Estimating hidden semi-Markov chains from discrete sequences

Yann GUÉDON

Unité Mixte de Recherche CIRAD/CNRS/INRA/Université Montpellier II
Botanique et Bioinformatique de l'Architecture des Plantes,
TA 40/PS2, 34398 Montpellier Cedex 5, France
E-mail: guedon@cirad.fr

ACKNOWLEDGMENTS

I would like to thank Avner Bar-Hen for his fruitful comments on an earlier version of this paper, a referee for many helpful suggestions that led to an improvement in the presentation of this paper and Yves Caraglio for the botanical drawings.

Estimating hidden semi-Markov chains from discrete sequences

ABSTRACT

We address the estimation of hidden semi-Markov chains from nonstationary discrete sequences. Hidden semi-Markov chains are particularly useful to model the succession of homogeneous zones or segments along sequences. A discrete hidden semi-Markov chain is composed of a non-observable state process, which is a semi-Markov chain, and a discrete output process. Hidden semi-Markov chains generalize hidden Markov chains and enable the modeling of various durational structures. From an algorithmic point of view, a new forward-backward algorithm is proposed whose complexity is similar to that of the Viterbi algorithm in terms of sequence length (quadratic in the worst case in time and linear in space). This opens the way to the maximum likelihood estimation of hidden semi-Markov chains from long sequences. This statistical modeling approach is illustrated by the analysis of branching and flowering patterns in plants.

KEY WORDS: Censoring; EM algorithm; Forward-backward algorithm; Hidden semi-Markov chain; Nonparametric maximum likelihood; Plant structure analysis; Smoothing algorithm; Viterbi algorithm.

1 Introduction

In this paper, we study the estimation of hidden semi-Markov chains from nonstationary discrete - possibly multivariate - sequences. In the type of discrete sequences in which we are interested, the local composition properties do not hold throughout the length of a given sequence. These sequences may rather be viewed as a succession of homogeneous zones or segments where the composition properties do not change substantially within each zone, but change markedly between zones. These homogeneous zones may either occur in a recurrent or a transient way. This type of structuring in sequences may be found in such diverse applications as speech unit modeling (Rabiner, 1989), DNA sequence analysis (Churchill, 1989; Braun and Müller, 1998) or the analysis of branching and flowering patterns in plants (Guédon et al., 2001).

The interest in hidden semi-Markov chains originates in the field of speech recognition where they were studied as a possible alternative to classical hidden Markov chains for speech unit modeling. Hidden Markov chains emerged in the 1970s in engineering and have since become a major tool for both pattern recognition applications, such as speech or handwriting recognition (see Poritz (1988) or Rabiner (1989) for tutorial introductions), and biological sequence analysis (see Churchill (1989) and Durbin et al. (1998)); see also the monograph of MacDonald and Zucchini (1997). Basically, a hidden Markov chain is a pair of discrete-time 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 Markov chain, by a probabilistic function or mapping denoted by f. Since the mapping f is such that 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 $\{X_t\}$. It should be noted that the output process $\{X_t\}$ may be either discrete or continuous, univariate or multivariate.

A major drawback with hidden Markov chains is the inflexibility in describing the time spent in a given state which is geometrically distributed. It is unlikely that such a type of implicit state occupancy distribution is an appropriate model for speech segment duration, the length of segments of a given C+G content along DNA sequences or the length of branching zones in plants. In a hidden semi-Markov chain, the state process $\{S_t\}$ is a finite-state semi-Markov chain while the conditional independence assumptions concerning the output process $\{X_t\}$ are the same as in a simple hidden Markov chain. A semi-Markov chain is composed of an embedded first-order Markov chain representing the transitions between distinct states, and discrete state occupancy distributions representing sojourn times in nonabsorbing states. Hidden semi-Markov chains with nonparametric state occupancy distributions were first proposed in the field of speech recognition by Ferguson (1980). After this pioneering work, the statistical inference problem related to hidden semi-Markov chains was further investigated by different authors (Russell and Moore, 1985; Levinson, 1986; Guédon and Cocozza-Thivent, 1990; Guédon, 1992) and different parametric hypotheses were put forward for the state occupancy distributions (Poisson, 'discrete' gamma).

Our treatment is in contrast with these previous proposals where it was implicitly assumed that the end of a sequence systematically coincides with the exit from a state, that is the sequence length is not independent of the process. This very specific assumption entails a simple writing of the likelihood functions but the corresponding hidden semi-Markov chains are misspecified in the sense that they cannot incorporate absorbing states and hence cannot be considered as true generalization of hidden Markov chains. We define hidden semi-Markov chains with absorbing states and thus define the likelihood of a state sequence generated by an underlying semi-Markov chain with a right censoring of the time spent in the last visited state.

We review carefully the implications of this right censoring in the design of the algorithms (forward-backward and Viterbi). We also propose a new forward-backward algorithm with complexities that are quadratic in the worst case in time and linear in space, in terms of sequence length. This is a major improvement compared to the proposal of Guédon and Cocozza-Thivent (1990) where the complexity in time was cubic in the worst case and the complexity in space was quadratic in the worst case. This opens the way to the application of the full machinery of hidden semi-Markov chains to long sequences such as DNA sequences. Up to now, the use of hidden semi-Markov chains for gene finding relied mainly on the Viterbi algorithm for determining the optimal homogeneous zones while the parameter estimates were obtained by various ad-hoc procedures (Burge and Karlin, 1997; Lukashin and Borodovsky, 1998).

The remainder of this paper is organized as follows. Discrete hidden semi-Markov chains are formally defined in Section 2. The estimation of a hidden semi-Markov chain from discrete sequences based on the application of the EM algorithm and the associated forward-backward algorithm, which forms the core of this paper, is presented in Section 3. In Section 4, complementary algorithms, including the Viterbi algorithm, which may be especially useful for the validation of hidden semi-Markov chains, are reviewed. The resulting data analysis methodology is illustrated in Section 5 by the analysis of branching and flowering patterns in plants. Section 6 consists of concluding remarks and a discussion of some perspectives.

2 Discrete hidden semi-Markov chain definition and notations

Let $\{S_t\}$ be a semi-Markov chain with finite state space $\{0, \ldots, J-1\}$; see Kulkarni (1995) for a general reference about semi-Markov models. In the case of a nonabsorbing state, the sojourn time in this state is a discrete non-negative random variable with an arbitrary distribution. A semi-Markov chain is constructed from an embedded first-order Markov chain. This J-state first-order Markov chain is defined by the following parameters:

- initial probabilities $\pi_j = P(S_0 = j)$ with $\sum_j \pi_j = 1$,
- transition probabilities
  - nonabsorbing state i: for each $j \neq i, p_{ij} = P(S_{t+1} = j | S_{t+1} \neq i, S_t = i)$ with $\sum_{j \neq i} p_{ij} = 1$ and $p_{ii} = 0$,
  - absorbing state i: $\widetilde{p}_{ii} = P(S_{t+1} = i | S_t = i) = 1$ and for each $j \neq i, \widetilde{p}_{ij} = 0$.

This embedded first-order Markov chain represents transitions between distinct states except in the absorbing state case.

An occupancy (or sojourn time) distribution is attached to each nonabsorbing state of the embedded first-order Markov chain

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

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, $D_j(u) = \sum_{v \geq u} d_j(v)$. The whole (first-order Markov chain + state occupancy distributions) constitutes a semi-Markov chain. It should be noted that the absorbing states keep a Markovian definition which contrasts with the definition of the nonabsorbing semi-Markovian states.

If the process starts out at $t = 0$ in a given nonabsorbing 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.

By replacing a first-order Markov chain by a semi-Markov chain, the Markovian property is transferred to the level of the embedded first-order Markov chain. In the semi-Markov chain case, the conditional independence between the past and the future is only ensured when the process moves from one state to another distinct state. This property holds at each time step in the case of a Markov chain.

A discrete hidden semi-Markov chain can be seen 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

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 $f(i) = f(j)$ may be satisfied for some different i, j, 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 $\{X_t\}$.

The output process $\{X_t\}$ is related to the semi-Markov chain $\{S_t\}$ by the observation (or emission) probabilities

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

These observation probabilities can be arranged as a $J \times Y$ matrix denoted by B with all rows summing to one (Y denotes the number of possible outputs).

The definition of the observation probabilities expresses the assumption that the output process at time t depends only on the underlying 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 $S_t = s_t$.

For the remainder of this paper, we need to introduce some notations. 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 $S_0^{\tau-1} = s_0^{\tau-1}$). The number of states visited in the sequence $s_0^{\tau-1}$ will be denoted by R. In the estimation framework, θ designates the vector of all parameters.

3 Estimation of a hidden semi-Markov chain

The proposed estimation procedure based on the application of the EM algorithm has the following properties:

- Hidden semi-Markov chains with absorbing states can be estimated from data,
- The complexity of the forward-backward algorithm that implements the E-step of the EM algorithm is similar to the complexity of the forward algorithm alone or of the Viterbi algorithm, that is $O(J\tau(J+\tau))$-time in the worst case and $O(J\tau)$-space,
- the proposed forward-backward algorithm is immune to numerical underflow problems and does not require ad-hoc scaling procedures. It is well known that the direct implementation of the originally proposed forward-backward algorithm (see Ferguson (1980) whose essential results are summarized in Rabiner (1989)) entails the multiplication of many probabilities (either transition, occupancy or observation probabilities) and thus generates underflow errors; see Devijver (1985) where this point is thoroughly discussed in the context of hidden Markov chains.
- this forward-backward 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 (Kitagawa, 1987), this forward-backward algorithm is a smoothing algorithm.

The proposal of Guédon and Cocozza-Thivent (1990) only had the two last properties. In particular, the complexity in space, which was quadratic in terms of the sequence length τ, effectively restricted the application of this first proposed algorithm to short sequences.

3.1 Application of the EM algorithm

The estimation problem is stated as a nonparametric maximum likelihood estimation problem which means that the state occupancy distributions are considered as nonparametric discrete distributions concentrated on finite sets of time points. In the following, we will state the estimation problem with a single observed sequence. The generalization to the practical case of a sample of sequences is straightforward (Guédon and Cocozza-Thivent, 1990; Guédon, 1992). Let us consider the complete-data likelihood where both the outputs $x_0^{\tau-1}$ and the states $s_0^{\tau-1}$ of the underlying semi-Markov chain are observed

$$f\left(s_0^{\tau-1}, x_0^{\tau-1}; \theta\right) = P\left(S_0^{\tau-1} = s_0^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}; \theta\right).$$

The contribution of the state sequence to the complete-data likelihood has always been written as

$$\pi_{s_0} d_{s_0}\left(u_0\right) \prod_{r \geq 1} p_{s_{r-1}s_r} d_{s_r}\left(u_r\right) I\left(\sum_r u_r = \tau\right), \tag{2}$$

where s_r is the $(r+1)$-th visited state, u_r is the time spent in state s_r and $I()$ denotes the indicator function; see for instance Russel and Moore (1985) Rabiner (1989), Guédon and Cocozza-Thivent (1990) or Burge and Karlin (1997).

Convention (2) implicitly means that the end of a sequence systematically coincides with the exit from a state. This very specific assumption has the undesirable consequence that only semi-Markov chains without absorbing states can make such a contribution to the likelihood. Since we wish to define semi-Markov chains as a true generalization of Markov chains, the contribution of the state sequence to the complete-data likelihood $f\left(s_0^{\tau-1}, x_0^{\tau-1}; \theta\right)$ can be defined as

$$\pi_{s_0} d_{s_0}\left(u_0\right) \left\{ \prod_{r=1}^{R-1} p_{s_{r-1}s_r} d_{s_r}\left(u_r\right) \right\} p_{s_{R-1}s_R} D_{s_R}\left(u_R\right) I\left(\sum_{r=0}^{R} u_r = \tau\right).$$

This new assumption corresponds to a more general statement of the problem but generates some difficulties regarding the final right-censored sojourn time interval which cannot be used in the estimation procedure. The rationale behind the corresponding estimator is somewhat similar to Cox's partial likelihood idea (Cox 1975) in the sense that it is derived by maximizing part of the likelihood function (see Section 3.3 where this point is illustrated). Nevertheless, the aim underlying the factorization of the likelihood is clearly different from that emphasized by Cox. In the sequel, this estimator will be referred to as the partial likelihood estimator and will serve as a reference.

Let us consider the complete-data likelihood where both the outputs $x_0^{\tau-1}$ and the states $s_0^{\tau+u}$ of the underlying semi-Markov chain are observed

$$\begin{aligned} & f\left(s_0^{\tau+u}, x_0^{\tau-1}; \theta\right) \\ = \; & P\left(S_0^{\tau-1} = s_0^{\tau-1}, S_{\tau-1+v} = s_{\tau-1}, v = 1, ..., u, S_{\tau+u} \neq s_{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}; \theta\right). \end{aligned}$$

In this new specification of the complete-data problem, the state sequence is completed up to the exit from the state occupied at time $\tau - 1$, which is assumed to be a nonabsorbing state. The estimator based on this specification of the complete-data problem will be termed the complete likelihood estimator.
The contribution of the state sequence to the complete-data likelihood is thus

$$\pi_{s_0} d_{s_0}(u_0) \prod_{r=1}^{R} p_{s_{r-1}s_r} d_{s_r}(u_r) I\left(\sum_{r=0}^{R-1} u_r < \tau \leq \sum_{r=0}^{R} u_r\right).$$

Note that, in the case of a final absorbing state j, the contribution of the state sequence to the complete-data likelihood ends simply with a product of $\widetilde{p}_{jj}$ up to time $\tau - 1$. This case is indeed trivial since there is no need to estimate transition probabilities or a state occupancy distribution.

The objective of the estimation procedure is to find the estimate of θ which maximizes the likelihood of the observed sequence $x_0^{\tau-1}$

$$L(\theta) = \sum_{s_0,...,s_{\tau-1}} \sum_{u} f\left(s_0^{\tau+u}, x_0^{\tau-1}; \theta\right),$$

where $\sum_{s_0,...,s_{\tau-1}}$ means sum on every possible state sequence of length τ and $\sum_u$ means sum on every supplementary duration from time τ spent in the state occupied at time $\tau - 1$.

