Package ‘AdvBinomApps’

January 20, 2025

Type Package

Title Upper Clopper-Pearson Confidence Limits for Burn-in Studies under Additional Available Information

Version 1.0

Depends GenBinomApps, rootSolve

Date 2016-04-05

Author Daniel Kurz [aut], Horst Lewitschnig [aut, cre]

Maintainer Horst Lewitschnig <horst.lewitschnig@infineon.com>

Description Functions to compute upper Clopper-Pearson confidence limits of early life failure probabilities and required sample sizes of burn-in studies under further available information, e.g. from other products or technologies.

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2016-04-07 07:45:08

Contents

`AdvBinomApps-package` *Upper Clopper-Pearson Confidence Limits For Burn-in Studies under Additional Available Information*

Description

Functions to compute upper Clopper-Pearson confidence limits of early life failure probabilities and required sample sizes of burn-in studies under further available information, e.g. from other products or technologies.

Details

Package:	AdvBinomApps
Type:	Package
Version:	1.0
Date:	2016-04-05
License:	GPL-3

Author(s)

Daniel Kurz, Horst Lewitschnig

Maintainer: Horst Lewitschnig <horst.lewitschnig@infineon.com>

References

C.J. Clopper and E.S. Pearson: *The use of confidence or fiducial limits illustrated in the case of the binomial*. Biometrika, 26(4): 404-413, 1934.

D. Kurz, H. Lewitschnig and J. Pilz: *Decision-Theoretical Model for Failures Tackled by Countermeasures*. IEEE Transactions on Reliability, 63(2): 583-592, 2014. DOI: 10.1109/TR.2014.2315952.

D. Kurz, H. Lewitschnig and J. Pilz: *Failure probability estimation under additional subsystem information with application to semiconductor burn-in*. Resubmitted to: Journal of Applied Statistics, 2015.

D. Kurz, H. Lewitschnig and J. Pilz: *An Advanced Area Scaling Approach for Semiconductor Burn-in*. Microelectronics Reliability, 55(1): 129-137, 2015. DOI: 10.1016/j.microrel.2014.09.007.

D. Kurz, H. Lewitschnig and J. Pilz: *Failure Probability Estimation with Differently Sized Reference Products for Semiconductor Burn-in Studies*. Applied Stochastic Models in Business and Industry, 31(5): 732-744, 2015. DOI: 10.1002/asmb.2100.

Acknowledgment

The work has been performed in the project EPT300, co-funded by grants from Austria, Germany, Italy, The Netherlands and the ENIAC Joint Undertaking. This project is co-funded within the programme "Forschung, Innovation und Technologie fuer Informationstechnologie" by the Austrian Ministry for Transport, Innovation and Technology.

See Also

GenBinomApps

Examples

```
#MULTIPLE REFERENCE PRODUCTS:
k<-c(1,2)
n<-c(110000,138000)
A.ref<-c(5.21,10.71)
A.follow<-8.5
p.target<-20e-06
ci.mult.ref(k,n,A.ref,A.follow,p.target=p.target)

#SYNERGIES
k<-c(0,1)
n<-c(110000,330000)
ci.syn(k,n,0.1,20e-06)

#SEPARATE AREA SCALING
k<-c(1,0)
n<-110000
A.ref<-c(1,1.5)
A.follow<-c(1,2)
p.target<-20e-06
ci.sas(k,n,A.ref,A.follow,0.1,p.target)
```

`ci.mult.ref` *Upper Clopper-Pearson confidence limits for area scaling with differently sized reference products*

Description

Function to compute upper Clopper-Pearson confidence limits of failure probabilities on the basis of differently sized reference products. Optionally, the required numbers of additional inspections for each reference product to reach a predefined target failure probability of the follower product are returned.

Usage

```
ci.mult.ref(k, n, A.ref, A.follow, alpha = 0.1, p.target = 1,
prec = 2, tailcut = 1e-08, tol = 1e-12)
```

Arguments

k	vector of total numbers of failures for each reference product.
n	vector of numbers of inspected devices for each reference product.
A.ref	vector of chip sizes for each reference product (in mm^2).
A.follow	size of follower product.

alpha	alpha-level (1-alpha confidence level, default: 0.1).
p.target	target failure probability of follower product (optional).
prec	precision for greatest common divisor is 10^-prec (default: 2).
tailcut	probabilities for scaled failures smaller than tailcut are set to zero for each reference product (default: 1e-08). Too small values for tailcut might cause increased computation times.
tol	tolerance of uniroot-function used for computing failure probability per greatest common chip size (default: 1e-12).

