Hidden hybrid Markov/semi-Markov chains.

Yann Guédon

Hidden hybrid Markov/semi-Markov chains

Yann Guédon

Unité Mixte de Recherche CIRAD/CNRS/INRA/IRD/Université Montpellier II
Botanique et Bioinformatique de l'Architecture des Plantes, TA 40/PS2,
34398 Montpellier Cedex 5, France

Abstract

Models that combine Markovian states with implicit geometric state occupancy distributions and semi-Markovian states with explicit state occupancy distributions, are investigated. This type of model retains the flexibility of hidden semi-Markov chains for the modeling of short or medium size homogeneous zones along sequences but also enables the modeling of long zones with Markovian states. The forward-backward algorithm, which in particular enables to implement efficiently the E-step of the EM algorithm, and the Viterbi algorithm for the restoration of the most likely state sequence are derived. It is also shown that macro-states, i.e. series-parallel networks of states with common observation distribution, are not a valid alternative to semi-Markovian states but may be useful at a more macroscopic level to combine Markovian states with semi-Markovian states. This statistical modeling approach is illustrated by the analysis of branching and flowering patterns in plants.

Key words: Forward-backward algorithm; Hidden Markov chain; Hidden semi-Markov chain; Macro-state; Plant structure analysis; Smoothing algorithm; Viterbi algorithm.

1 Introduction

One drawback of hidden semi-Markov chains is the time complexity of the main algorithms (forward-backward and Viterbi) which is quadratic in the worst case in terms of sequence length instead of linear for simple hidden Markov chains (Guédon, 2003). This may limit the potential application of this type of model for the analysis of sequences including long homogeneous zones (for instance some intronic zones in DNA sequences). In some cases, it was also noted that the lengths of some zones of interest in different biological applications are approximately geometrically distributed. This is the case for intronic zones in the human genome (Burge and Karlin,

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

1997) and non-coding zones in bacterial genomes (Lukashin and Borodovsky, 1998). In this paper, we show that the lengths of some branching/axillary flowering zones are also geometrically distributed. For these two reasons, it is interesting to develop efficient computational methods for hidden hybrid Markov/semi-Markov chains, i.e. models where some states are semi-Markovian (with an explicit state occupancy distribution) while the others are simply Markovian (with an implicit geometric state occupancy distribution). It should be noted that hidden semi-Markov chains as defined in Guédon (2003) can be seen as hidden hybrid Markov/semi-Markov chains since the absorbing states are by nature Markovian. In this paper, the Markovian nature of states is no longer restricted to absorbing states since nonabsorbing Markovian states can now be defined. In the context of the application to gene finding, as illustrated by systems such as Genie (Kulp et al., 1996), GENSCAN (Burge and Karlin, 1997) and GeneMark.hmm (Lukashin and Borodovsky, 1998), the incorporation of non-absorbing Markovian states is critical since the distributions of the lengths of the longest homogeneous zones are approximately geometric.

For combining Markovian states with semi-Markovian states in hidden Markov models, one can either consider a semi-Markovian framework where the occupancy distributions of some non-absorbing states are constrained to be geometric or a Markovian framework where some states are replaced by series-parallel networks of states with common observation distribution; see Cook and Russell (1986) and Durbin et al. (1998). This latter approach is similar to the 'method of stages' well-known in queueing system theory (Kleinrock, 1975). The occupancy distributions of the macro-states defined in this way are built from the implicit geometric occupancy distributions of the elementary Markovian states. These geometric distributions are combined by convolution in the case of states in series, while they are combined by mixture in the case of (series of) states in parallel. A key property of this macro-state approach is that the conditional independence assumptions within the process are preserved with respect to hidden Markov chains. Hence, the hidden Markov chain algorithms for parameter estimation, and for computing most likely state sequences still apply. In this paper, we show that macro-states are not a valid alternative to semi-Markovian states but may be useful at a more macroscopic level to combine Markovian states with semi-Markovian states.

Burge (1997) proposed adaptations of the forward-backward algorithm and the Viterbi algorithm for a specific class of hidden hybrid Markov/semi-Markov chains where Markovian and semi-Markovian states alternated. The proposed algorithms entailed notably to consider three successive states in the recursions and relied fairly heavily on the specific structure of the model considered. We propose in this paper a general solution for the main algorithms of hidden hybrid Markov/semi-Markov chains. The main technical outcome of this work is that algorithms for hidden hybrid Markov/semi-Markov chains are basically the juxtaposition of the basic algorithms for hidden Markov chains (Devijver, 1985; Rabiner, 1989; Ephraim and Merhav, 2002) and hidden-semi Markov chains (Guédon, 2003) with simple connections in the case of transitions from a Markovian state to a semi-Markovian state or vice versa. As a consequence, the forward-backward an Viterbi algorithms for hidden hybrid Markov/semi-Markov chains keep the time complexity of the forward-backward

and Viterbi algorithms for simple hidden Markov chains in the case of Markovian states.

The remainder of this paper is organized as follows. Discrete hidden hybrid Markov/semi-Markov chains are formally defined in Section 2. Marco-states are discussed as a possible alternative to semi-Markovian states in Section 3. The estimation of a hidden hybrid Markov/semi-Markov chain from discrete sequences based on the application of the EM algorithm and the associated forward-backward algorithm is presented in Section 4. The Viterbi algorithm for the restoration of the most likely state sequence is presented in Section 5. The resulting data analysis methodology is illustrated in Section 6 by the reanalysis of branching and flowering patterns on apricot tree growth units originally analyzed by a hidden semi-Markov chain. Section 7 consists of concluding remarks.

2 Discrete 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

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

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

For a nonabsorbing Markovian state j, we have

$$P\left(S_{t+1} = k | S_{t+1} \neq j, S_t = j\right) = \frac{\widetilde{p}_{jk}}{1 - \widetilde{p}_{jj}},$$

and the implicit state occupancy distribution is the '1-shifted' geometric distribution with parameter $1 - \widetilde{p}_{jj}$

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

Hybrid Markov/semi-Markov chains can be seen as a sub-class of semi-Markov chains where the occupancy distributions of some nonabsorbing states are constrained to be geometric distributions. For a nonabsorbing state, it is possible to adopt a semi-Markovian parameterization of a Markovian state (Burge 1997). We did not adopt this solution since it cannot be transposed to absorbing states. Furthermore, we will show in Sections 4.1 and 5 that the parameterization chosen leads to simple algorithmic solutions both for the forward-backward algorithm and the Viterbi algorithm.

A discrete hidden hybrid Markov/semi-Markov chain can be viewed as a pair of stochastic processes $\{S_t, X_t\}$ where the discrete 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 $X_t = f\left(S_t\right)$). Since the mapping f is such that $f\left(j\right) = f\left(k\right)$ 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 $\{X_t\}$.

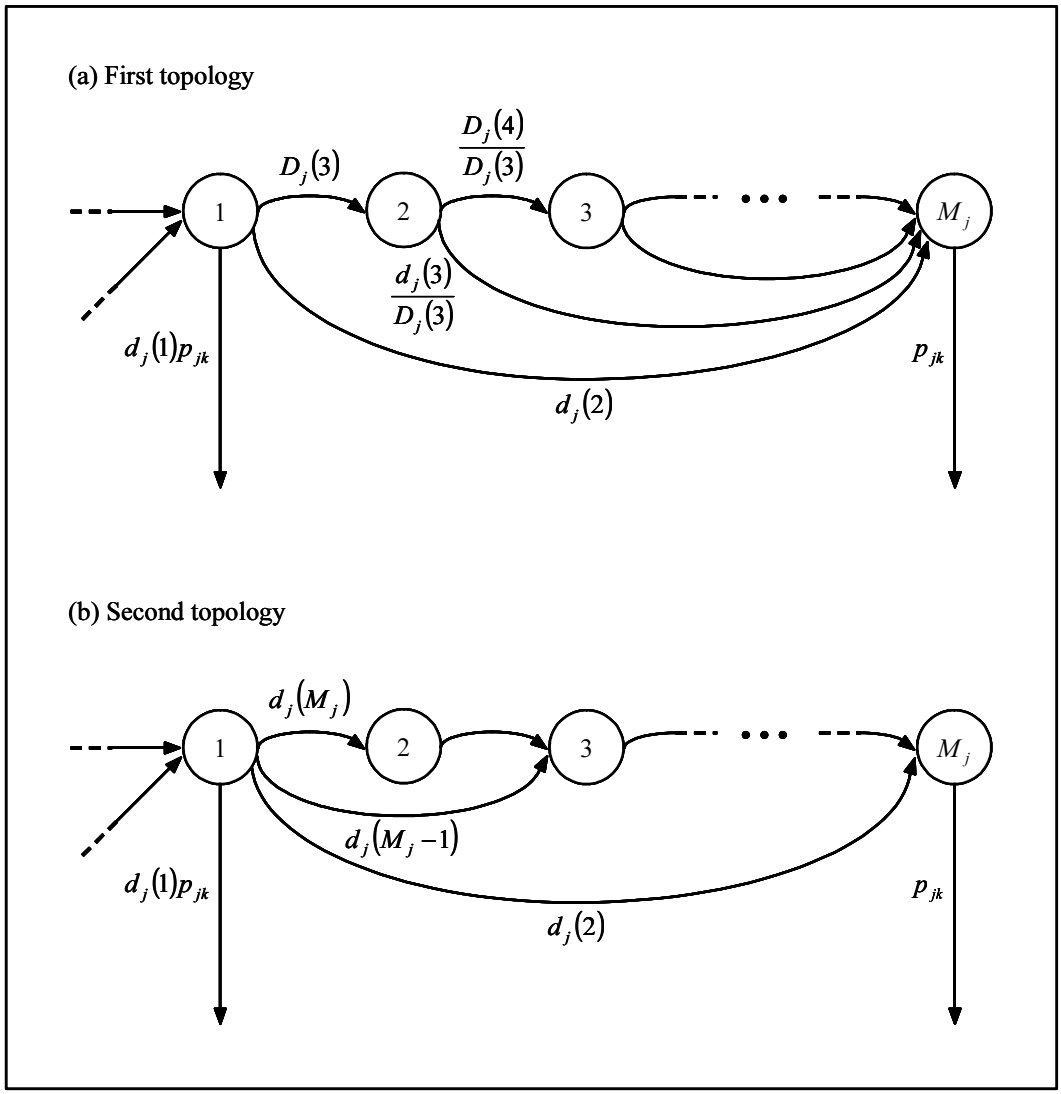

Fig. 1. Nonparametric macro-states: the transition probabilities within the macro-states are parameterized in a semi-Markovian manner.

