, that differ from the most probable state sequence and from other state sequences previously extracted on the basis of the same criteria. These structural differences between state sequences are naturally highlighted in the state profile computed by the Viterbi forward-backward algorithm. The output of the Viterbi forward-backward algorithm can be viewed as the superposition (from the less probable to the most probable) of all the state sequences in a $J \times \tau$ array. Hence, segments of individual state sequences and in particular the entire most probable state sequence are directly apparent in the state profiles computed by the Viterbi forward-backward algorithm while individual state sequences are not apparent in the states profiles computed by the forward-backward algorithm. Since the superposition rule is different for state profiles and the two types of state change profiles (corresponding to state entering or state exit) computed by the Viterbi forward-backward algorithm, state profiles and state change profiles may highlight different structures in state sequences.

The entropy of the state sequence that explains an observed sequence, which is the most obvious measure of state sequence uncertainty, can be computed as a byproduct of the forward-backward algorithm. Hence the Viterbi forward-backward algorithm and the forward-backward algorithm play complementary roles for exploring the state sequence space structure. We feel that both the Viterbi forward-backward algorithm, the generalized Viterbi algorithm and the forward-backward algorithm for sampling state sequences should be standard in hidden Markovian model implementation where currently only the forward-backward and Viterbi algorithms are typically implemented.

Methods for exploring the state sequence space in the case of hidden hybrid Markov/semi-Markov chains are fully implemented in the AMAPmod software which is freely available at http://amap.cirad.fr.

Acknowledgments

The author thanks Yves Caraglio and Céline Meredieu for providing the Corsican pine data, Patrick Heuret for providing the red oak data, and Jean-Baptiste Durand for helpful comments.

Appendix A: Pseudo-code of the Viterbi forward-backward algorithm

The following convention is adopted in the presentation of the pseudo-code of the Viterbi forward-backward algorithm: The operator ':=' denotes the assignment of a value to a variable (or the initialization of a variable with a value) and the working variables $\text{Forward}_j(t)$, Observ, $\text{StateIn}_j(t+1)$, Transition_{ij}, $\text{Backward}_j(t)$, $\text{Occupancy}_j(u)$ and $\text{Aux}_j(t+1)$ are introduced for this implementation. $\text{Forward}_j(t)$ is used to compute $\alpha_j(t)$ for a semi-Markovian state and $\widetilde{\alpha}_j(t)$ for a Markovian

state, $\text{Backward}_j(t)$ is used to compute $\beta_j(t)$ for a semi-Markovian state and $\widetilde{\beta}_j(t)$ for a Markovian state while $\text{Aux}_j(t+1)$ is used to compute $\xi_j(t+1)$ for a semi-Markovian state and $\widetilde{\xi}_j(t+1)$ for a Markovian state. Transition_{ij} corresponds to p_{ij} for a semi-Markovian state and to $\widetilde{p}_{ij}$ for a Markovian state. This highlights the natural mixing of the forward (respectively backward) recursion for semi-Markovian and Markovian states. The other variables correspond to the quantities already introduced in Section 3.

Forward recursion

for $t := 0$ to $\tau - 1$ **do**
  for $j := 0$ to $J - 1$ **do**
    if state j is semi-Markovian **then**
      $\text{Forward}_j(t) := 0$
      $\text{Observ} := 1$

      if $t < \tau - 1$ **then**
        for $u := 1$ to $\min(t+1, M_j)$ **do**
          $\text{Observ} := \text{Observ}\, b_j(x_{t-u+1})$
          if $u < t+1$ **then**
            if $\text{Observ}\, d_j(u)\, \text{StateIn}_j(t-u+1) > \text{Forward}_j(t)$ **then**
              $\text{Forward}_j(t) := \text{Observ}\, d_j(u)\, \text{StateIn}_j(t-u+1)$
            end if
          else $\{u = t+1\}$
            if $\text{Observ}\, d_j(t+1)\, \pi_j > \text{Forward}_j(t)$ **then**
              $\text{Forward}_j(t) := \text{Observ}\, d_j(t+1)\, \pi_j$
            end if
          end if
        end for