Value

p.ref	vector of upper Clopper-Pearson confidence limits for each reference product (without the other reference products).
p.mm	upper Clopper-Pearson confidence limit of the failure probability per mm^2 (on the basis of all reference products).
p.follow	upper Clopper-Pearson confidence limit of the failure probability of the follower product (on the basis of all reference products).
n.add	vector of required numbers of additional inspections for each reference product in order to reach p.target for the follower product.

Author(s)

Daniel Kurz, Horst Lewitschnig

Maintainer: Horst Lewitschnig <horst.lewitschnig@infineon.com>

References

D. Kurz, H. Lewitschnig and J. Pilz: *Failure Probability Estimation with Differently Sized Reference Products for Semiconductor Burn-in Studies*. Applied Stochastic Models in Business and Industry, 31(5): 732-744, 2015. DOI: 10.1002/asmb.2100.

See Also

phi.mult.ref ci.mult.ref.cm phi.mult.ref.cm

Examples

```
k<-c(1,2)
n<-c(110000,138000)
A.ref<-c(5.21,10.71)
A.follow<-8.5
p.target<-20e-06
ci.mult.ref(k,n,A.ref,A.follow,p.target=p.target)

k<-c(1,2,2)
n<-c(110000,138000,170000)
A.ref<-c(5.21,10.71,7.89)
A.follow<-8.5
```

```
p.target<-20e-06
ci.mult.ref(k,n,A.ref,A.follow,p.target=p.target)
```

`ci.mult.ref.cm` *Upper Clopper-Pearson confidence limits for area scaling with differently sized reference products and countermeasures*

Description

Function to compute upper Clopper-Pearson confidence limits of failure probabilities on the basis of differently sized reference products and failures tackled by countermeasures. Optionally, the required numbers of additional inspections for each reference product to reach a predefined target failure probability of the follower product are returned.

Usage

```
ci.mult.ref.cm(k, n, A.ref, A.follow, K, theta, alpha = 0.1,
p.target = 1, prec = 2, tailcut = 1e-08, tol = 1e-12)
```

Arguments

`k`	vector of total numbers of failures for each reference product.
`n`	vector of numbers of inspected devices for each reference product.
`A.ref`	vector of chip sizes for each reference product (in mm^2).
`A.follow`	size of follower product.
`K`	matrix with entries K[j,i] denoting the number of failures of the j-th reference product tackled with the i-th countermeasure. If two or more countermeasures have the same efficiency, they can be handled as one countermeasure for several failures. If the i-th countermeasure does not apply to the j-th reference product, then set K[j,i]=0. If there is no countermeasure for a failure at all, then it does not need to be considered in `K` (the failure itself is already considered in `k`).
`theta`	vector of (different) effectivenesses of countermeasures.
`alpha`	alpha-level (1-alpha confidence level, default: 0.1).
`p.target`	target failure probability of follower product (optional).
`prec`	precision for greatest common divisor is 10^-`prec` (default: 2).
`tailcut`	probabilities for scaled failures smaller than `tailcut` are set to zero for each reference product (default: 1e-08). Too small values for `tailcut` might cause increased computation times.
`tol`	tolerance of `uniroot`-function used for computing failure probability per greatest common chip size with countermeasures (default: 1e-12).

Value

p.ref.cm
: vector of upper Clopper-Pearson confidence limits for each reference product with countermeasures (without the other reference products).

p.mm.cm
: upper Clopper-Pearson confidence limit of the failure probability per mm^2 with countermeasures (on the basis of all reference products).

p.follow.cm
: upper Clopper-Pearson confidence limit of the failure probability of the follower product with countermeasures (on the basis of all reference products).

n.add.cm
: vector of required numbers of additional inspections for each reference product in order to reach p.target for the follower product with countermeasures.

