Package ‘mixingsimulation’

November 25, 2020

Type Package

Title Simulation for powder mixing process

Version 0.0.0

Authors Mayooran Thevaraja [aut, cre], Kondaswamy Govindaraju [aut], Mark Bebbington [aut]

URL `https://github.com/Mayooran1987/mixingsimulation`

BugReports `https://github.com/Mayooran1987/mixingsimulation/issues`

Description This package develops for simulating powder mixing process for microbial risk assessment in the bulk material production process.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports ggplot2, ggthemes, plyr, reshape2, stats

Suggests testthat

RoxygenNote 7.1.1

Depends R (>= 3.2)

Maintainer Mayooran Thevaraja <`mayooran@eng.jfn.ac.lk`>

Language en-US

R topics documented:

compare_mixing 2
sim_multiple 3
sim_single 4

Index **6**

`compare_mixing` *The graphical comparison between different mixing schemes with varying parameters of mixing by simulation results.*

Description

This function provides the graphical displays for a different set of mixing plans for comparison purpose. (to be finished later on)

Usage

```
compare_mixing(n_iter, mu, sigma, b, k, distribution)
```

Arguments

`n_iter`	number of iterations
`mu`	average number of colony-forming units in a primary sample which is in logarithmic scale if we use a lognormal distribution
`sigma`	log standard deviation of the colony-forming units in a primary sample
`b`	concentration parameter
`k`	number of small portions/ primary samples
`distribution`	what suitable distribution type we have employed for simulation such as 'Poisson-fair' or 'Poisson-beta' or 'Lognormal-fair' or 'Lognormal-beta'

Details

Let N' be the number of colony-forming units in the mixed sample which is produced by mixing of k primary sample and $N' = \sum(N_i)$ and i be the number of colony-forming units in the i^{th} primary sample; where $i = 1, 2, ...k$ and $y_i = x_i / \sum(x_i) = q_i/Q$; where x_i follows $gamma(b, 1)$ (to be finished later on)

- Case 1 (Poisson-fair): N_i follows $Poisson(\mu)$
- Case 2 (Poisson-beta): N_i follows $Poisson(\mu * y_i)$
- Case 3 (Lognormal-fair): N_i follows $Binomial(M_i, 1/k)$; where M_i follows $Lognormal(\mu, \sigma)$
- Case 4 (Lognormal-beta): N_i follows $Binomial(M_i, y_i)$; where M_i follows $Lognormal(\mu, \sigma)$

Value

graphical comparison between different mixing schemes

References

- Nauta, M.J., 2005. Microbiological risk assessment models for partitioning and mixing during food handling. International Journal of Food Microbiology 100, 311-322.

See Also

sim_multiple, sim_single

Examples

```
n_iter <- 200000
mu <- c(log(100),log(100),log(100))
sigma <- c(0.8,0.8,0.8)
b <- c(0.1,1,10)
k <- c(10,10,10)
distribution <-  c("Lognormal-beta","Lognormal-beta","Lognormal-beta")
compare_mixing (n_iter, mu, sigma, b, k, distribution )
```

`sim_multiple` *The total number of colony-forming units in the mixed sample by simulation result (in the multiple mixing plan).*

Description

This function calculates the resulting total number of colony forming units in the mixed sample in the multiple mixing plans. (to be finished later on)

Usage

```
sim_multiple(n_iter, mu, sigma, b, k, distribution)
```

Arguments

`n_iter`	number of iterations
`mu`	average number of colony-forming units in a primary sample which is in logarithmic scale if we use a lognormal distribution
`sigma`	log standard deviation of the colony-forming units in a primary sample
`b`	concentration parameter
`k`	number of small portions/ primary samples
`distribution`	what suitable distribution type we have employed for simulation such as 'Poisson-fair' or 'Poisson-beta' or 'Lognormal-fair' or 'Lognormal-beta'

Details

Let N' be the number of colony-forming units in the mixed sample which is produced by mixing of k primary sample and $N' = \sum(N_i)$ (to be finished later on)

Value

total number of colony forming units in the multiple mixing scheme

References

- Nauta, M.J., 2005. Microbiological risk assessment models for partitioning and mixing during food handling. International Journal of Food Microbiology 100, 311-322.

See Also

sim_single, compare_mixing

Examples

```
n_iter <- 200000
mu <- c(20,30,50,60)
sigma <- c(0.8,0.8,0.8,0.8)
b <- c(0.1,0.1,0.1,0.1)
k <- c(10,10,10,10)
distribution <-  c("Poisson-fair","Poisson-fair","Poisson-fair","Poisson-fair")
head(sim_multiple( n_iter, mu, sigma, b, k, distribution))
```

sim_single — *The total number of colony-forming units in the mixed sample by simulation result (in the single mixing plan).*

Description

This function calculates the resulting total number of colony forming units in the mixed sample in the single mixing plan. (to be finished later on)

Usage

```
sim_single(n_iter, mu, sigma, b, k, distribution, summary = FALSE)
```

Arguments

n_iter	number of iterations
mu	average number of colony-forming units in a primary sample which is in logarithmic scale if we use a lognormal distribution
sigma	log standard deviation of the colony-forming units in a primary sample
b	concentration parameter
k	number of small portions/ primary samples
distribution	what suitable distribution type we have employed for simulation such as 'Poisson-fair' or 'Poisson-beta' or 'Lognormal-fair' or 'Lognormal-beta'
summary	if we need to get the mean and standard deviation of simulated N', use summary = TRUE (default summary =FALSE).

Details

Let N' be the number of colony-forming units in the mixed sample which is produced by mixing of k primary sample and $N' = \sum(N_i)$ (to be finished later on)

Value

total number of colony forming units in the single mixing plan

References

- Nauta, M.J., 2005. Microbiological risk assessment models for partitioning and mixing during food handling. International Journal of Food Microbiology 100, 311-322.

See Also

sim_multiple, compare_mixing

Examples

```
n_iter <- 200000
mu <- c(20,30,50,60)
sigma <- c(0.8,0.8,0.8,0.8)
b <- c(0.1,0.1,0.1,0.1)
k <- c(10,10,10,10)
distribution <-  c("Poisson-fair","Poisson-beta","Lognormal-fair","Lognormal-beta")
sim_single( n_iter, mu[2], sigma[2], b[2], k[2], distribution[2], summary = TRUE)
```

Index

compare_mixing, *2*, *3*, *5*

sim_multiple, *2*, 3, *5*
sim_single, *2*, *3*, 4