# drimmR : R Package for Estimation, Simulation and Reliability of drifting Markov Models

**Vlad Stefan Barbu**
University of Rouen

**Geoffray Brelurut**
University of Rouen

**Annthomy Gilles**
University of Rouen

**Arnaud Lefebvre**
University of Rouen

**Victor Mataigne**
University of Rouen

**Alexandre Seiller**
University of Rouen

**Nicolas Vergne**
University of Rouen

**Abstract**

The aim of this paper is to present the R package drimmR devoted to the estimation, simulation and associated reliability measures of drifting Markov models (DMMs). These are particular non-homogeneous Markov chains introduced in **?** and further developed in **?**, for which the Markov transition matrix is a linear/polynomial function of two/several Markov transition matrices. Several statistical frameworks are taken into account (one or several samples, complete or incomplete samples, models of the same length or not) and two types of estimations are proposed when starting from several samples. We also compute the probabilities of appearance of a word along a sequence under a given model.

*Keywords*: Markov models, drifting Markov models, Non-parametric estimation, Simulation, Reliability, AIC, BIC.

## 1. Introduction

In this work we focus on multi-state systems modeled by means of a particular class of non-homogeneous Markov processes introduced in **?**, called drifting Markov processes. Most of the estimation methods and reliability results are developed in **?**.

Note that in many mathematical models it is assumed the homogeneity with respect to time, which is inappropriate in most of the applications. But, considering general non-homogeneous processes could be unrealistic from a practical point of view. For this reason, the drifting Markov chains introduced in **?** assume that the Markov transition matrix is a linear/polynomial function of two/several Markov transition matrices. Thus we obtain a "smooth" non-homogeneity, with sensibly less parameters than in the general case.

Few R packages have been developed to handle similar types of models of Markov, semi-Markov or hidden semi-Markov type, useful in reliability or DNA analysis. For semi-Markov models we have `semiMarkov` R package **?** that performs maximum likelihood estimation for parametric continuous-time semi-Markov processes, `smm` R package (**?**) which performs parametric and non-parametric estimation and simulation for multi-state discrete-time semi-Markov processes. Two R packages are also dedicated to hidden semi-Markov models, implementing estimation and prediction methods: the `hsmm` R package **?** and the `mhsmm` R package

**?**.

Note that there is no `R` package developed for drifting Markov models (DMMs). Thus the purpose of this paper is to present an `R` package that we have developed, called `drimmR`, which performs estimation and simulation for such models, as well as the estimation of associated reliability measures. The aim of this paper is to describe the different possibilities of this package. To summarize, the package `drimmR` that we present deals with different problems:

- We consider one or several sample paths; for several sample paths, two estimation methods are proposed: one is the usual LSE, the other one is the average of LSEs obtained on each sample;
- The samples paths are complete or incomplete;
- The sample paths come from drifting Markov chains that are of the same length or of different lengths (between the same Markov transition probability matrices);
- We derive exact computations of reliability/survival analysis measures (reliability or survival function, availability, maintainability, failure rates);
- We compute the probabilities of appearance of a word in a given sequence under a given model.

We would like to mention that a web interface called `WebDRIMM` has been also developed (cf. **?**) for simulating and estimating drifting Markov models, as well as associated reliability indicators; it is available at `http://bioinfo.univ-rouen.fr/WebDRIMM`

The paper is organized as follows. Section **??** describes the drifting Markov models used in this package, present associated reliability indicators and corresponding estimation results and techniques. Section **??** illustrates the different functions of the `drimmR` package and we end the paper by presenting some concluding remarks on this `R` package in Section **??**.

# 2. Drifting Markov models, estimation and associated reliability measures

Let us consider a random system with finite state space $E = \{1, \ldots, s\}$, $s < \infty$. We assume that the time evolution of a system is governed by a discrete-time stochastic process with values in $E$. In the following we will introduce a class of processes according to **?** in Section **??**, we will briefly present associated reliability indicators according to **?** in Section **??** and estimation of the parameters in Section **??**, according to **??**.

## 2.1. Drifting Markov models

Let $\Pi_0 = (\Pi_0(u, v))_{u,v \in E}$ and $\Pi_1 = (\Pi_1(u, v))_{u,v \in E}$ be two Markov transition matrices of order 1 over the state space $E$.

**Definition 1 (linear drifting Markov chain of order 1 and of length $n$)** *A sequence* $X_0$, $X_1, \ldots, X_n$ *with state space* $E = \{1, 2, \ldots, s\}$ *is said to be a* linear drifting Markov chain (of order 1) *of length $n$ between the Markov transition matrices* $\Pi_0$ *and* $\Pi_1$ *if the distribution of* $X_t$, $t = 1, \ldots, n$, *is defined by*

$$\mathbb{P}(X_t = v \mid X_{t-1} = u, X_{t-2}, \ldots) = \Pi_{\frac{t}{n}}(u, v),\ u, v \in E, \tag{1}$$

*where*

$$\Pi_{\frac{t}{n}}(u, v) = \left(1 - \frac{t}{n}\right) \Pi_0(u, v) + \frac{t}{n}\Pi_1(u, v),\ u, v \in E. \tag{2}$$

Let us denote by $\alpha = (\alpha(1), \ldots, \alpha(s))$ the *initial distribution of the chain*, that is the distribution of $X_0$, $\alpha(u) = \mathbb{P}(X_0 = u)$ for any state $u \in E$.

The *linear drifting Markov model* of order 1 can be generalized to *polynomial drifting Markov model* of order $k$ and degree $d$. Let $\Pi_{\frac{i}{d}} = (\Pi_{\frac{i}{d}}(u_1, \ldots, u_k, v))_{u_1,\ldots,u_k,v\in E}$ be $d$ Markov transition matrices (of order $k$) over a state space $E$.

**Definition 2 (polynomial drifting Markov chain of order $k$ and of length $n$)** *A sequence* $X_0$, $X_1$, $\ldots$, $X_n$ *with state space* $E = \{1, 2, \ldots, s\}$ *is said to be a* polynomial drifting Markov chain of order k *and of length $n$ if the distribution of* $X_t$, $t = 1, \ldots, n$, *is defined by*

$$\mathbb{P}(X_t = v \mid X_{t-1} = u_k, X_{t-2} = u_{k-1}, \ldots) = \Pi_{\frac{t}{n}}(u_1, \ldots, u_k, v),\ u_1, \ldots, u_k, v \in E, \tag{3}$$

*where*

$$\Pi_{\frac{t}{n}}(u_1, \ldots, u_k, v) = \sum_{i=0}^{d} A_i(t)\Pi_{\frac{i}{d}}(u_1, \ldots, u_k, v),\ u_1, \ldots, u_k, v \in E, \tag{4}$$

*with $A_i$ polynomials of degree $d$ such as, for any* $i, j \in \{0, 1, \ldots, d\}$, $A_i(\frac{nj}{d}) =_{\{i=j\}}$ .