Author(s)

Daniel Kurz, Horst Lewitschnig

Maintainer: Horst Lewitschnig <horst.lewitschnig@infineon.com>

References

D. Kurz, H. Lewitschnig and J. Pilz: *Failure Probability Estimation with Differently Sized Reference Products for Semiconductor Burn-in Studies*. Applied Stochastic Models in Business and Industry, 31(5): 732-744, 2015. DOI: 10.1002/asmb.2100.

D. Kurz, H. Lewitschnig and J. Pilz: *Decision-Theoretical Model for Failures Tackled by Countermeasures*. IEEE Transactions on Reliability, 63(2): 583-592, 2014. DOI: 10.1109/TR.2014.2315952.

See Also

phi.mult.ref ci.mult.ref phi.mult.ref.cm

Examples

```
#Reference product 1: 1 failure - failure tackled with 80% efficiency.
#Reference product 2: 2 failures - 1 failure tackled with 80%,
#1 failure with 60% efficiency.
k<-c(1,2)
K<-matrix(c(1,0,1,1),2,2,byrow=TRUE)
n<-c(110000,138000)
theta<-c(0.8,0.6)
A.ref<-c(5.21,10.71)
A.follow<-8.5
p.target<-20e-06
ci.mult.ref.cm(k,n,A.ref,A.follow,K,theta,p.target=p.target)

#Reference product 1: 1 failure - failure tackled with 20% efficiency.
#Reference product 2: 2 failures - 1 failure tackled with 20%,
#1 failure with 40% efficiency.
#Reference product 3: 2 failures - both tackled with 60% efficiency.
k<-c(1,2,2)
n<-c(110000,138000,170000)
K<-matrix(c(1,0,0,1,1,0,0,0,2),3,3,byrow=TRUE)
theta<-c(0.2,0.4,0.6)
```

```
A.ref<-c(5.21,10.71,7.89)
A.follow<-8.5
p.target<-20e-06
ci.mult.ref.cm(k,n,A.ref,A.follow,K,theta,p.target=p.target)

#Reference product 1: 1 failure - failure tackled with 20% efficiency.
#Reference product 2: 2 failures - 1 failure tackled with 40% efficiency,
#1 failure without countermeasure.
#Reference product 3: 3 failures - 1 failure tackled with 60% efficiency,
#2 failures without countermeasures.
k<-c(1,2,3)
n<-c(110000,138000,170000)
K<-matrix(c(1,0,0,0,1,0,0,0,1),3,3,byrow=TRUE)
theta<-c(0.2,0.4,0.6)
A.ref<-c(5.21,10.71,7.89)
A.follow<-8.5
p.target<-20e-06
ci.mult.ref.cm(k,n,A.ref,A.follow,K,theta,p.target=p.target)
```

`ci.sas` *Separate area scaling for upper Clopper-Pearson confidence limits*

Description

Function to compute upper Clopper-Pearson confidence limits of failure probabilities of follower products by means of separate area scaling (SAS). Furthermore, the validity of the SAS in comparison to the classical area scaling (CAS) is evaluated. Optionally, the required numbers of additional inspections of the reference product in order to reach the predefined target failure probability of the follower product according to the CAS and SAS are returned.

Usage

```
ci.sas(k, n, A.ref, A.follow, alpha = 0.1, p.target = 1, atol = 1e-08)
```

Arguments

`k`	vector of numbers of failures for each subset on reference product (total number of failures on reference product = sum of entries of `k`).
`n`	number of inspected devices in burn-in study of reference product.
`A.ref`	vector of sizes for each subset on reference product (in mm^2).
`A.follow`	vector of sizes for each subset on follower product (in mm^2).
`alpha`	alpha-level (1-alpha confidence level, default: 0.1).
`p.target`	target failure probability of follower product (optional).
`atol`	tolerance of `multiroot`-function used for computing `p.sas` (default: 1e-08). For further details, see description of `multiroot`-function.

Details

Function makes use of `multiroot`-function of the package `rootSolve` to solve non-linear equation system for the subset failure probabilities.

Appropriate starting values for `multiroot` are chosen automatically.

In case of non-convergence of `multiroot`-function, `NA` is returned.

Function designed and verified for number of subsets < 6.

