

# Package ‘drimmR’

February 18, 2021

**Type** Package

**Title** Estimation, Simulation and Reliability of drifting Markov Models

**Version** 1.0.0

**Author** Vlad Stefan Barbu,  
Geoffray Brelurut,  
Annthomy Gilles,  
Arnaud Lefebvre,  
Victor Mataigne,  
Alexandre Seiller,  
Nicolas Vergne

**Maintainer** Nicolas Vergne <Nicolas.Vergne@univ-rouen.fr>

**Description** This package introduce the drifting Markov models (DMMs) which are inhomogeneous Markov models designed for modeling the heterogeneities of sequences in a more flexible way than homogeneous Markov chains or even hidden Markov. In this context, we developed a R package dedicated to the estimation, simulation and reliability of drifting Markov models. The implemented methods are described in Vergne, N. (2008), <doi:10.2202/1544-6115.1326>, Barbu, V.S., Vergne, N. (2019) <doi:10.1007/s11009-018-9682-8>

**License** GPL-3 + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Suggests** testthat,  
knitr,  
rmarkdown

**Imports** seqinr,  
Biostrings,  
testthat,  
stats,  
ggplot2,  
utils,  
parallel,  
foreach,  
future,  
doParallel,  
doSNOW,  
tidyverse

**RoxygenNote** 7.1.1  
**VignetteBuilder** knitr  
**Depends** R (>= 2.10)

**R topics documented:**

A	2
aic	3
aic.dmmsum	4
bic	5
bic.dmmsum	5
Distribution_evol	6
dmmsum	7
errorRate	8
getDistribution	10
getDistribution.dmmsum	10
getStationaryLaw	11
getStationaryLaw.dmmsum	12
getTransitionMatrix	13
getTransitionMatrix.dmmsum	14
lambda	14
length_probas	15
loglik	16
loglik.dmmsum	16
M	17
R	18
simulate	19
simulate.dmmsum	19
stationaryLaw_evol	20
words_probas	21
word_proba	22
word_probas	23
<b>Index</b>	<b>24</b>

---

A	<i>Evaluate Availability</i>
---	------------------------------

---

**Description**

Estimation of the pointwise (or instantaneous) availability for (ergodic) repairable systems at time  $l \in N$

**Usage**

A(x, k1, k2, s1, output\_file = NULL, plot = FALSE)

**Arguments**

x	An object of class "dmm"
k1	start position (numeric)
k2	end position (numeric)
s1	Character vector of the subspace working states among the state space vector s.t. $s_1 < s$
output_file	A file containing matrix of availability probabilities
plot	FALSE (no figure plot of availability by position); TRUE (figure plot)

**Details**

The pointwise (or instantaneous) availability is the probability that the system is in a working state at time  $l$ , independently of the fact that the system worked or not during the time interval  $[0; l)$

**Value**

A matrix with Availability score at each position

**Author(s)**

Alexandre Seiller

**See Also**

[dmmsum](#), [getTransitionMatrix](#), [R](#), [M](#)

**Examples**

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda, 1, 1, c("a", "c", "g", "t"))
k1 <- 1
k2 <- 200
s1 <- c("c", "t") # vector of working states
AVA.out <- "C:\\...\\file.txt"
A(dmm, k1, k2, s1, output_file=AVA.out, plot=FALSE)
```

---

aic

---

*Akaike Information Criterion (AIC)*


---

**Description**

Generic function computing the Akaike Information Criterion of the model 'x', with the list of sequences 'sequences'.

**Usage**

```
aic(x, sequence)
```

**Arguments**

- x** An object of class "dmm" for which the loglikelihood can be computed.
- sequence** A vector of character or a list of vector of character representing the sequences for which the AIC criterion must be computed.

**Value**

A numeric value giving the value of the AIC.

**Author(s)**

Victor Mataigne, Alexandre Seiller

---

aic.dmmsum	<i>Evaluate AIC</i>
------------	---------------------

---

**Description**

Evaluate AIC

**Usage**