Instead of the successively visited states, the sojourn times and the outputs emitted in these states, only the outputs are observed. Hence, we are faced with an incomplete-data problem and the EM algorithm (Baum et al., 1970; Dempster, Laird, and Rubin, 1977; McLachlan and Krishnan, 1997) is a natural candidate for deriving the nonparametric maximum likelihood estimator. Let $\theta^{(k)}$ denote the current value of θ at iteration k. The conditional expectation of the complete-data log-likelihood is thus given by

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

The EM algorithm maximizes $L(\theta)$ by iteratively maximizing $Q(\theta|\theta^{(k)})$ over θ. The next value $\theta^{(k+1)}$ is chosen as

$$\theta^{(k+1)} = \arg\max_{\theta} \left\{Q(\theta|\theta^{(k)})\right\}.$$

Each iteration of the EM algorithm increases $L(\theta)$ and, generally, the sequence of reestimated parameters $\theta^{(k)}$ converge to a local maximum of $L(\theta)$. The conditional expectation $Q(\theta|\theta^{(k)})$

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

$$\begin{aligned} Q(\theta|\theta^{(k)}) &= Q_\pi\left(\{\pi_j\}_{j=0}^{J-1}|\theta^{(k)}\right) + \sum_{i=0}^{J-1} Q_p\left(\{p_{ij}\}_{j=0}^{J-1}|\theta^{(k)}\right) \\ &\quad + \sum_{j=0}^{J-1} Q_d\left(\{d_j(u)\}|\theta^{(k)}\right) I(p_{jj}=0) + \sum_{j=0}^{J-1} Q_b\left(\{b_j(y)\}_{y=0}^{Y-1}|\theta^{(k)}\right) \end{aligned} \tag{3}$$

with

$$Q_\pi\left(\{\pi_j\}_{j=0}^{J-1}|\theta^{(k)}\right) = \sum_j P\left(S_0=j|X_0^{\tau-1}=x_0^{\tau-1};\theta^{(k)}\right)\log\pi_j, \tag{4}$$

$$Q_p\left(\{p_{ij}\}_{j=0}^{J-1}|\theta^{(k)}\right) = \sum_{j\neq i}\sum_{t=0}^{\tau-2} P\left(S_{t+1}=j, S_t=i|X_0^{\tau-1}=x_0^{\tau-1};\theta^{(k)}\right)\log p_{ij}, \tag{5}$$

$$\begin{aligned} & Q_d\left(\{d_j(u)\}|\theta^{(k)}\right) \\ = & \sum_u \left\{\sum_{t=0}^{\tau-2} P\left(S_{t+u+1}\neq j, S_{t+u-v}=j, v=0,\ldots,u-1, S_t\neq j|X_0^{\tau-1}=x_0^{\tau-1};\theta^{(k)}\right)\right. \\ & \left. + P\left(S_u\neq j, S_{u-v}=j, v=1,\ldots,u|X_0^{\tau-1}=x_0^{\tau-1};\theta^{(k)}\right)\right\}\log d_j(u) \end{aligned} \tag{6}$$

and

$$Q_b\left(\{b_j(y)\}_{y=0}^{Y-1}|\theta^{(k)}\right) = \sum_{y=0}^{Y-1}\sum_{t=0}^{\tau-1} P\left(X_t=y, S_t=j|X_0^{\tau-1}=x_0^{\tau-1};\theta^{(k)}\right)\log b_j(y). \tag{7}$$

Reestimation formulae are obtained by independently maximizing each of these terms. In the sequel, the quantities involved in (4) (5) (6) (7) will be termed reestimation quantities. Therefore, the practical implementation of the E-step of the EM algorithm by the forward-backward algorithm consists in computing these reestimation quantities for all sequences of the sample, all times t and all states j.

3.2 Forward-backward algorithm

In the hidden Markov chain case, the forward-backward algorithm is based on the following decomposition of the smoothed probability $L_j(t)$

$$\begin{aligned} L_j(t) &= P\left(S_t=j|X_0^{\tau-1}=x_0^{\tau-1}\right) \\ &= \frac{P\left(X_{t+1}^{\tau-1}=x_{t+1}^{\tau-1}|S_t=j\right)}{P\left(X_{t+1}^{\tau-1}=x_{t+1}^{\tau-1}|X_0^t=x_0^t\right)} P\left(S_t=j|X_0^t=x_0^t\right) \\ &= B\text{HMC}_j(t)\, F\text{HMC}_j(t) \end{aligned} \tag{8}$$

which expresses the conditional independence between the past and the future of the process at each time t. Devijver (1985) showed that the quantities $F_{\text{HMC}_j}(t)$ can be computed by a forward pass through the observed sequence $x_0^{\tau-1}$ (i.e. from 0 to $\tau-1$) while either the quantities $B_{\text{HMC}_j}(t)$ or $L_j(t)$ can be computed by a backward pass through $x_0^{\tau-1}$ (i.e. from $\tau-1$ to 0). This gives an algorithm whose complexity is $O\left(J^2\tau\right)$-time and which is immune to numerical underflow problems.

In the case of a hidden semi-Markov chain, the forward-backward algorithm is based on the following decomposition

$$\begin{aligned} L1_j(t) &= P\left(S_{t+1} \neq j, S_t = j | X_0^{\tau-1} = x_0^{\tau-1}\right) \\ &= \frac{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1} | S_{t+1} \neq j, S_t = j\right)}{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1} | X_0^t = x_0^t\right)} P\left(S_{t+1} \neq j, S_t = j | X_0^t = x_0^t\right) \\ &= B_j(t) F_j(t) \end{aligned} \tag{9}$$

which expresses the conditional independence between the past and the future of the process at state change times.

In the case of a hidden Markov chain, decomposition (8) naturally fits the EM estimate requirements while, in the case of a hidden semi-Markov chain, decomposition (9) does not directly fit the EM estimate requirements. The fact that the initially proposed forward-backward algorithm (Ferguson, 1980) only allowed computation of $P\left(S_{t+1} \neq j,\ S_t = j |\ X_0^{\tau-1} = x_0^{\tau-1}\right)$ for each time t and each state j instead of $P\left(S_t = j | X_0^{\tau-1} = x_0^{\tau-1}\right)$ resulted in a very complex reestimation formula for the observation probabilities.

Guédon and Cocozza-Thivent (1990) showed that the quantities $F_j(t)$ can be computed by a forward pass through the observed sequence $x_0^{\tau-1}$ while either the quantities $B_j(t)$ or $L1_j(t)$ can be computed by a backward pass through $x_0^{\tau-1}$. The backward recursion can then be adapted to compute the smoothed probabilities $L_j(t) = P\left(S_t = j | X_0^{\tau-1} = x_0^{\tau-1}\right)$ for each time t and each state j. This indirect fit of the conditional independence properties of a hidden semi-Markov chain with the EM estimate requirements is one of the key difficulties when estimating hidden semi-Markov chains. In the proposal of Guédon and Cocozza-Thivent (1990), the price paid for this indirect fit was a backward recursion whose complexity in time was cubic instead of quadratic in the worst case for the forward recursion. In the following, we will show that it is possible to design a backward recursion 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 $L_j(t) = P\left(S_t = j | X_0^{\tau-1} = x_0^{\tau-1}\right)$ instead of $L1_j(t) = P\left(S_{t+1} \neq j,\ S_t = j |\ X_0^{\tau-1} = x_0^{\tau-1}\right)$ does not entail a change in the order of magnitude of the algorithm complexity.

The forward recursion is given by (see Appendix A for details of the derivation),

$t = 0, \ldots, \tau-2; j = 0, \ldots, J-1$:

$$
\begin{aligned}
F_j(t) &= P\left(S_{t+1} \neq j, S_t = j | X_0^t = x_0^t\right) \\
&= \frac{b_j(x_t)}{N_t}\left[\sum_{u=1}^{t}\left\{\prod_{v=1}^{u-1}\frac{b_j(x_{t-v})}{N_{t-v}}\right\} d_j(u) \sum_{i\neq j} p_{ij} F_i(t-u)\right. \\
&\quad \left. + \left\{\prod_{v=1}^{t}\frac{b_j(x_{t-v})}{N_{t-v}}\right\} d_j(t+1)\,\pi_j\right], \qquad (10)
\end{aligned}
$$

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

A key difference with respect to the presentation in Guédon and Cocozza-Thivent (1990) concerns the censoring at time $\tau - 1$ of the sojourn time in the last visited state. Using a similar argument as in (10), we obtain for time $\tau - 1$,
$j = 0, \ldots, J-1$:

$$
\begin{aligned}
F_j(\tau-1) &= P\left(S_{\tau-1} = j | X_0^{\tau-1} = x_0^{\tau-1}\right) \\
&= \frac{b_j(x_{\tau-1})}{N_{\tau-1}}\left[\sum_{u=1}^{\tau-1}\left\{\prod_{v=1}^{u-1}\frac{b_j(x_{\tau-1-v})}{N_{\tau-1-v}}\right\} D_j(u) \sum_{i\neq j} p_{ij} F_i(\tau-1-u)\right. \\
&\quad \left. + \left\{\prod_{v=1}^{\tau-1}\frac{b_j(x_{\tau-1-v})}{N_{\tau-1-v}}\right\} D_j(\tau)\,\pi_j\right]. \qquad (11)
\end{aligned}
$$

The exact time spent in the last visited state is unknown, only the minimum time spent in this state is known. Therefore, the probability mass functions of the sojourn times in state j of the general forward recursion formula (10) are replaced by the corresponding survivor functions in (11).

The normalizing factor N_t is directly obtained during the forward recursion. Using a similar argument as in (10), we obtain,
$t = 0, \ldots, \tau - 1$:

$$
\begin{aligned}
N_t &= P\left(X_t = x_t | X_0^{t-1} = x_0^{t-1}\right) \\
&= \sum_j P\left(S_t = j, X_t = x_t | X_0^{t-1} = x_0^{t-1}\right) \\
&= \sum_j b_j(x_t)\left[\sum_{u=1}^{t}\left\{\prod_{v=1}^{u-1}\frac{b_j(x_{t-v})}{N_{t-v}}\right\} D_j(u) \sum_{i\neq j} p_{ij} F_i(t-u)\right. \\
&\quad \left. + \left\{\prod_{v=1}^{t}\frac{b_j(x_{t-v})}{N_{t-v}}\right\} D_j(t+1)\,\pi_j\right]. \qquad (12)
\end{aligned}
$$

The backward recursion consists of computing $L_j(t) = P\left(S_t = j | X_0^{\tau-1} = x_0^{\tau-1}\right)$ for each state j backward from time $\tau - 1$ to time 0. The backward recursion is initialized for $t = \tau - 1$ by,
$j = 0, \ldots, J-1$:

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

Compared to the proposal of Guédon and Cocozza-Thivent (1990), the major change consists of a new derivation of the quantities $L_j(t)$. The key point here lies in the rewriting of $L_j(t)$ as three terms, $L1_j(t)$, $L_j(t+1)$ computed at the previous step and a third term which expresses the entrance into state j

$$\begin{aligned} L_j(t) &= P\left(S_t=j|X_0^{\tau-1}=x_0^{\tau-1}\right) \\ &= P\left(S_{t+1}\neq j, S_t=j|X_0^{\tau-1}=x_0^{\tau-1}\right)+P\left(S_{t+1}=j|X_0^{\tau-1}=x_0^{\tau-1}\right) \\ &\quad -P\left(S_{t+1}=j, S_t\neq j|X_0^{\tau-1}=x_0^{\tau-1}\right) \\ &= L1_j(t)+L_j(t+1)-P\left(S_{t+1}=j, S_t\neq j|X_0^{\tau-1}=x_0^{\tau-1}\right). \end{aligned} \tag{13}$$

The backward recursion is based on $L1_j(t)$ (see Appendix A for details of the derivation), $t=\tau-2,\ldots,0; j=0,\ldots,J-1$:

$$\begin{aligned} L1_j(t) &= \left[\sum_{k\neq j}\left[\sum_{u=1}^{\tau-2-t}\frac{L1_k(t+u)}{F_k(t+u)}\left\{\prod_{v=0}^{u-1}\frac{b_k(x_{t+u-v})}{N_{t+u-v}}\right\}d_k(u)\right.\right. \\ &\quad \left.\left.+\left\{\prod_{v=0}^{\tau-2-t}\frac{b_k(x_{\tau-1-v})}{N_{\tau-1-v}}\right\}D_k(\tau-1-t)\right]p_{jk}\right]F_j(t). \end{aligned} \tag{14}$$

The third term in (13) is given by (see Appendix A for details of the derivation), $t=\tau-2,\ldots,0; j=0,\ldots,J-1$:

$$\begin{aligned} &P\left(S_{t+1}=j, S_t\neq j|X_0^{\tau-1}=x_0^{\tau-1}\right) \\ &= \left[\sum_{u=1}^{\tau-2-t}\frac{L1_j(t+u)}{F_j(t+u)}\left\{\prod_{v=0}^{u-1}\frac{b_j(x_{t+u-v})}{N_{t+u-v}}\right\}d_j(u)\right. \\ &\quad \left.+\left\{\prod_{v=0}^{\tau-2-t}\frac{b_j(x_{\tau-1-v})}{N_{\tau-1-v}}\right\}D_j(\tau-1-t)\right]\sum_{i\neq j}p_{ij}F_i(t). \end{aligned} \tag{15}$$

The computation of $L_j(t)$ may appear at first sight relatively intricate but, in fact, the computations of $L1_j(t)=P\left(S_{t+1}\neq j,\ S_t=j|\ X_0^{\tau-1}=x_0^{\tau-1}\right)$ in (14) and $P\left(S_{t+1}=j,\ S_t\neq j|\ X_0^{\tau-1}=x_0^{\tau-1}\right)$ in (15) may easily be performed by introducing the following auxiliary quantities

$$\begin{aligned} G_j(t+1,u) &= \frac{L1_j(t+u)}{F_j(t+u)}\left\{\prod_{v=0}^{u-1}\frac{b_j(x_{t+u-v})}{N_{t+u-v}}\right\}d_j(u), \qquad u=1,\ldots,\tau-2-t, \\ G_j(t+1,\tau-1-t) &= \left\{\prod_{v=0}^{\tau-2-t}\frac{b_j(x_{\tau-1-v})}{N_{\tau-1-v}}\right\}D_j(\tau-1-t), \end{aligned}$$

and

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

At each time t, these auxiliary quantities should be precomputed.
Then,

$$L1_j(t) = \left\{ \sum_{k \neq j} G_k(t+1) p_{jk} \right\} F_j(t), \tag{16}$$

and

$$\begin{aligned} & P\left(S_{t+1} = j, S_t \neq j | X_0^{\tau-1} = x_0^{\tau-1}\right) \\ = & \frac{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1} | S_{t+1} = j, S_t \neq j\right)}{P\left(X_{t+1}^{\tau-1} = x_{t+1}^{\tau-1} | X_0^t = x_0^t\right)} P\left(S_{t+1} = j, S_t \neq j | X_0^t = x_0^t\right) \\ = & G_j(t+1) \sum_{i \neq j} p_{ij} F_i(t). \end{aligned}$$

An implementation of this forward-backward algorithm is proposed in Appendix B in pseudo-code form where issues concerning computational complexity are discussed.