Value

`p.cas`	vector of upper Clopper-Pearson confidence limits of failure probabilities for each subset on reference product according to CAS.
`p.sas`	vector of upper Clopper-Pearson confidence limits of failure probabilities for each subset on reference product according to SAS.
`p.follow.cas`	upper Clopper-Pearson confidence limit of the failure probability of the follower product according to CAS.
`p.follow.sas`	upper Clopper-Pearson confidence limit of the failure probability of the follower product according to SAS.
`delta`	evidence factor against CAS.
`n.add.cas`	required number of additional inspections of the reference product in order to reach `p.target` of the follower product according to CAS.
`n.add.sas`	required number of additional inspections of the reference product in order to reach `p.target` of the follower product according to SAS.

Author(s)

Daniel Kurz, Horst Lewitschnig

Maintainer: Horst Lewitschnig <horst.lewitschnig@infineon.com>

References

D. Kurz, H. Lewitschnig and J. Pilz: *An Advanced Area Scaling Approach for Semiconductor Burn-in*. Microelectronics Reliability, 55(1): 129-137, 2015. DOI: 10.1016/j.microrel.2014.09.007.

See Also

`ci.sas.cm`

Examples

```
k<-c(1,0)
n<-100000
A.ref<-c(1,1.5)
A.follow<-c(1,2)
p.target<-20e-06
ci.sas(k,n,A.ref,A.follow,0.1,p.target)

k<-c(2,1,0)
```

```
n<-100000
A.ref<-c(2,3,4)
A.follow<-c(1,2,3)
p.target<-20e-06
ci.sas(k,n,A.ref,A.follow,0.1,p.target)

k<-c(1,1,0,2)
n<-100000
A.ref<-c(1,1,1,1)
A.follow<-c(3,4,5,6)
p.target<-20e-06
ci.sas(k,n,A.ref,A.follow,0.1,p.target)
```

---

ci.sas.cm *Separate area scaling for upper Clopper-Pearson confidence limits with countermeasures*

Description

Function to compute upper Clopper-Pearson confidence limits of failure probabilities of follower products by means of separate area scaling (SAS) with failures tackled by countermeasures. Furthermore, the validity of the SAS in comparison to the classical area scaling (CAS) is evaluated taking into account the implemented countermeasures. Optionally, the required numbers of additional inspections of the reference product in order to reach the predefined target failure probability of the follower product according to the CAS and SAS with countermeasures are returned.

Usage

```
ci.sas.cm(k, n, A.ref, A.follow, K, theta, alpha = 0.1,
p.target = 1, atol = 1e-08)
```

Arguments

k	vector of total numbers of failures for each subset on reference product (total number of failures on reference product = sum of entries of k).
n	number of inspected devices in burn-in study of reference product.
A.ref	vector of sizes for each subset on reference product (in mm^2).
A.follow	vector of sizes for each subset on follower product (in mm^2).
K	matrix with entries K[j,i] denoting the number of failures in the j-th subset tackled with the i-th countermeasure. If two or more countermeasures have the same efficiency, they can be handled as one countermeasure for several failures. If the i-th countermeasure does not apply to the j-th subset, then set K[j,i]=0. If there is no countermeasure for a failure at all, then it does not need to be considered in K (the failure itself is already considered in k).
theta	vector of (different) effectivenesses of countermeasures.
alpha	alpha-level (1-alpha confidence level, default: 0.1).

`p.target`	target failure probability of follower product (optional).
`atol`	tolerance of `multiroot`-function used for computing `p.sas` (default: 1e-08). For further details, see description of `multiroot`-function.

Details

Function makes use of `multiroot`-function of the package `rootSolve` to solve non-linear equation system for the subset failure probabilities.

Appropriate starting values for `multiroot` are chosen automatically.

In case of non-convergence of `multiroot`-function, `NA` is returned.

Function designed and verified for number of subsets < 6.

Value

`p.cas.cm`	vector of upper Clopper-Pearson confidence limits of failure probabilities for each subset on reference product according to CAS with countermeasures.
`p.sas.cm`	vector of upper Clopper-Pearson confidence limits of failure probabilities for each subset on reference product according to SAS with countermeasures.
`p.follow.cas.cm`	upper Clopper-Pearson confidence limit of the failure probability of the follower product according to CAS with countermeasures.
`p.follow.sas.cm`	upper Clopper-Pearson confidence limit of the failure probability of the follower product according to SAS with countermeasures.
`delta.cm`	evidence factor against CAS with countermeasures.
`n.add.cas.cm`	required number of additional inspections of the reference product in order to reach `p.target` of the follower product according to CAS with countermeasures.
`n.add.sas.cm`	required number of additional inspections of the reference product in order to reach `p.target` of the follower product according to SAS with countermeasures.