```
## S3 method for class 'dmmsum'
aic(x, sequences)
```

**Arguments**

- x** An object of class "dmm", [dmmsum](#)
- sequence** A character vector or a list of character vector representing the sequence

**Value**

A list of numeric, AIC

**Author(s)**

Victor Mataigne, Alexandre Seiller

**See Also**

[dmmsum](#), [getTransitionMatrix](#), [loglik](#), [aic](#)

**Examples**

```
data(lambda, package = "drimmR")
sequence <- c("a","g","g","t","c","g","a","t","a","a","a")
dmm <- point_estimate(lambda, 1, 1, c('a','c','g','t'), 1000000)
aic(dmm, sequence)
```

---

bic	<i>Bayesian Information Criterion (BIC)</i>
-----	---

---

**Description**

Generic function computing the Bayesian Information Criterion of the model 'x', with the list of sequences 'sequences'.

**Usage**

```
bic(x, sequence)
```

**Arguments**

x	An object of class "dmm" for which the loglikelihood can be computed.
sequence	A vector or a list of vector of character representing the sequences for which the BIC criterion must be computed.

**Value**

A numeric value giving the value of the BIC.

**Author(s)**

Victor Mataigne, Alexandre Seiller

---

bic.dmmsum	<i>Evaluate BIC</i>
------------	---------------------

---

**Description**

Evaluate BIC

**Usage**

```
## S3 method for class 'dmmsum'
bic(x, sequences)
```

**Arguments**

x	An object of class "dmm", <a href="#">dmmsum</a>
sequence	A character vector or a list of character vector representing the sequence

**Value**

A numeric, BIC

**Author(s)**

Victor Mataigne, Alexandre Seiller

**See Also**

[dmmsum](#), [getTransitionMatrix](#), [loglik](#), [bic](#)

**Examples**

```
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")
bic(Dmm,sequence)
```

---

Distribution_evol	<i>Plot distributions for a range of positions between &lt;start&gt; and &lt;end&gt;</i>
-------------------	--

---

**Description**

Plot distributions for a range of positions between <start> and <end>

**Usage**

```
Distribution_evol(
  x,
  start = 1,
  end = NULL,
  step = NULL,
  output_file = NULL,
  plot = FALSE
)
```

**Arguments**

x	An object of class "dmm"
start	Start position (numeric)
end	End position (numeric)
step	A step (numeric)
output_file	A file containing matrix of distributions
plot	FALSE (no figure plot of dist evolution); TRUE (figure plot)

**Value**

A matrix of distributions with position and probability of states

**Author(s)**

Alexandre Seiller

**See Also**

[dmmsum](#), [getDistribution](#), [getStationaryLaw](#)

## Examples

```
#' data(lambda, package = "drimmR")
dmm <- dmmsum(lambda, 1, 1, c('a','c','g','t'), init.estim = "freq")
DIST.out <- "C:\\...\\file.txt"
Distributionevol(dmm,start=1,end=length(lambda)-1,step=10000, output_file=DIST.out, plot=FALSE)
```

dmmsum

*Point by point estimates of a k-th order Drifting-Markov Model*

## Description

Estimation of  $d+1$  points of support transition matrices and  $|E|^k$  initial law of a  $k$ -th order Drifting Markov Model starting from one or several sequences.

## Usage