We would like to stress that the coherence between notations implies the choice of the notation $\Pi_{\frac{i}{d}}$ and that, in fact, $A_i$ are Lagrange polynomials; see **?** for more details on these two points.

## 2.2. Reliability of drifting Markov models

In order to undertake a reliability analysis of a system modeled by a DMM, let us assume that the state space of the system is partitioned into working and failure states, $E = U \cup D$, with $U \cap D = \emptyset$, where $U = \{1, \ldots, s_1\}$ represents the working states and $D = \{s_1 + 1, \ldots, s\}$ the failure states of the system. According to this partition of the state space we partition any matrix of vector we are working with and we denote the corresponding partitions accordingly (e.g., $\Pi_0^{UU}$, $\Pi_0^{DU}$, $\alpha^U$ etc.).

For a linear drifting Markov chain of order 1 $(X_t)_{0\leq t\leq n}$, the reliability at time $l$, $l \in \mathbb{N}$, is given by

$$R(l) = \alpha^U \prod_{t=1}^{l} \left( \left(1 - \frac{t}{n}\right) \Pi_0^{UU} + \frac{t}{n}\Pi_1^{UU} \right) 1^U,\ \text{where } 1^U = (\underbrace{1, \cdots, 1}_{s_1})^\top. \tag{5}$$

For a linear drifting Markov chain of order 1 $(X_t)_{0\leq t\leq n}$, the pointwise (or instantaneous) availability at time $l$, $l \in \mathbb{N}$, is given by

$$A(l) = \alpha \prod_{t=1}^{l} \left( \left(1 - \frac{t}{n}\right) \Pi_0 + \frac{t}{n}\Pi_1 \right) 1^{E,U},\ \text{where } 1^{E,U} = (\underbrace{1, \cdots, 1}_{s_1}, \underbrace{0, \cdots, 0}_{s-s_1})^\top. \tag{6}$$

For a linear drifting Markov chain of order 1 $(X_t)_{0\leq t\leq n}$, the maintainability at time $l$, $l \in \mathbb{N}$, is given by

$$M(l) = 1 - \alpha^D \prod_{t=1}^{l} \left( \left(1 - \frac{t}{n}\right) \Pi_0^{DD} + \frac{t}{n} \Pi_1^{DD} \right) 1^D, \text{ where } 1^D = (\underbrace{1, \cdots, 1}_{s-s_1})^\top. \tag{7}$$

For a linear drifting Markov chain of order 1 $(X_t)_{0\leq t\leq n}$, the BMP-failure rate at time $lattime$l,l $\in \mathbb{N}$, is given by

$$\lambda(l) \begin{cases} 1 - \frac{\mu_0^U \ \prod_{t=1}^{l}( \ (1-\frac{t}{n})\pi_0^{UU} + (\frac{t}{n})\pi_1^{UU}) \ \nVdash^U}{\mu_0^U \ \prod_{t=1}^{l-1}( \ (1-\frac{t}{n})\pi_0^{UU} + (\frac{t}{n})\pi_1^{UU}) \ \nVdash^U} & \text{, si R(l-1) != 0} \\ 0 \ , \ otherwise \end{cases} \tag{8}$$

For a linear drifting Markov chain of order 1 $(X_t)_{0\leq t\leq n}$, the RG-failure rate at time $lattime$l,l $\in \mathbb{N}$, is given by

$$r(l) \begin{cases} -\ln \frac{\mu_0^U \ \prod_{t=1}^{l}( \ (1-\frac{t}{n})\pi_0^{UU} + (\frac{t}{n})\pi_1^{UU}) \ \nVdash^U}{\mu_0^U \ \prod_{t=1}^{l-1}( \ (1-\frac{t}{n})\pi_0^{UU} + (\frac{t}{n})\pi_1^{UU}) \ \nVdash^U} & , \ if \ l \geq 1 \ , \\ -\ln R(0) \ , \ if \ \ l = 0 \end{cases} \tag{9}$$

See [**BaVe2018**] for more details and [**Bar2004b**, **Bar2008b**] for similar questions in a discrete-time semi-Markov framework.

## 2.3. Estimation of drifting Markov models

In this section we will consider different types of data for which the estimators of the characteristics of a drifting Markov chain and of the associated reliability indicators will be derived.

One can observe one sample path, that will be denoted by $\mathcal{H}(m, n) := (X_0, X_1, \ldots, X_m)$, where $m$ denotes the length of the sample path and $n$ the length of the drifting Markov chain. Two cases can be considered: (a1) $m = n$ (a complete sample path); (a2) $m < n$ (an incomplete sample path).

One can also observe $H$ i.i.d. sample paths, $\mathcal{H}_i(m_i, n_i), i = 1, \ldots, H$. Four cases can be considered here: (b1) $m_i = n_i = n$ for all $i = 1, \ldots, H$ (complete sample paths of drifting Markov chains of the same length) ; (b2) $n_i = n$ for all $i = 1, \ldots, H$ (incomplete sample paths of drifting Markov chains of the same length); (b3) $m_i = n_i$ for all $i = 1, \ldots, H$ (complete sample paths of drifting Markov chains of different lengths) ; (b4) $m_i \leq n_i$ for all $i = 1, \ldots, H$ (incomplete sample paths of drifting Markov chains of different lengths).

We have developed (cf. **??**) mean square estimators starting from data under these frameworks; we present here only the case of a linear drifting Markov chain of order 1 under the sample framework (b1) (complete sample paths of drifting Markov chains of the same length).

Under the setting (b1), starting with $H$ complete sample paths of drifting Markov chains of the same length of a linear drifting Markov chain between two Markov transition matrices (of order 1) $\Pi_0$ and $\Pi_1$, for any states $u, v \in E$, the estimators of $\Pi_0(u, v)$ and $\Pi_1(u, v)$ are given by:

$$\widehat{\Pi}_{0;(n,H)}(u,v) = \frac{P_1(H,m,n)P_2(H,m,n) - P_3(H,m,n)P_4(H,m,n)}{P_5(H,m,n)P_1(H,m,n) - P_3(H,m,n)^2}$$
$$\widehat{\Pi}_{1;(n,H)}(u,v) = \frac{P_5(H,m,n)P_4(H,m,n) - P_3(H,m,n)P_2(H,m,n)}{P_5(H,m,n)P_1(H,m,n) - P_3(H,m,n)^2},$$

where we have introduced the following notation:
$P_1(H,m,n) = \sum_{t=1}^{m}\sum_{h=1}^{H} {}_{\{X_{t-1}^h=u\}}(\frac{t}{n})^2$, $P_2(H,m,n) = \sum_{t=1}^{m}\sum_{h=1}^{H} {}_{\{X_{t-1}^h=u, X_t^h=v\}}(1-\frac{t}{n})$,
$P_3(H,m,n) = \sum_{t=1}^{m}\sum_{h=1}^{H} {}_{\{X_{t-1}^h=u\}}(1-\frac{t}{n})(\frac{t}{n})$, $P_4(H,m,n) = \sum_{t=1}^{m}\sum_{h=1}^{H} {}_{\{X_{t-1}^h=u, X_t^h=v\}}(\frac{t}{n})$,
$P_5(H,m,n) = \sum_{t=1}^{m}\sum_{h=1}^{H} {}_{\{X_{t-1}^h=u\}}(1-\frac{t}{n})^2$, with $m = (m_1,\ldots,m_H)$ and $n = (n_1,\ldots,n_H)$.
Note we can adapt the estimation procedures that we have previously obtained in order to get estimators of the drifting Markov models in the other cases; one can see **?** for more details. Using the expression of the reliability indicators of a drifting Markov chain previously obtained and the estimators of the characteristics of a drifting Markov chain, one immediately obtains the associated plug-in estimators of the reliability metrics.