      else $\{t = \tau - 1\}$
        for $u := 1$ to $\min(\tau, M_j)$ **do**
          $\text{Observ} := \text{Observ}\, b_j(x_{\tau-u})$
          if $u < \tau$ **then**
            if $\text{Observ}\, D_j(u)\, \text{StateIn}_j(\tau-u) > \text{Forward}_j(\tau-1)$ **then**
              $\text{Forward}_j(\tau-1) := \text{Observ}\, D_j(u)\, \text{StateIn}_j(\tau-u)$
            end if
          else $\{u = \tau\}$
            if $\text{Observ}\, D_j(\tau)\, \pi_j > \text{Forward}_j(\tau-1)$ **then**
              $\text{Forward}_j(\tau-1) := \text{Observ}\, D_j(\tau)\, \pi_j$
            end if
          end if
        end for
      end if

```
      else {state j is Markovian}
        if t = 0 then
          Forward_j(0) := b_j(x_0) π_j
        else {t > 0}
          Forward_j(t) := b_j(x_t) StateIn_j(t)
        end if
      end if
    end for

    if t < τ − 1 then
      for j := 0 to J − 1 do
        StateIn_j(t + 1) := 0
        for i := 0 to J − 1 do
          if Transition_ij Forward_i(t) > StateIn_j(t + 1) then
            StateIn_j(t + 1) := Transition_ij Forward_i(t)
          end if
        end for
      end for
    end if
  end for
```

In a first step, the quantities $\text{Forward}_j(t) = \max_{s_0,\ldots,s_{t-1}} P(S_{t+1} \neq j,\ S_t = j,\ S_0^{t-1} = s_0^{t-1}, X_0^t = x_0^t)$ are computed for each semi-Markovian state and the quantities $\text{Forward}_j(t) = \max_{s_0,\ldots,s_{t-1}} P(S_t = j,\ S_0^{t-1} = s_0^{t-1}, X_0^t = x_0^t)$ are computed for each Markovian state. In a second step, the quantities $\text{StateIn}_j(t+1)$ are computed. For a semi-Markovian state j, $\text{StateIn}_j(t+1) = \max_{s_0,\ldots,s_t} P(S_{t+1} = j,\ S_t \neq j, S_0^t = s_0^t, X_0^t = x_0^t)$ while for a Markovian state j, $\text{StateIn}_j(t+1) = \max_{s_0,\ldots,s_t} P(S_{t+1} = j, S_0^t = s_0^t, X_0^t = x_0^t)$. The quantities $\text{Forward}_j(t)$ and $\text{StateIn}_j(t+1)$ should be stored for each time t and each state j.

Backward recursion

```
  for j := 0 to J − 1 do
    Backward_j(τ − 1) := 1
    γ_j(τ − 1) := Forward_j(τ − 1)
  end for

  for t := τ − 2 to 0 do
    for j := 0 to J − 1 do
      if state j is semi-Markovian then
        Observ := 1
        for u := 1 to min(τ − 1 − t, M_j) do
          Observ := Observ b_j(x_{t+u})
          if u < τ − 1 − t then
            Occupancy_j(u) := Backward_j(t + u) Observ d_j(u)
```

```
        else {u = τ − 1 − t}
          Occupancy_j(u) := Observ D_j(τ − 1 − t)
        end if
      end for

      Aux_j(t + 1) := 0
      for u := min(τ − 1 − t, M_j) to 1 do
        if Occupancy_j(u) > Aux_j(t + 1) then
          Aux_j(t + 1) := Occupancy_j(u)
        end if
        if Aux_j(t + 1) StateIn_j(t + 1) > γ_j(t + u) then
          γ_j(t + u) := Aux_j(t + 1) StateIn_j(t + 1)
        end if
      end for

    else {state j is Markovian}
      Aux_j(t + 1) := Backward_j(t + 1) b_j(x_{t+1})
    end if
  end for

  for j := 0 to J − 1 do
    Backward_j(t) := 0
    for k := 0 to J − 1 do
      if Aux_k(t + 1) Transition_jk > Backward_j(t) then
        Backward_j(t) := Aux_k(t + 1) Transition_jk
      end if
    end for
    γ_j(t) := Backward_j(t) Forward_j(t)
  end for
end for
```