Because for each $t < \tau - 1$, $L1_j(t) = B_j(t) F_j(t)$, the backward recursion based on $B_j(t)$ is directly deduced from (14)

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

and the third term in (13) can be rewritten as

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

Hence, a variant of the backward recursion presented above can also be built on $B_j(t)$.

3.3 Parameter reestimation

The reestimation formulae for the parameters of a hidden semi-Markov chain are obtained by maximizing the different terms of $Q(\theta|\theta^{(k)})$ (see the decomposition (3)), each term depending on a given subset of θ. In the following, for each parameter subset, we simply give the reestimation formula which is directly deduced from the maximization of (4) (5) (6) (7) in the nonparametric framework.

For the parameters of the embedded first-order Markov chain, we obtain:

- initial probabilities

$$\pi_j^{(k+1)} = P\left(S_0 = j|X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right) = L_j(0), \tag{17}$$

- transition probabilities

$$\begin{aligned} p_{ij}^{(k+1)} &= \frac{\sum_{t=0}^{\tau-2} P\left(S_{t+1} = j, S_t = i|X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right)}{\sum_{t=0}^{\tau-2} P\left(S_{t+1} \neq i, S_t = i|X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right)} \\ &= \frac{\sum_{t=0}^{\tau-2} G_j(t+1)\, p_{ij} F_i(t)}{\sum_{t=0}^{\tau-2} L1_i(t)}. \end{aligned} \tag{18}$$

The numerator quantity in (18) is directly extracted from the computation of $L1_i(t)$ (16).

For each nonabsorbing state j, we have for the state occupancy distribution

$$\begin{aligned} & Q_d\left(\{d_j(u)\}|\theta^{(k)}\right) \\ = & \sum_u \left\{ \sum_{t=0}^{\tau-2} P\left(S_{t+u+1} \neq j, S_{t+u-v} = j, v = 0, \ldots, u-1, S_t \neq j|X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right) \right. \\ & \left. + P\left(S_u \neq j, S_{u-v} = j, v = 1, \ldots, u|X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right) \right\} \log d_j(u) \qquad (19) \\ = & \sum_u \eta_{j,u}^{(k)} \log d_j(u). \qquad (20) \end{aligned}$$

The general term in (19) for $u \leq \tau - 2 - t$ is directly extracted from the computation of $L_j(t)$

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

while for $u > \tau - 2 - t$, we obtain

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

The computation of these quantities is easily mixed with the computation of (see Appendix A)

$$
\begin{aligned}
& P\left(S_{\tau-1-v} = j, v = 0, \ldots, \tau-2-t, S_t \neq j | X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right) \\
= & \left\{ \prod_{v=0}^{\tau-2-t} \frac{b_j\left(x_{\tau-1-v}\right)}{N_{\tau-1-v}} \right\} D_j(\tau-1-t) \sum_{i \neq j} p_{ij} F_i(t).
\end{aligned}
$$

The term in (19) corresponding to the time spent in the initial state requires some supplementary computation at time $t = 0$,
$u \leq \tau - 1$:

$$
\begin{aligned}
& P\left(S_u \neq j, S_{u-v} = j, v = 1, \ldots, u | X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right) \\
= & \frac{L1_j(u-1)}{F_j(u-1)} \left\{ \prod_{v=1}^{u} \frac{b_j\left(x_{u-v}\right)}{N_{u-v}} \right\} d_j(u) \pi_j,
\end{aligned}
$$

$u > \tau - 1$:

$$
P\left(S_u \neq j, S_{u-v} = j, v = 1, \ldots, u | X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right) = \left\{ \prod_{v=1}^{\tau} \frac{b_j\left(x_{\tau-v}\right)}{N_{\tau-v}} \right\} d_j(u) \pi_j.
$$

By noting that

$$
\begin{aligned}
& \sum_u \left\{ \sum_{t=0}^{\tau-2} P\left(S_{t+u+1} \neq j, S_{t+u-v} = j, v = 0, \ldots, u-1, S_t \neq j | X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right) \right. \\
& \left. + P\left(S_u \neq j, S_{u-v} = j, v = 1, \ldots, u | X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right) \right\} \\
= & \sum_{t=0}^{\tau-2} P\left(S_{t+1} \neq j, S_t = j | X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right) + P\left(S_{\tau-1} = j | X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right) \\
= & \sum_{t=0}^{\tau-2} L1_j(t) + L_j(\tau-1),
\end{aligned}
$$

the reestimated state occupancy probabilities are then given by

$$
\begin{aligned}
d_j^{(k+1)}(u) &= \frac{\eta_{j,u}^{(k)}}{\sum_v \eta_{j,v}^{(k)}} \\
&= \frac{\eta_{j,u}^{(k)}}{\sum_{t=0}^{\tau-2} L1_j(t) + L_j(\tau-1)}. \qquad (21)
\end{aligned}
$$

The reestimated observation probabilities are given by

$$
\begin{aligned}
b_j^{(k+1)}(y) &= \frac{\sum_{t=0}^{\tau-1} P\left(X_t = y, S_t = j | X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right)}{\sum_{t=0}^{\tau-1} P\left(S_t = j | X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right)} \\
&= \frac{\sum_{t=0}^{\tau-1} L_j(t) I(x_t = y)}{\sum_{t=0}^{\tau-1} L_j(t)}. \qquad (22)
\end{aligned}
$$

It should be noted that all the quantities involved in the reestimation formulae (17) (18) (21) (22) are directly extracted from the backward recursion with only a few additional computations (the only supplementary computations concern the contributions at time $t = 0$ and the contributions of the time spent in the last visited state to the reestimation quantities of the state occupancy distributions).

The only difference between the complete likelihood estimator and the partial likelihood estimator (see Section 3.1) lies in the reestimation of the state occupancy distributions. In the case of the partial likelihood estimator, the information relative to the time spent in the last visited state is not used in the estimation procedure. The term of $Q(\theta|\theta^{(k)})$ for the state j occupancy distribution is thus given by

$$
\begin{aligned}
& \widetilde{Q}_d\left(\{d_j(u)\}|\theta^{(k)}\right) \\
= & \sum_u \left\{ \sum_{t=0}^{\tau-2-u} P\left(S_{t+u+1} \neq j, S_{t+u-v} = j, v = 0, \ldots, u-1, S_t \neq j | X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right) \right. \\
& \left. + P\left(S_u \neq j, S_{u-v} = j, v = 1, \ldots, u | X_0^{\tau-1} = x_0^{\tau-1}; \theta^{(k)}\right) I(u \leq \tau-1) \right\} \log d_j(u) \\
= & \sum_u \widetilde{\eta}_{j,u}^{(k)} \log d_j(u)
\end{aligned}
$$

with

$$
\sum_u \widetilde{\eta}_{j,u}^{(k)} = \sum_{t=0}^{\tau-2} L1_j(t).
$$

3.4 Practical aspects

To provide both a regularization capability to the estimator and obtain parsimonious models, we propose, for the state occupancy distributions, to replace the nonparametric M-step of the EM

algorithm (21) by a parametric M-step in the practical estimation procedure. Model parsimony is a critical issue for the small sample case studies discussed in Section 5. Recall that the EM algorithm alternates two steps, the E-step which consists in calculating $Q(\theta|\theta^{(k)})$ and the M-step which consists in choosing the next parameter value $\theta^{(k+1)}$ that maximizes $Q(\theta|\theta^{(k)})$ over θ (Dempster et al., 1977; McLachlan and Krishnan, 1997). In our context (see Section 3.3), the outputs of the E-step for the state j occupancy distribution are the reestimation quantities $\left\{\eta_{j,u}^{(k)}\right\}$ (see 20). Hence, the reestimation quantities $\eta_{j,u}^{(k)}$ can be considered as a pseudo-sample (with real frequencies) generated by a given parametric state j occupancy distribution in order to design a parametric M-step.

In the following, we define as possible parametric state occupancy distributions binomial distributions, Poisson distributions and negative binomial distributions with an additional shift parameter d ($d \geq 1$) which defines the minimum sojourn time in a given state.

The binomial distribution with parameters d, n and p ($q = 1 - p$), $\mathrm{B}(d, n, p)$ where $0 \leq p \leq 1$, is defined by

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

The Poisson distribution with parameters d and λ, $\mathrm{P}(d, \lambda)$, where λ is a real number ($\lambda > 0$), is defined by

$$d_j(u) = \frac{e^{-\lambda}\lambda^{u-d}}{(u-d)!}, \qquad u = d, d+1, \ldots$$

The negative binomial distribution with parameters d, r and p , $\mathrm{NB}(d, r, p)$, where r is a real number ($r > 0$) and $0 < p \leq 1$, is defined by

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

The shift parameter d being fixed, the parameters n and p of the binomial distribution $\mathrm{B}(d, n, p)$, λ of the Poisson distribution $\mathrm{P}(d, \lambda)$ and r and p of the negative binomial distribution $\mathrm{NB}(d, r, p)$ are estimated by classical point estimation procedures from the reestimation quantities $\left\{\eta_{j,u}^{(k)}\right\}$ (Johnson, Kotz, and Kemp, 1993). For a given nonabsorbing state j, a parametric state occupancy distribution is estimated for each possible shift parameter value $d = 1, \ldots, \min\left(u : \eta_{j,u} > 0\right)$. The state occupancy distribution which gives the maximum likelihood of the reestimation quantities is retained. This procedure can be extended by testing not only different possible shift parameters but also different parametric hypotheses (chosen from binomial, Poisson and negative binomial). It should be noted that the proposed approach for a parametric M-step is somewhat ad-hoc (due mainly to the estimation of discrete parameters that define bounds to the support of the state occupancy distributions) but very useful in practice for samples of limited size.

The convergence of the estimation procedure is monitored upon the increase over iterations of the log-likelihood of the observed sequences. This is a direct consequence of one of the main properties of the EM algorithm (see McLachlan and Krishnan (1997), pp. 82-84). The forward

recursion (10) (11) (12) can be used to compute the likelihood of the observed sequence $x_0^{\tau-1}$

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

The log-likelihood of the observed sequence is thus given by

$$\log P\left(X_0^{\tau-1}=x_0^{\tau-1};\theta\right)=\sum_{t=0}^{\tau-1}\log N_t.$$

4 Complementary algorithms for building hidden semi-Markov chains from discrete sequences

The estimation algorithm presented earlier constitutes the core of a coherent methodology for building hidden semi-Markov chains from discrete sequences. But, a model-building methodology is not restricted to the inference stage and is likely to include other classes of algorithms or other uses of previously introduced algorithms, especially, for the validation stage. For instance, the output of the forward-backward algorithm presented in Section 3.2 is basically the state profile for an observed sequence $x_0^{\tau-1}$ given by the smoothed probabilities $L_j(t)$ as a function of the index parameter t. This constitutes a relevant diagnostic tool (Churchill, 1989), especially to detect ambiguous zones where more than a single state is likely to explain the outputs observed in a given zone.