# 3. The drimmR package

The drimmR package is principally devoted to the simulation and estimation of drifting Markov models, as well as to the estimation of associated reliability measures and to the computation of the probabilities of a word occurrence under a given model. All the different possibilities of the package are illustrated in Figure **??**.

## 3.1. Estimation of drifting Markov models

The estimation of DMMs is carried out by the functions `dmmsum`, when starting from one (several) sample path (and obtain LSE). We will describe this function in the sequel.

1. The function `dmmsum`

The different **arguments** of this function are:

- `sequences`: A list of character vector(s) representing one (several) sequence(s) from which the estimation is carried out
- `order`: Order of the Markov chain
- `degree`: Degree of the polynomials (e.g., linear drifting if degree=1, etc.)
- `states`: Vector of states space of length s > 1
- `init.estim`: Method used to estimate the initial law.

Here we have an example of an estimation of a drifting Markov model of order 1 and degree 1 using the function `dmmsum`, starting from a DNA sequence called `lambda`.

```
data(lambda, package = "drimmR")
states <- c("a","c","g","t")
dmm <- dmmsum(lambda, 1, 1, states, init.estim="freq")
```

```
$states
[1] "a" "c" "g" "t"

$order
[1] 1

$degree
[1] 1

$Polynomials
    t^0 t^1
A_0   1  -1
A_1   0   1

$length
[1] 48502

$matrices
$matrices$Pi0
          a         c         g         t
a 0.2548330 0.2495270 0.2593907 0.2362493
c 0.2353305 0.2465529 0.3360221 0.1820945
g 0.2313610 0.3008693 0.2879023 0.1798674
t 0.1356776 0.2174431 0.4129533 0.2339260

$matrices$Pi1
          a         c         g         t
a 0.3398025 0.1715507 0.1870320 0.3016148
c 0.3339305 0.1911400 0.2075919 0.2673376
g 0.2801379 0.2600478 0.2021377 0.2576765
t 0.2218149 0.2283648 0.2305139 0.3193065


$init.estim

        a         c         g         t
0.2543400 0.2342171 0.2642778 0.2471651

attr(,"class")
[1] "dmm"    "dmmsum"
```

*Estimation of corresponding model characteristics*

Once a Markov drifting model has been estimated/constructed as described in the previous subsection, various characteristics of the model can be computed. These are: the log-likelihood of one or several sequences, the AIC and BIC information criteria, the stationary distribution on an entire sequence or only on a part of it, the distribution of the chain on the entire sequence or only on a part of it.

The functions `loglik`, `aic` and `bic` have the following **arguments**:

- `x`: Object of class `dmm`
- `sequences`: A list of character vector(s) representing one (several) sequence(s)

They return the numerical values of the log-likelihood, the AIC, the BIC of the sequence, respectively.

```
data(lambda, package = "drimmR")
sequence <- c("a","g","g","t","c","g","a","t","a","a","a")
dmm <-dmmsum(lambda, 1, 1, c(’a’,’c’,’g’,’t’), init.estim = "freq")

loglik(dmm, sequence)
[[1]]
[1] -16.08783

aic(dmm, sequence)
[[1]]
[1] 56.17567

bic(dmm, sequence)
[[1]]
[1] 60.95041
```

The function `getTransitionMatrix` evaluates the transition matrix at a given position. The function has the following arguments :

- `x`: Object of class `dmm`
- `pos`: position along the sequence (integer)

```
data(lambda)
dmm <- dmmsum(lambda, 1, 1, c(’a’,’c’,’g’,’t’),init.estim = "freq")
t <- 10
getTransitionMatrix(dmm,pos=t)
```

```
          a         c         g         t
a 0.2548505 0.2495109 0.2593757 0.2362628
c 0.2353509 0.2465415 0.3359956 0.1821121
g 0.2313710 0.3008609 0.2878846 0.1798834
t 0.1356954 0.2174453 0.4129157 0.2339436
```

The function `getStationaryLaw` evaluates the stationary law at a given position or for all positions along the list of sequence(s).

- `x`: Object of class `dmm`
- `pos`: position along the sequence (integer)
- `all.pos`: FALSE (default, evaluation at pos index) ; TRUE (evaluation for all pos index)
- `internal`: FALSE (default) ; TRUE (for internal use of dmmsum initial law)

```
data(lambda, package = "drimmR")
sequence <- sample(lambda, 30, replace=TRUE )
dmm <- dmmsum(sequence, 1, 1, c(’a’,’c’,’g’,’t’), init.estim = "freq")
t <- 10
getStationaryLaw(dmm,pos=t)
```

```
        a         c         g         t
0.2641443 0.2794671 0.2091486 0.2472401
```

```
getStationaryLaw(dmm,all.pos=TRUE)
```

```
                a         c         g         t
pos 1  0.2529924 0.2551165 0.3350840 0.1568071
pos 2  0.2497715 0.2615672 0.3302433 0.1584180
pos 3  0.2462983 0.2677077 0.3259821 0.1600119
pos 4  0.2426070 0.2735575 0.3222392 0.1615964
pos 5  0.2387271 0.2791341 0.3189611 0.1631778
pos 6  0.2346847 0.2844522 0.3161009 0.1647623
pos 7  0.2305025 0.2895249 0.3136173 0.1663552
pos 8  0.2262009 0.2943637 0.3114738 0.1679617
pos 9  0.2217974 0.2989783 0.3096378 0.1695864
pos 10 0.2173081 0.3033775 0.3080805 0.1712339
pos 11 0.2127468 0.3075688 0.3067761 0.1729082
pos 12 0.2081263 0.3115587 0.3057015 0.1746135
pos 13 0.2034578 0.3153529 0.3048358 0.1763536
pos 14 0.1987514 0.3189560 0.3041603 0.1781322
pos 15 0.1940163 0.3223724 0.3036581 0.1799531
pos 16 0.1892609 0.3256053 0.3033139 0.1818199
pos 17 0.1844925 0.3286576 0.3031137 0.1837362
pos 18 0.1797180 0.3315315 0.3030449 0.1857055
pos 19 0.1749438 0.3342288 0.3030958 0.1877316
pos 20 0.1701755 0.3367505 0.3032560 0.1898180
pos 21 0.1654185 0.3390975 0.3035156 0.1919684
pos 22 0.1606776 0.3412698 0.3038658 0.1941868
pos 23 0.1559573 0.3432675 0.3042984 0.1964769
pos 24 0.1512618 0.3450896 0.3048058 0.1988428
pos 25 0.1465952 0.3467353 0.3053808 0.2012887
pos 26 0.1419612 0.3482029 0.3060171 0.2038188
pos 27 0.1373632 0.3494905 0.3067086 0.2064377
pos 28 0.1328047 0.3505956 0.3074495 0.2091502
pos 29 0.1282888 0.3515155 0.3082347 0.2119610
pos 30 0.1238188 0.3522467 0.3090591 0.2148755
```