```
dmmsum(
  sequences,
  order,
  degree,
  states,
  init.estim = c("mle", "freq", "prod", "stationary", "unif", ...)
)
```

## Arguments

sequences	A list of character vector(s) representing one (several) sequence(s)
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	Default="mle". Method used to estimate the initial law. If 'init.estim' = "mle", then the classical Maximum Likelihood Estimator is used, if 'init.estim' = "freq", then, the initial distribution 'init.estim' is estimated by taking the frequencies of the words of length 'k' for all sequences. If 'init.estim' = "prod", then, 'init.estim' is estimated by using the product of the frequencies of each letter (for all the sequences) in the word of length 'k'. If init.estim = "stationary", then 'init.estim' is estimated by using the stationary law of the point of support transition matrices of each letter. If 'init.estim' = "unif", then, 'init.estim' of each letter is estimated by using $\frac{1}{s}$ . Or init.estim= customisable vector of length $ E ^k$ . See Details for the formulas.

## Details

The 'dmmsum' function creates a Drifting-Markov model object ('dmm').

Let  $E = 1, \dots, s$ ,  $s < \infty$  be random system with finite state space, with a time evolution governed by discrete-time stochastic process of values in  $E$ . A sequence  $X_0, X_1, \dots, X_n$  with state space  $E = 1, 2, \dots, s$  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  $X_t$ ,  $t = 1, \dots, n$ , is defined by  $P(X_t = v \mid X_{t-1} = u, X_{t-2}, \dots) = \Pi_{\frac{t}{n}}(u, v)$ ,  $u, v \in E$ , where  $\Pi_{\frac{t}{n}}(u, v) = (1 - \frac{t}{n})\Pi_0(u, v) +$

$\frac{t}{n} \Pi_1(u, v)$ ,  $u, v \in E$ . The linear drifting Markov model of order 1 can be generalized to polynomial Drifting-Markov model of order  $k$  and degree  $d$ . Let  $\Pi_{\frac{i}{d}} = (\Pi_{\frac{i}{d}}(u_1, \dots, u_k, v))_{u_1, \dots, u_k, v \in E}$  be  $d$  Markov transition matrices (of order  $k$ ) over a state space  $E$ .

The initial distribution of a  $k$ -th order Drifting Markov Model is defined as  $\mu_i = P(X_1 = i)$ . The initial distribution of the  $k$  first letters is freely be customisable by the user, but five methods are proposed for the estimation of the latter :

**Estimation based on the Maximum Likelihood Estimator:** The Maximum Likelihood Estimator for the initial distribution. The formula is:  $\hat{\mu}_i = \frac{N_{start_i}}{L}$ , where  $N_{start_i}$  is the number of occurrences of the word  $i$  (of length  $k$ ) at the beginning of each sequence and  $L$  is the number of sequences. This estimator is reliable when the number of sequences  $L$  is high.

**Estimation based on the frequency:** The initial distribution is estimated by taking the frequencies of the words of length ' $k$ ' for all sequences. The formula is  $\hat{\mu}_i = \frac{N_i}{N}$ , where  $N_i$  is the number of occurrences of the word  $i$  (of length  $k$ ) in the sequences and  $N$  is the sum of the lengths of the sequences.

**Estimation based on the product of the frequencies of each state:** The initial distribution is estimated by using the product of the frequencies of each state (for all the sequences) in the word of length  $k$ .

**Estimation based on the stationary law of point of support transition matrix for a word of length  $k$  :**  
The initial distribution is estimated using  $\mu(\Pi_{\frac{k-1}{n}})$

**Estimation based on the uniform law :**  $\frac{1}{s}$

## Value

An object of class "dmm", [dmmsum](#)

## Author(s)

Geoffray Brelurut, Alexandre Seiller

## Examples

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

---

errorRate

*Evaluate error rates*

---

## Description

Estimation of two different definition of the failure rate : the BMP-failure rate and RG-failure rate



**Usage**

```
errorRate(
  x,
  k1,
  k2,
  s1,
  error.rate = c("BMP", "RG"),
  output_file = NULL,
  plot = FALSE
)
```

**Arguments**

x	An object of class "dmm"
k1	start position (numeric)
k2	end position (numeric)
s1	Character vector of the subspace working states among the state space vector s.t. $s1 < s$
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.
output_file	A file containing matrix of error rates at each position
plot	FALSE (no figure plot of error rates by position); TRUE (figure plot)

**Details**

The BMP-failure rate denoted by  $\lambda(l), l \in N$  is usually considered for continuous time systems.

The RG-failure rate denoted by  $r(l), l \in N$  is adapted to work in discrete time systems.

**Value**

A matrix with error rate score at each position

**Author(s)**

Alexandre Seiller

**See Also**

[dmmsum](#), [getTransitionMatrix](#), [R](#)

**Examples**

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda, 1, 1, c("a", "c", "g", "t"))
k1 <- 1
k2 <- 200
s1 <- c("c", "t") # vector of working states
ER.out <- "C:\\...\\file.txt"
errorRate(dmm, k1, k2, s1, error.rate="BMP", output_file=ER.out, plot=FALSE)
```