Author(s)

Daniel Kurz, Horst Lewitschnig

Maintainer: Horst Lewitschnig <horst.lewitschnig@infineon.com>

References

D. Kurz, H. Lewitschnig and J. Pilz: *An Advanced Area Scaling Approach for Semiconductor Burn-in*. Microelectronics Reliability, 55(1): 129-137, 2015. DOI: 10.1016/j.microrel.2014.09.007.

D. Kurz, H. Lewitschnig and J. Pilz: *Decision-Theoretical Model for Failures Tackled by Countermeasures*. IEEE Transactions on Reliability, 63(2): 583-592, 2014. DOI: 10.1109/TR.2014.2315952.

See Also

`ci.sas`

Examples

```
#Reference product: 1 failure.
#Subset 1: 1 failure - failure tackled with 80% efficiency.
#Subset 2: no failures.
k<-c(1,0)
K<-matrix(c(1,0),2,1)
n<-100000
A.ref<-c(1,1)
theta<-0.8
A.follow<-c(1,2)
p.target<-20e-06
ci.sas.cm(k,n,A.ref,A.follow,K,theta,0.1,p.target)

#Reference product: 3 failures.
#Subset 1: 2 failures - 1 failure tackled with 80%,
#1 failure with 70% efficiency.
#Subset 2: 1 failure - failure tackled with 70% efficiency.
#Subset 3: no failures.
k<-c(2,1,0)
K<-matrix(c(1,1,0,1,0,0),3,2,byrow=TRUE)
n<-100000
A.ref<-c(2,3,4)
theta<-c(0.8,0.7)
A.follow<-c(1,2,3)
p.target<-20e-06
ci.sas.cm(k,n,A.ref,A.follow,K,theta,0.1,p.target)

#Reference product: 4 failures.
#Subset 1: 2 failures - 1 failure tackled with 80% efficiency,
#1 failure without countermeasure.
#Subset 2: 1 failure - failure tackled with 70% efficiency.
#Subset 3: 1 failure - failure without countermeasure.
k<-c(2,1,1)
K<-matrix(c(1,0,0,1,0,0),3,2,byrow=TRUE)
n<-100000
A.ref<-c(2,3,4)
theta<-c(0.8,0.7)
A.follow<-c(1,2,3)
p.target<-20e-06
ci.sas.cm(k,n,A.ref,A.follow,K,theta,0.1,p.target)
```

---

ci.syn *Upper Clopper-Pearson confidence limits under chip synergies*

Description

Function to compute upper Clopper-Pearson confidence limits of failure probabilities on the basis of burn-in studies for each subset of a chip. Optionally, the required number of additional inspections for reaching a predefined target failure probability is returned.

Usage

```
ci.syn(k, n, alpha = 0.1, p.target = 1, tol = 1e-10)
```

Arguments

`k`	vector of numbers of failures for each subset.
`n`	vector of numbers of inspections for each subset.
`alpha`	alpha-level (1-alpha confidence level, default: 0.1).
`p.target`	target failure probability (optional).
`tol`	tolerance of `uniroot`-function used for computing `p.hat` (default: 1e-10).

Value

`p.hat`	upper Clopper-Pearson confidence limit of the failure probability of the assembled devices.
`n.add`	required number of additional inspections of each subset for reaching `p.target`.

Author(s)

Daniel Kurz, Horst Lewitschnig

Maintainer: Horst Lewitschnig <horst.lewitschnig@infineon.com>

References

D. Kurz, H. Lewitschnig and J. Pilz: *Failure probability estimation under additional subsystem information with application to semiconductor burn-in*. Resubmitted to: Journal of Applied Statistics, 2015.