The output process $\{X_t\}$ is related to the hybrid Markov/semi-Markov chain $\{S_t\}$ by the observation (or emission or state dependent) 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 $S_t = s_t$.

Some notations need to be introduced for the remainder of this paper. The observed sequence of length τ, $X_0 = x_0, \ldots, X_{\tau-1} = x_{\tau-1}$ will be abbreviated $X_0^{\tau-1} = x_0^{\tau-1}$ (this convention transposes to the state sequence $S_0^{\tau-1} = s_0^{\tau-1}$). In the estimation framework, θ designates the vector of all parameters.

3 Hidden Markov chains with macro-states

To compare hidden Markov chains with macro-states with hidden semi-Markov chains, it is interesting to consider both nonparametric and parametric definitions of macro-states.

Nonparametric macro-states:

Consider a macro-state made of a series of M_j states with no self-transition. This type of macro-state can model any occupancy distribution supported by $[1, M_j]$ and hence can be compared to a semi-Markovian state. The two topologies presented in Fig. 1 can be proposed (a single preceding macro-state and a single subsequent macro-state are assumed to simplify matters and the transition probabilities within the macro-states are parameterized in a semi-Markovian manner to help the comparison). In the two cases, $2\,(J-2)+M_j-1$ independent transition probabilities are attached to each macro-state (instead of $J-2$ independent transition probabilities and $M_j - 1$ independent occupancy probabilities for a semi-Markovian state with a nonparametric occupancy distribution). The redundancy in the parameters lies in the duplication of the exit transitions in the first and last states due to the specific situation of staying one time step in a given macro-state which cannot be handled in a similar way as the situation of staying more than one time step. The use of this type of macro-state suffers from a major drawback:

- The complexity in space (both for the forward-backward and the Viterbi algorithms) is $O((\sum_j M_j)\tau)$ instead of $O(J\tau)$ for a hidden semi-Markov chain; see Section 4.1 and Guédon (2003). This drastically limits the application of this type of macro-state model to short homogeneous zones.

Other drawbacks are:

- Tying constraints should be imposed between transitions from state 1 and state M_j (to state 1 of a given macro-state $k \neq j$).
- The regularization of macro-state occupancy distributions leads to impose some complex tying constraints across parameters and is far more difficult to manage than in the case of semi-Markovian states; see the proposal of different methods for the regularization of the occupancy distributions of semi-Markovian states in Guédon (2003).
- The redimensioning of a macro-state between two iterations of the estimation

procedure cannot be handled simply while the size of the support of the explicit occupancy distribution in the case of a semi-Markovian state may be reduced simply if the tail probability (beyond a given value) tends towards zero. This is further penalizing since the initial number of elementary states of a macro-state cannot be underestimated and should be overestimated.

It should also be noted that the naive implementation of a macro-state model entail managing a large ($\sum_j M_j \times \sum_j M_j$) but sparse transition probability matrix. Hence, specific implementations of the forward-backward and the Viterbi algorithms should be designed.

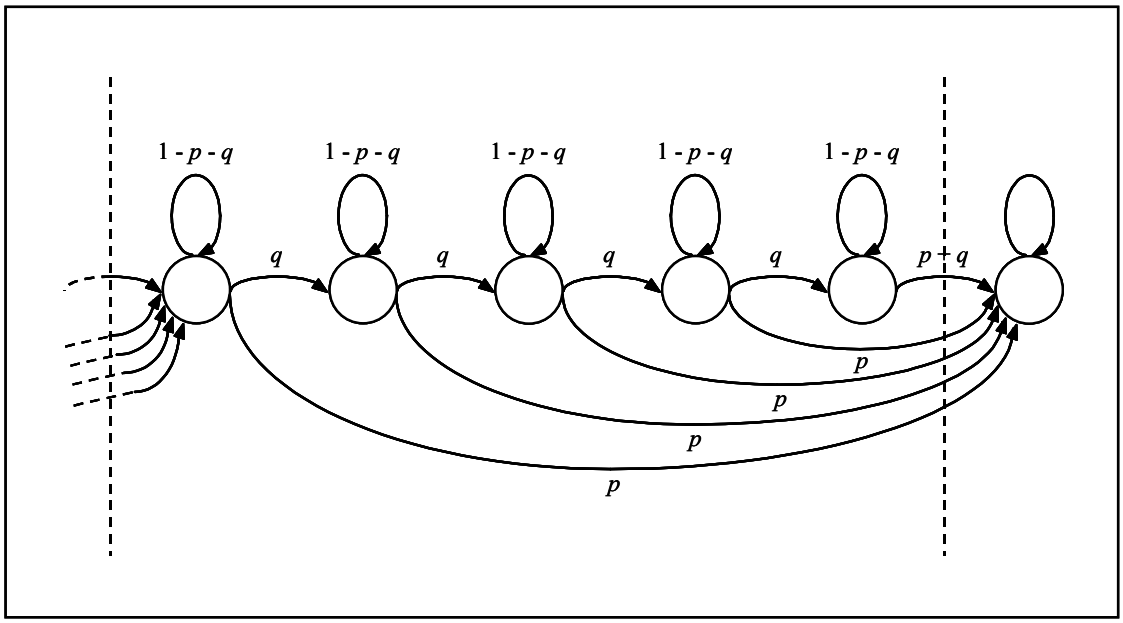

Fig. 2. Parametric macro-state topology.

Moreover, there is no gain in time complexity when using a hidden Markov chain with macro-states compared to the equivalent hidden semi-Markov chain. For instance, the elementary step of both a forward or a backward recursion for a given time step and a given state requires $2\,(J-1)+1+2(M_j-2)$ operations in the case of a macro-state instead of $J-1+M_j$ operations in the case of a semi-Markovian state with generally $J \ll M_j$.

Parametric macro-states:

Let us define the negative binomial distribution with parameters d, r and p, $\mathrm{NB}(d,r,p)$, where d ($d \geq 1$) is a shift parameter that defines the minimum sojourn time in state j, r is a real number ($r > 0$) and $0 < p \leq 1$

$$d_j\,(u) = \binom{u-d+r-1}{r-1} p^r q^{u-d}, \qquad u = d, d+1, \ldots \tag{2}$$

'Parametric' macro-states can be illustrated by the example depicted in Fig. 2 (Guédon, 1992). The macro-state occupancy distribution is defined by two free parameters p and q corresponding respectively to the exit probability and to the next-state transition probability. To simplify matters, we consider a single subsequent macro-state. Let $\{U_k; k = 1, \ldots, r\}$ be a sequence of r mutually independent random variables representing state occupancies with common geometric distribution $\mathrm{NB}(1, 1, p + q)$ and N be the random variable representing the number of U_k summed where N is independent of the U_k. The distribution of $U_1 + \cdots + U_n$ is the n-fold convolution of $\mathrm{NB}(1, 1, p + q)$ and thus is $\mathrm{NB}(n, n, p + q)$. Since the number of states is bounded, the distribution of N is $\mathrm{NB}(1, 1, p/(p + q))$ truncated at $N = r$ in the sense that the survivor function is concentrated at r

$$\begin{aligned} P(N = n) &= \frac{p}{p+q}\left(\frac{q}{p+q}\right)^{n-1}, \qquad n = 1, \ldots, r-1, \\ P(N = r) &= \left(\frac{q}{p+q}\right)^{r-1}. \end{aligned}$$

The resulting macro-state occupancy distribution is the compound (Feller, 1968) or stopped-sum distribution (Johnson et al., 1993)

$$\begin{aligned} & P(U = u) \\ &= \sum_{n=1}^{r} P(N = n)\, P(U_1 + \cdots + U_n = u) \\ &= \sum_{n=1}^{r-1} \frac{p}{p+q}\left(\frac{q}{p+q}\right)^{n-1} \binom{u-1}{n-1} (p+q)^n (1-p-q)^{u-n} \\ &\quad + \left(\frac{q}{p+q}\right)^{r-1} \binom{u-1}{r-1} (p+q)^r (1-p-q)^{u-r} \\ &= \sum_{n=1}^{r-1} \binom{u-1}{n-1} (1-p-q)^{u-n} q^{n-1} p + \binom{u-1}{r-1} (1-p-q)^{u-r} q^{r-1} (p+q), \quad u = 1, 2, \ldots \end{aligned}$$

If $p = 0$, the macro-state occupancy distribution is the negative binomial distribution $\mathrm{NB}(r, r, q)$

$$P(U = u) = \binom{u-1}{r-1} q^r (1-q)^{u-r}, \qquad u = r, r+1, \ldots$$

This case corresponds to a series of r states with the shortcoming that, the minimum time spent in the macro-state is r.

