Package ‘proteomicdesign’

February 20, 2015

Type Package

Title Optimization of a multi-stage proteomic study

Version 2.0

Date 2012-12-30

Author Irene SL Zeng

Maintainer Irene Zeng <zeng@stat.auckland.ac.nz>

Description This package provides functions to identify the optimal solution that maximizes numbers of detectable differentiated proteins from a multi-stage clinical proteomic study.

License GPL-3

Depends MASS

Repository CRAN

Date/Publication 2013-01-04 12:04:40

NeedsCompilation no

R topics documented:

`proteomicdesign-package` *Optimization of a multi stage proteomic study*

Description

This package provides functions to identify the optimal solution that maximizes numbers of detectable disease-differentiated proteins from a multi-stage proteomic study. This package has three main functions to derive the optimize design parameters under different scenarios in a multi-stage study setting. It assumes that the first stage study has already conducted and the protein parameters and stage I sample size are available. The current functions are designed for the matched case-control or paired samples design throughout the multi-stages.

Details

Package:	proteomicdesign
Type:	Package
Version:	2.0
Date:	2012-12-30
License:	What license is it under? GPL-3
Depends:	R,library(MASS)

Optim.two.stage.group() is used when group information for each protein is available through the biological pathway or other analyses. It uses Monte Carlo simulation to estimate the expected number of proteins with true effects. Optim.two.stage.appr() is used when group information for each protein is available as described as above. It uses an approximated analytical function to estimate the expected number of proteins with true effects. Optim.two.stage.single() is used when no group information is available. It uses Monte Carlo simulation to estimate the expected number of true effects from the multi-stage design. The package has two auxiliary functions that can be use to estimate the cost at the last two stages from the multi-stage study. power.group.cost() is used to estimate the stage II and III cost when the design solutions are derived from optim.two.stage.group() and optim.two.stage.appr(). power.single.cost() is used to estimate the stage II and III cost when the design solutions are derived from optim.two.stage.single().

Author(s)

Irene SL Zeng under the supervision of Professor Thomas Lumley Maintainer: Irene Zeng<i.zeng@auckland.ac.nz>

References

Babak Abbasi, S. T. A. N., Mehrzad Abdi Khalife, Yasser Faize (2011). "A hyprid Variable neighborhood search and simulated annealing algorithm to estimate the three parameters of the Weibull distribution." Expert Systems with Application 38: 700-708. Belisle, C. J. P. (1992). "Convergence theorems for a class of simulated annealing algorithm on Rd " Journal of Applied Probability 29: 885-895. Jorge Nocedal, S. J. W. (1999). Numerical Optimization. New York, Springer.

`calculate.cost` *Function to calculate the total cost of the multi-stage design*

Description

A function to calculate the total cost of the multi-stage design. It is used in the optim.two.stage.appr() as a sub-function.

Usage

```
calculate.cost(n2, n3, p1, p2)
```

Arguments

n2	the stage II sample size
n3	the stage III sample size
p1	the number of proteins selected from stage I
p2	the number of proteins selected from stage II

`calculate.n3` *A function to calculate the stage III sample size based on the current design parameters*

Description

A function to calculate the stage III sample size based on the current design parameters. It is used to reduce the dimensions of the design parameters in the optimization. The sample size of stage III is calculated by used the known budget, a slack term for the inequality constraint, and the current design parameters from the search.

Usage

```
calculate.n3(n2, p2, p3)
```

Arguments

n2	stage II sample size
p2	number of proteins selected from stage I
p3	number of proteins selected from stage II

`do.one.experiment` *Simulation function for calculate the expected number of detectable proteins from the three stage proteomic study using group information*

Description

The Monte Carlo simulated objective function for the optim.two.stage.group().

Usage

```
do.one.experiment(initial, optimize = TRUE)
```

Arguments

`initial`	The design parameters selected at the current search
`optimize`	The logical variable for the output: when it is true, the output is the expected number of proteins with the true effects.

`do.one.experiment.t` *Simulation function for calculating the expected number of detectable proteins from the three stage proteomic study*

Description

Simulation function for calculating the expected number of detectable proteins from the three stage proteomic study. It is used in the optim.two.stage.single().

Usage

```
do.one.experiment.t(initial)
```

Arguments

`initial`	The design parameters selected at the current search

`Ftest.Ttest` *A function to perform the paired t test for each protein, and the Hotelling T test for the group that the protein is assigned.*

Description

The function calculate the t test for each protein and the Hoteling T test for each group.

Usage

```
Ftest.Ttest(sample, n, m)
```

Arguments

`sample`	The protein sample simulated from the protein matrix
`n`	The number of biological samples
`m`	The number of proteins

`genseq.appr` *A sub function to generate the sub solution space*

Description

A function to generate the sub solution space in the optim.two.stage.app() function

Usage

```
genseq.appr(initial)
```

Arguments

`initial`	The boundary for the sub solution space

Details

This function inputs the user defined boundary for the design parameter, the radius for the current sub-solution space. It outputs the selected solution.

`genseq.group` *A sub function to generate the sub solution space*

Description

A function to generate the sub solution space in the optim.two.stage.group() function

Usage

```
genseq.group(initial)
```

Arguments

`initial`	The boundary for the sub solution space

Details

This function inputs the user defined boundary for the design parameter, the radius for the current sub-solution space. It outputs the selected solution.