*Analysis of a model*

Once a Markov drifting model has been estimated/constructed as described at the beginning of this section, the obtained model can be analyzed by means of several functions.

1. The computation of word probabilities according to an estimated/constructed DMM is carried out by means of the functions `word_proba`, `word_probas` and `words_probas`.

The function `word_proba` computes the probability of a word at a given position; it has the following **arguments**:

- `word`: A word, i.e., a subsequence string of characters
- `pos`: A position of the word along the sequence (numeric)
- `x`: An object of class `dmm`
- `output_file`: A file containing the probability
- `internal`: FALSE (default) ; TRUE (for internal use of word applications)

It returns the numerical value of the probability.

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda, 1, 1, c(’a’,’c’,’g’,’t’), init.estim = "freq")
PROB.out <- "C:\\...\\file.txt"
word_proba("cgt",10,dmm,output_file=PROB.out)
```

```
       cgt
0.01563115
```

The function `word_probas` computes the probabilities of a word at given positions. It has the following **arguments**:

- `word`: A word, i.e., a subsequence string of characters
- `pos`: Vector of integer positions of the word along the sequence; it is given by a `start` and `end` point
- `x`: An object of class `dmm`
- `output_file`: A file containing the probabilities
- `internal`: FALSE (default) ; TRUE (for internal use of word applications)
- `plot`: display a figure plot of word probabilities along the positions if TRUE

It returns a numerical vector of probabilities computed at the positions given in `pos`.

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda, 1, 1, c(’a’,’c’,’g’,’t’), init.estim = "freq")
PROB.out <- "C:\\...\\file.txt"
res <- word_probas("cgt",c(100,length(lambda)-2),mod,
output_file=PROB.out, plot=TRUE)
head(res[[1]], n=10)
```

```
   position probability
1       100  0.01562696
2       101  0.01562692
3       102  0.01562687
4       103  0.01562682
5       104  0.01562678
6       105  0.01562673
7       106  0.01562668
8       107  0.01562664
9       108  0.01562659
10      109  0.01562654
```

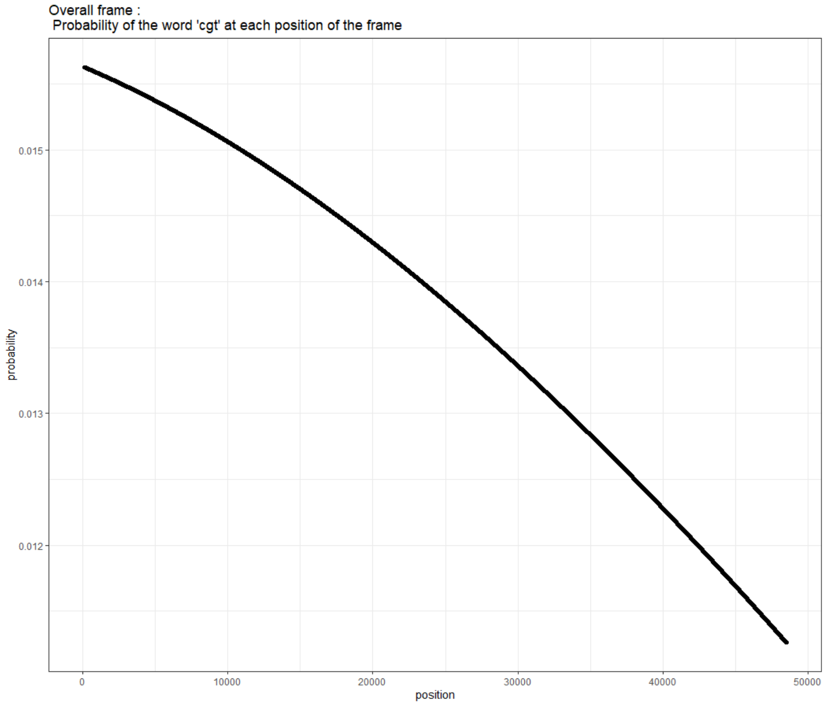

Figure 1: Figure plot of word probabilities along the sequence between <start> and <end> of `pos` argument

The function `words_probas` computes the probabilities of a word at given positions; it has the following **arguments**:

- `words`: A vector of characters containing words
- `pos`: Vector of integer positions of the word along the sequence; it is given by a `start` and `end` point
- `x`: An object of class `dmm`
- `output_file`: A file containing the matrix of probabilities
- `plot`: display a figure plot of word probabilities along the positions if TRUE

It returns a data frame of probabilities computed for each element of `words` at the positions given in `pos`.

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda, 1, 1, c(’a’,’c’,’g’,’t’), init.estim = "freq")
PROB.out <- "C:\\...\\file.txt"
res <- words_probas(c("atcgattc", "taggct", "ggatcgg"),c(100,length(lambda)-10),
mod, output_file=PROB.out, plot=TRUE)
head(res[[1]], n=10)
```

```
    position probability word ’atcgattc ’ probability word ’taggct ’
1        100                 1.048882e-05                0.0001132973
2        101                 1.048907e-05                0.0001132992
3        102                 1.048932e-05                0.0001133011
4        103                 1.048957e-05                0.0001133030
5        104                 1.048983e-05                0.0001133049
6        105                 1.049008e-05                0.0001133068
7        106                 1.049033e-05                0.0001133088
8        107                 1.049059e-05                0.0001133107
9        108                 1.049084e-05                0.0001133126
10       109                 1.049109e-05                0.0001133145