---

getDistribution	<i>Get distribution at a given position or at every positions</i>
-----------------	---

---

**Description**

Evaluate distribution at a given position or at every position

**Usage**

```
getDistribution(x, pos, all.pos = FALSE, internal = FALSE)
```

**Arguments**

x	An object of class "dmm"
pos	An integer, a position
all.pos	FALSE (evaluation at pos index) ; TRUE (evaluation for all pos indices)
internal	FALSE (default) ; TRUE (for internal use of distrib_evol function)

**Details**

Distribution at position  $l$  is evaluated by  $\mu_l = \mu_0 \prod_{t=k}^l \pi_{\frac{t}{n}}, \forall l \geq k, k \in N^*$  order of the DMM

**Value**

A vector or matrix of distribution probabilities

**Author(s)**

Alexandre Seiller

**Examples**

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda, 1, 1, c('a','c','g','t'), init.estim = "freq")
t <- 10
getDistribution(dmm,pos=t)
```

---

getDistribution.dmmsum	<i>Get distribution</i>
------------------------	-------------------------

---

**Description**

Evaluate distribution at a given position or at every position

**Usage**

```
## S3 method for class 'dmmsum'
getDistribution(x, pos, all.pos = FALSE, internal = FALSE)
```

**Arguments**

x	An object of class "dmm", <a href="#">dmmsum</a>
pos	An integer for position
all.pos	FALSE (default, evaluation at position index) ; TRUE (evaluation for all position indices)
internal	FALSE (default) ; TRUE (for internal use of distrib_evol function)

**Details**

Distribution at position  $l$  is evaluated by  $\mu_l = \mu_0 \prod_{t=k}^l \pi_{\frac{t}{n}}, \forall l \geq k, k \in N^*$  order of the DMM

**Value**

A vector or matrix of distribution probabilities

**Author(s)**

Alexandre Seiller

**See Also**

[dmmsum](#), [getTransitionMatrix](#), [Distribution\\_evol](#), [getStationaryLaw](#)

**Examples**

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda, 1, 1, c('a','c','g','t'), init.estim = "freq")
t <- 10
getDistribution(dmm,pos=t)
```

---

getStationaryLaw

*Get stationary law at a given position or at every positions*


---

**Description**

Evaluate stationary law at a given position or at every position

**Usage**

```
getStationaryLaw(x, pos, all.pos = FALSE, internal = FALSE)
```

**Arguments**

x	An object of class "dmm"
pos	An integer, a position
all.pos	FALSE (default, evaluation at position index) ; TRUE (evaluation for all position indices)
internal	FALSE (default) ; TRUE (for internal use of dmmsum initial law)

**Details**

Stationary law at position  $l$  is evaluated by solving  $\mu_k \pi_{\frac{k}{n}} = \mu$

**Value**

A vector or matrix of stationary law probabilities

**Author(s)**

Alexandre Seiller

**Examples**

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda, 1, 1, c('a','c','g','t'), init.estim = "freq")
t <- 10
getStationaryLaw(dmm,pos=t)
```

---

```
getStationaryLaw.dmmsum
```

*Get stationary law*

---

**Description**

Evaluate stationary law at a given position or at every position

**Usage**

```
## S3 method for class 'dmmsum'
getStationaryLaw(x, pos, all.pos = FALSE, internal = FALSE)
```

**Arguments**

x	An object of class "dmm", <a href="#">dmmsum</a>
pos	An integer for position
all.pos	FALSE (default, evaluation at position index) ; TRUE (evaluation for all position indices)
internal	FALSE (default) ; TRUE (for internal use of dmmsum initial law and word applications)

**Details**

Stationary law at position  $l$  is evaluated by solving  $\mu_k \pi_{\frac{k}{n}} = \mu$

**Value**

A vector or matrix of stationary law probabilities

**Author(s)**

Alexandre Seiller

**See Also**

[dmmsum](#), [getTransitionMatrix](#), [stationaryLaw\\_evol](#), [getDistribution](#)

**Examples**

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda, 1, 1, c('a','c','g','t'), init.estim = "freq")
t <- 10
getStationaryLaw(dmm,pos=t)
```