4.1 Viterbi algorithm

It may be interesting in different contexts to have the knowledge of the most likely state sequence associated with the observed sequence $x_0^{\tau-1}$. As an example, this can be used to segment the observed sequence, each successive segment corresponding to a given non-observable state. The most likely state sequence can be obtained by a dynamic programming method, usually referred to as the Viterbi algorithm (Guédon and Cocozza-Thivent, 1990).
Because the state process is a semi-Markov chain, we have for all t

$$\begin{aligned}
&\max_{s_0,\ldots,s_{\tau-1};s_{t+1}\neq s_t}P\left(S_0^{\tau-1}=s_0^{\tau-1},X_0^{\tau-1}=x_0^{\tau-1}\right)\\
=\;&\max_{s_t}\left\{\max_{s_{t+1},\ldots,s_{\tau-1}}P\left(X_{t+1}^{\tau-1}=x_{t+1}^{\tau-1},S_{t+1}^{\tau-1}=s_{t+1}^{\tau-1}|S_{t+1}\neq s_t,S_t=s_t\right)\right.\\
&\left.\times\max_{s_0,\ldots,s_{t-1}}P\left(S_{t+1}\neq s_t,S_0^t=s_0^t,X_0^t=x_0^t\right)\right\}. \qquad (23)
\end{aligned}$$

Let us define

$$\alpha_j(t)=\max_{s_0,\ldots,s_{t-1}}P\left(S_{t+1}\neq j,S_t=j,S_0^{t-1}=s_0^{t-1},X_0^t=x_0^t\right).$$

Hence, decomposition (23) can be rewritten as

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

On the basis of this decomposition, we can build the following recursion,
$t=0,\ldots,\tau-2;j=0,\ldots,J-1:$

$$\begin{aligned} \alpha_j\left(t\right) &= \max_{s_0,\ldots,s_{t-1}} P\left(S_{t+1}\neq j,S_t=j,S_0^{t-1}=s_0^{t-1},X_0^t=x_0^t\right) \\ &= b_j\left(x_t\right)\max\left[\max_{1\leq u\leq t}\left[\left\{\prod_{v=1}^{u-1}b_j\left(x_{t-v}\right)\right\}d_j\left(u\right)\max_{i\neq j}\left\{p_{ij}\alpha_i\left(t-u\right)\right\}\right],\right. \\ &\quad \left.\left\{\prod_{v=1}^{t}b_j\left(x_{t-v}\right)\right\}d_j\left(t+1\right)\pi_j\right]. \end{aligned} \tag{24}$$

The right censoring of the sojourn time in the last visited state makes particular the case
$t=\tau-1,$
$j=0,\ldots,J-1:$

$$\begin{aligned} \alpha_j\left(\tau-1\right) &= \max_{s_0,\ldots,s_{\tau-2}} P\left(S_{\tau-1}=j,S_0^{\tau-2}=s_0^{\tau-2},X_0^{\tau-1}=x_0^{\tau-1}\right) \\ &= b_j\left(x_{\tau-1}\right)\max\left[\max_{1\leq u\leq\tau-1}\left[\left\{\prod_{v=1}^{u-1}b_j\left(x_{\tau-1-v}\right)\right\}D_j\left(u\right)\max_{i\neq j}\left\{p_{ij}\alpha_i\left(\tau-1-u\right)\right\}\right],\right. \\ &\quad \left.\left\{\prod_{v=1}^{\tau-1}b_j\left(x_{\tau-1-v}\right)\right\}D_j\left(\tau\right)\pi_j\right]. \end{aligned} \tag{25}$$

The likelihood of the optimal state sequence associated with the observed sequence $x_0^{\tau-1}$ is $\max_j\left\{\alpha_j\left(\tau-1\right)\right\}$.

The Viterbi recursion is the equivalent in terms of dynamic programming of the forward recursion (summation in (10) (11) replaced by maximization in (24) (25)). Therefore, the proposals made for an efficient implementation of the forward recursion in Appendix B can be directly transposed to the Viterbi algorithm. For instance, the quantities $\max_{i\neq j}\left\{p_{ij}\alpha_i\left(t-u\right)\right\}$ can be computed once (at time $t-u$) and stored for further use, or the (partial) products $\prod_{v=1}^{u-1}b_j\left(x_{t-v}\right)$ can be computed recursively during the maximization on u in (24) (25). As for the forward recursion, the complexity is $O\left(J\tau\left(J+\tau\right)\right)$-time in the worst case and $O\left(J\tau\right)$-space. For each time t and each state j, two backpointers can be recorded, the first giving the optimal preceding state and the second the optimal preceding time of transition from this preceding state. These backpointers can be used in a second stage - often referred to as 'backtracking' - to retrieve the optimal state sequence. The backtracking procedure consists in tracing backward along the couple of backpointers from the optimal final state (at time $\tau-1$) to the optimal initial state (at time 0).

4.2 Families of characteristic distributions of a discrete hidden semi-Markov chain

For samples of discrete sequences, Guédon (1998, 1999) proposed a validation methodology relying on the fit of different types of characteristic distributions computed from model parameters to their empirical equivalents which can be extracted from data. The three points of view used for the specification of point processes (Cox and Isham, 1980), i.e. the intensity, interval and counting points of view, were transposed to define characteristic distributions of a discrete hidden semi-Markov chain. These characteristic distributions can be defined both for the state process and for the output process. In the former case, these characteristic distributions can be fitted to their empirical equivalents extracted from the optimal state sequences computed by the Viterbi algorithm (see Section 4.1). Note that in the point process context, 'intensity' refers to conditional distributions, while in our context, the intensity characteristics are unconditional distributions. Intensity refers to the random state/output occupied at a fixed time step while interval refers to the random time taken to reach a fixed state/output or to the random time spent in a fixed state/output. Finally, counting refers to the random number of occurrences of a fixed pattern in a sequence of fixed length. In the case of discrete-time discrete-state-space stochastic processes, characteristics take the form of families of discrete distributions, one distribution per time step for the intensity point of view and one distribution per state/output for the interval and counting points of view. In the case of a multivariate output process, families of characteristic distributions are defined for each elementary output process which are assumed to be mutually independent.

The most obvious characteristic distributions are the unconditional distributions of being in output y at successive times t (intensity point of view)

$$\left(P\left(X_t = y\right); y = 0, \ldots, Y-1\right).$$

For each output y, we define the three following types of interval and the associated distributions:

- time to the first occurrence of output y (or first passage time in output y)

$$h_y\left(t\right) = P\left(X_t = y, X_{t-v} \neq y, v = 1, \ldots, t\right), \quad t = 0, 1, \ldots,$$

- recurrence time in output y

$$f_{yy}\left(u\right) = P\left(X_{t+u} = y, X_{t+u-v} \neq y, v = 1, \ldots, u-1 | X_t = y\right), \quad u = 1, 2, \ldots,$$

- sojourn time in output y (or run length of output y)

$$d_y\left(u\right) = P\left(X_{t+u+1} \neq y, X_{t+u-v} = y, v = 0, \ldots, u-2 | X_{t+1} = y, X_t \neq y\right), \quad u = 1, 2, \ldots \quad (26)$$

In the semi-Markov chain case, the sojourn time distributions belong to the model definition while their transpositions to the output process are characteristic distributions. First passage times and recurrence times are counted in number of transitions while sojourn times are counted in number of time steps.

For each output y, we also define the two following types of counting measure and the associated distributions:

- Number of runs (or clumps) of output y per sequence of length τ

$$P\left(N_y\left(\tau-1\right)=n\right)=P\left(\sum_{t=1}^{\tau-1}I\left(x_t=y,x_{t-1}\neq y\right)+I\left(x_0=y\right)=n\right),\quad n=0,\ldots,\frac{\tau+1}{2}.$$

In this definition, the start of runs are counted. Hence, both the complete time intervals, such as defined in (26), and the final right-censored time intervals are counted.

- Number of occurrences of output y per sequence of length τ

$$P\left(N_y\left(\tau-1\right)=n\right)=P\left(\sum_{t=0}^{\tau-1}I\left(x_t=y\right)=n\right),\quad n=0,\ldots,\tau.$$

In the practical case of a sample of sequences of different lengths, the counting distributions become finite mixtures where the mixing weights are the probabilities of each possible sequence length

$$P\left(N_y=n\right)=\sum_{\tau}P\left(N_y\left(\tau-1\right)=n|\Upsilon=\tau\right)P\left(\Upsilon=\tau\right).$$

The families of characteristic distributions play different roles in the validation of estimated models. The probabilities of the outputs as a function of the index parameter (intensity point of view) give an overview of process 'dynamics'. This overview is complemented for the initial transient phases by the distributions of the time to the first occurrence of an output. The local dependencies are expressed both in the recurrence time distributions, the sojourn time distributions and the distributions of the number of runs of an output per sequence. These three types of characteristic distributions can help to highlight the scattered or aggregate distribution of a given output along sequences. Algorithms for computing characteristic distributions both for the state process and the output process are fully detailed in Guédon (1999).

5 Application to the analysis of branching and flowering patterns

We will now illustrate the use of hidden semi-Markov chains by the analysis of branching and flowering patterns in plants using two examples: branching on apple tree and branching and axillary flowering on apricot tree. In this context, the model is viewed as a useful tool with

which complex branching/flowering patterns contained in the studied samples of sequences can be summarized and compared. This type of application of hidden semi-Markov chains is reviewed in Guédon et al. (2001) on the basis of an enlarged set of examples and with a deeper discussion of biological issues.

5.1 Branching of apple tree trunk annual shoots

Seven apple cultivars (*Malus domestica* Borkh, *Rosaceae*) chosen for their diverse growth and fruiting habits were planted in Montpellier (south of France). Twenty trees per cultivar, grafted on rootstock M.7, were planted in the field and cut back to one bud one year after transplantation. The trees were then allowed to develop without pruning. The location of the immediate offspring shoots (offspring shoots developed without delay with respect to the parent node establishment date) was recorded after one year of growth while the location of one-year-delayed offspring shoots was recorded after two years of growth. Among these one-year-delayed offspring shoots, short shoots, long shoots and flowering shoots were distinguished. In these measurements, we qualified both the immediate branching which follows the establishment growth from the base to the top and the one-year-delayed branching organized from the top of the parent shoot. After an exploratory analysis of the sample of sequences, we chose to focus on the one-year-delayed branching structure and describe the first annual shoots of the trunks node by node from the top to the base where, for each node, the type of axillary production chosen among latent bud (0), one-year-delayed short shoot (1), one-year-delayed long shoot (2), one-year-delayed flowering shoot (3), and immediate shoot (4) was recorded (see Figure 1). The branching structure of the first annual shoot of the trunks after two years of growth is assumed to be a good predictor of the adult structure of the tree.

In this example, we will mainly focus on the sample of sequences corresponding to the cultivar 'Reinette B.' and, to a lesser extent, on that corresponding to the cultivar 'Belrène'. The three sequences shown in Figure 2 illustrate the measurements for the cultivar 'Reinette B.'. These three sequences exhibit a succession of six well-differentiated zones. Each of these six zones is characterized by a given mixture of axillary productions: (1, 2) for the first zone, (0, 3) for the second, 4 for the third, 0 for the fourth, (0, 1, 2) for the fifth and 0 for the sixth.

For the specification of the initial model $\theta^{(0)}$, we made the hypothesis of an embedded 'left-right' first-order Markov chain composed of 6 successive transient states and a final absorbing state (since some sequences start with a short unbranched apical zone). For each nonabsorbing state, we made the hypothesis of a geometric state occupancy distribution, which is equivalent to making the hypothesis of an underlying first-order Markov chain.

The convergence of the EM algorithm required 22 iterations. The estimation of model parameters conserved only the transitions between consecutive states, except the transition between state 2 and state 4 (see Figure 3); in the initial model specification, transitions from a given state to the three following states were allowed. The state occupancy distributions, particularly from state 2, have a low dispersion which expresses strong structuring in the succession of zones along the annual shoots. Each state is markedly differentiated from the immediately preceding

and following states by the attached observation probabilities. The accuracy of the model is mainly evaluated by the fit of characteristic distributions computed from the model parameters to the corresponding characteristics extracted from the observed sequences (Figures 4 and 5; the histograms represent the characteristics extracted from the sequences). It may be noted that the sample sizes for the characteristics extracted from the sequences are very variable: while there is a single data item per sequence for the counting characteristics (Figures 5c, 5d), there are on average many data items per sequence for the interval characteristics (Figures 5a, 5b). Since the most likely state sequences capture most of the likelihood of the observed sequences, the evaluation of model accuracy and the interpretation of the underlying biological phenomena may also rely on the most likely state sequences computed from the observed sequences by the Viterbi algorithm (Section 4.1). The optimal segmentation of three sequences is presented in Figure 2. States 1 to 6 clearly correspond to six well-differentiated successive zones. The lengths of the segmented zones and the axillary productions observed in these zones reflect the corresponding state occupancy and observation distributions shown in Figure 3. This examination of each individual sequence can indeed be complemented by the fits of the characteristic distributions at the state level to the corresponding characteristics extracted from the most likely state sequences (Figure 6).

The detailed comparison of two apple cultivars ('Reinette B.' and 'Belrène') on the basis of model parameters and characteristics is illustrated in Figures 7 and 8. The main difference between these two cultivars lies in the location of one-year-delayed short shoots. For 'Reinette B.', short shoots are mainly located on the basal part of the main shoot (between ranks 40 and 70 counted from the top) (Figure 4) while they are located mainly on the apical part of the main shoot for 'Belrène' (before rank 25) (Figure 8). The structures of the two models are very similar (see Figures 3 and 7), the main differences being the supplementary initial state (state 1) for 'Belrène' and the different balances between short and long shoots in the basal and apical zones (state 1 of 'Reinette B.' compared to state 2 of 'Belrène' and state 5 of 'Reinette B.' compared to state 6 of 'Belrène'). Most of the similarities between the branching structures of these two cultivars extend to the other cultivars.

5.2 Branching and flowering of apricot tree growth units

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 (Figure 9). 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 strongly 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). The structure of the estimated hidden semi-Markov chain is represented in Figure 10: only the transitions whose probability is greater than 0.03 are represented. The dotted edges correspond to the less probable transitions while the dotted vertices correspond to the less probable states. The underlying semi-Markov chain is composed of two transient states followed by a five-state recurrent class. 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 (before rank 11) while the second transient state corresponds to the end of the transient phase for the flowering variable (see Figure 11). 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.

The main outcome of this study is that the recurrent class is structured by the flowering variable. The number of flowers increases from 1 to 2 and from 2 to 3 flowers but almost never directly from 1 to 3 and, conversely, decreases from 3 to 2 and from 2 to 1 but almost never directly from 3 to 1. This result can be checked precisely by estimating the one-step transition probabilities from the sub-sequences - for the flowering variable - corresponding to the recurrent class extracted by the Viterbi algorithm (the initial phases corresponding to the two initial transient states are removed in this way). We thus obtain $\widehat{p}_{13} = 0.014$ and $\widehat{p}_{31} = 0.038$.