probability word ’ggatcgg ’
0.0001056253
0.0001056236
0.0001056220
0.0001056203
0.0001056186
0.0001056169
0.0001056152
0.0001056135
0.0001056118
0.0001056101
```

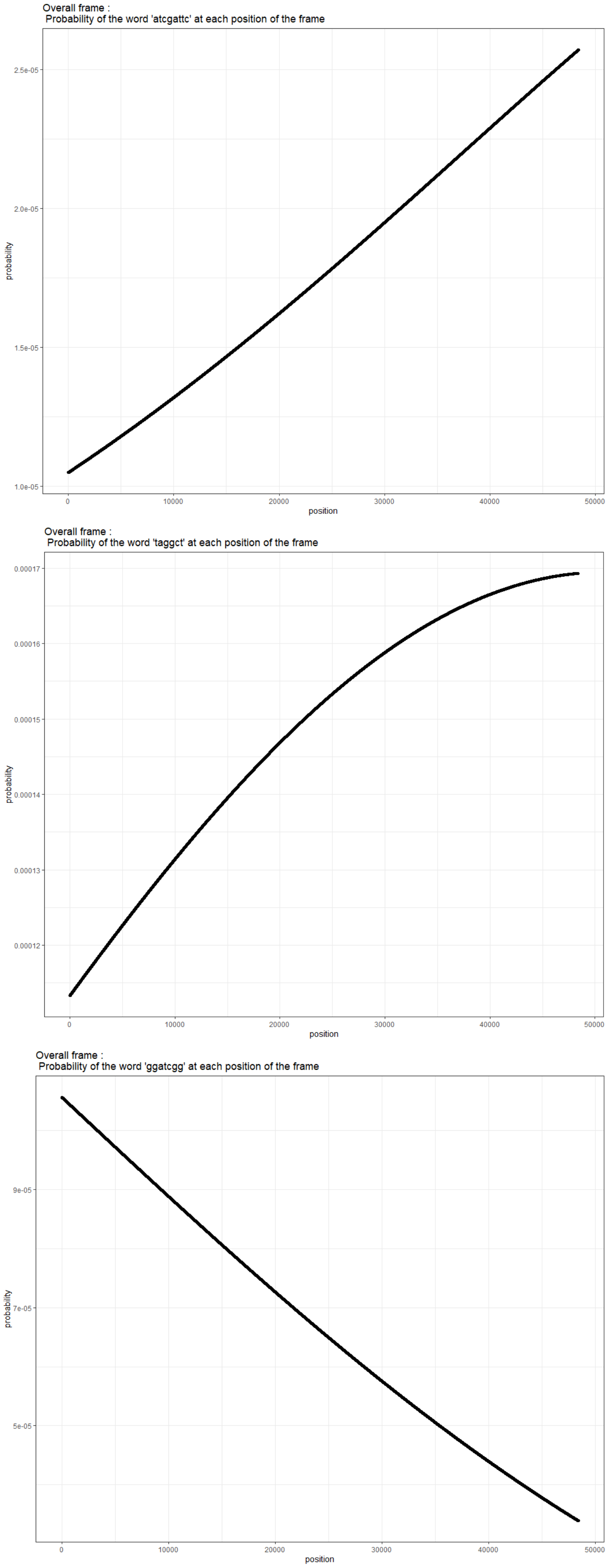

Figure 2: Figure plot of probabilities for each word along the sequence between <start> and <end> of `pos` argument

2. The function `length_probas` computes the probabilities of occurrence of the observed word of given size in a sequence at several positions. This function has the following **arguments**:

- `n`: Integer, the given length of the word
- `sequence`: Vector of characters representing the sequence
- `pos`: Vector of integer positions of the word along the sequence. It it is given by a `start` and `end` point
- `x`: An object of class `dmm`
- `output_file` A file containing the vector of probabilities
- `plot` display several figure plots of probabilities of appearance of words along the positions if TRUE

It returns a data frame of probability by position

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda, 1, 1, c(’a’,’c’,’g’,’t’), init.estim = "freq")
PROB.out <- "C:\\...\\file.txt"
res <- length_probas(n=2, lambda, c(100,length(lambda)-1), dmm,
output_file=PROB.out, plot=TRUE)
```

```
   position word        probability
1          1   gg 0.0920149143338398
2          2   gg 0.0920137134362522
3          3   gc 0.0961580205303848
4          4   cg 0.0869006631431165
5          5   gg 0.092010107824605
6          6   gc 0.0961552204073731
7          7   cg 0.0868975633651543
8          8   ga 0.0739442789031447
9          9   ac 0.0543989570401243
10        10   cc 0.0637602384067068
```