In a first step, for each semi-Markovian state j, the auxiliary quantities $\text{Occupancy}_j(u) = \max_{s_{t+u+1},\ldots,s_{\tau-1}} P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1},\ S_{t+u+1}^{\tau-1} = s_{t+u+1}^{\tau-1}, S_{t+u+1} \neq j, S_{t+u-v} = j, v = 0,\ldots,u-2 | S_{t+1} = j, S_t \neq j\right)$ are computed for increasing values of u, the auxiliary quantity $\text{Aux}_j(t+1)$ is then computed for decreasing values of u and the quantity $\text{Aux}_j(t+1)\,\text{StateIn}_j(t+1) = \max_{v \geq u} \max_{s_0,\ldots,s_t} \max_{s_{t+v+1},\ldots,s_{\tau-1}} P\left(S_0^t = s_0^t,\ S_t \neq j,\right.$ $\left. S_{t+w} = j, w = 1,\ldots,v, S_{t+v+1} \neq j, S_{t+v+1}^{\tau-1} = s_{t+v+1}^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right)$ is compared to the current evaluation of $\gamma_j(t+u) = \max_{s_0,\ldots,s_{t+u-1}} \max_{s_{t+u+1},\ldots,s_{\tau-1}} P\left(S_0^{t+u-1} = s_0^{t+u-1},\right.$ $\left. S_{t+u} = j, S_{t+u+1}^{\tau-1} = s_{t+u+1}^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right)$. Then in the second step, the quantities $\text{Backward}_j(t)$ are extracted. It should be noted that $\gamma_j(t)$ is initialized with $\beta_j(t)\,\alpha_j(t)$. The quantities $\text{Backward}_j(t)$ and $\gamma_j(t)$ should be stored for each time t and each state j while the auxiliary quantities $\text{Occupancy}_j(u)$ need only be stored for each sojourn time u and the auxiliary quantities $\text{Aux}_j(t+1)$ need only be stored for each state j.

Supplementary computation for $t = 0$ (computation of $\xi_j(0)\,\pi_j$)

for $j := 0$ to $J - 1$ **do**
  if state j is semi-Markovian **then**
    $\text{Observ} := 1$
    for $u := 1$ to $\min(\tau, M_j)$ **do**
      $\text{Observ} := \text{Observ}\, b_j(x_{u-1})$
      if $u < \tau$ **then**
        $\text{Occupancy}_j(u) := \text{Backward}_j(u-1)\,\text{Observ}\, d_j(u)$
      else $\{u = \tau\}$
        $\text{Occupancy}_j(u) := \text{Observ}\, D_j(\tau)$
      end if
    end for

    $\text{Aux}_j(0) := 0$
    for $u := \min(\tau, M_j)$ to 1 **do**
      if $\text{Occupancy}_j(u) > \text{Aux}_j(0)$ **then**
        $\text{Aux}_j(0) := \text{Occupancy}_j(u)$
      end if
      if $\text{Aux}_j(0)\,\pi_j > \gamma_j(u-1)$ **then**
        $\gamma_j(u-1) := \text{Aux}_j(0)\,\pi_j$
      end if
    end for
  end if
end for

For numerical stability, a log transformation should be applied to all the model parameters. As a consequence, all the products are turned into sums in a practical computer implementation.

Appendix B: Pseudo-code of the generalized Viterbi algorithm

For each time t, each state j and each rank n, three backpointers must be recorded, the first $\text{PreviousState}_j^n(t)$ giving the optimal preceding state, the second $\text{PreviousRank}_j^n(t)$ giving the associated rank and the third $\text{Occupancy}_j^n(t)$ giving the optimal preceding time of transition from this preceding state. We need also to introduce the working variables $\text{State}_j^n(t+1)$ and $\text{Rank}_j^n(t+1)$.

Forward recursion

$L_0 := 1$

for $t := 0$ to $\tau - 1$ **do**
  for $j := 0$ to $J - 1$ **do**

if state j is semi-Markovian **then**
  $\text{Observ}(1) := b_j(x_t)$
  for $u := 2$ to $\min(t+1, M_j)$ **do**
    $\text{Observ}(u) := \text{Observ}(u-1)\, b_j(x_{t-u+1})$
  end for
  for $u := 1$ to $\min(t+1, M_j)$ **do**
    $r(u) := 1$
  end for

  for $n := 1$ to L_t **do**
    $\text{Forward}_j^n(t) := 0$
    if $t < \tau - 1$ **then**
      for $u := 1$ to $\min(t+1, M_j)$ **do**
        if $u < t+1$ **then**
          if $\text{Observ}(u)\, d_j(u)\, \text{StateIn}_j^{r(u)}(t-u+1) > \text{Forward}_j^n(t)$ **then**
            $\text{Forward}_j^n(t) := \text{Observ}(u)\, d_j(u)\, \text{StateIn}_j^{r(u)}(t-u+1)$
            $\text{PreviousState}_j^n(t) := \text{State}_j^{r(u)}(t-u+1)$
            $\text{PreviousRank}_j^n(t) := \text{Rank}_j^{r(u)}(t-u+1)$
            $\text{Occupancy}_j^n(t) := u$
          end if
        else $\{u = t+1\}$
          if $\text{Observ}(t+1)\, d_j(t+1)\, \pi_j > \text{Forward}_j^n(t)$ **then**
            $\text{Forward}_j^n(t) := \text{Observ}(t+1)\, d_j(t+1)\, \pi_j$
            $\text{Occupancy}_j^n(t) := t+1$
          end if
        end if
      end for