See Also

`phi.syn` `phi.syn.cm` `ci.syn.cm`

Examples

```
k<-c(0,1)
n<-c(110000,330000)
ci.syn(k,n,0.1,20e-06)

k<-c(1,0,1,5)
n<-c(330000,240000,240000,400000)
ci.syn(k,n,0.1,20e-06)
```

ci.syn.cm *Upper Clopper-Pearson confidence limits under chip synergies and countermeasures*

Description

Function to compute upper Clopper-Pearson confidence limits of failure probabilities on the basis of burn-in studies with countermeasures for each subset of a chip. Optionally, the required number of additional inspections for reaching a predefined target failure probability with countermeasures is returned.

Usage

```
ci.syn.cm(k, n, K, theta, alpha = 0.1, p.target = 1, tol = 1e-10)
```

Arguments

`k`	vector of numbers of failures for each subset.
`n`	vector of numbers of inspections for each subset.
`K`	matrix with entries K[j,i] denoting the number of failures in the j-th subset tackled with the i-th countermeasure. If two or more countermeasures have the same efficiency, they can be handled as one countermeasure for several failures. If the i-th countermeasure does not apply to the j-th subset, then set K[j,i]=0. If there is no countermeasure for a failure at all, then it does not need to be considered in `K` (the failure itself is already considered in `k`).
`theta`	vector of (different) effectivenesses of countermeasures.
`alpha`	alpha-level (1-alpha confidence level, default: 0.1).
`p.target`	target failure probability (optional).
`tol`	tolerance of `uniroot`-function used for computing `p.hat.cm` (default: 1e-10).

Value

`p.hat.cm`	upper Clopper-Pearson confidence limit of the failure probability of the assembled devices with countermeasures.
`n.add.cm`	required number of additional inspections of each subset for reaching `p.target` with countermeasures.

Author(s)

Daniel Kurz, Horst Lewitschnig

Maintainer: Horst Lewitschnig <horst.lewitschnig@infineon.com>

References

D. Kurz, H. Lewitschnig and J. Pilz: *Failure probability estimation under additional subsystem information with application to semiconductor burn-in*. Resubmitted to: Journal of Applied Statistics, 2015.

D. Kurz, H. Lewitschnig and J. Pilz: *Decision-Theoretical Model for Failures Tackled by Countermeasures*. IEEE Transactions on Reliability, 63(2): 583-592, 2014. DOI: 10.1109/TR.2014.2315952.

See Also

phi.syn ci.syn phi.syn.cm

Examples

```
#Subset 1: no failures.
#Subset 2: 1 failure - failure tackled with 80% efficiency.
k<-c(0,1)
K<-matrix(c(0,1),2,1,byrow=TRUE)
theta<-0.8
n<-c(110000,330000)
ci.syn.cm(k,n,K,theta,0.1,20e-06)

#Subset 1: 1 failure - failure tackled with 80% efficiency.
#Subset 2: 1 failure - failure tackled with 70% efficiency.
#Subset 3: 2 failures - 1 failure tackled with 80%,
#1 failure with 70% efficiency.
k<-c(1,1,2)
K<-matrix(c(1,0,0,1,1,1),3,2,byrow=TRUE)
theta<-c(0.8,0.7)
n<-c(110000,150000,220000)
ci.syn.cm(k,n,K,theta,0.1,20e-06)

#Subset 1: 1 failure - failure tackled with 80% efficiency.
#Subset 2: 1 failure - failure without countermeasure.
#Subset 3: 2 failures - 1 failure tackled with 70% efficiency,
#1 failure without countermeasure.
k<-c(1,1,2)
K<-matrix(c(1,0,0,0,0,1),3,2,byrow=TRUE)
theta<-c(0.8,0.7)
n<-c(110000,150000,220000)
ci.syn.cm(k,n,K,theta,0.1,20e-06)
```

---

gcd.mult.ref *Greatest common divisor of chip sizes*

Description

Function to compute the greatest common divisor of the chip sizes of the reference products at a fixed precision.

Usage

```
gcd.mult.ref(A, prec = 2)
```

Arguments

A	vector of chip sizes for each reference product (in mm^2).
prec	precision for greatest common divisor is 10^-prec (default: 2).

Value

A.gcd	greatest common divisor of the sizes in mm^2 (at specified precision).