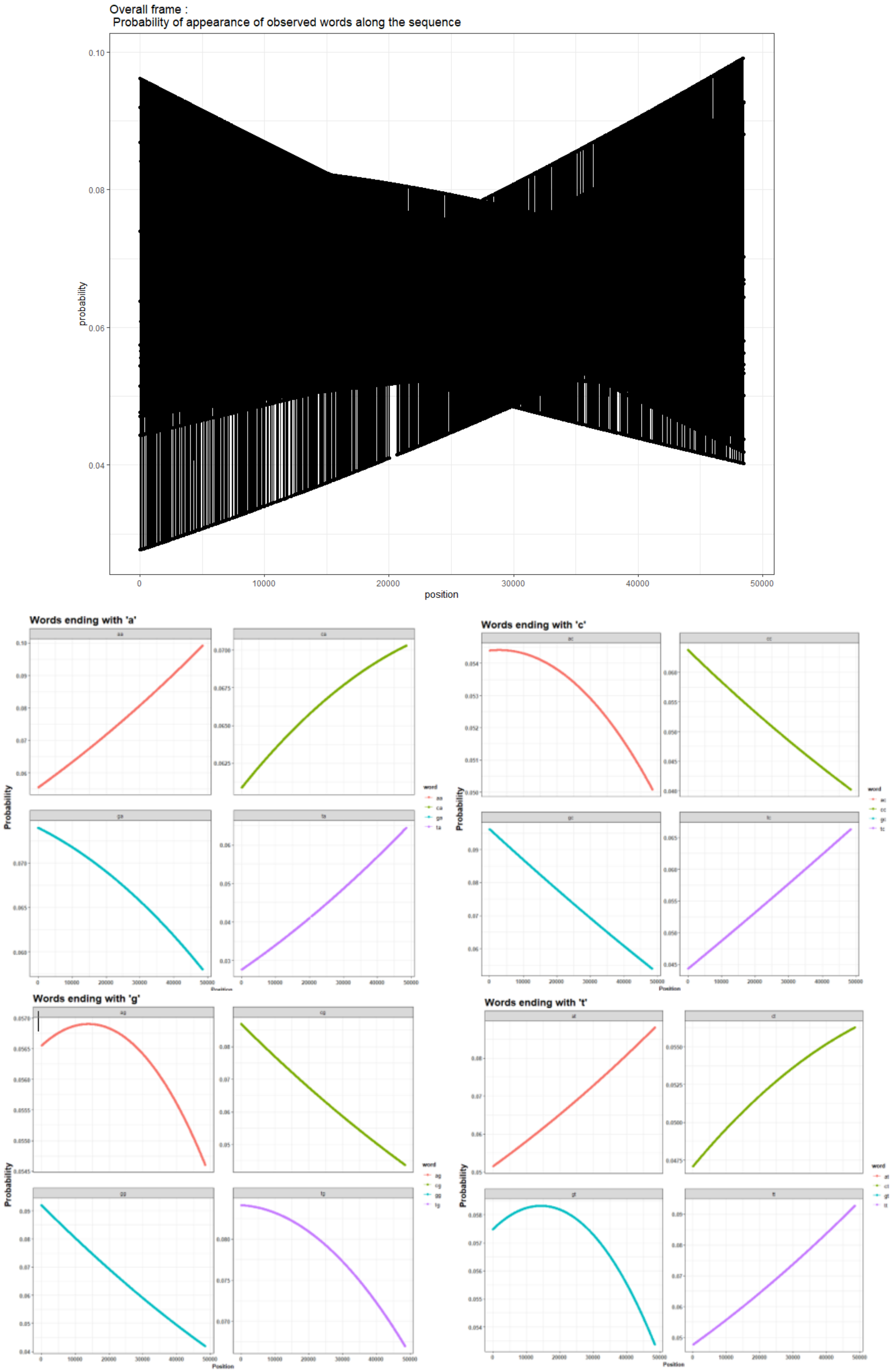

Figure 3: Figure plot of probabilities of appearance of words of size `n` along the sequence between <start> and <end> of `pos` argument

### 3.2. Simulation of drifting Markov models

The simulation of drifting Markov models is carried out by the function `simulation` that has the following **arguments**:

- `DRIMM`: An object of class `dmm`
- `output_file`: File containing the simulated sequence
- `model_size`: Integer, the size of the model

```
data(lambda, package = "drimmR")
SIM.out <- "C:\\...\\file.txt"
dmm <- dmmsum(lambda, 1, 1, c('a','c','g','t'), init.estim = "freq")
# simulate a sequence of length 20 000 from dmm
simulate(dmm,SIM.out,20000)
```

```
 "Write a simulated file from the model"
   [1] "t" "a" "g" "g" "g" "t" "c" "a" "t" "c" "g" "c" "c" "g" "g" "c" "g" "c"
   "c" "g" "c" "c" "a" "a" "a" "g" "g" "a" "t" "t" "c" "g" "t" "t" "t" "g" "a"
  [38] "g" "c" "c" "g" "t" "c" "t" "c" "g" "g" "g" "a" "t" "c" "a" "t" "c" "g"
  "g" "g" "g" "c" "t" "t" "c" "t" "g" "c" "t" "g" "c" "g" "t" "c" "t" "c" "c"
  [75] "g" "g" "a" "c" "g" "g" "t" "c" "t" "g" "a" "t" "t" "t" "g" "g" "c" "g"
  "g" "g" "g" "g" "a" "g" "c" "t" "g" "g" "a" "g" "g" "g" "c" "g" "g" "g" "t"
  ...
```

## 3.3. Reliability of drifting Markov models

The reliability measures are computed by means of the following functions: `A` (availability), `R` (reliability,survival function), `M` (maintainability), `errorRate` (the classical failure rate, called BMP-failure rate and a more recent failure rate adapted to discrete data, called RG-failure rate). For more details on these reliability measures one can see [**Bar2004b**, **Bar2008b**]. All these functions have the following **arguments**:

- `x`: An object of class `dmm`
- `k1`: An integer, start position for the computation of the corresponding reliability measure
- `k2`: An integer, end position for the computation of the corresponding reliability measure
- `s1`: Character vector of the subspace working states (up-states) among the state space vector s.t. $s1 < s$
- `output_file`: File containing the estimated/computed values of the corresponding reliability measure at each position of the selected frame
- `plot`: display figure plot of the corresponding reliability measure along the positions if TRUE

The function `errorRate` enables to select the type of evaluated failure rate with an additional argument :

- `error.rate`: Default="BMP", then BMP-failure-rate is the method used to estimate the error rate. If error.rate= "RG", then RG-failure rate is the method used to estimate the error rate.

These functions return vectors with the values of the corresponding reliability measure.

*Estimation of reliability*

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda,1,1,c("a","c","g","t"))
REL.out <- "C:\\...\\file.txt"
R(dmm,k1=1, k2=10,s1=c("a","c"),output_file = REL.out, plot=TRUE)
```

```
     positions  reliability
 [1,]         0 1.0000000000
 [2,]         1 0.4885571729
 [3,]         2 0.2411445408
 [4,]         3 0.1188995038
 [5,]         4 0.0586239943
 [6,]         5 0.0289048794
 [7,]         6 0.0142517252
 [8,]         7 0.0070269076
 [9,]         8 0.0034646676
[10,]         9 0.0017082814
[11,]        10 0.0008422825
```

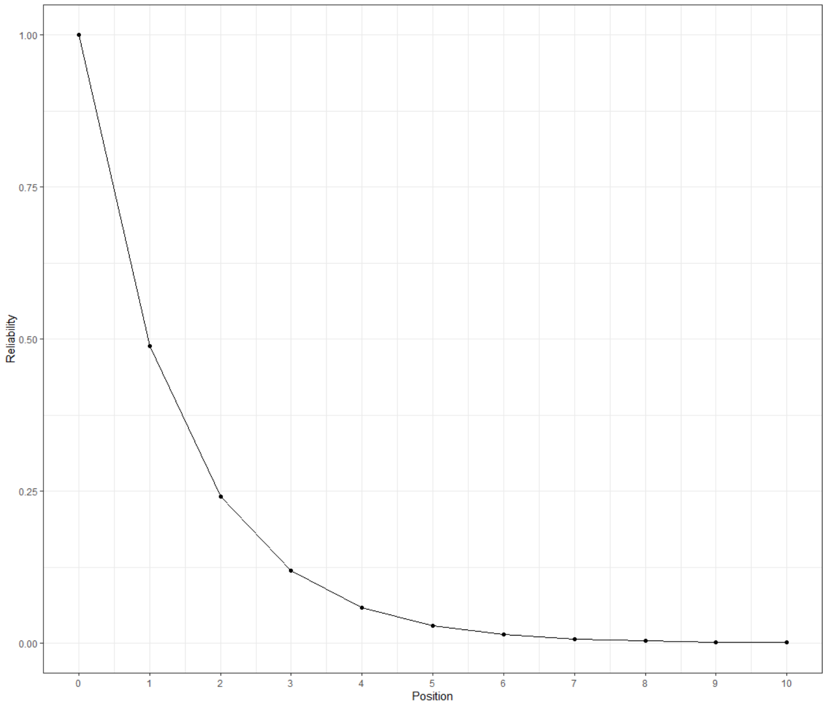

Figure 4: Figure plot of reliability probabilities along the sequence between `k1` and `k2` position arguments for the working states subspace `s1`

*Estimation of availability*

```
 data(lambda, package = "drimmR")
dmm <- dmmsum(lambda,1,1,c("a","c","g","t"))
AVA.out <- "C:\\...\\file.txt"
A(dmm,k1=1, k2=10,s1=c("a","c"),output_file = AVA.out, plot=TRUE)
```