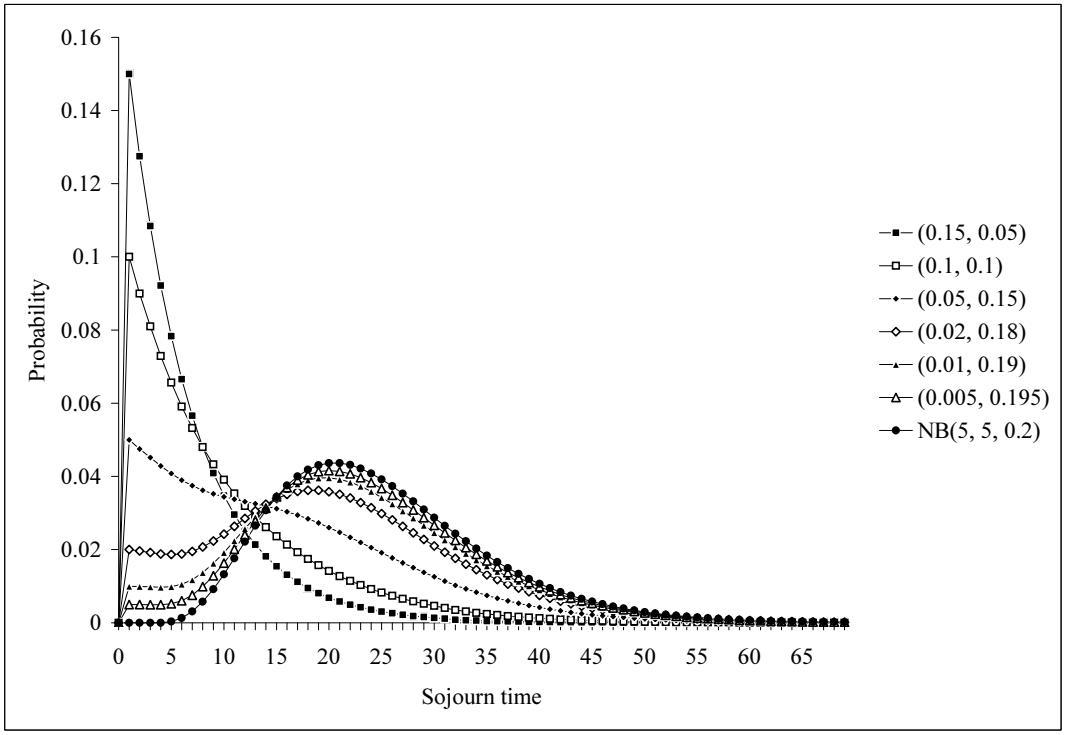

Fig. 3. Parametric macro-state occupancy distributions for different (p, q) values with $p + q = 0.2$.

Let $g_{U_k}(s)$ be the common generating function of the U_k and $g_N(s)$ the generating function of N. If $r \to +\infty$, the generating function of the random sum $U = U_1 + \cdots + U_N$ is the compound function (Feller, 1968)

$$\begin{aligned} g_U(s) &= g_N(g_{U_k}(s)) \\ &= \frac{p}{p+q}\left\{\frac{(p+q)s}{1-(1-p-q)s}\right\} \Big/ \left[1 - \left(\frac{q}{p+q}\right)\left\{\frac{(p+q)s}{1-(1-p-q)s}\right\}\right] \\ &= \frac{ps}{1-(1-p-q)s-qs} \\ &= \frac{ps}{1-(1-p)s}. \end{aligned}$$

Hence, the macro-state occupancy distribution is the geometric distribution $\mathrm{NB}(1, 1, p)$. In practical cases, if the distribution of N is close to a geometric distribution (which means that the weight of the truncation is low), the resulting compound distribution is also close to a geometric distribution.

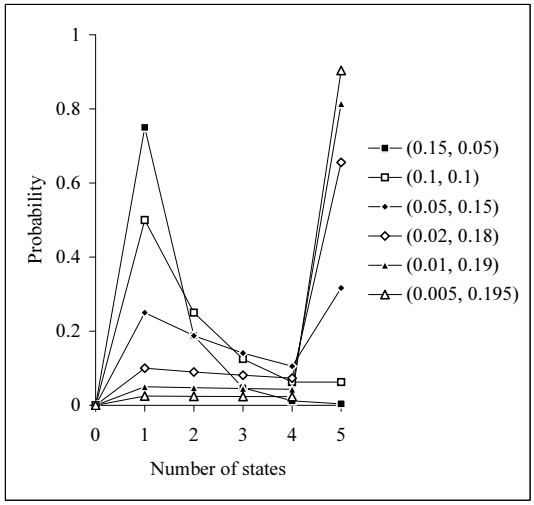

Fig. 4. Distribution of the number of visited states in a parametric macro-state for different (p, q) values with $p + q = 0.2$.

The possible shapes of the macro-state occupancies can be illustrated using the family defined by $r = 5$ and $p + q = 0.2$ and represented in Fig. 3 for selected pairs of (p, q) values. The macro-state occupancy distribution for $(p, q) = (0.15, 0.05)$ is strikingly close to the geometric distribution NB$(1, 1, 0.15)$ since the sum distribution is strikingly close to the geometric distribution NB$(1, 1, 0.75)$ (Fig. 4) while the macro-state occupancy distribution for $(p, q) = (0, 0.2)$ is the negative binomial distribution NB$(5, 5, 0.2)$. The shapes of some of the intermediate distributions seem quite inappropriate for the modeling of zone lengths (Fig. 3). For a fixed number of states r of a macro-state, both the shape and the dispersion are far more constrained than in the case of the explicit occupancy distribution of a semi-Markovian state. These parametric macro-states suffer from a lack of flexibility compared to semi-Markovian states.

4 Estimation of an hidden hybrid Markov/semi-Markov chain

Since macro-states are not a valid alternative to semi-Markovian states, it is necessary to develop efficient algorithms for parameter estimation of hidden hybrid Markov/semi-Markov chains.

With reference to hidden semi-Markov chains (Guédon, 2003), the adaptation of the EM algorithm is straightforward. Recall that hidden hybrid Markov/semi-Markov chains can be seen as a sub-class of hidden semi-Markov chains. Hence, the statement of the estimation problem is unchanged.

Let us consider the complete data where both the outputs $x_0^{\tau-1}$ and the states $s_0^{\tau-1+u}$ of the underlying semi-Markov chain are observed. In the specification of the complete data, the state sequence is completed up to the exit from the state

occupied at time $\tau - 1$, which is assumed to be a semi-Markovian state. If the last visited state is Markovian, it is only necessary to consider the state sequence up to time $\tau - 1$. Let $\theta^{(k)}$ denote the current value of θ at iteration k. In the hidden semi-Markov chain case, the conditional expectation of the complete-data log-likelihood is given by

$$Q(\theta|\theta^{(k)}) = E\left\{\log f\left(S_0^{\tau-1+u}, X_0^{\tau-1}; \theta\right)|X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right\}.$$

This conditional expectation can be rewritten as a sum of terms, each term depending on a given subset of parameters

$$\begin{aligned} Q(\theta|\theta^{(k)}) = & Q_\pi\left(\{\pi_j\}_{j=0}^{J-1}|\theta^{(k)}\right) + \sum_{i=0}^{J-1} Q_p\left(\{p_{ij}\}_{j=0}^{J-1}|\theta^{(k)}\right) \\ & + \sum_{j=0}^{J-1} Q_d\left(\{d_j(u)\}|\theta^{(k)}\right) I(p_{jj} = 0) + \sum_{j=0}^{J-1} Q_b\left(\{b_j(y)\}_{y=0}^{Y-1}|\theta^{(k)}\right). \end{aligned}$$

The reestimation of the initial probabilities, the state occupancy distributions and the observation probabilities remain unchanged in the case of hidden hybrid Markov/semi-Markov chain but it is necessary to consider the two alternative definitions of the terms attached to the transition probabilities. For the transition probabilities attached to a semi-Markovian state i, we obtain

$$Q_p\left(\{p_{ij}\}_{j=0}^{J-1}|\theta^{(k)}\right) = \sum_{j\neq i}\sum_{t=0}^{\tau-2} P\left(S_{t+1} = j, S_t = i|X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right)\log p_{ij}, \qquad (3)$$

and for the transition probabilities attached to a Markovian state i, we obtain

$$Q_{\widetilde{p}}\left(\{\widetilde{p}_{ij}\}_{j=0}^{J-1}|\theta^{(k)}\right) = \sum_{j}\sum_{t=0}^{\tau-2} P\left(S_{t+1} = j, S_t = i|X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right)\log \widetilde{p}_{ij}. \qquad (4)$$

4.1 Forward-backward algorithm

The forward-backward algorithm that implements the E-step of the EM algorithm basically computes the smoothed probabilities $L_j(t) = P\left(S_t = j|X_0^{\tau-1} = x_0^{\tau-1}\right)$ as a function of the index parameter t. Hence, in the vocabulary of state-space models, the forward-backward algorithm is a smoothing algorithm. The time complexity of the forward-backward algorithm is $O(J\tau(J+\tau))$ in the worst case for hidden semi-Markov chains and $O(J^2\tau)$ for hidden Markov chains (and the space complexity is

$O(J\tau)$ in both cases). Our objective is to ensure that the forward-backward algorithm for hidden hybrid Markov/semi-Markov chains retains the time complexity of the forward-backward algorithm for hidden Markov chains in the case of Markovian states.