This result is also expressed in the estimated hidden semi-Markov chain (see Figure 10) by the combination of the transition probabilities between states 4 ('1 flower'), 5 ('2 flowers') and 6 ('3 flowers') and the observation probabilities; see the last three rows of the estimated observation probability matrix for the flowering variable $\widehat{B}_f$ corresponding to states 4, 5 and 6

$$\widehat{B}_f = \begin{pmatrix} & \cdots & & \\ 0.29 & \mathbf{0.65} & 0.05 & 0.01 \\ 0.01 & 0.13 & \mathbf{0.85} & 0.01 \\ 0.01 & 0.01 & 0.21 & \mathbf{0.77} \end{pmatrix}.$$

It should be noted that '1 flower' and '3 flowers' cannot be observed together in a single state (see the 2nd and the 4th columns of $\widehat{B}_f$). For these three states, the observation distributions for the branching variable are far less contrasted with a majority of one-year-delayed short shoots; see the last three rows of the estimated observation probability matrix for the branching variable $\widehat{B}_b$ corresponding to states 4, 5 and 6 (the last column corresponding to immediate shoot with systematically a zero entry for these three states is not shown)

$$\widehat{B}_b = \begin{pmatrix} & \cdots & \\ 0.23 & \mathbf{0.63} & 0.14 \\ 0.11 & \mathbf{0.82} & 0.07 \\ 0.09 & \mathbf{0.85} & 0.06 \end{pmatrix}.$$

Note that one-year-delayed shorts shoots are preferentially associated with a high number of flowers while one-year-delayed long shoots are preferentially associated with a small number of flowers; see the observation distributions for state 4 in $\widehat{B}_b$ and $\widehat{B}_f$ compared to the observation distributions for states 5 and 6. This may be interpreted as an inhibitory effect of the differentiated flowers on the vegetative development of the corresponding offspring shoot (note that flower differentiation occurs before the vegetative development of the offspring shoot). At the more macroscopic level, the biological interpretation of the 'remanent' character of the flowering along the growth units has not yet been fully elucidated.

6 Concluding remarks

Determining the appropriate number of states of the embedded first-order Markov chain is a critical issue in the initial model specification. This point is illustrated in Figure 12 which shows the fit of the intensity characteristics for a five-state hidden semi-Markov chain estimated from the apple tree sequences (cultivar 'Reinette B.'). The transient phases at the beginning of the sequences are in this case poorly modeled with respect to the seven-state hidden semi-Markov chain (Figure 4).

Due to the final recurrent class composed of more than one state, the apricot tree example can be used to compare the partial likelihood and the complete likelihood estimates. In the former case, we obtain $2 \log L = -7758.6$ and in the latter case $2 \log L = -7745.7$ which gives a difference of either AIC or BIC of 12.9 (since the number of free parameters and the sample size are identical for the two models compared). The rules of thumb of Jeffreys (1961, Appendix B) suggest that a difference of BIC of at least $2 \log 100 = 9.2$ is needed to deem the model with the higher BIC substantially better. Ignoring the final right-censored sojourn times biases the estimated state occupancy distributions downwards, since a long sojourn time is more likely to contain the censoring time than a short one (a phenomenon called length bias). As an illustration, we obtain for the means of the state 4, 5 and 6 occupancy distributions $\mu_4 = 5.1, \mu_5 = 7, \mu_6 = 9.6$ in the case of the partial likelihood estimates, and $\mu_4 = 7, \mu_5 = 10, \mu_6 = 10.6$ in the case of the complete likelihood estimates. It is well known that censoring at a time which is not a stopping time may introduce bias; see for instance Aalen and Husebye (1991). The end of the last complete sojourn time is not a stopping time because of the need to look beyond the stopping time before deciding to stop.

For the regularization of the state occupancy distributions, a potential solution would consist in incorporating penalty terms in the likelihood. In the framework of the EM algorithm, the E-step is unchanged but for the M-step, the maximization of each term $Q_d\left(\{d_j(u)\}\,|\theta^{(k)}\right)$ corresponding to each nonabsorbing state is replaced by the maximization of

$$Q_d\left(\{d_j(u)\}|\theta^{(k)}\right) - \lambda_j J\left(\{d_j(u)\}\right), \tag{27}$$

where λ_j is a tuning constant and $J(\{d_j(u)\})$ is a roughness penalty. For instance $J()$ may be the sum of squared second differences $J(\{d_j(u)\}) = \sum_u \{(d_j(u+1) - d_j(u)) - (d_j(u) - d_j(u-1))\}^2$.

Green (1990) demonstrated the computational economy and accelerated convergence yielded by employing the one-step-late (OSL) algorithm. For each nonabsorbing state, the OSL algorithm solves

$$DQ_d\left(\{d_j(u)\}|\theta^{(k)}\right) - \lambda_j DJ\left(\left\{d_j^{(k)}(u)\right\}\right) = 0, \tag{28}$$

where D denotes the derivative operator.

The only difference between equation (28) and equating the derivatives of expression (27) to 0 is that in equation (28), the derivatives of the penalty are evaluated at the current value $\left\{d_j^{(k)}(u)\right\}$.

As an alternative to the Viterbi algorithm which determines the state sequence $\widetilde{s}_0^{\tau-1}$ that globally maximizes $P\left(S_0^{\tau-1} = s_0^{\tau-1}, X_0^{\tau-1} = x_0^{\tau-1}\right)$, it is also possible to compute the most likely state sequence on the basis of a local criterion (see Rabiner (1989) and Fredkin and Rice (1992) for discussions of this method in the framework of hidden Markov chains), that is to determine for each t the most likely state

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

The forward-backward algorithm presented in Section 3.2 can be used to compute in this way the optimal state sequence. One strong restriction with this use of the forward-backward algorithm lies in the more global dependency structure induced by the underlying semi-Markov chain compared to a simple Markov chain. Hence, this local optimization is likely to generate short runs of states that intersperse a longer run of another state. In some cases, the resulting sequence may not even be a valid state sequence.

The relevance of the most likely state sequence obtained by the Viterbi algorithm is strongly related to the structural properties of the embedded first-order Markov chain. With an irreducible Markov chain, many different state sequences have approximately the same likelihood, while with a 'left-right' Markov chain (i.e. composed of a succession of transient states and a final absorbing state), the most likely state sequence captures most of the likelihood of the observed sequence. This should be kept in mind in the different possible practical uses of the most likely state sequences. In particular, it was proposed to base an approximate EM iteration on the Viterbi algorithm (see Rabiner (1989) for the application of this principle to hidden Markov chains). The principle is the following: On the basis of observed sequences and corresponding optimal state sequences, event counts can be made for each parameter of the hidden semi-Markov chain from which estimates are directly deduced. One of the main justifications of this alternative estimation algorithm was to avoid the numerical difficulties of the initially proposed forward-backward algorithm. This justification has since become irrelevant.

In summary, the use of the forward-backward algorithm should be restricted to the implementation of the E-step of the EM algorithm and to the computation of the state profiles, while the Viterbi algorithm should not be use as a basis for estimation procedures.

The proposed analysis methodology based on hidden semi-Markov chains is fully implemented in the AMAPmod software (Godin et al., 1997; Godin, Guédon, and Costes, 1999).

References

Aalen, O. O. and Husebye, E. (1991), "Statistical analysis of repeated events forming renewal processes," *Statistics in Medicine* 10, 1227-1240.

Baum, L. E., Petrie, T., Soules, G. and Weiss, N. (1970), "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Annals of Mathematical Statistics*, 41, 164-171.

Braun, J. V. and Müller, H.-G. (1998), "Statistical methods for DNA sequence segmentation," *Statistical Science*, 13, 142-162.

Burge, C., and Karlin, S. (1997), "Prediction of complete gene structures in human genomic DNA," *Journal of Molecular Biology*, 268, 78-94.

Churchill, G. A. (1989), "Stochastic models for heterogeneous DNA sequences," *Bulletin of Mathematical Biology*, 51, 79-94.

Cox, D.R. (1975), "Partial likelihood," *Biometrika*, 62, 269-276.

Cox, D. R., and Isham, V. (1980), *Point Processes*, London: Chapman and Hall.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum likelihood from incomplete data via the EM algorithm (with discussion)," *Journal of the Royal Statistical Society*, Ser. B, 39, 1-38.

Devijver, P. A. (1985), "Baum's forward-backward algorithm revisited," *Pattern Recognition Letters*, 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: Cambridge University Press.

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

Fredkin, D. R., and Rice, J. A. (1992), "Bayesian restoration of single-channel patch clamp recordings," *Biometrics,* 48, 427-448.

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 *Plants to Ecosystems - Advances in Computational Life Sciences*, ed. M. T. Michalewicz, Volume 1, Collingwood, Victoria: CSIRO Publishing, pp. 53-84.

Green P. J. (1990), "On the use of the EM algorithm for penalized likelihood estimation," *Journal of the Royal Statistical Society*, Ser. B, 52, 443-452.

Guédon, Y. (1992), "Review of several stochastic speech unit models," *Computer Speech and Language*, 6, 377-402.

Guédon, Y. (1998), "Hidden semi-Markov chains: A new tool for analyzing nonstationary discrete sequences," In *Proceedings of the 2nd International Symposium on Semi-Markov Models: Theory and Applications*, eds. J. Janssen and N. Limnios, Compiègne, France.

Guédon, Y. (1999), "Computational methods for discrete hidden semi-Markov chains," *Applied Stochastic Models in Business and Industry*, 15, 195-224.

Guédon, Y., Barthélémy, D., Caraglio, Y., and Costes, E. (2001), "Pattern analysis in branching and axillary flowering sequences," *Journal of Theoretical Biology*, 212, 481-520.

Guédon, Y., and Cocozza-Thivent, C. (1990), "Explicit state occupancy modelling by hidden semi-Markov models: Application of Derin's scheme," *Computer Speech and Language*, 4, 167-192.

Jeffreys, H. (1961), *Theory of Probability* (3rd ed.), Oxford: Oxford University Press.

Johnson, N. L., Kotz, S., and Kemp, A. W. (1993), *Univariate Discrete Distributions* (2nd ed.), New York: Wiley.

Kitagawa, G. (1987), "Non-gaussian state-space modeling of nonstationary time series (with discussion)," *Journal of the American Statistical Association*, 82, 1032-1063.

Kulkarni, V. G. (1995), *Modeling and Analysis of Stochastic Systems*, London: Chapman & Hall.

Levinson, S. E. (1986), "Continuously variable duration hidden Markov models for automatic speech recognition," *Computer Speech and Language*, 1, 29-45.

Lukashin, A. V., and Borodovsky, M. (1998), "GeneMark.hmm: new solutions for gene finding," *Nucleic Acids Research*, 26, 1107-1115.

MacDonald, I. L., and Zucchini, W. (1997), *Hidden Markov and Other Models for Discrete-valued Time Series*, London: Chapman and Hall.

McLachlan, G. J., and Krishnan, T. (1997), *The EM Algorithm and Extensions*, New York: Wiley.

Poritz, A. B. (1988), "Hidden Markov models: A guided tour," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, New York, pp. 7-13.

Rabiner, L. R. (1989), "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, 77, 257-286.

Russell, M. J., and Moore, R. K. (1985), "Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Tampa, Florida, pp. 5-8.

Appendix A: derivation of the forward-backward algorithm

The forward recursion is given by,
$t = 0, \ldots, \tau - 2; j = 0, \ldots, J - 1$:

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

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

$$\begin{aligned}
L1_j(t) &= P\left(S_{t+1} \neq j, S_t = j | X_0^{\tau-1} = x_0^{\tau-1}\right) \\
&= \sum_{k \neq j} \left\{ \sum_{u=1}^{\tau-2-t} P\left(S_{t+u+1} \neq k, S_{t+u-v} = k, v = 0, \ldots, u-1, S_t = j | X_0^{\tau-1} = x_0^{\tau-1}\right) \right. \\
&\quad \left. + P\left(S_{\tau-1-v} = k, v = 0, \ldots, \tau-2-t, S_t = j | X_0^{\tau-1} = x_0^{\tau-1}\right) \right\}. \qquad (29)
\end{aligned}$$

For the general term in (29), we have the following decomposition

$$
\begin{aligned}
& P\left(S_{t+u+1}\neq k,S_{t+u-v}=k,v=0,\ldots,u-1,S_t=j|X_0^{\tau-1}=x_0^{\tau-1}\right)\\
= & \frac{P\left(S_{t+u+1}\neq k,S_{t+u-v}=k,v=0,\ldots,u-1,S_t=j,X_0^{\tau-1}=x_0^{\tau-1}\right)}{P\left(S_{t+u+1}\neq k,S_{t+u}=k,X_0^{\tau-1}=x_0^{\tau-1}\right)}\\
& \times P\left(S_{t+u+1}\neq k,S_{t+u}=k|X_0^{\tau-1}=x_0^{\tau-1}\right)\\
= & \frac{P\left(X_{t+u+1}^{\tau-1}=x_{t+u+1}^{\tau-1}|S_{t+u+1}\neq k,S_{t+u}=k\right)P\left(S_{t+u+1}\neq k,S_{t+u}=k|X_0^{\tau-1}=x_0^{\tau-1}\right)}{P\left(X_{t+u+1}^{\tau-1}=x_{t+u+1}^{\tau-1}|S_{t+u+1}\neq k,S_{t+u}=k\right)P\left(S_{t+u+1}\neq k,S_{t+u}=k|X_0^{t+u}=x_0^{t+u}\right)}\\
& \times\frac{P\left(X_{t+1}^{t+u}=x_{t+1}^{t+u}|S_{t+u-v}=k,v=0,\ldots,u-1\right)}{P\left(X_{t+1}^{t+u}=x_{t+1}^{t+u}|X_0^{t}=x_0^{t}\right)}\\
& \times P\left(S_{t+u+1}\neq k,S_{t+u-v}=k,v=0,\ldots,u-2|S_{t+1}=k,S_t\neq k\right)\\
& \times P\left(S_{t+1}=k|S_{t+1}\neq j,S_t=j\right)P\left(S_{t+1}\neq j,S_t=j|X_0^{t}=x_0^{t}\right) \qquad (30)\\
= & \frac{L1_k\left(t+u\right)}{F_k\left(t+u\right)}\left\{\prod_{v=0}^{u-1}\frac{b_k\left(x_{t+u-v}\right)}{N_{t+u-v}}\right\}d_k\left(u\right)p_{jk}F_j\left(t\right).
\end{aligned}
$$

Using a similar argument, we obtain for the second term in (29) which corresponds to the last visited state

$$
\begin{aligned}
& P\left(S_{\tau-1-v}=k,v=0,\ldots,\tau-2-t,S_t=j|X_0^{\tau-1}=x_0^{\tau-1}\right)\\
= & \left\{\prod_{v=0}^{\tau-2-t}\frac{b_k\left(x_{\tau-1-v}\right)}{N_{\tau-1-v}}\right\}D_k\left(\tau-1-t\right)p_{jk}F_j\left(t\right).
\end{aligned}
$$

Finally, we obtain for $L1_j\left(t\right)$

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

Using a similar decomposition as in (30), we obtain for the third term in (13)

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

Appendix B: 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 Observ and $\text{StateIn}_j\,(t+1)$ are introduced for this implementation. The other variables correspond to the quantities already introduced in Section 3.2.