```
     positions availability
 [1,]         0    0.4885572
 [2,]         1    0.4690808
 [3,]         2    0.4761674
 [4,]         3    0.4766474
 [5,]         4    0.4766457
 [6,]         5    0.4766422
 [7,]         6    0.4766424
 [8,]         7    0.4766428
 [9,]         8    0.4766432
[10,]         9    0.4766436
[11,]        10    0.4766440
```

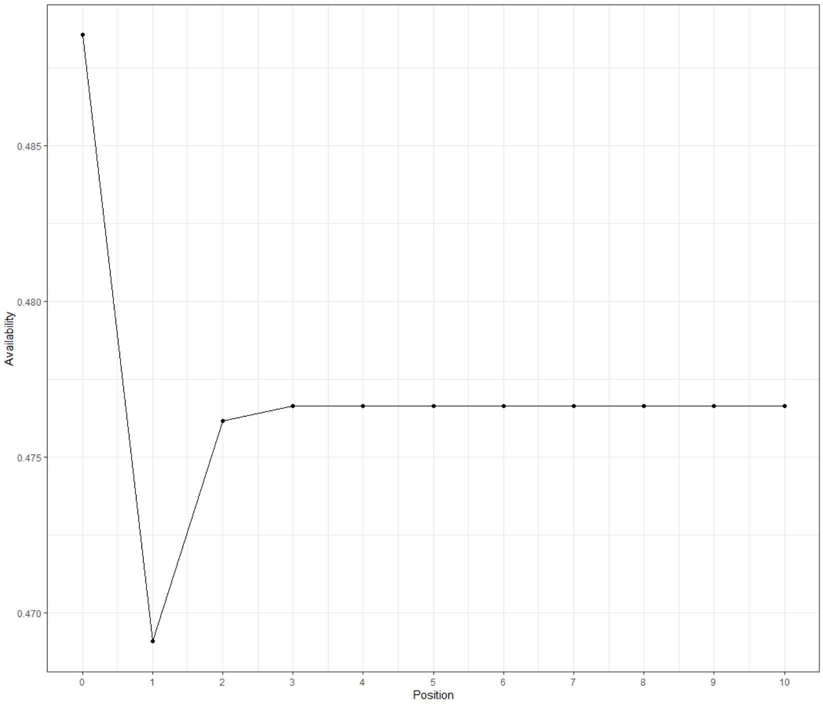

Figure 5: Figure plot of availability probabilities along the sequence between `k1` and `k2` position arguments for the working states subspace `s1`

*Estimation of maintainability*

```
 data(lambda, package = "drimmR")
dmm <- dmmsum(lambda,1,1,c("a","c","g","t"))
MAIN.out <- "C:\\...\\file.txt"
M(dmm,k1=1, k2=10,s1=c("a","c"),output_file = MAIN.out, plot=TRUE)
```

```
      positions maintainability
 [1,]         0       0.0000000
 [2,]         1       0.4885572
 [3,]         2       0.7164935
 [4,]         3       0.8485148
 [5,]         4       0.9189865
 [6,]         5       0.9566753
 [7,]         6       0.9768307
 [8,]         7       0.9876094
 [9,]         8       0.9933737
[10,]         9       0.9964564
[11,]        10       0.9981049
```

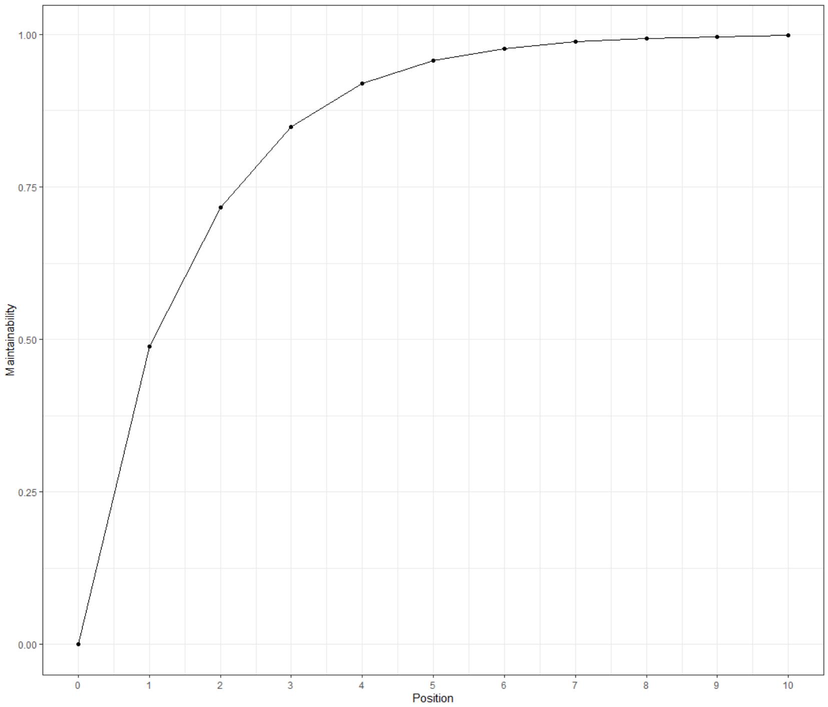

Figure 6: Figure plot of availability probabilities along the sequence between `k1` and `k2` position arguments for the working states subspace `s1`

*Estimation of failure rates*

```
 data(lambda, package = "drimmR")
dmm <- dmmsum(lambda,1,1,c("a","c","g","t"))
ER.out <- "C:\\...\\file.txt"
errorRate(mod,1,10,c("a","c"), error.rate="BMP", output_file=ER.out, plot=TRUE)
```