---

getTransitionMatrix	<i>Get transition matrix at a given position</i>
---------------------	--

---

**Description**

Get transition matrix at a given position

**Usage**

```
getTransitionMatrix(x, pos)
```

**Arguments**

x	An object of class "dmm"
pos	An integer, a position

**Value**

A transition matrix at a given position

**Author(s)**

Victor Mataigne, Alexandre Seiller

**Examples**

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

---

```
getTransitionMatrix.dmmsum
```

*Get transition matrix at a given position*

---

### Description

Get transition matrix at a given position

### Usage

```
## S3 method for class 'dmmsum'
getTransitionMatrix(x, pos)
```

### Arguments

x	An object of class "dmm", <a href="#">dmmsum</a>
pos	position along the sequence (integer)

### Value

A transition matrix at a given position

### Author(s)

Victor Mataigne, Alexandre Seiller

### See Also

[dmmsum](#)

### Examples

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

---

lambda

*lambda genome*

---

### Description

Complete data from phage genome [WT71] of length 48502

### Usage

```
data(lambda)
```

### Format

A vector object of class "Rdata".

**Examples**

```
data(lambda)
```

---

length_probas	<i>Probability of occurrence of the Observed word of size n in a sequence at several positions</i>
---------------	--

---

**Description**

Probability of occurrence of the Observed word of size n in a sequence at several positions

**Usage**

```
length_probas(n, sequence, pos, x, output_file = NULL, plot = FALSE)
```

**Arguments**

n	An integer, the length word
sequence	A vector of characters
pos	A vector of integer positions
x	An object of class "dmm"
output_file	A file containing the vector of probabilities
plot	FALSE (no figure plot of words probabilities); TRUE (figure plot)

**Value**

a dataframe of probability by position (and probability plots)

**Author(s)**

Victor Mataigne, Alexandre Seiller

**See Also**

[dmmsum](#), [getTransitionMatrix](#), [word\\_proba](#)

**Examples**

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

---

loglik	<i>Loglikelihood</i>
--------	----------------------

---

**Description**

Generic function computing the loglikelihood of the model 'x', with the list of sequences 'sequences'.

**Usage**

```
loglik(x, sequences)
```

**Arguments**

x	An object of class "dmm" for which the loglikelihood can be computed.
sequences	A vector of character or list of vectors representing the sequences for which the log-likelihood of the model must be computed.

**Value**

A numeric, the log-likelihood

**Author(s)**

Annthomy Gilles, Alexandre Seiller

---

loglik.dmmsum	<i>Evaluate Log-likelihood</i>
---------------	--------------------------------

---

**Description**

Evaluate Log-likelihood

**Usage**

```
## S3 method for class 'dmmsum'
loglik(x, sequences)
```

**Arguments**

x	An object of class "dmm", <a href="#">dmmsum</a>
sequences	A character vector or a list of character vectors representing the sequence

**Value**

A list of log-likelihood (numeric)

**Author(s)**

Annthomy Gilles, Alexandre Seiller



**See Also**

[dmmsum](#), [getTransitionMatrix](#)

**Examples**

```
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)
```

---

M	<i>Evaluate Maintainability</i>
---	---------------------------------

---

**Description**

Estimation of maintainability for (ergodic) repairable system at time  $k \in N$ .

**Usage**

```
M(x, k1, k2, s1, output_file = NULL, plot = FALSE)
```

**Arguments**

x	An object of class "dmm"
k1	start position (numeric)
k2	end position (numeric)
s1	Character vector of the subspace working states among the state space vector s.t. $s1 < s$
output_file	A file containing matrix of maintainability probabilities
plot	FALSE (no figure plot of maintainability by position); TRUE (figure plot)

**Details**

The maintainability at time  $k \in N$  of a repairable system is the probability that the system is repaired up to time  $l$ , given that it has failed at time  $l = 0$ .