    else $\{t = \tau - 1\}$
      for $u := 1$ to $\min(\tau, M_j)$ **do**
        if $u < \tau$ **then**
          if $\text{Observ}(u)\, D_j(u)\, \text{StateIn}_j^{r(u)}(\tau-u) > \text{Forward}_j^n(\tau-1)$ **then**
            $\text{Forward}_j^n(\tau-1) := \text{Observ}(u)\, D_j(u)\, \text{StateIn}_j^{r(u)}(\tau-u)$
            $\text{PreviousState}_j^n(\tau-1) := \text{State}_j^{r(u)}(\tau-u)$
            $\text{PreviousRank}_j^n(\tau-1) := \text{Rank}_j^{r(u)}(\tau-u)$
            $\text{Occupancy}_j^n(\tau-1) := u$
          end if
        else $\{u = \tau\}$
          if $\text{Observ}(\tau)\, D_j(\tau)\, \pi_j > \text{Forward}_j^n(\tau-1)$ **then**
            $\text{Forward}_j^n(\tau-1) := \text{Observ}(\tau)\, D_j(\tau)\, \pi_j$
            $\text{Occupancy}_j^n(\tau-1) := \tau$
          end if
        end if
      end for
    end if

$r(\text{Occupancy}_j^n(t)) := r(\text{Occupancy}_j^n(t)) + 1$
end for

else {state j is Markovian}
for $n := 1$ to L_t **do**
if $t = 0$ **then**
$\text{Forward}_j^n(0) := b_j(x_0)\,\pi_j$
else $\{t > 0\}$
$\text{Forward}_j^n(t) := b_j(x_t)\,\text{StateIn}_j^n(t)$
$\text{PreviousState}_j^n(t) := \text{State}_j^n(t)$
$\text{PreviousRank}_j^n(t) := \text{Rank}_j^n(t)$
end if
$\text{Occupancy}_j^n(t) := 1$
end for
end if

for $n := L_t + 1$ to L **do**
$\text{Forward}_j^n(t) := 0$
end for
end for

if $t < \tau - 1$ **then**
if $J\,L_t < L$ **then**
$L_{t+1} := J\,L_t$
else $\{J\,L_t \geq L\}$
$L_{t+1} := L$
end if

for $j := 0$ to $J - 1$ **do**
for $i := 0$ to $J - 1$ **do**
$r(i) := 1$
end for

for $n := 1$ to L_{t+1} **do**
$\text{StateIn}_j^n(t+1) := 0$
for $i := 0$ to $J - 1$ **do**
if $\text{Transition}_{ij}\,\text{Forward}_i^{r(i)}(t) > \text{StateIn}_j^n(t+1)$ **then**
$\text{StateIn}_j^n(t+1) := \text{Transition}_{ij}\,\text{Forward}_i^{r(i)}(t)$
$\text{State}_j^n(t+1) := i$
$\text{Rank}_j^n(t+1) := r(i)$
end if
end for

$r(\text{State}_j^n(t+1)) := r(\text{State}_j^n(t+1)) + 1$

```
        end for

        for n := L_{t+1} + 1 to L do
          StateIn_j^n(t+1) := 0
        end for
      end for
    end if
  end for
```

In a first step, the quantities $\text{Forward}_j^n(t)$ are computed for each state j and each rank n. In a second step, the quantities $\text{StateIn}_j^n(t+1)$ are computed for each state j and each rank n. The rank propagation decomposes into two steps since the ranks attached to $\text{Forward}_j^n(t)$ are deduced from the ranks attached to $\left(\text{StateIn}_j^{r(u)}(t-u+1);\right.$ $\left.u := 1, \ldots, \min(t+1, M_j)\right)$ and the ranks attached to $\text{StateIn}_j^n(t+1)$ are deduced from the ranks attached to $\left(\text{Forward}_i^{r(i)}(t);\ i = 0, \ldots, J-1\right)$. The increasing number of partial state sequences for small values of t is taken into account by initializing at 0 $\text{Forward}_j^n(t)$ for each state j and for $n = L_t + 1, \ldots, L$, and $\text{StateIn}_j^n(t+1)$ for each state j and for $n = L_{t+1} + 1, \ldots, L$. The quantities $\text{Forward}_j^n(t)$ should be stored for each state j and each rank n while the quantities $\text{PreviousState}_j^n(t)$, $\text{PreviousRank}_j^n(t)$, $\text{Occupancy}_j^n(t)$, $\text{StateIn}_j^n(t+1)$, $\text{State}_j^n(t+1)$ and $\text{Rank}_j^n(t+1)$ should be stored for each time t, each state j and each rank n.