Author(s)

Daniel Kurz, Horst Lewitschnig

Maintainer: Horst Lewitschnig <horst.lewitschnig@infineon.com>

Examples

```
A<-c(48, 30, 42)
gcd.mult.ref(A)

A<-c(2.2, 3.01, 5)
gcd.mult.ref(A, prec = 2)
gcd.mult.ref(A, prec = 1)
gcd.mult.ref(A, prec = 0)
```

---

phi.mult.ref *Downscaling of failures to greatest common chip size*

Description

Function to scale the numbers of failures in burn-in studies of differently sized reference products down to the greatest common chip size of the products and to merge the downscaled information.

Usage

```
phi.mult.ref(k, n, A.ref, prec = 2, tailcut = 1e-08)
```

Arguments

k	vector of total numbers of failures for each reference product.
n	vector of numbers of inspected devices for each reference product.
A.ref	vector of chip sizes for each reference product (in mm^2).
prec	precision for greatest common divisor is 10^-prec (default: 2).
tailcut	probabilities for scaled failures smaller than tailcut are set to zero for each reference product (default: 1e-08). Too small values for tailcut might cause increased computation times.

Value

`phi`	data frame with possible numbers of failures `k.gcd` and probabilities `phi(k.gcd)`. Only failure numbers `k.gcd` with `phi(k.gcd)>0` are printed out.
`A.gcd`	greatest common divisor of the sizes of the reference products.

Author(s)

Daniel Kurz, Horst Lewitschnig

Maintainer: Horst Lewitschnig <horst.lewitschnig@infineon.com>

References

D. Kurz, H. Lewitschnig and J. Pilz: *Failure Probability Estimation with Differently Sized Reference Products for Semiconductor Burn-in Studies*. Applied Stochastic Models in Business and Industry, 31(5): 732-744, 2015. DOI: 10.1002/asmb.2100.

See Also

`ci.mult.ref` `phi.mult.ref.cm` `ci.mult.ref.cm`

Examples

```
k<-c(1,2)
n<-c(10,15)
A.ref<-c(2,3)
phi.mult.ref(k,n,A.ref)

k<-c(1,1)
n<-c(110000,220000)
A.ref<-c(5.21,10.71)
phi.mult.ref(k,n,A.ref)

k<-c(1,2,3,4)
n<-c(10,15,20,30)
A.ref<-c(1,2,3,4)
phi.mult.ref(k,n,A.ref)
```

`phi.mult.ref.cm` *Downscaling of failures tackled by countermeasures to greatest common chip size*

Description

Function to scale failures tackled by countermeasures in burn-in studies of differently sized reference products down to the greatest common chip size of the products and to merge the downscaled information.

Usage

```
phi.mult.ref.cm(k, n, A.ref, K, theta, prec = 2, tailcut = 1e-08)
```

Arguments

`k`	vector of total numbers of failures for each reference product.
`n`	vector of numbers of inspected devices for each reference product.
`A.ref`	vector of chip sizes for each reference product (in mm^2).
`K`	matrix with entries K[j,i] denoting the number of failures of the j-th reference product tackled with the i-th countermeasure. If two or more countermeasures have the same efficiency, they can be handled as one countermeasure for several failures. If the i-th countermeasure does not apply to the j-th reference product, then set K[j,i]=0. If there is no countermeasure for a failure at all, then it does not need to be considered in `K` (the failure itself is already considered in `k`).
`theta`	vector of (different) effectivenesses of countermeasures.
`prec`	precision for greatest common divisor is 10^-`prec` (default: 2).
`tailcut`	probabilities for scaled failures smaller than `tailcut` are set to zero for each reference product (default: 1e-08). Too small values for `tailcut` might cause increased computation times.

Value

`phi.cm`	data frame with possible number of failures `k.gcd` (after the implementation of countermeasures) and probabilities `phi.cm(k.gcd)`. Only failure numbers `k.gcd` with `phi.cm(k.gcd)`>0 are printed out.
`A.gcd`	greatest common divisor of the sizes of the reference products.

Author(s)

Daniel Kurz, Horst Lewitschnig

Maintainer: Horst Lewitschnig `<horst.lewitschnig@infineon.com>`

References

D. Kurz, H. Lewitschnig and J. Pilz: *Failure Probability Estimation with Differently Sized Reference Products for Semiconductor Burn-in Studies*. Applied Stochastic Models in Business and Industry, 31(5): 732-744, 2015. DOI: 10.1002/asmb.2100.