The proposed forward-backward algorithm is basically a combination of the forward-backward algorithm proposed by Guédon (2003) for hidden semi-Markov chains and the forward-backward algorithm proposed by Devijver (1985) for hidden Markov chains. These two 'parent' forward-backward algorithms share the property of being immune to numerical underflow problems. The algorithm described below directly inherits this property.

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

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

$$\begin{aligned} F_j(t) &= P\left(S_{t+1} \neq j, S_t = j | X_0^t = x_0^t\right) \\ &= \frac{b_j(x_t)}{N_t} \left[\sum_{u=1}^{t} \left\{ \prod_{v=1}^{u-1} \frac{b_j(x_{t-v})}{N_{t-v}} \right\} d_j(u) \sum_{i \neq j} p_{ij} F_i(t-u) \right. \\ &\quad \left. + \left\{ \prod_{v=1}^{t} \frac{b_j(x_{t-v})}{N_{t-v}} \right\} d_j(t+1) \pi_j \right], \end{aligned} \tag{5}$$

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 $t = \tau - 1$

$$\begin{aligned} F_j(\tau-1) &= P\left(S_{\tau-1} = j | X_0^{\tau-1} = x_0^{\tau-1}\right) \\ &= \frac{b_j(x_{\tau-1})}{N_{\tau-1}} \left[\sum_{u=1}^{\tau-1} \left\{ \prod_{v=1}^{u-1} \frac{b_j(x_{\tau-1-v})}{N_{\tau-1-v}} \right\} D_j(u) \sum_{i \neq j} p_{ij} F_i(\tau-1-u) \right. \\ &\quad \left. + \left\{ \prod_{v=1}^{\tau-1} \frac{b_j(x_{\tau-1-v})}{N_{\tau-1-v}} \right\} D_j(\tau) \pi_j \right]. \end{aligned} \tag{6}$$

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 (Devijver, 1985),

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

$$\begin{aligned}\widetilde{F}_j(t) &= P\left(S_t = j | X_0^t = x_0^t\right) \\ &= \frac{b_j(x_t)}{N_t} \sum_i \widetilde{p}_{ij} \widetilde{F}_i(t-1). \end{aligned} \tag{7}$$

In the vocabulary of state-space models, this forward algorithm for hidden Markov chains is a filtering algorithm.

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

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

For a semi-Markovian state j, Guédon (2003) proposed for the computation of the normalizing factor,

$t = 0, \ldots, \tau - 1$:

$$\begin{aligned}&P\left(S_t = j, X_t = x_t | X_0^{t-1} = x_0^{t-1}\right) \\ &= b_j(x_t) \left[\sum_{u=1}^{t} \left\{ \prod_{v=1}^{u-1} \frac{b_j(x_{t-v})}{N_{t-v}} \right\} D_j(u) \sum_{i \neq j} p_{ij} F_i(t-u) \right. \\ &\quad \left. + \left\{ \prod_{v=1}^{t} \frac{b_j(x_{t-v})}{N_{t-v}} \right\} D_j(t+1) \pi_j \right]. \end{aligned} \tag{8}$$

Transposing the decomposition (11) of the smoothed probabilities in the backward recursion, we propose here the following alternative solution in order to save computation time (this change entails storing the quantities $P(S_t = j |\ X_0^t = x_0^t)$ for each semi-Markovian state j; see the appendix)

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

and,

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

$$
\begin{aligned}
&P\left(S_t = j, X_t = x_t | X_0^{t-1} = x_0^{t-1}\right) \\
&= P\left(X_t = x_t | S_t = j\right) \left\{ P\left(S_t = j, S_{t-1} \neq j | X_0^{t-1} = x_0^{t-1}\right) \right. \\
&\quad \left. - P\left(S_t \neq j, S_{t-1} = j | X_0^{t-1} = x_0^{t-1}\right) + P\left(S_{t-1} = j | X_0^{t-1} = x_0^{t-1}\right) \right\} \\
&= b_j\left(x_t\right) \left\{ \sum_{i \neq j} p_{ij} F_i\left(t-1\right) - F_j\left(t-1\right) + P\left(S_{t-1} = j | X_0^{t-1} = x_0^{t-1}\right) \right\}, \qquad (9)
\end{aligned}
$$

where $P\left(S_{t-1} = j | X_0^{t-1} = x_0^{t-1}\right) = P\left(S_{t-1} = j, X_{t-1} = x_{t-1} | X_0^{t-2} = x_0^{t-2}\right) / N_{t-1}$. In this way, the computation of the normalizing factors do not depend of the maximum times spent in the different states. Hence, the computation of the normalizing factors in (9) is of 'Markovian' complexity instead of 'semi-Markovian' complexity as in (8).

For a Markovian state j

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

and,

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

$$
P\left(S_t = j, X_t = x_t | X_0^{t-1} = x_0^{t-1}\right) = b_j\left(x_t\right) \sum_i \widetilde{p}_{ij} \widetilde{F}_i\left(t-1\right). \qquad (10)
$$

The combination of the forward recursion for hidden semi-Markov chains and the forward recursion for hidden Markov chains relies on the following fact.

Fact 1 *For a semi-Markovian state $i \neq j$*

$$
\begin{aligned}
&P\left(S_{t+1} = j, S_t = i | X_0^t = x_0^t\right) \\
&= P\left(S_{t+1} = j | S_{t+1} \neq i, S_t = i\right) P\left(S_{t+1} \neq i, S_t = i | X_0^t = x_0^t\right) \\
&= p_{ij} F_i\left(t\right),
\end{aligned}
$$

and for a Markovian state i

$$
\begin{aligned}
P\left(S_{t+1} = j, S_t = i | X_0^t = x_0^t\right) &= P\left(S_{t+1} = j | S_t = i\right) P\left(S_t = i | X_0^t = x_0^t\right) \\
&= \widetilde{p}_{ij} \widetilde{F}_i\left(t\right).
\end{aligned}
$$

Note that state j may be Markovian or semi-Markovian.

The quantity $p_{ij}F_i(t-u)$ in (5) and (6) - respectively the quantity $p_{ij}F_i(t-1)$ in (9) - should be replaced by $\widetilde{p}_{ij}\widetilde{F}_i(t-u)$ - respectively $\widetilde{p}_{ij}\widetilde{F}_i(t-1)$ - if state i is Markovian and, conversely, the quantity $\widetilde{p}_{ij}\widetilde{F}_i(t-1)$ in (7) and (10) should be replaced by $p_{ij}F_i(t-1)$ if state i is semi-Markovian. The resulting forward algorithm computes in parallel $F_j(t) = P(S_{t+1} \neq j,\ S_t = j|\ X_0^t = x_0^t)$ for semi-Markovian states and $\widetilde{F}_j(t) = P(S_t = j|\ X_0^t = x_0^t)$ for Markovian states. Note that $F_j(\tau-1) = \widetilde{F}_j(\tau-1) = P\left(S_{\tau-1} = j|\ X_0^{\tau-1} = x_0^{\tau-1}\right)$. As usual, the likelihood of the observed sequence is directly computed as a byproduct of the forward recursion since

$$P\left(X_0^{\tau-1} = x_0^{\tau-1};\theta\right) = \prod_{t=0}^{\tau-1} P\left(X_t = x_t|X_0^{t-1} = x_0^{t-1};\theta\right) = \prod_{t=0}^{\tau-1} N_t.$$

In practice, the log-likelihood of the observed sequence given by $\log P\left(X_0^{\tau-1} = x_0^{\tau-1};\theta\right) = \sum_{t=0}^{\tau-1}\log N_t$ is computed. This is useful both to monitor the convergence of the EM algorithm or as a (similarity) measure to affect an unknown sequence to a class represented by a previously estimated model in the context of pattern recognition applications.

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

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

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

For a semi-Markovian state j, the backward recursion relies on the following decomposition of the smoothed probability $L_j(t)$ (Guédon, 2003):

$$\begin{aligned} L_j(t) &= P\left(S_t = j|X_0^{\tau-1} = x_0^{\tau-1}\right) \\ &= P\left(S_{t+1} \neq j, S_t = j|X_0^{\tau-1} = x_0^{\tau-1}\right) + P\left(S_{t+1} = j|X_0^{\tau-1} = x_0^{\tau-1}\right) \\ &\quad - P\left(S_{t+1} = j, S_t \neq j|X_0^{\tau-1} = x_0^{\tau-1}\right) \\ &= L1_j(t) + L_j(t+1) - P\left(S_{t+1} = j, S_t \neq j|X_0^{\tau-1} = x_0^{\tau-1}\right). \end{aligned} \tag{11}$$

The backward recursion is based on the quantities $L1_j(t)$,

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

$$
\begin{aligned}
L1_j(t) = & \left[\sum_{k\neq j}\left[\sum_{u=1}^{\tau-2-t}\frac{L1_k(t+u)}{F_k(t+u)}\left\{\prod_{v=0}^{u-1}\frac{b_k(x_{t+u-v})}{N_{t+u-v}}\right\}d_k(u)\right.\right. \\
& \left.\left.+\left\{\prod_{v=0}^{\tau-2-t}\frac{b_k(x_{\tau-1-v})}{N_{\tau-1-v}}\right\}D_k(\tau-1-t)\right]p_{jk}\right]F_j(t) \\
= & \left\{\sum_{k\neq j}G_k(t+1)\,p_{jk}\right\}F_j(t), \qquad (12)
\end{aligned}
$$

where

$$
G_k(t+1) = \frac{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1} | S_{t+1} = k, S_t \neq k\right)}{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1} | X_0^t = x_0^t\right)}.
$$