**Value**

A matrix with Maintainability score at each position

**Author(s)**

Alexandre Seiller

**See Also**

[dmmsum](#), [getTransitionMatrix](#)

## Examples

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda,1,1,c("a","c","g","t"))
k1 <- 1
k2 <- 200
s1 <- c("c","t") # vector of working states
MAIN.out <- "C:\\...\\file.txt"
M(dmm,k1,k2,s1, output_file=MAIN.out,plot=FALSE)
```

---

R

---

*Evaluate Reliability*


---

## Description

Estimation of reliability for (ergodic) repairable or (non-ergodic) non-repairable systems at time  $l \in N$

## Usage

```
R(x, k1, k2, s1, output_file = NULL, plot = FALSE)
```

## Arguments

x	An object of class "dmm"
k1	start position (numeric)
k2	end position (numeric)
s1	Character vector of the subspace working states among the state space vector s.t. $s1 < s$
output_file	A file containing matrix of reliability probabilities
plot	FALSE (no figure plot of reliability by position); TRUE (figure plot)

## Details

The reliability at time  $l \in N$  is the probability thaht the system has functioned without failure in the period  $[0, l]$

## Value

A matrix with Reliability score at each position

## Author(s)

Alexandre Seiller

## See Also

[dmmsum](#), [getTransitionMatrix](#)

**Examples**

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda,1,1,c("a","c","g","t"))
k1 <- 1
k2 <- 200
s1 <- c("c","t") # vector of working states
REL.out <- "C:\\...\\file.txt"
R(dmm,k1,k2,s1, output_file=REL.out,plot=FALSE)
```

simulate

*Simulate a sequence with the Drifting Markov Model***Description**

Simulate a sequence with the Drifting Markov Model

**Usage**

```
simulate(x, output_file, model_size = 1e+05)
```

**Arguments**

x	An object of class "dmm"
output_file	A file containing matrix of probabilities
model_size	Size of the model

**Author(s)**

Annthomy Gilles, Alexandre Seiller

**Examples**

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

simulate.dmmsum

*Simulate a sequence with the Drifting Markov Model***Description**

Simulate a sequence with the Drifting Markov Model

**Usage**

```
## S3 method for class 'dmmsum'
simulate(x, output_file, model_size = NULL)
```

**Arguments**

x	An object of class "dmm", <a href="#">dmmsum</a>
output_file	File containing the simulated sequence
model_size	Size of the model

**Value**

the vector of simulated sequence

**Author(s)**

Annthomy Gilles, Alexandre Seiller

**See Also**

[dmmsum](#), [getTransitionMatrix](#), [getStationaryLaw](#)

**Examples**

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

---

stationaryLaw_evol	<i>Plot stationary laws for a range of positions between &lt;start&gt; and &lt;end&gt;</i>
--------------------	--

---

**Description**

Plot stationary laws for a range of positions between <start> and <end>

**Usage**

```
stationaryLaw_evol(
  x,
  start = 1,
  end = NULL,
  step = NULL,
  output_file = NULL,
  plot = FALSE
)
```

**Arguments**

x	An object of class "dmm"
start	Start position (numeric)
end	End position (numeric)
step	A step (numeric)
output_file	A file containing matrix of stationary laws
plot	FALSE (no figure plot of SL evolution); TRUE (figure plot)

**Value**

A matrix of probabilities with position and probability of states (and figure plot)

**Author(s)**

Alexandre Seiller

**See Also**

[dmmsum](#), [getStationaryLaw](#)

**Examples**

```
#' data(lambda, package = "drimmR")
dmm <- dmmsum(lambda, 1, 1, c('a','c','g','t'), init.estim = "freq")
SL.out <- "C:\\...\\file.txt"
stationaryLaw_evol(dmm,start=10,end=1000,step=301, output_file=SL.out, plot=FALSE)
```