```
      positions       BMP
 [1,]         0 0.0000000
 [2,]         1 0.5114428
 [3,]         2 0.5064149
 [4,]         3 0.5069368
 [5,]         4 0.5069450
 [6,]         5 0.5069446
 [7,]         6 0.5069440
 [8,]         7 0.5069434
 [9,]         8 0.5069428
[10,]         9 0.5069422
[11,]        10 0.5069416
```

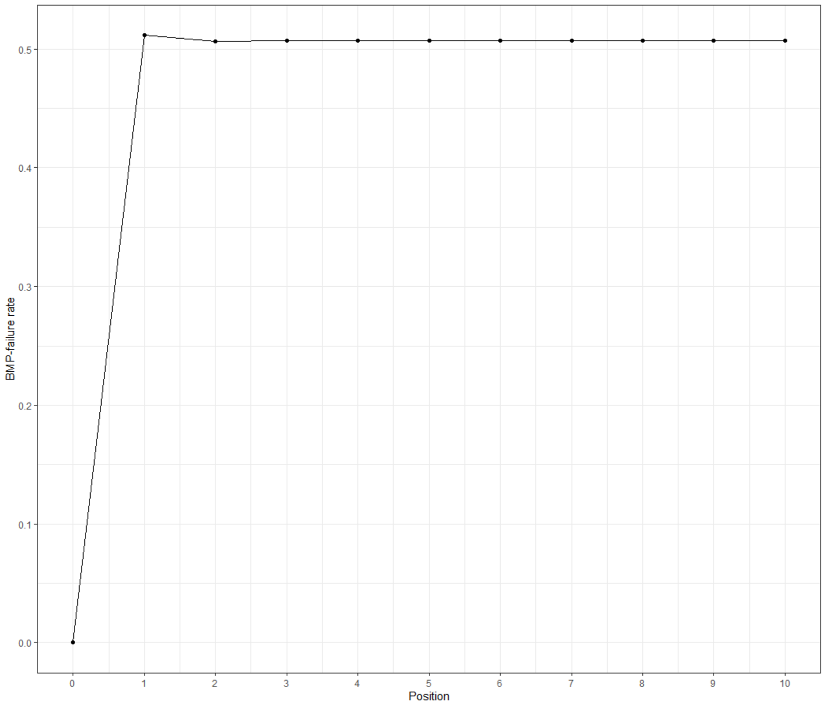

Figure 7: Figure plot of BMP-failure rates along the sequence between `k1` and `k2` position arguments for the working states subspace `s1`

## 4. Concluding remarks

To conclude, in this paper we have presented `drimmR`, an `R` package for simulation, estimation and reliability and survival analysis of drifting Markov models. These are versatile stochastic models of Markov type capable of taking into account a time non-homogeneity of a known, controlled shape. For this reason these models can represent interesting modeling alternatives to classical models (like Markov models, semi-Markov models, etc.) and can be useful for researchers, practitioners and engineers in various fields.

The package addresses several items important from practical point of view, when carrying out the estimation: we consider one or several samples, the sample paths can be complete or not, can come from models of the same length or of different lengths.

The fields of application of the DMMs can be numerous; we only want to point out three of them.

- *Survival analysis* and *reliability theory*: note that important indicators like reliability or survival function, availability, maintainability and failure rates are computed in our package;
- *Bioinformatics*: note that important quantities for OMICS data in general, DNA analysis in particular, are computed by the proposed package. Thus we have the computation of the probabilities and expectations of appearance of a given word along the sequence.

drimmr

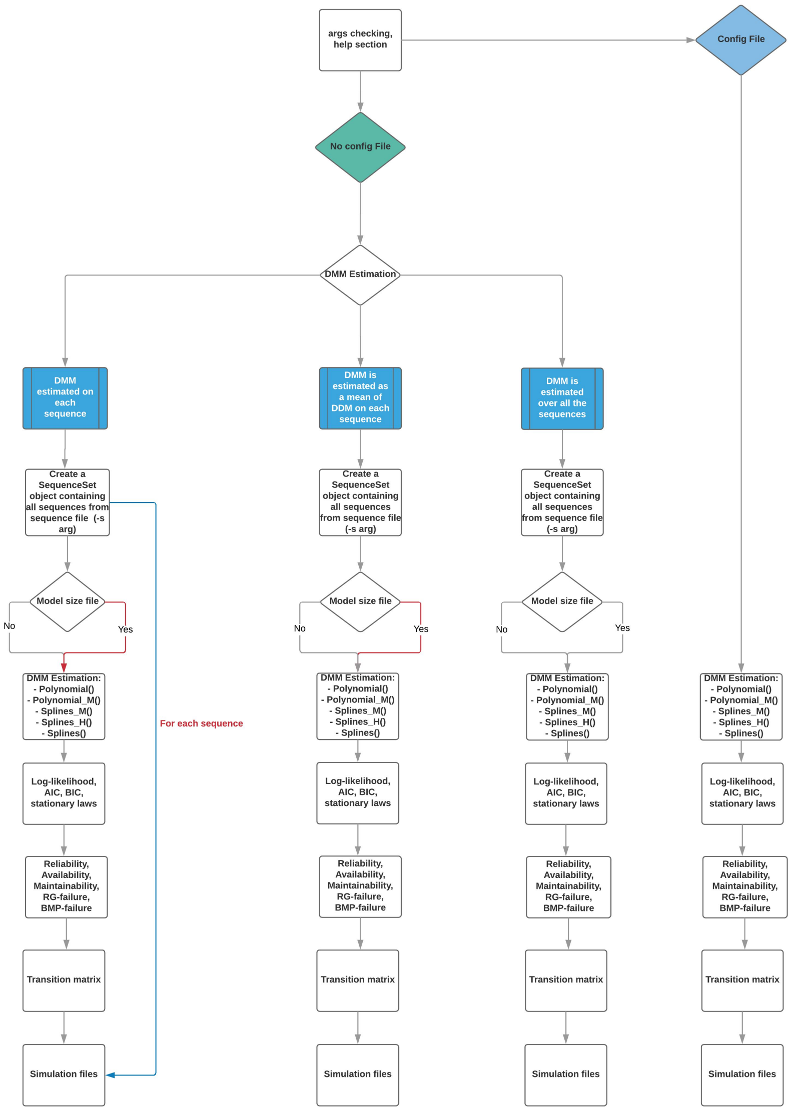
args checking, help section
Config File
No config File
DMM Estimation
DMM estimated on each sequence
DMM is estimated as a mean of DDM on each sequence
DMM is estimated over all the sequences
Create a SequenceSet object containing all sequences from sequence file (-s arg)
Create a SequenceSet object containing all sequences from sequence file (-s arg)
Create a SequenceSet object containing all sequences from sequence file (-s arg)
Model size file
No
Yes
Model size file
No
Yes
Model size file
No
Yes
For each sequence
DMM Estimation: - Polynomial() - Polynomial_M() - Splines_M() - Splines_H() - Splines()
DMM Estimation: - Polynomial() - Polynomial_M() - Splines_M() - Splines_H() - Splines()
DMM Estimation: - Polynomial() - Polynomial_M() - Splines_M() - Splines_H() - Splines()
DMM Estimation: - Polynomial() - Polynomial_M() - Splines_M() - Splines_H() - Splines()
Log-likelihood, AIC, BIC, stationary laws
Log-likelihood, AIC, BIC, stationary laws
Log-likelihood, AIC, BIC, stationary laws
Log-likelihood, AIC, BIC, stationary laws
Reliability, Availability, Maintainability, RG-failure, BMP-failure
Reliability, Availability, Maintainability, RG-failure, BMP-failure
Reliability, Availability, Maintainability, RG-failure, BMP-failure
Reliability, Availability, Maintainability, RG-failure, BMP-failure
Transition matrix
Transition matrix
Transition matrix
Transition matrix
Simulation files
Simulation files
Simulation files
Simulation files

Figure 8: Flowchart of the `drimmR package`.

**Affiliation:**

Firstname Lastname
Affiliation
Address, Country
E-mail: name@address
URL: http://link/to/webpage/