The third term in (11) is given by,

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

$$
\begin{aligned}
& P\left(S_{t+1} = j, S_t \neq j | X_0^{\tau-1} = x_0^{\tau-1}\right) \\
= & \left[\sum_{u=1}^{\tau-2-t}\frac{L1_j(t+u)}{F_j(t+u)}\left\{\prod_{v=0}^{u-1}\frac{b_j(x_{t+u-v})}{N_{t+u-v}}\right\}d_j(u)\right. \\
& \left.+\left\{\prod_{v=0}^{\tau-2-t}\frac{b_j(x_{\tau-1-v})}{N_{\tau-1-v}}\right\}D_j(\tau-1-t)\right]\sum_{i\neq j}p_{ij}F_i(t) \\
= & G_j(t+1)\sum_{i\neq j}p_{ij}F_i(t).
\end{aligned}
$$

For a Markovian state j, the backward recursion is given by (Devijver, 1985),

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

$$
\begin{aligned}
L_j(t) = & P\left(S_t = j | X_0^{\tau-1} = x_0^{\tau-1}\right) \\
= & \left\{\sum_k \frac{L_k(t+1)\,b_k(x_{t+1})\,\widetilde{p}_{jk}}{\widetilde{F}_k(t+1)\,N_{t+1}}\right\}\widetilde{F}_j(t) \\
= & \left\{\sum_k \widetilde{G}_k(t+1)\,\widetilde{p}_{jk}\right\}\widetilde{F}_j(t), \qquad (13)
\end{aligned}
$$

where

$$\widetilde{G}_k(t+1) = \frac{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1} | S_{t+1} = k\right)}{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1} | X_0^t = x_0^t\right)}.$$

Fact 2 *For a semi-Markovian state $k \neq j$*

$$\begin{aligned}
&P\left(S_{t+1} = k, S_t = j | X_0^{\tau-1} = x_0^{\tau-1}\right) \\
&= \frac{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1} | S_{t+1} = k, S_t \neq k\right)}{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1} | X_0^t = x_0^t\right)} P\left(S_{t+1} = k, S_t = j | X_0^t = x_0^t\right) \\
&= G_k(t+1)\, P\left(S_{t+1} = k, S_t = j | X_0^t = x_0^t\right),
\end{aligned}$$

and for a Markovian state k

$$\begin{aligned}
&P\left(S_{t+1} = k, S_t = j | X_0^{\tau-1} = x_0^{\tau-1}\right) \\
&= \frac{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1} | S_{t+1} = k\right)}{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1} | X_0^t = x_0^t\right)} P\left(S_{t+1} = k, S_t = j | X_0^t = x_0^t\right) \\
&= \widetilde{G}_k(t+1)\, P\left(S_{t+1} = k, S_t = j | X_0^t = x_0^t\right).
\end{aligned}$$

where the computation of $P(S_{t+1} = k, S_t = j | X_0^t = x_0^t)$ depends on the nature of state j (either Markovian or semi-Markovian); see Fact 1.

Hence, $G_k(t+1)$ in (12) should be replaced by $\widetilde{G}_k(t+1)$ if state k is Markovian and, conversely, $\widetilde{G}_k(t+1)$ in (13) should be replaced by $G_k(t+1)$ if state k is semi-Markovian. An implementation of this forward-backward algorithm is proposed in the appendix in pseudo-code form where common computations between semi-Markovian states and Markovian states are highlighted.

4.2 Parameter reestimation

The reestimation of the initial probabilities, the state occupancy distributions and the observation probabilities (M-step of the EM algorithm) remains unchanged with reference to the hidden semi-Markov chain case described in Guédon (2003). The reestimation formulas for the transition probabilities are directly deduced from the maximizations of (3) and (4).

For a semi-Markovian state i, we have the following reestimation formula for the transition probabilities

$$
\begin{aligned}
p_{ij}^{(k+1)} &= \frac{\sum_{t=0}^{\tau-2} P\left(S_{t+1}=j, S_t=i | X_0^{\tau-1}=x_0^{\tau-1}; \theta^{(k)}\right)}{\sum_{t=0}^{\tau-2} P\left(S_{t+1}\neq i, S_t=i | X_0^{\tau-1}=x_0^{\tau-1}; \theta^{(k)}\right)} \\
&= \frac{\sum_{t=0}^{\tau-2} G_j\left(t+1\right) p_{ij} F_i\left(t\right)}{\sum_{t=0}^{\tau-2} L1_i\left(t\right)},
\end{aligned} \tag{14}
$$

while for a Markovian state i, we have

$$
\begin{aligned}
\widetilde{p}_{ij}^{(k+1)} &= \frac{\sum_{t=0}^{\tau-2} P\left(S_{t+1}=j, S_t=i | X_0^{\tau-1}=x_0^{\tau-1}; \theta^{(k)}\right)}{\sum_{t=0}^{\tau-2} P\left(S_t=i | X_0^{\tau-1}=x_0^{\tau-1}; \theta^{(k)}\right)} \\
&= \frac{\sum_{t=0}^{\tau-2} \widetilde{G}_j\left(t+1\right) \widetilde{p}_{ij} \widetilde{F}_i\left(t\right)}{\sum_{t=0}^{\tau-2} L_i\left(t\right)}.
\end{aligned} \tag{15}
$$

The quantity $G_j\left(t+1\right)$ in (14) should be replaced by $\widetilde{G}_j\left(t+1\right)$ if state j is Markovian and, conversely, the quantity $\widetilde{G}_j\left(t+1\right)$ in (15) should be replaced by $G_j\left(t+1\right)$ if state j is semi-Markovian.

5 Viterbi algorithm

The manner in which the forward recursion for hidden semi-Markov chains and the forward recursion for hidden Markov chains are combined in the case of hybrid models directly transposes to the Viterbi algorithm.

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

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

$$
\begin{aligned}
\alpha_j\left(t\right) &= \max_{s_0,\ldots,s_{t-1}} P\left(S_{t+1}\neq j, S_t=j, S_0^{t-1}=s_0^{t-1}, X_0^t=x_0^t\right) \\
&= b_j\left(x_t\right) \max\left[\max_{1\leq u\leq t}\left[\left\{\prod_{v=1}^{u-1} b_j\left(x_{t-v}\right)\right\} d_j\left(u\right) \max_{i\neq j}\left\{p_{ij}\alpha_i\left(t-u\right)\right\}\right],\right. \\
&\quad \left.\left\{\prod_{v=1}^{t} b_j\left(x_{t-v}\right)\right\} d_j\left(t+1\right)\pi_j\right].
\end{aligned} \tag{16}
$$

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

$$
\begin{aligned}
&\alpha_j(\tau-1)\\
&=\max_{s_0,\ldots,s_{\tau-2}} P\left(S_{\tau-1}=j, S_0^{\tau-2}=s_0^{\tau-2}, X_0^{\tau-1}=x_0^{\tau-1}\right)\\
&=b_j(x_{\tau-1})\max\left[\max_{1\le u\le \tau-1}\left[\left\{\prod_{v=1}^{u-1} b_j(x_{\tau-1-v})\right\} D_j(u)\max_{i\neq j}\{p_{ij}\alpha_i(\tau-1-u)\}\right],\right.\\
&\left.\left\{\prod_{v=1}^{\tau-1} b_j(x_{\tau-1-v})\right\} D_j(\tau)\,\pi_j\right]. \qquad (17)
\end{aligned}
$$

For a Markovian state j, the Viterbi 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 (Rabiner, 1989),

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

$$
\begin{aligned}
\widetilde{\alpha}_j(t) &= \max_{s_0,\ldots,s_{t-1}} P\left(S_t=j, S_0^{t-1}=s_0^{t-1}, X_0^t=x_0^t\right)\\
&= b_j(x_t)\max_i\{\widetilde{p}_{ij}\widetilde{\alpha}_i(t-1)\}. \qquad (18)
\end{aligned}
$$

The likelihood of the optimal state sequence associated with the observed sequence $x_0^{\tau-1}$ is $\max_j\{\alpha_j(\tau-1)\}$ (it should be noted that $\alpha_j(\tau-1)=\widetilde{\alpha}_j(\tau-1)$).

Fact 3 *For a semi-Markovian state $i\neq j$*

$$
\begin{aligned}
&\max_{s_0,\ldots,s_{t-1}} P\left(S_{t+1}=j, S_t=i, S_0^{t-1}=s_0^{t-1}, X_0^t=x_0^t\right)\\
&=P(S_{t+1}=j|S_{t+1}\neq i, S_t=i)\max_{s_0,\ldots,s_{t-1}} P\left(S_{t+1}\neq i, S_t=i, S_0^{t-1}=s_0^{t-1}, X_0^t=x_0^t\right)\\
&=p_{ij}\alpha_i(t),
\end{aligned}
$$

and for a Markovian state i

$$
\begin{aligned}
&\max_{s_0,\ldots,s_{t-1}} P\left(S_{t+1}=j, S_t=i, S_0^{t-1}=s_0^{t-1}, X_0^t=x_0^t\right)\\
&=P(S_{t+1}=j|S_t=i)\max_{s_0,\ldots,s_{t-1}} P\left(S_t=i, S_0^{t-1}=s_0^{t-1}, X_0^t=x_0^t\right)\\
&=\widetilde{p}_{ij}\widetilde{\alpha}_i(t).
\end{aligned}
$$

Note that state j may be Markovian or semi-Markovian.