---

words\_probas

---

*Probability of several words at several positions of a DMM*


---

**Description**

Probability of several words at several positions of a DMM

**Usage**

```
words_probas(words, pos, x, output_file = NULL, plot = FALSE)
```

**Arguments**

words	A vector of characters containing words
pos	A vector of integer positions
x	An object of class "dmm"
output_file	A file containing the matrix of probabilities
plot	FALSE (no figure plot of words probabilities); TRUE (figure plot)

**Value**

a dataframe of word probabilities along the positions of the sequence

**Author(s)**

Victor Mataigne, Alexandre Seiller

**See Also**

[dmmsum](#), [getTransitionMatrix](#), [word\\_proba](#), [word\\_probas](#)

**Examples**

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda, 1, 1, c('a','c','g','t'), init.estim = "freq")
PROB.out <- "C:\\...\\file.txt"
words_probas(c("atcgattc", "taggct", "ggatcgg"),c(100,300),dmm, output_file=PROB.out, plot=FALSE)
```

---

word_proba	<i>Probability of a word at a position t of a DMM</i>
------------	---

---

**Description**

Probability of a word at a position t of a DMM

**Usage**

```
word_proba(word, pos, x, output_file = NULL, internal = FALSE)
```

**Arguments**

word	A subsequence (string of characters)
pos	A position (numeric)
x	An object of class "dmm"
output_file	A file containing the probability
internal	FALSE (default) ; TRUE (for internal use of word applications)

**Value**

A numeric, probability of word

**Author(s)**

Victor Mataigne, Alexandre Seiller

**See Also**

[dmmsum](#), [getTransitionMatrix](#), [word\\_probas](#), [words\\_probas](#)

**Examples**

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda, 1, 1, c('a','c','g','t'), init.estim = "freq")
PROB.out <- "C:\\...\\file.txt"
word_proba("aggctga",4,dmm, output_file=PROB.out)
```

---

word_probas	<i>Probabilities of a word at several positions of a DMM</i>
-------------	--

---

**Description**

Probabilities of a word at several positions of a DMM

**Usage**

```
word_probas(word, pos, x, output_file = NULL, plot = FALSE)
```

**Arguments**

word	A subsequence (string of characters)
pos	A vector of integer positions
x	An object of class "dmm"
output_file	A file containing the vector of probabilities
plot	FALSE (no figure plot of word probabilities); TRUE (figure plot)

**Value**

A numeric vector, probabilities of word

**Author(s)**

Victor Mataigne, Alexandre Seiller

**See Also**

[dmmsum](#), [getTransitionMatrix](#), [word\\_proba](#), [words\\_probas](#)

**Examples**

```
data(lambda, package = "drimmR")
dmm <- dmmsum(lambda, 1, 1, c('a','c','g','t'), init.estim = "freq")
PROB.out <- "C:\\...\\file.txt"
word_probas("aggctga",c(100,300),dmm, output_file=PROB.out, plot=FALSE)
```

# Index

## \* datasets

lambda, 14

A, 2

aic, 3, 4

aic.dmmsum, 4

bic, 5, 6

bic.dmmsum, 5

Distribution\_evol, 6, 11

dmmsum, 3–6, 7, 8, 9, 11–18, 20–23

errorRate, 8

getDistribution, 6, 10, 13

getDistribution.dmmsum, 10

getStationaryLaw, 6, 11, 11, 20, 21

getStationaryLaw.dmmsum, 12

getTransitionMatrix, 3, 4, 6, 9, 11, 13, 13,  
15, 17, 18, 20–23

getTransitionMatrix.dmmsum, 14

lambda, 14

length\_probas, 15

loglik, 4, 6, 16

loglik.dmmsum, 16

M, 3, 17

R, 3, 9, 18

simulate, 19

simulate.dmmsum, 19

stationaryLaw\_evol, 13, 20

word\_proba, 15, 21, 22, 23

word\_probas, 21, 22, 23

words\_probas, 21, 22, 23