Forward recursion

For $t := 0$ to $\tau - 1$ Do
  $N_t := 0$
  For $j := 0$ to $J - 1$ Do
    $F_j\,(t) := 0$
    Observ $:= b_j\,(x_t)$

    If $(t < \tau - 1)$ Then
      For $u := 1$ to $\min\,(t+1, M_j)$ Do
        If $(u < t + 1)$ Then
          $F_j\,(t) := F_j\,(t) +$ Observ $d_j\,(u)$ $\text{StateIn}_j\,(t - u + 1)$
          $N_t := N_t +$ Observ $D_j\,(u)$ $\text{StateIn}_j\,(t - u + 1)$
          Observ := Observ $b_j\,(x_{t-u})\,/N_{t-u}$
        Else $(u = t + 1)$
          $F_j\,(t) := F_j\,(t) +$ Observ $d_j\,(t+1)$ π_j
          $N_t := N_t +$ Observ $D_j\,(t+1)$ π_j
        EndIf
      EndFor

    Else $(t = \tau - 1)$
      For $u := 1$ to $\min\,(\tau, M_j)$ Do
        If $(u < \tau)$ Then
          $F_j\,(\tau - 1) := F_j\,(\tau - 1) +$ Observ $D_j\,(u)$ $\text{StateIn}_j\,(\tau - u)$
          Observ := Observ $b_j\,(x_{\tau-1-u})\,/N_{\tau-1-u}$
        Else $(u = \tau)$
          $F_j\,(\tau - 1) := F_j\,(\tau - 1) +$ Observ $D_j\,(\tau)$ π_j
        EndIf
      EndFor
      $N_{\tau-1} := N_{\tau-1} + F_j\,(\tau - 1)$
    EndIf
  EndFor

  For $j := 0$ to $J - 1$ Do
    $F_j\,(t) := F_j\,(t)\,/N_t$
  EndFor

  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
        $\text{StateIn}_j\,(t+1) := \text{StateIn}_j\,(t+1) + p_{ij} F_i\,(t)$
      EndFor
    EndFor

EndIf
EndFor

In a first step, the quantities $P\left(S_{t+1} \neq j,\ S_t = j, X_t = x_t |\ X_0^{t-1} = x_0^{t-1}\right)$ and $P\left(S_t = j,\ X_t = x_t |\ X_0^{t-1} = x_0^{t-1}\right)$ are simultaneously computed (the only difference is the replacement of the probability mass function $d_j(u)$ by the survivor function $D_j(u)$; see (10) and (12)). The (partial) products $\prod_{v=1}^{u-1} b_j(x_{t-v}) / N_{t-v}$ are computed recursively during the summation on u using the variable Observ. In a second step, the forward probabilities $F_j(t)$ are extracted as $P\left(S_{t+1} \neq j,\ S_t = j, X_t = x_t |\ X_0^{t-1} = x_0^{t-1}\right) / N_t$. Finally, in a third step, the quantities $P\left(S_{t+1} = j,\ S_t \neq j |\ X_0^t = x_0^t\right) = \sum_{i \neq j} p_{ij} F_i(t)$ are extracted using the variable $\text{StateIn}_j(t+1)$. 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. Hence, the complexity of the forward recursion is $O(J\tau(J+\tau))$-time in the worst case and $O(J\tau)$-space.

Backward recursion

For $j := 0$ to $J-1$ Do
  $L_j(\tau - 1) := F_j(\tau - 1)$
EndFor

For $t := \tau - 2$ to 0 Do
  For $j := 0$ to $J-1$ Do
    $G_j(t+1) := 0$
    Observ $:= 1$
    For $u := 1$ to $\min(\tau - 1 - t, M_j)$ Do
      Observ := Observ $b_j(x_{t+u}) / N_{t+u}$
      If $(u < \tau - 1 - t)$ Then
        $G_j(t+1) := G_j(t+1) + L1_j(t+u)$ Observ $d_j(u) / F_j(t+u)$
      Else $(u = \tau - 1 - t)$
        $G_j(t+1) :=$ $G_j(t+1) +$ Observ $D_j(\tau - 1 - t)$
      EndIf
    EndFor
  EndFor

  For $j := 0$ to $J-1$ Do
    $L1_j(t) := 0$
    For $k := 0$ to $J-1$ Do
      $L1_j(t) := L1_j(t) + G_k(t+1)\, p_{jk}$
    EndFor
    $L1_j(t) := L1_j(t)\, F_j(t)$
    $L_j(t) := L1_j(t) + L_j(t+1) - G_j(t+1)$ $\text{StateIn}_j(t+1)$
  EndFor
EndFor

In a first step, the auxiliary quantities $G_j(t+1)$ are computed. In the same manner as for the forward recursion, the (partial) products $\prod_{v=0}^{u-1} b_k(x_{t+u-v}) / N_{t+u-v}$ are computed recursively during the summation on u using the variable Observ. Then in the second step, the quantities $L1_j(t)$ and $L_j(t)$ are extracted. The quantities $L1_j(t)$ should be stored for each time t and each

state j while the smoothed probabilities $L_j(t)$ and the auxiliary quantities $G_j(t+1)$ need only be stored for each state j. The complexity of the backward recursion is $O(J\tau(J+\tau))$-time in the worst case and $O(J\tau)$-space.

The summation on the preceding times in the forward recursion (respectively the following times in the backward recursion) are performed over a limited range which corresponds to the possible sojourn times in the state of interest. Hence, the worst case complexities of both the forward and the backward recursions are not always reached and, in practice, these complexities are on average much lower.

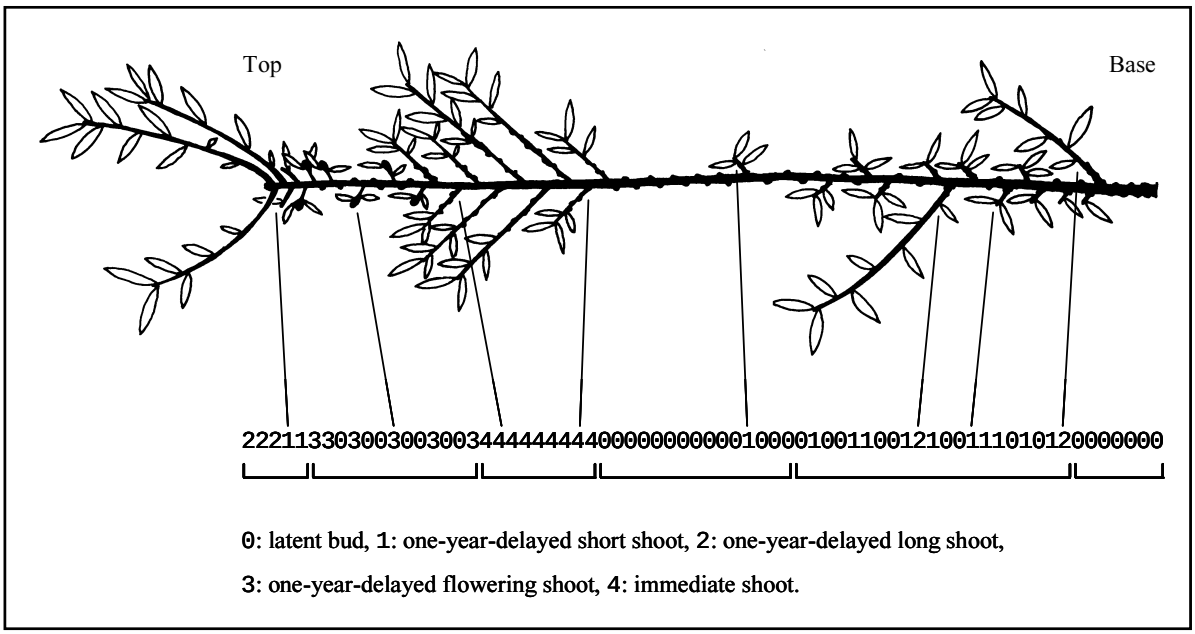

Figure 1. Apple tree (cultivar 'Reinette B.'): First annual shoot of the trunk where the nature of the axillary production was recorded for each successive node (drawing Yves Caraglio).

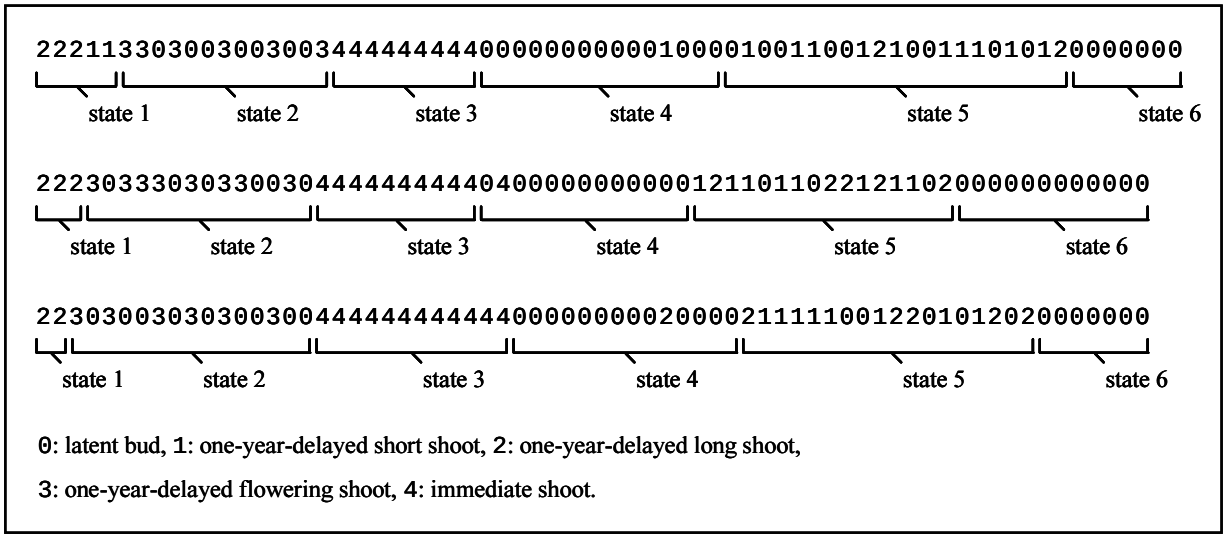

Figure 2. Apple tree (cultivar 'Reinette B.'): optimal segmentation of three observed sequences.

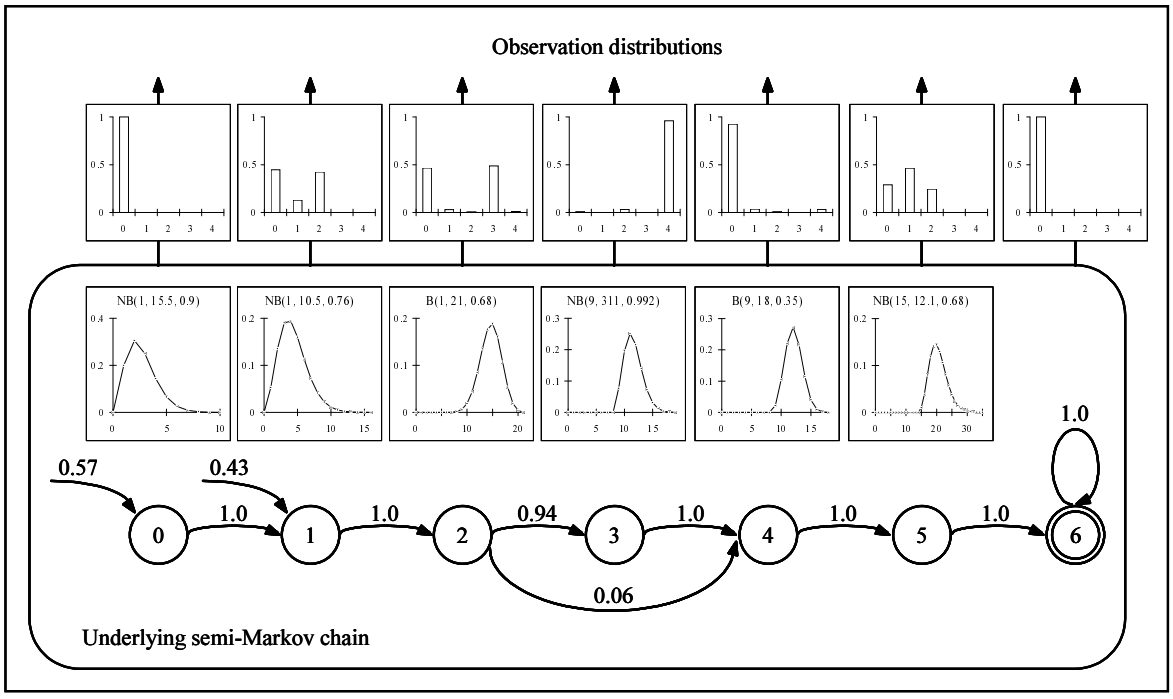

Figure 3. Apple tree (cultivar 'Reinette B.'): estimated hidden semi-Markov chain.