The quantity $p_{ij}\alpha_i(t-u)$ in (16) and (17) 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 (18) should be replaced by $p_{ij}\alpha_i(t-1)$ if state i is semi-Markovian. The resulting Viterbi algorithm computes in parallel $\alpha_j(t) = \max_{s_0,\ldots,s_{t-1}} P(S_{t+1} \neq j,\ S_t = j, S_0^{t-1} = s_0^{t-1}, X_0^t = x_0^t)$ for semi-Markovian states and $\widetilde{\alpha}_j(t) = \max_{s_0,\ldots,s_{t-1}} P(S_t = j,\ S_0^{t-1} = s_0^{t-1}, X_0^t = x_0^t)$ for Markovian states. Note that $\alpha_j(\tau-1) = \widetilde{\alpha}_j(\tau-1) = \max_{s_0,\ldots,s_{\tau-2}} P(S_{\tau-1} = j,\ S_0^{\tau-2} = s_0^{\tau-2}, X_0^{\tau-1} = x_0^{\tau-1})$. The Viterbi recursion is the equivalent in terms of dynamic programming of the forward recursion (summation in (5) (6) (7) replaced by maximization in (16) (17) (18)). Therefore, the proposals made for an efficient implementation of the forward recursion in the appendix directly transpose to the Viterbi algorithm.

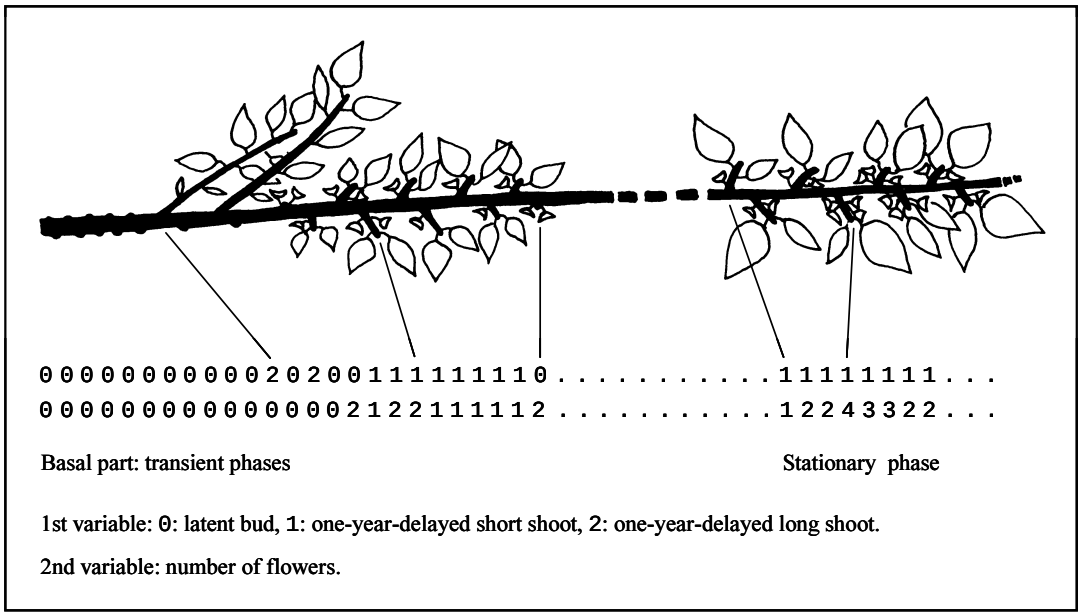

Fig. 5. Growth unit of cultivar 'Lambertin' where the nature of the axillary production and the number of associated flowers were recorded for each successive node (drawing by Yves Caraglio).

If the objective is to retrieve the optimal state sequence, the recursion described above should be complemented by a backtracking procedure. In the case of semi-Markovian states, the backtracking procedure operates by jumps on the basis of two backpointers, the first giving the optimal preceding state and the second 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 a single backpointer giving the optimal preceding state.

6 Application to the analysis of branching and flowering patterns

The use of hybrid models is illustrated by the reanalysis of a sample of sequences originally analyzed by a hidden semi-Markov chain (Guédon et al., 2001; Guédon, 2003). A sample of 48 growth units (portion of a leafy axis established between two resting phases) of apricot tree (*Prunus armeniaca*, *Rosaceae*), cultivar 'Lambertin', grafted on rootstock 'Manicot' was described node by node from the base to the top. The type of axillary production - chosen among latent bud (0), one-year-delayed short shoot (1), one-year-delayed long shoot (2) and immediate shoot (3) - and the number of associated flowers (0, 1, 2, 3 flowers or more) were recorded for each node (Fig. 5). The branching and the flowering variables correspond to events that do not occur simultaneously in plant development and were thus measured at two different dates (beginning of the growth period for the flowering and end of the growth period for the branching). These are nevertheless assumed to be closely related since the flowers are always borne by the offspring shoots in positions corresponding to prophylls (the two first foliar organs of an offspring shoot).

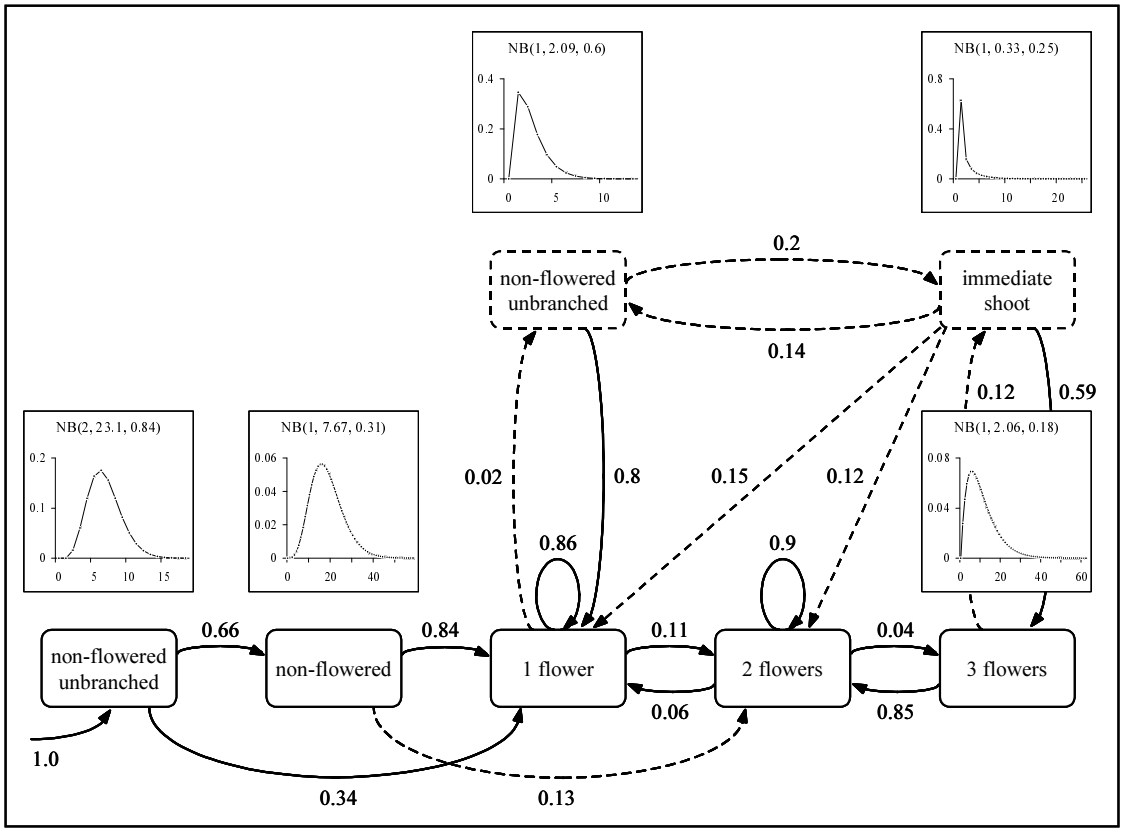

Fig. 6. Estimated hidden hybrid Markov/semi-Markov chain.

The estimated hybrid model is represented in Fig. 6: only the transitions with probabilities exceeding 0.03 are represented. The dotted edges correspond to the less probable transitions (the convention is not the same for Markovian states and semi-

Markovian states) while the dotted vertices correspond to the less probable states. The underlying hybrid Markov/semi-Markov chain is composed of two transient states followed by a five-state recurrent class mainly structured on the basis of the flowering variable. An interpretation is associated with each state, summarizing the combination of the estimated observation probabilities. The first transient state corresponds to the initial transient phases for both variables while the second transient state corresponds to the end of the transient phase for the flowering variable; see Guédon et al. (2001) and Guédon (2003). The two less probable states in the recurrent class are the direct expression of biological hypotheses and were a priori defined in the specification stage by appropriate constraints on model parameters: the 'resting' state (unbranched, non-flowered) corresponds to zones of slowdown in the growth of the parent shoot. The immediate branching state corresponds to a rare event in this context and immediate branching follows very different rules compared to one-year-delayed branching and these two types of branching should not therefore be mixed in a given state.