Backtracking

```
for j := 0 to J − 1 do
  r(j) := 1
end for

for n := 1 to L_{τ−1} do
  MaxForward := 0
  for j := 0 to J − 1 do
    if Forward_j^{r(j)}(τ − 1) > MaxForward then
      MaxForward := Forward_j^{r(j)}(τ − 1)
      s̃_{τ−1}^n := j
    end if
  end for

  m := r(s̃_{τ−1}^n)
  r(s̃_{τ−1}^n) := r(s̃_{τ−1}^n) + 1
  t := τ − 1
  repeat
    j := s̃_t^n
    u := Occupancy_j^m(t)
    for v := 1 to u − 1 do
      s̃_{t−v}^n := j
```

end for
if $t \geq u$ **then**
$\tilde{s}_{t-u}^{n} := \text{PreviousState}_{j}^{m}(t)$
$m := \text{PreviousRank}_{j}^{m}(t)$
end if
$t := t - u$
until $t \geq 0$
end for

For numerical stability, a log transformation should be applied to all the model parameters. As a consequence, all the products are turned into sums in a practical computer implementation.

References

Brushe, G.D., Mahony, R.E., Moore, J.B., 1998. A soft output hybrid algorithm for ML/MAP sequence estimation. IEEE Trans. Inf. Theory 44(7), 3129-3134.

Burge, C., Karlin, S., 1997. Prediction of complete gene structures in human genomic DNA. J. Mol. Biol. 268, 78-94.

Cawley, S.L., Pachter, L., 2003. HMM sampling and applications to gene finding and alternative splicing. Bioinformatics 19 (Suppl. 2), ii36-ii41.

Chib, S., 1996. Calculating posterior distributions and modal estimates in Markov mixture models. J. Econometrics 75, 79-97.

Churchill, G.A., 1989. Stochastic models for heterogeneous DNA sequences. Bull. Math. Biol. 51(1), 79-94.

Cover, T.M., Thomas, J.A., 1991. Elements of Information Theory. Wiley, New York.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm (with discussion). J. Royal Statist. Soc. Ser. B 39(1), 1-38.

Ephraim, Y., Merhav, N., 2002. Hidden Markov processes. IEEE Trans. Inf. Theory 48(6), 1518-1569.

Ferguson, J.D., 1980. Variable duration models for speech. In: Ferguson, J.D. (Ed.), Proceedings of the Symposium on the Applications of Hidden Markov Models to Text and Speech, Princeton, New Jersey, 143-179.

Foreman, L.A., 1993. Generalization of the Viterbi algorithm. IMA J. Math. Appl. Bus. Ind. 4, 351-367.

Guédon, Y., 2003. Estimating hidden semi-Markov chains from discrete sequences. J. Comput. Graphical Statist. 12(3), 604-639.

Guédon, Y., 2005. Hidden hybrid Markov/semi-Markov chains. Comput. Statist. Data Anal. 49(3), 663-688.

Guédon, Y., Barthélémy, D., Caraglio, Y., Costes, E., 2001. Pattern analysis in branching and axillary flowering sequences. J. Theor. Biol. 212, 481-520.

Hernando, D., Crespi, V., Cybenko, G., 2005. Efficient computation of the hidden Markov model entropy for a given observation sequence. IEEE Trans. Inf. Theory 51(7), 2681-2685.

Heuret, P., Guédon, Y., Guérard, N., Barthélémy, D., 2003. Analysing branching pattern in plantations of young red oak trees (Quercus rubra L., Fagaceae). Ann. Bot. 91(4), 479-492.
Kulkarni, V.G., 1995. Modeling and Analysis of Stochastic Systems. Chapman & Hall, London.
Lukashin, A.V., Borodovsky, M., 1998. GeneMark.hmm: new solutions for gene finding. Nucleic Acids Res. 26(4), 1107-1115.
McLachlan, G.J., Krishnan, T., 1997. The EM Algorithm and Extensions. Wiley, New York.
Sansom, J., Thomson, P., 2001. Fitting hidden semi-Markov models to breakpoint rainfall data. J. Applied Proba. 38A, 142-157.
Schmidler, S.C., Liu, J.S., Brutlag, D.L., 2000. Bayesian segmentation of protein secondary structure. J. Comput. Biol. 7(1/2), 233-248.
Scott, S.L., 2002. Bayesian methods for hidden Markov models: Recursive computing in the 21st century. J. Amer. Statist. Assoc. 97(457), 337-351.
Tanner, M.A., 1996. Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions, 3rd Edition. Springer, New York.