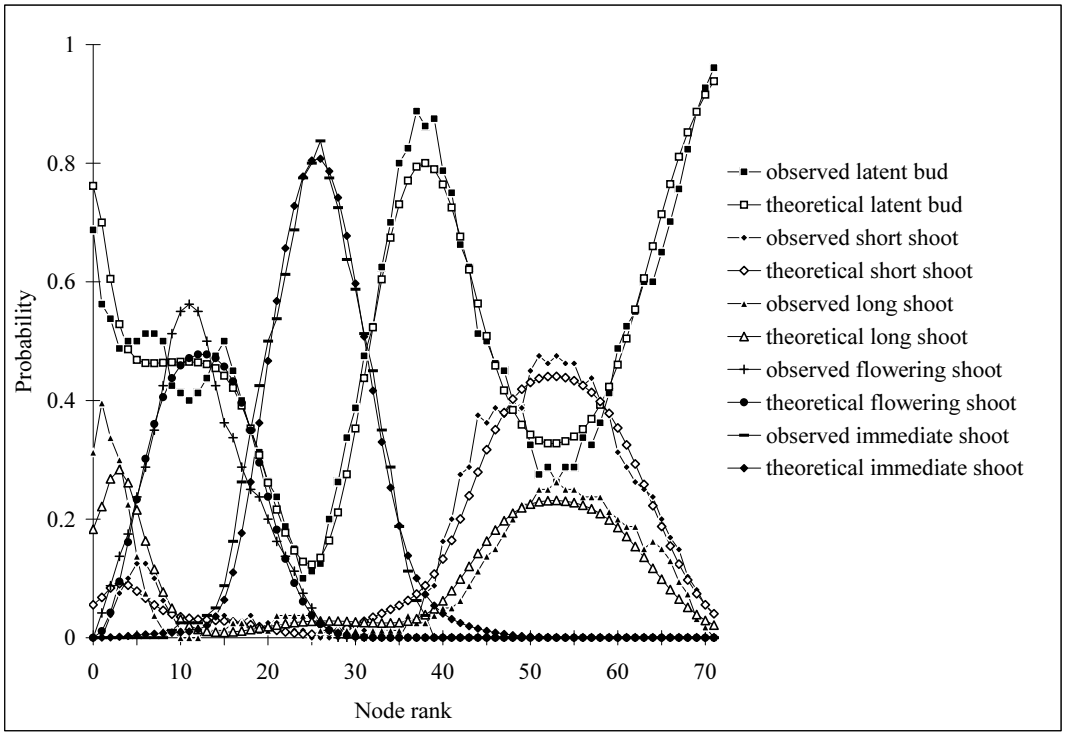

Figure 4. Apple tree (cultivar 'Reinette B.'): intensity point of view.

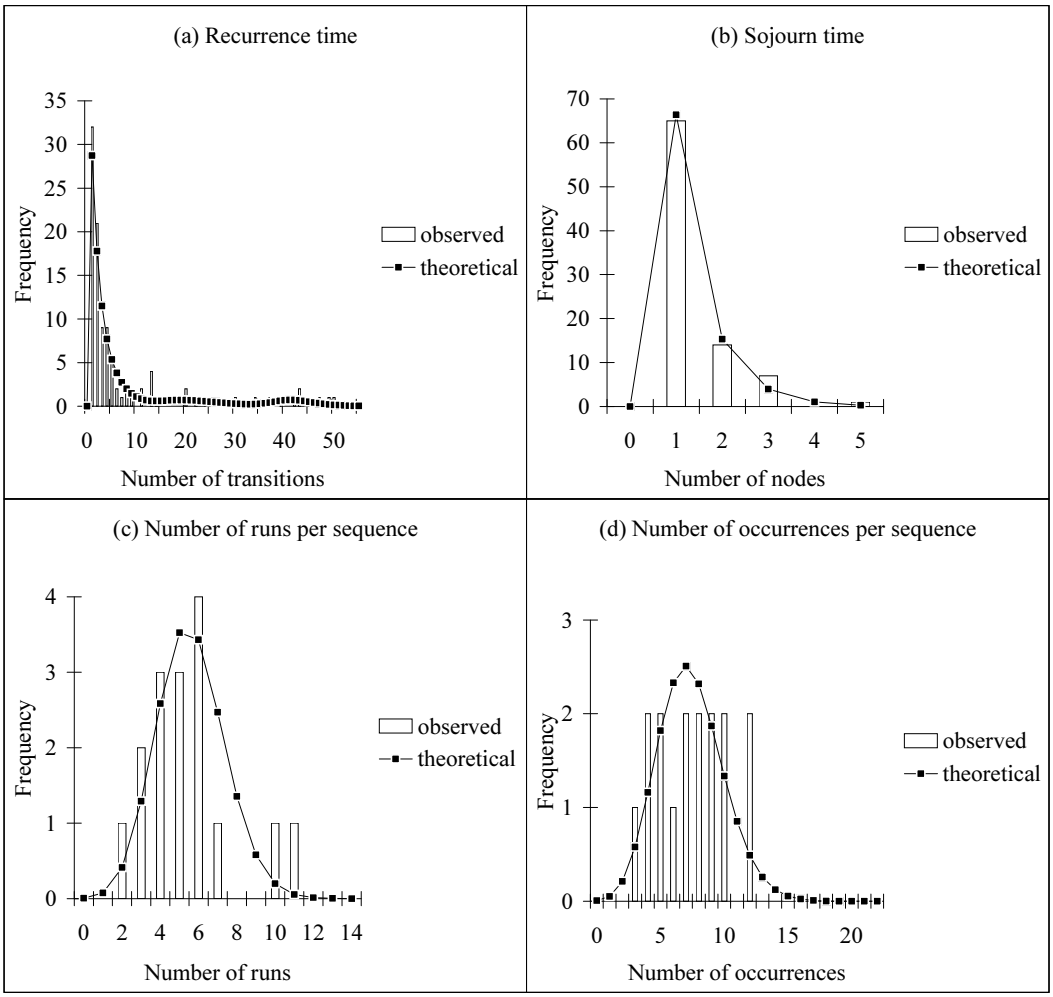

Figure 5. Apple tree (cultivar 'Reinette B.'): interval and counting points of view for long shoots.

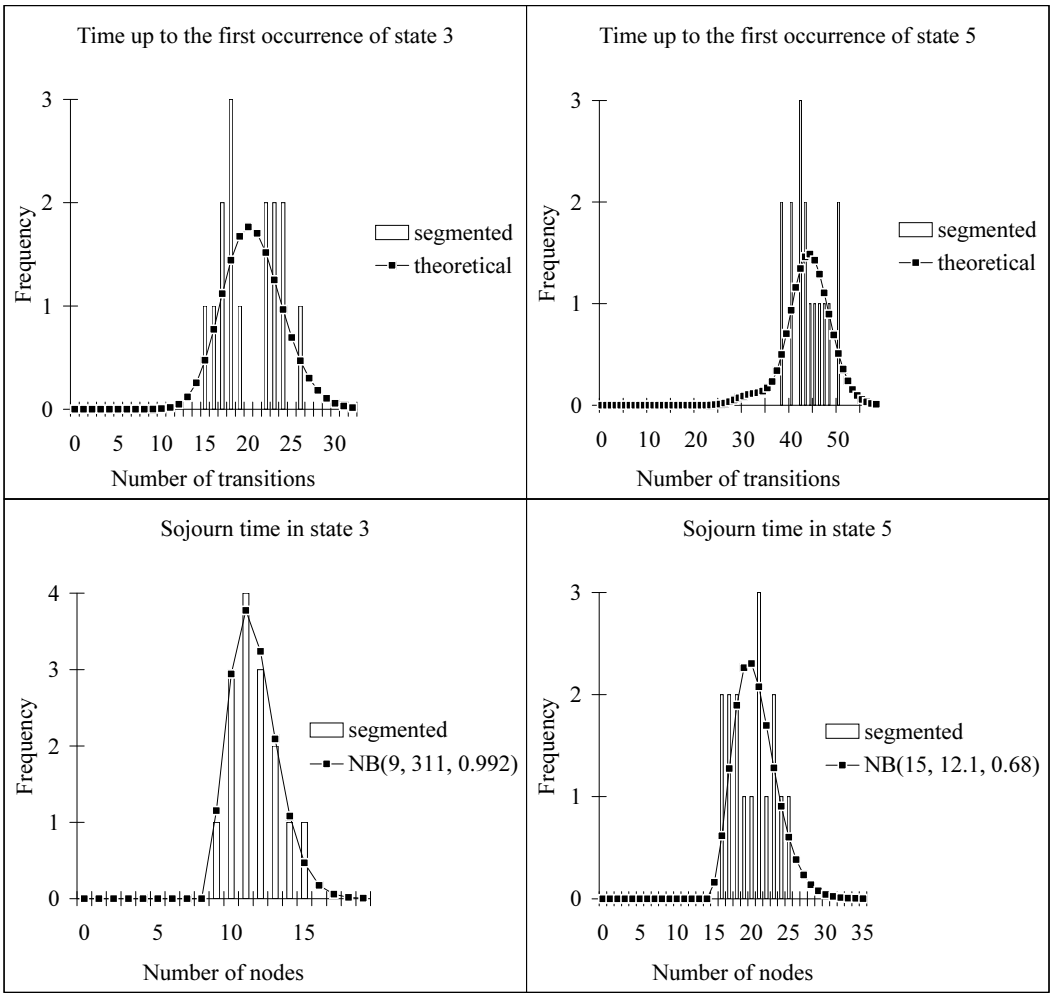

Figure 6. Apple tree (cultivar 'Reinette B.'): interval point of view at the state level.

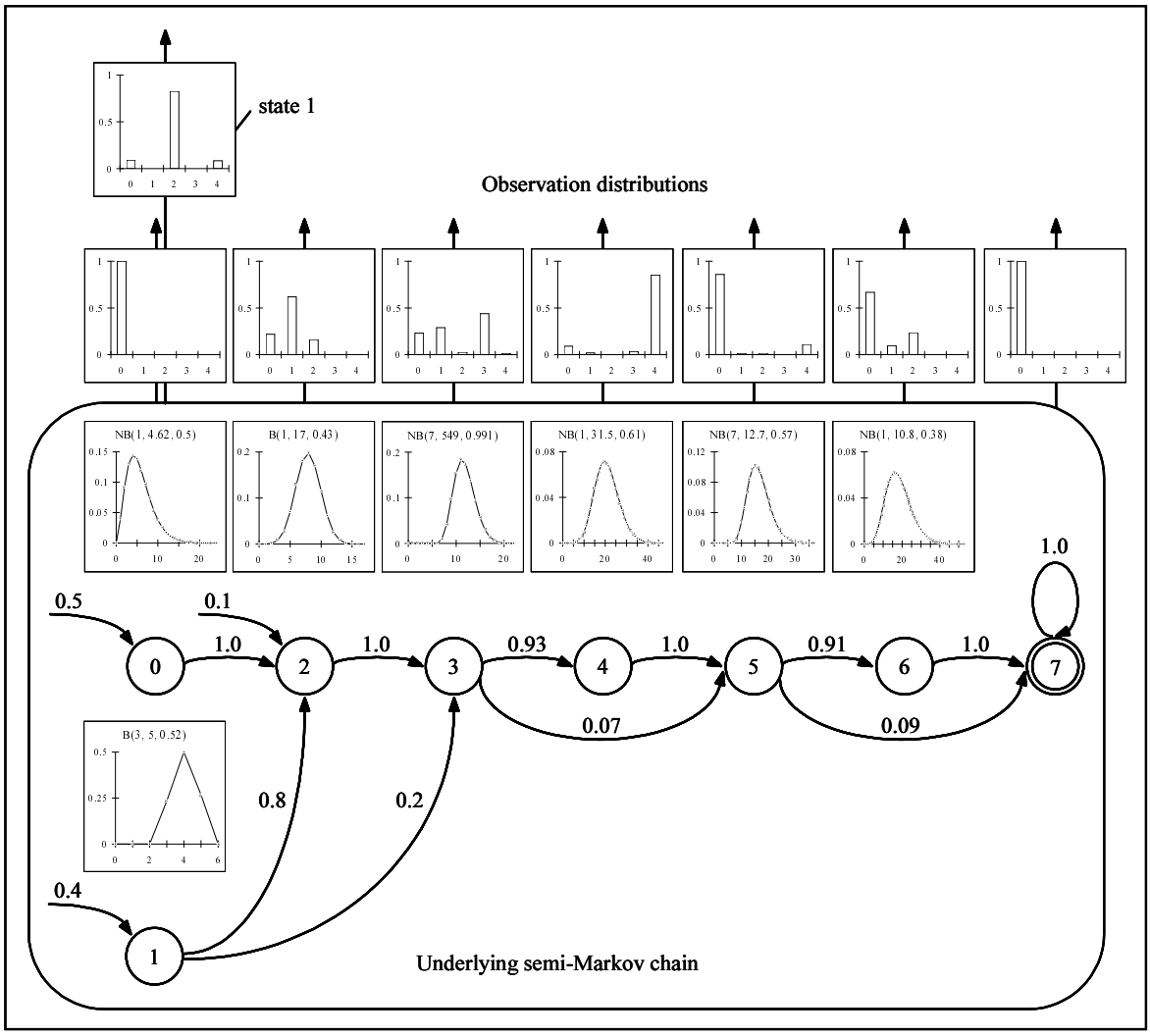

Figure 7. Apple tree (cultivar 'Belrène'): estimated hidden semi-Markov chain.