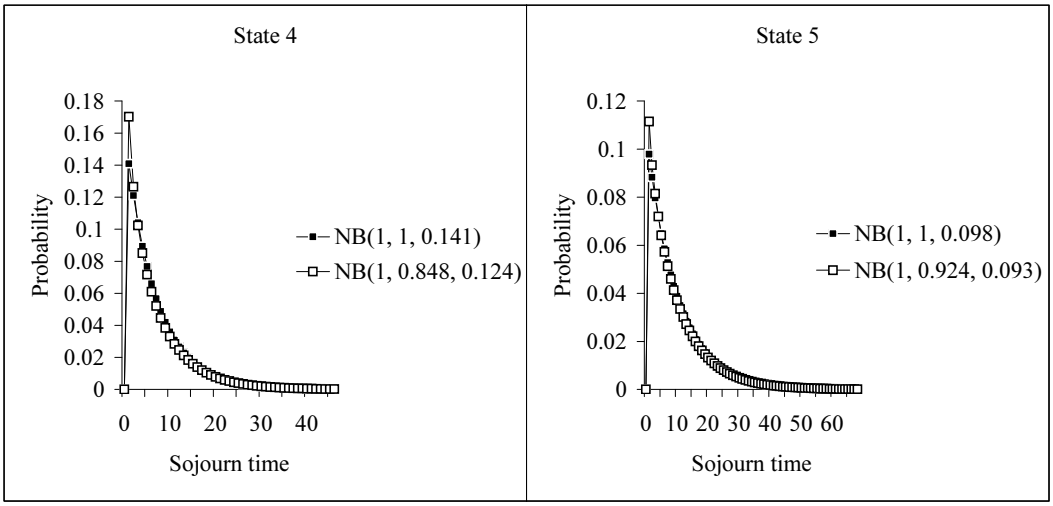

Fig. 7. Comparison of state occupancy distributions between the hidden hybrid Markov/semi-Markov chain and the hidden semi-Markov chain.

In the originally estimated hidden semi-Markov chain, it could be seen that the state occupancy distributions estimated for states 4 ('1 flower') and 5 ('2 flowers') were very close to '1-shifted' geometric distributions NB(1, 1, p): NB(1, 0.848, 0.124) and NB(1, 0.924, 0.094) respectively, i.e. '1-shifted' negative binomial distributions (see the definition (2)) with parameter close to 1. Moreover, state 5 and state 4 (in this order) were the two most represented states with a total weight of 55% on the basis of the means of the counting distributions (number of occurrences of a given state per sequence) related to the sequence length distribution. Hence, we chose to estimate an hybrid model where states 4 and 5 were Markovian.

The convergence of the EM algorithm required 19 iterations for the hidden semi-Markov chain and 20 iterations for the hybrid model. The convergence of the EM

algorithm is monitored upon the increase over iterations of the log-likelihood of the observed sequences which is directly obtained as a byproduct of the forward recursion; see Section 4.1. The implicit geometric state occupancy distributions are very close to the explicit state occupancy distributions estimated for the hidden semi-Markov chain; see Fig. 7. A detailed comparison of the other parameters showed that they are almost identical in the estimated hidden semi-Markov chain and in the estimated hybrid model. Therefore, the log-likelihoods of the sequences for the two models are also almost identical ($2 \log L = -7745.7$ for the hidden semi-Markov chain and $2 \log L = -7745.1$ for the hybrid model). As a consequence, the most likely state sequences computed with the Viterbi algorithm for the two estimated models are strikingly similar (3 differences for a cumulated sequence length of 2881).

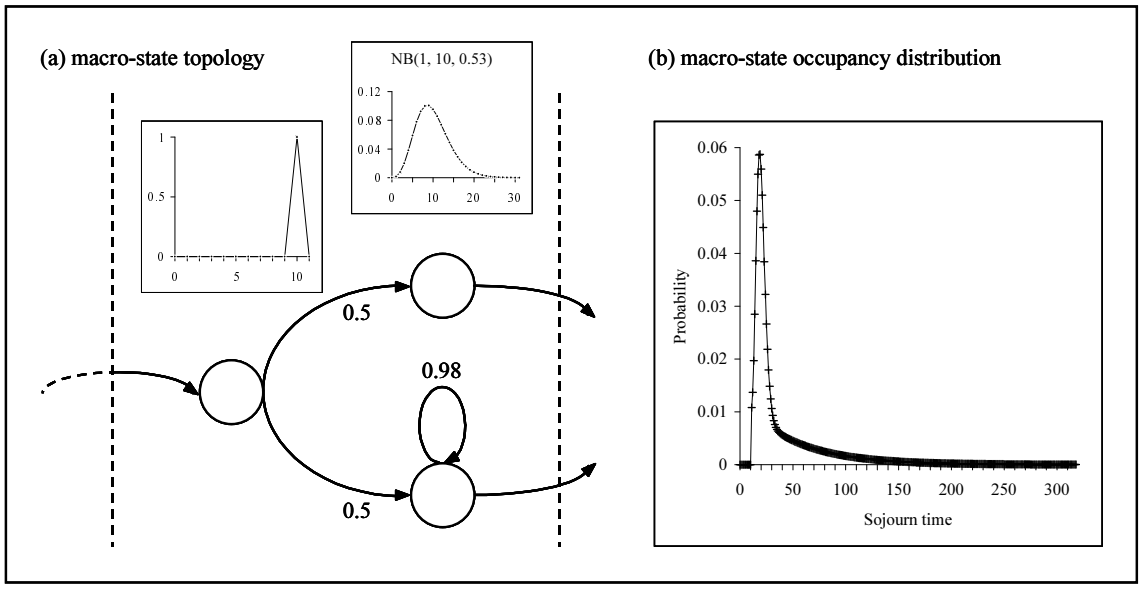

Fig. 8. Example of a macro-state built from a Markovian state and semi-Markovian states with the associated macro-state occupancy distribution.

A validation methodology relying on the fit of different types of characteristic distributions computed from model parameters to their empirical equivalents extracted from data is illustrated by diverse examples, including this apricot tree example, in Guédon et al. (2001) and Guédon (2003). It should be noted that the algorithms for computing characteristic distributions of hidden semi-Markov chains (Guédon, 1999) - for instance interval and counting distributions for the different possible outputs - can be modified for hybrid models in the same manner in which the forward recursion is modified; see Section 4.1.

7 Concluding remarks

Macro-states should not be considered as a valid alternative to semi-Markovian states for the modeling of short or medium size homogeneous zones as shown in Section 3. For long zones, Markovian states are mandatory because of algorithmic complexity constraints. Nevertheless, the shape of the implicit geometric state occupancy distribution may be too constraining and to remedy this shortcoming, macro-states combining Markovian states with semi-Markovian states may be included in hidden hybrid Markov/semi-Markov chains. A zone of highly variable length - for instance corresponding to introns in DNA sequences; see Kulp et al. (1996) and Burge and Karlin (1997) - can be modeled by a series-parallel network of Markovian and semi-Markovian states with common observation distribution. This point is illustrated by the example in Fig. 8 where the macro-state is composed of a 'degenerated' semi-Markovian state with a fixed sojourn time (to model the minimum sojourn time spent in the macro-state) followed by two elementary states in parallel, a Markovian state for long zones and a semi-Markovian state for shorter zones.

Hence, Markovian states, semi-Markovian states and macro-states (for combining Markovian states with semi-Markovian states) are the building blocks of flexible state processes with precise guidelines and algorithmic solutions for their combination. The algorithms described in Sections 4.1 and 5 still apply in the case of macro-states, the only minor modification being the management of tying constraints within macro-states for the reestimation of the observation distributions. This point of view is in accordance with the development of very flexible hidden Markov models which can also incorporate various sub-models as output processes; see Burge (1997) and Burge and Karlin (1997).

The principle of the computation of the normalizing factors (see Section 4.1) can be directly transposed to the computation of the marginal state distributions as a function of the index parameter. In the case of a semi-Markov chain, the computation of the probabilities of being in a given state at time t requires the prior computation of the probabilities of leaving a given state at time t. Thus, the auxiliary quantities $\alpha_j(t) = P(S_{t+1} \neq j, S_t = j)$ are computed for each successive time t and each state j by the following 'forward' recursion (Guédon, 1999),

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

$$\begin{aligned}\alpha_j(t) &= P(S_{t+1} \neq j, S_t = j) \\ &= \sum_{u=1}^{t} d_j(u) \sum_{i \neq j} p_{ij} \alpha_i(t-u) + d_j(t+1)\, \pi_j.\end{aligned}$$

For computing the probability of being in state j at time t, Guédon (1999) proposed,

$t = 0, \ldots, \tau - 1$:

$$P(S_t = j) = \sum_{u=1}^{t} D_j(u) \sum_{i \neq j} p_{ij} \alpha_i(t-u) + D_j(t+1)\, \pi_j.$$

In a way similar to the computation of the normalizing factors in the forward recursion (see (9)), we propose here the following alternative solution in order to save computation time

$$P(S_0 = j) = \pi_j,$$

and,

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

$$\begin{aligned} P(S_t = j) &= P(S_t = j, S_{t-1} \neq j) - P(S_t \neq j, S_{t-1} = j) + P(S_{t-1} = j) \\ &= \sum_{i \neq j} p_{ij} \alpha_i(t-1) - \alpha_j(t-1) + P(S_{t-1} = j). \end{aligned}$$

This algorithm can be adapted to the case of hybrid models incorporating Markovian states in the same manner in which the forward recursion of hidden semi-Markov chains is adapted to hybrid models; see Section 4.1. The output distributions can be directly deduced since

$$P(X_t = y) = \sum_j b_j(y)\, P(S_t = j).$$

The fits of state or output probabilities as a function of the index parameter are valuable validation tools as illustrated in Guédon et al. (2001) and Guédon (2003).

