

# Package: pmultinom (via r-universe)

September 2, 2024

**Type** Package

**Title** One-Sided Multinomial Probabilities

**Version** 1.0.0

**Author** Alexander Davis

**Maintainer** Alexander Davis <ajdavis2@mdanderson.org>

**Description** Implements multinomial CDF ( $P(N_1 \leq n_1, \dots, N_k \leq n_k)$ ) and tail probabilities ( $P(N_1 > n_1, \dots, N_k > n_k)$ ), as well as probabilities with both constraints ( $P(l_1 < N_1 \leq u_1, \dots, l_k < N_k \leq u_k)$ ). Uses a method suggested by Bruce Levin (1981) <doi:10.1214/aos/1176345593>.

**License** AGPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**Imports** fftw

**Suggests** testthat

**NeedsCompilation** no

**Date/Publication** 2018-04-24 15:42:54 UTC

**Repository** <https://alex-l-m.r-universe.dev>

**RemoteUrl** <https://github.com/cran/pmultinom>

**RemoteRef** HEAD

**RemoteSha** da5a1e170b5ca367941fc715828a06709577f701

## Contents

invert.pmultinom . . . . .	2
pmultinom . . . . .	3
<b>Index</b>	<b>6</b>

---

invert.pmultinom	<i>Calculate the sample size such that the probability of a result is a given amount.</i>
------------------	---

---

### Description

Calculate the sample size such that the probability of a result is a given amount.

### Usage

```
invert.pmultinom(lower = -Inf, upper = Inf, probs, target.prob, method)
```

### Arguments

lower	Vector of lower bounds. Lower bounds are excluded
upper	Vector of upper bounds. Upper bounds are included
probs	Cell probabilities
target.prob	The probability of the event, at the output sample size.
method	Method used for computation. Only method currently implemented is "exact"

### Details

If only lower is given, then the result is the smallest size such that `pmultinom(lower=lower, size=size, probs=probs) >= target.prob`. If only upper is given, then the result is the smallest size such that `pmultinom(upper=upper, size=size, probs=probs) <= target.prob`. Behavior when both lower and upper are given is not yet implemented.

### Value

The sample size parameter at which the the target probability of the given event is achieved.

### References

Casasent, A. K., Schalck, A., Gao, R., Sei, E., Long, A., Pangburn, W., ... & Navin, N. E. (2018). Multiclonal Invasion in Breast Tumors Identified by Topographic Single Cell Sequencing. *Cell*. doi:10.1016/j.cell.2017.12.007

### See Also

[pmultinom](#)

**Examples**

```

# How many cells must be sequenced to have a 95% chance of
# observing at least 2 from each subclone of a tumor? (Data
# from Casasent et al (2018); see vignette("pmultinom") for
# details of this example)

# Input:
ncells <- 204
subclone.freqs <- c(43, 20, 82, 17, 5, 37)/ncells
target.number <- c(2, 2, 2, 2, 2, 0)
lower.bound <- target.number - 1
invert.pmultinom(lower=lower.bound, probs=subclone.freqs,
                 target.prob=.95, method="exact")

# Output:
# [1] 192

```

---

pmultinom	<i>Calculate the probability that a multinomial random vector is between, elementwise, two other vectors.</i>
-----------	---

---

**Description**

Calculate the probability that a multinomial random vector is between, elementwise, two other vectors.

**Usage**

```
pmultinom(lower = -Inf, upper = Inf, size, probs, method)
```

**Arguments**

lower	Vector of lower bounds. Lower bounds are excluded
upper	Vector of upper bounds. Upper bounds are included
size	Number of draws
probs	Cell probabilities
method	Method used for computation. Only method currently implemented is "exact"

**Details**

The calculation follows the scheme suggested in Levin (1981): begin with the equivalent probability for a Poisson random vector, and update it by conditioning on the sum of the vector being equal to the size parameter, using Bayes' theorem. This requires computation of the distribution of a sum of truncated Poisson random variables, which is accomplished using convolution, as per Levin's suggestion for an exact calculation. Levin's suggestion for an approximate calculation, using Edgeworth expansions, may be added to a later version. Fast convolution is achieved using the fastest Fourier transform in the west (Frigo, Johnson 1998).

**Value**

The probability that a multinomial random vector is greater than all the lower bounds, and less than or equal all the upper bounds:

$$P(l_1 < N_1 \leq u_1, \dots, l_k < N_k \leq u_k)$$

If only the upper bounds are given, then this is the multinomial CDF:

$$P(N_1 \leq u_1, \dots, N_k \leq u_k)$$

If only the lower bounds are given, then this is the multinomial tail probability:

$$P(N_1 > l_1, \dots, N_k > l_k)$$
**References**

Fougere, P. F. (1988). Maximum entropy calculations on a discrete probability space. In *Maximum-Entropy and Bayesian Methods in Science and Engineering* (pp. 205-234). Springer, Dordrecht. doi:10.1007/978-94-009-3049-0\_10

Frigo, Matteo, and Steven G. Johnson. (2005). The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2), 216-231. doi:10.1109/JPROC.2004.840301

Levin, Bruce. (1981). "A Representation for Multinomial Cumulative Distribution Functions". *Annals of Statistics* 9 (5): 1123–6. doi:10.1214/aos/1176345593

**See Also**

[invert.pmultinom](#)

**Examples**

```
# To determine the bias of a die, Rudolph Wolf rolled it
# 20,000 times. Side 2 was the most frequently observed, and
# was observed 3631 times. What is the probability that a
# fair die would have a side observed this many times or
# more?
```

```
# Input:
1 - pmultinom(upper=rep.int(3630, 6), size=20000,
              probs=rep.int(1/6, 6), method="exact")
```

```
# Output:
# [1] 7.379909e-08
```

```
# Therefore we conclude that the die is biased. Fougere
# (1988) attempted to account for these biases by assuming
# certain manufacturing errors. Repeating the calculation
# with the distribution Fougere derived:
```

```
# Input:
theoretical.dist <- c(.17649, .17542, .15276, .15184, .17227, .17122)
1 - pmultinom(upper=rep.int(3630, 6), size=20000,
              probs=theoretical.dist, method="exact")
```

```
# Output:
# [1] 0.043362
```

```
# Therefore we conclude that the die still seems more biased  
# than Fougere's model can explain.
```

# Index

`invert.pmultinom`, 2, 4

`pmultinom`, 2, 3