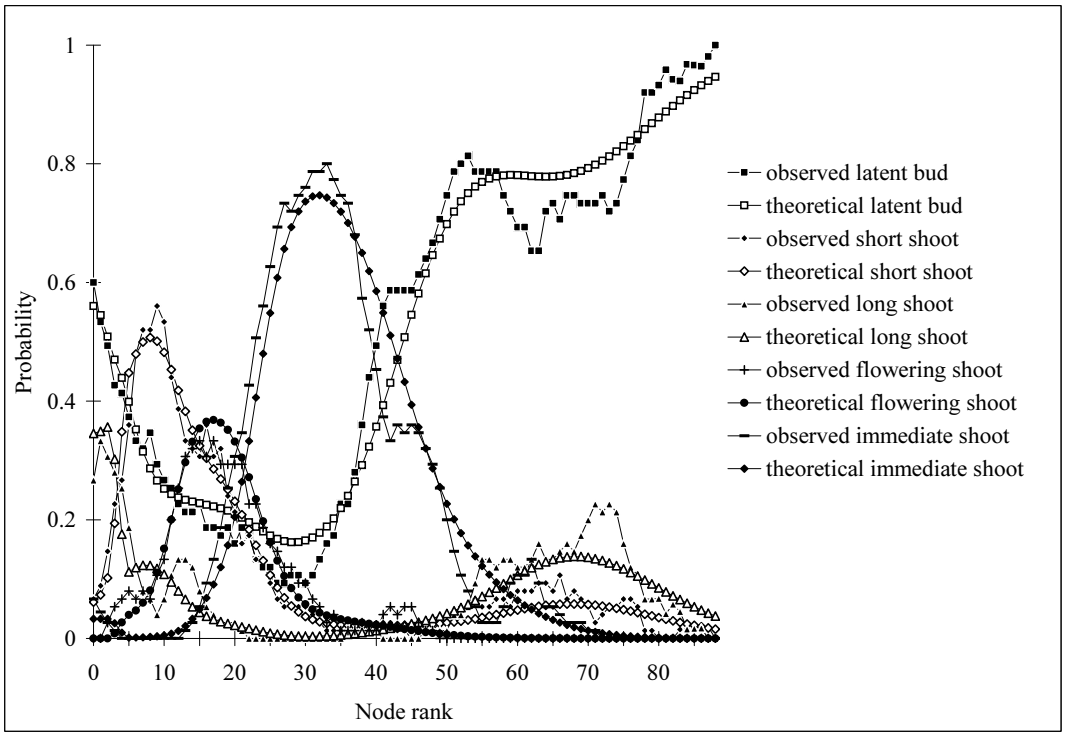

Figure 8. Apple tree (cultivar 'Belrène'): intensity point of view.

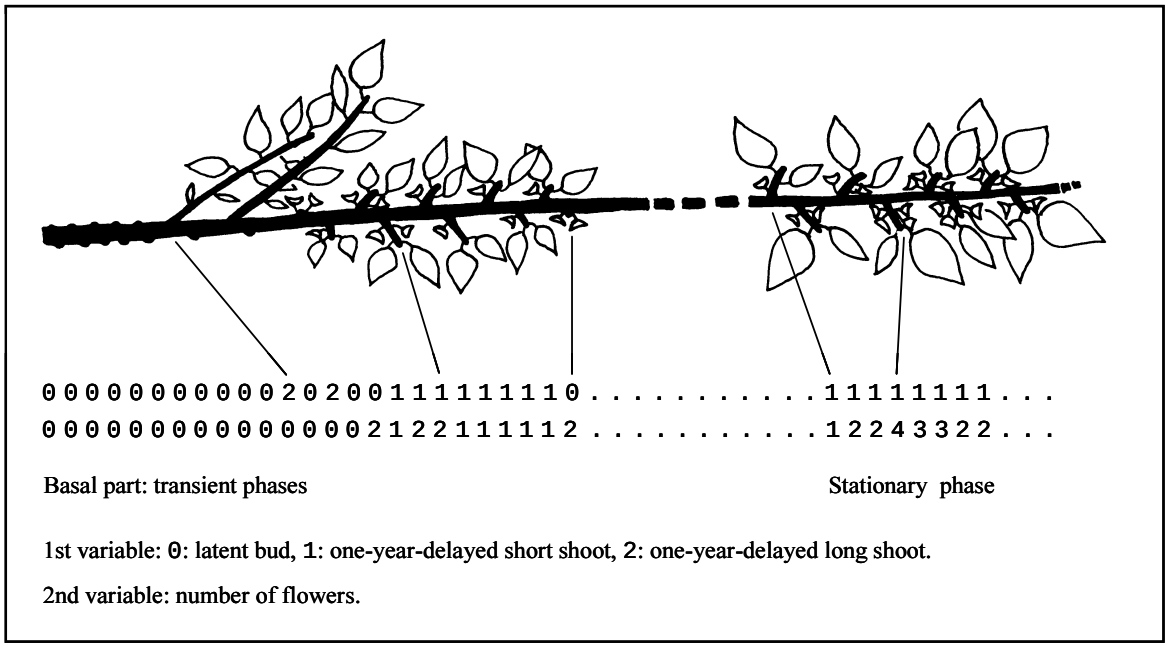

Figure 9. Apricot tree: Growth unit of cultivar 'Lambertin' where the nature of the axillary production and the number of associated flowers were recorded for each successive node (drawing Yves Caraglio).

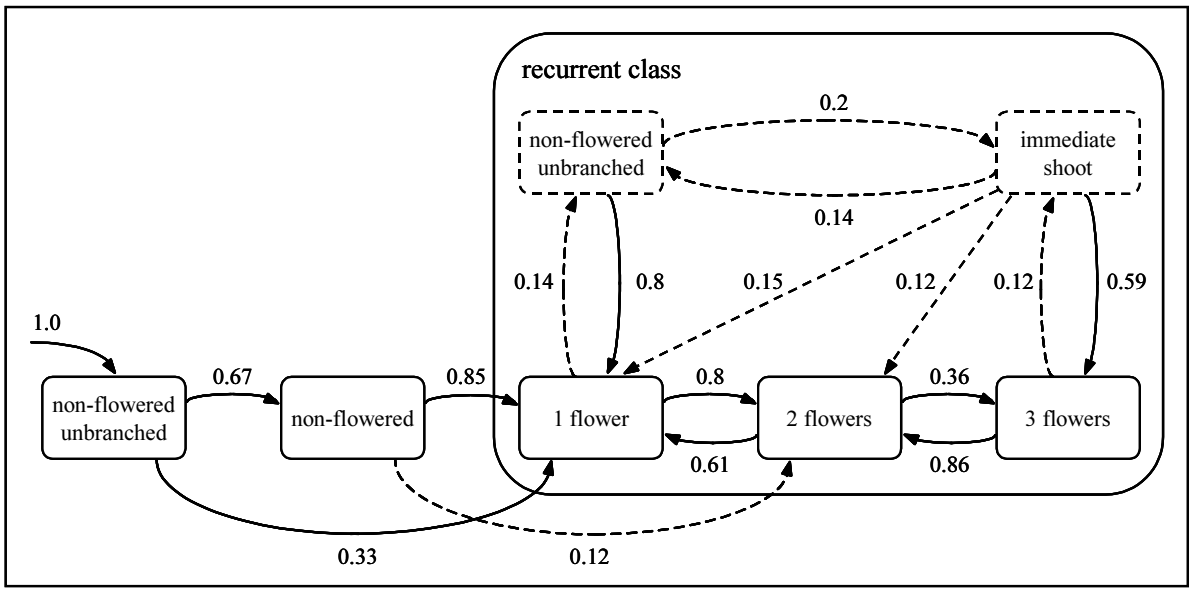

Figure 10. Apricot tree: structure of the estimated hidden semi-Markov chain.

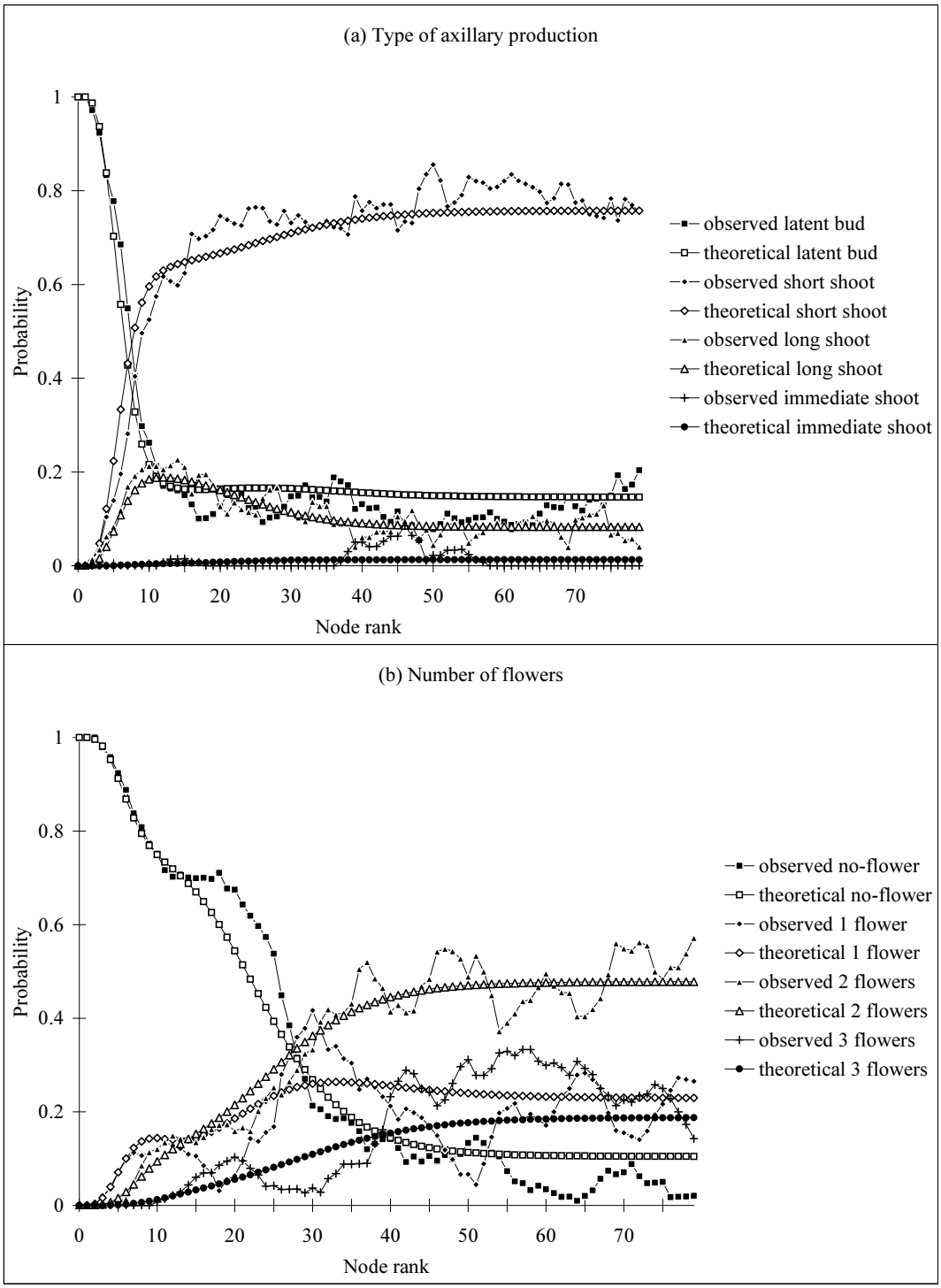

Figure 11. Apricot tree: intensity point of view.

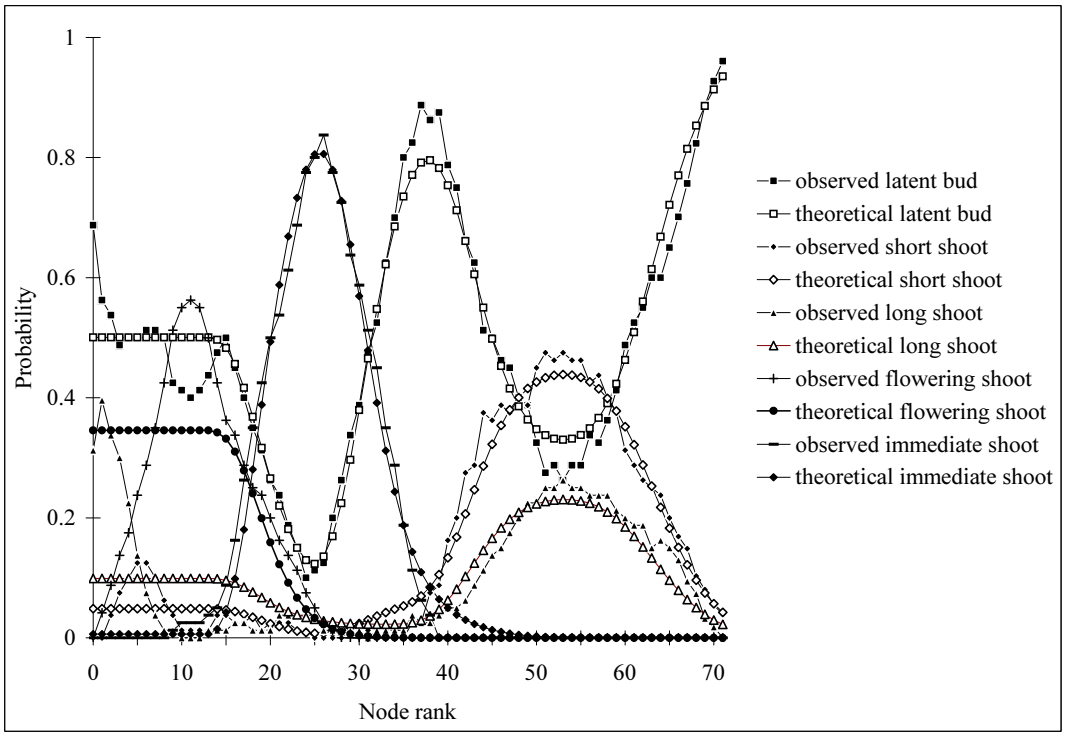

Figure 12. Apple tree (cultivar 'Reinette B.'): intensity point of view for an estimated five-state hidden semi-Markov chain.