Computational methods for hidden hybrid Markov/semi-Markov chains are fully implemented in the AMAPmod software (Godin *et al.* 1997, 1999) which is freely available at http://amap.cirad.fr.

Acknowledgments

The author thanks Dominique Cellier for his helpful comments and Yves Caraglio for the botanical drawing.

Appendix: Pseudo-code of the forward-backward algorithm

The following convention is adopted in the presentation of the pseudo-code of the 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 $\mathrm{Norm}_j(t)$, $\mathrm{Forward}_j(t)$, Observ, $\mathrm{StateIn}_j(t+1)$, $\mathrm{Transition}_{ij}$, $\mathrm{Backward}_j(t)$ and $\mathrm{Aux}_j(t+1)$ are introduced for this implementation. Note that $\mathrm{Forward}_j(t)$ is used to compute $F_j(t)$ for a semi-Markovian state and $\widetilde{F}_j(t)$ for a Markovian state, $\mathrm{Backward}_j(t)$ is used to compute $L1_j(t)$ for a semi-Markovian state and $L_j(t)$ for a Markovian state while $\mathrm{Aux}_j(t+1)$ is used to compute $G_j(t+1)$ for a semi-Markovian state and $\widetilde{G}_j(t+1)$ for a Markovian state. $\mathrm{Transition}_{ij}$ 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 4.1.

Forward recursion

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

    if state j is semi-Markovian **then**
      if $t = 0$ **then**
        $\mathrm{Norm}_j(0) := b_j(x_0)\,\pi_j$
      else $\{t > 0\}$
        $\mathrm{Norm}_j(t) := b_j(x_t)\{\mathrm{StateIn}_j(t) - \mathrm{Forward}_j(t-1) + \mathrm{Norm}_j(t-1)\}$
      end if
      $N_t := N_t + \mathrm{Norm}_j(t)$

    else {state j is Markovian}
      if $t = 0$ **then**
        $\mathrm{Forward}_j(0) := b_j(x_0)\,\pi_j$
      else $\{t > 0\}$
        $\mathrm{Forward}_j(t) := b_j(x_t)\,\mathrm{StateIn}_j(t)$
      end if
      $N_t := N_t + \mathrm{Forward}_j(t)$
    end if
  end for

  for $j := 0$ to $J - 1$ **do**
    if state j is semi-Markovian **then**
      $\mathrm{Norm}_j(t) := \mathrm{Norm}_j(t)/N_t$
    else {state j is Markovian}
      $\mathrm{Forward}_j(t) := \mathrm{Forward}_j(t)/N_t$
    end if
  end for

  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**
$\text{Observ} := \text{Observ}\, b_j(x_{t-u+1})/N_{t-u+1}$
if $u < t+1$ **then**
$\text{Forward}_j(t) := \text{Forward}_j(t) + \text{Observ}\, d_j(u)\, \text{StateIn}_j(t-u+1)$
else $\{u = t+1\}$
$\text{Forward}_j(t) := \text{Forward}_j(t) + \text{Observ}\, d_j(t+1)\, \pi_j$
end if
end for

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

if $t < \tau - 1$ **then**
for $j := 0$ to $J-1$ **do**
$\text{StateIn}_j(t+1) := 0$
for $i := 0$ to $J-1$ **do**
$\text{StateIn}_j(t+1) := \text{StateIn}_j(t+1) + \text{Transition}_{ij}\, \text{Forward}_i(t)$
end for
end for
end if
end for

For semi-Markovian states, the auxiliary quantities $\text{Norm}_j(t)$ are introduced to compute the normalizing factor N_t. In a first step, the quantities $P\left(S_t = j,\ X_t = x_t \middle|\ X_0^{t-1} = x_0^{t-1}\right)$ are computed (using the variable $\text{Norm}_j(t)$ for a semi-Markovian state j or the variable $\text{Forward}_j(t)$ for a Markovian state j). In a second step, the quantities $P\left(S_t = j \middle|\ X_0^t = x_0^t\right)$ (which are the forward probabilities $F_j(t)$ in the case of Markovian states) are extracted as $P\left(S_t = j,\ X_t = x_t \middle|\ X_0^{t-1} = x_0^{t-1}\right)/N_t$. In a third step, the quantities $\text{Forward}_j(t) = P\left(S_{t+1} \neq j,\ S_t = j, \middle|\ X_0^t = x_0^t\right)$ are computed for each semi-Markovian state. In a fourth step, the quantities $\text{StateIn}_j(t+1)$ are

extracted. For a semi-Markovian state j, $\text{StateIn}_j(t+1) = P(S_{t+1} = j,\ S_t \neq j |\ X_0^t = x_0^t)$ while for a Markovian state j $\text{StateIn}_j(t+1) = P(S_{t+1} = j |\ X_0^t = x_0^t)$ (predicted probability in the vocabulary of state space models). The forward probabilities $F_j(t)$ and the companion quantities $\text{StateIn}_j(t+1)$ should be stored for each time t and each state j and the normalizing quantities N_t should be stored for each time t. The auxiliary quantities $\text{Norm}_j(t)$ need only be stored for each semi-Markovian state j.

Backward recursion

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

for $t := \tau-2$ to 0 **do**
  for $j := 0$ to $J-1$ **do**

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

    else {state j is Markovian}
      $\text{Aux}_j(t+1) := \text{Backward}_j(t+1)/\text{StateIn}_j(t+1)$ (see the remark below)
    end if
  end for

  for $j := 0$ to $J-1$ **do**
    $\text{Backward}_j(t) := 0$
    for $k := 0$ to $J-1$ **do**
      $\text{Backward}_j(t) := \text{Backward}_j(t) + \text{Aux}_k(t+1)\,\text{Transition}_{jk}$
    end for
    $\text{Backward}_j(t) := \text{Backward}_j(t)\,\text{Forward}_j(t)$

if state j is semi-Markovian **then**
$L_j(t) := \text{Backward}_j(t) + L_j(t+1) - \text{Aux}_j(t+1)\,\text{StateIn}_j(t+1)$
else {state j is Markovian}
$L_j(t) := \text{Backward}_j(t)$
end if
end for
end for

For a Markovian state j, $\widetilde{F}_j(t+1)\,N_{t+1}/b_j(x_{t+1}) = P(S_{t+1} = j|X_0^t = x_0^t) = \sum_i \widetilde{p}_{ij}\widetilde{F}_i(t)$; see (7).

In a first step, the auxiliary quantities $\text{Aux}_j(t+1)$ are computed. Then in the second step, the quantities $\text{Backward}_j(t)$ and $L_j(t)$ are extracted. The quantities $\text{Backward}_j(t)$ should be stored for each time t and each state j while the smoothed probabilities $L_j(t)$ and the auxiliary quantities $\text{Aux}_j(t+1)$ need only be stored for each state j.

References

Burge, C. (1997). Identification of Genes in Human Genomic DNA. Phd Thesis, Stanford University, Stanford, CA.

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

Cook, A. E. and Russel, M. J. (1986). Improved duration modelling in hidden Markov models using series-parallel configurations of states. In: Proceedings of the Institute of Acoustics Autumn Conference on Speech and Hearing, Windermere, Vol. 8: Part 7, 299-306.

Devijver, P. A. (1985). Baum's forward-backward algorithm revisited. Pattern Recogn. Lett. 3, 369-373.

Durbin, R., Eddy, S. R., Krogh, A. and Mitchison, G. J. (1998). Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, Cambridge.

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

Feller, W. (1968). An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd edn. Wiley, New York.

Godin, C., Guédon, Y. and Costes, E. (1999). Exploration of a plant architecture database with the AMAPmod software illustrated on an apple tree hybrid family. Agronomie 19, 163-184.

Godin, C., Guédon, Y., Costes, E. and Caraglio, Y. (1997). Measuring and analysing plants with the AMAPmod software. In: Michalewicz, M. T. (Ed.), Plants to Ecosystems - Advances in Computational Life Sciences, Vol. 1, CSIRO Publishing, Collingwood, Victoria, 53-84.

Guédon, Y. (1992). Review of several stochastic speech unit models. Comput. Speech Lang. 6, 377-402.

Guédon, Y. (1999). Computational methods for discrete hidden semi-Markov chains. Appl. Stochastic Models Bus. Ind. 15, 195-224.
Guédon, Y. (2003). Estimating hidden semi-Markov chains from discrete sequences. J. Comput. Graphical Statist. 12(3), 604-639.
Guédon, Y., Barthélémy, D., Caraglio, Y. and Costes, E. (2001). Pattern analysis in branching and axillary flowering sequences. J. Theor. Biol. 212, 481-520.
Johnson, N. L., Kotz, S. and Kemp, A. W. (1993). Univariate Discrete Distributions, 2nd edn. Wiley, New York.
Kleinrock, L. (1975). Queueing Systems, Vol. 1: Theory. Wiley, New York.
Kulkarni, V. G. (1995). Modeling and Analysis of Stochastic Systems. Chapman & Hall, London.
Kulp, D., Haussler, D., Reese, M. G. and Eeckman, F. H. (1996). A generalized hidden Markov model for the recognition of human genes in DNA. In: States, D. J., Agarwal, P., Gaasterland, T., Hunter, L. and Smith, R. F. (Eds.), Proceedings of the fourth International Conference on Intelligent Systems for Molecular Biology, AAAI Press, Menlo Park, CA, 134-142.
Lukashin, A. V. and Borodovsky, M. (1998). GeneMark.hmm: new solutions for gene finding. Nucleic Acids Res. 26(4), 1107-1115.
Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE 77(2), 257-286.