D. Kurz, H. Lewitschnig and J. Pilz: *Decision-Theoretical Model for Failures Tackled by Countermeasures*. IEEE Transactions on Reliability, 63(2): 583-592, 2014. DOI: 10.1109/TR.2014.2315952.

See Also

`phi.mult.ref` `ci.mult.ref` `ci.mult.ref.cm`

Examples

```
k<-c(1,2)
n<-c(10,10)
K<-matrix(c(1,0,1,1),2,2,byrow=TRUE)
theta<-c(0.7,0.8)
A.ref<-c(1,2)
phi.mult.ref.cm(k,n,A.ref,K,theta)

k<-c(1,2)
n<-c(110000,220000)
K<-matrix(c(1,0,0,1),2,2,byrow=TRUE) #no CM for one fail!
theta<-c(0.7,0.8)
A.ref<-c(2,3)
phi.mult.ref.cm(k,n,A.ref,K,theta)
```

---

phi.syn *Assembling of devices*

Description

Function to compute probability of having a certain number of failures out of `min(n)` devices, which are randomly assembled out of a certain number of chip subsets.

Usage

```
phi.syn(k, n)
```

Arguments

k	vector of numbers of failures for each subset.
n	vector of numbers of inspections for each subset.

Value

phi	data frame with possible numbers of failures (out of `min(n)` assembled devices) and associated probabilities.

Author(s)

Daniel Kurz, Horst Lewitschnig

Maintainer: Horst Lewitschnig <horst.lewitschnig@infineon.com>

References

D. Kurz, H. Lewitschnig and J. Pilz: *Failure probability estimation under additional subsystem information with application to semiconductor burn-in*. Resubmitted to: Journal of Applied Statistics, 2015.

See Also

ci.syn phi.syn.cm ci.syn.cm

Examples

```
k<-c(1,2)
n<-c(10,15)
phi.syn(k,n)

k<-c(0,1,1,1)
n<-c(110000,220000,150000,330000)
phi.syn(k,n)
```

---

phi.syn.cm *Assembling of devices with countermeasures*

Description

Function to compute the probability of having a certain number of failures out of `min(n)` randomly assembled devices with countermeasures.

Usage

```
phi.syn.cm(k, n, K, theta)
```

Arguments

k	vector of total numbers of failures for each subset.
n	vector of numbers of inspections for each subset.
K	matrix with entries K[j,i] denoting the number of failures of the j-th subset tackled with the i-th countermeasure. If two or more countermeasures have the same efficiency, they can be handled as one countermeasure for several failures. If the i-th countermeasure does not apply to the j-th subset, then set K[j,i]=0. If there is no countermeasure for a failure at all, then it does not need to be considered in `K` (the failure itself is already considered in `k`).
theta	vector of (different) effectivenesses of countermeasures.

Value

phi.cm	data frame with possible numbers of failures (out of `min(n)` assembled devices) and associated probabilities considering the implemented countermeasures.

Author(s)

Daniel Kurz, Horst Lewitschnig

Maintainer: Horst Lewitschnig <horst.lewitschnig@infineon.com>

References

D. Kurz, H. Lewitschnig and J. Pilz: *Failure probability estimation under additional subsystem information with application to semiconductor burn-in*. Resubmitted to: Journal of Applied Statistics, 2015.

D. Kurz, H. Lewitschnig and J. Pilz: *Decision-Theoretical Model for Failures Tackled by Countermeasures*. IEEE Transactions on Reliability, 63(2): 583-592, 2014. DOI: 10.1109/TR.2014.2315952.

See Also

phi.syn ci.syn ci.syn.cm

Examples

```
k<-c(0,1)
K<-matrix(c(0,1),2,1,byrow=TRUE)
theta<-0.8
n<-c(110000,330000)
phi.syn.cm(k,n,K,theta)

k<-c(1,1,2)
K<-matrix(c(0,0,0,1,1,1),3,2,byrow=TRUE)
theta<-c(0.7,0.5)
n<-c(10,15,20)
phi.syn.cm(k,n,K,theta)
```

Index