`genseq.single` *A sub function to generate the sub solution space*

Description

A function to generate the sub solution space in the optim.two.stage.group() function

Usage

```
genseq.single(initial)
```

Arguments

`initial`	The boundary for the sub solution space

Details

This function inputs the user defined boundary for the design parameter, the radius for the current sub-solution space. It outputs the selected solution.

`optim.two.stage.appr` *Optimize numbers of discoveries by using an approximated analytical objective function in a multi-stage clinical proteomic study that utilizes biological group information*

Description

This function provides a design solution of a three-stage proteomic study. The solution comprises of the significance levels at stage I/II and sample size at stage II/III, which maximizes the numbers of proteins with true effects in the three-stage proteomic study. The differences of this function from the optim.two.stage.group()is that, it uses an approximated analytical objective function instead of the Monte-Carlo simulated objective function to calculate the expected numbers of proteins with true effects. It uses the simulation annealing method that bases on the Metropolis function to determine if a solution shall be accepted at each step in the optimization. The input parameters are the mean difference and its standard deviation in the relative intensity/intensity between the matched case-control for each protein identified from the stage I discovery, the stage I sample size, the cost functions and a vector to define the technical artifact correction. A nested simulation annealing method is used to construct multiple sub-searching spaces from the global solution space.

Usage

```
optim.two.stage.appr(budget, protein, n1, artifact, iter.number, assaycost2.function, assaycost3.func
```

Arguments

`budget` The budget of the three-stage proteomic study for the stage II verification and stage III clinical validation: It dose not include stage I cost.

`protein` The protein dataset: it needs to have four variables: proteinid,beta,sigma,and group. proteinid is the numerical id for each protein. beta is the mean difference. sigma is the standard deviation of the difference. group is the group that the protein is being assigned. The protein dataset needs to have at least 2 proteins.

`n1` The stage I sample size

`artifact` The technical artifact correction factor for each protein

`iter.number` The number of iterations for the nested-simulation annealing

`assaycost2.function`
The assay cost function of number of proteins (p) and number of patients(n) at stage II

`assaycost3.function`
The assay cost function of number of proteins (p) ONLY at stage III

`recruit` The recruitment cost for a patient. It is assumed to be the same at each stage. The default value is 100.

`a1.t.min` The minimal value of stage I t test p value for the solution: The default value is 0.01.

`a1.t.max` The maximal value of the stage I t test p value for the solution: The default value is 0.25.

`a1.f.min` The minimal value of the stage I F test p value for the solution: The default value is 0.01.

`a1.f.max` The maximal value of the stage I F test p value for the solution: The default value is 0.25.

`a1.step` The interval of the p values for t and F test of stage I: The default value is 0.025.

`a2.t.min` The minimal value of the stage II t test p value for the solution: The default value is 0.01.

`a2.t.max` The maximal value of the stage II t test p value for the solution: The default value is 0.05.

`a2.f.min` The minimal value of the stage II F test p value for the solution: The default value is 0.05.

`a2.f.max` The maximal value of the stage II F test p value for the solution: The default value is 0.05.

`a2.step` The interval of the p value for stage II: The default value is 0.025.

`n2.min` The minimal value of sample size at stage II: The default value is 100.

`n2.max` The maximal value of sample size at stage II: The default value is 1000.

`n2.step` The interval of sample size at stage II: The default value is 100.

`n3.min` The minimal value of sample size at stage III: The default value is 100.

`n3.max` The maximal value of sample size at stage III: The default value is 1000.

`n3.step` The interval of sample size at stage III: The default value is 100.

Value

solution.matrix

`comp1` The number of true positives detected from the third stage

`comp2` The stage I T test p value decision threshold

`comp3` The stage I F test p value decision threshold

`comp4` The stage II T test p value decision threshold

`comp5` The stage II F test p value decision threshold

`comp6` The sample size at stage II

`comp6` The sample size at stage III

`comp7` The radius of the search subspace at the final iteration

`comp8` from the number 8 component to last component are all solution space boundary parameters

Author(s)

Irene Suilan Zeng, Thomas Lumley

References

Babak Abbasi, S. T. A. N., Mehrzad Abdi Khalife, Yasser Faize (2011). "A hyprid Variable neighborhood search and simulated annealing algorithm to estimate the three parameters of the Weibull distribution." Expert Systems with Application 38: 700-708. Belisle, C. J. P. (1992). "Convergence theorems for a class of simulated annealing algorithm on Rd " Journal of Applied Probability 29: 885-895. Jorge Nocedal, S. J. W. (1999). Numerical Optimization. New York, Springer.

See Also

optim.two.stage.single(), optim.two.stage.group(), power.group.cost()

Examples

```
## An example of 5 proteins from an immunology proteomic study
assaycost2=function(n,p){280*p+1015*n}
assaycost3=function(p){200*p}
protein<-data.frame(proteinid=c(100,101,103,104,105),beta=c(2.4,2.6,0.5,2.6,0.7),sigma=c(0.6,0.7,0.3,0.7,0.4),
optim.two.stage.appr(budget=500000,protein=protein,n1=30,artifact=rep(1,5),iter.number=1,assaycost2.function=a
```

`optim.two.stage.group` *Optimization of the design parameters in the discovery , verification and validation stage from a multi-stage clinical proteomic study using biological grouping information*

Description

This function provides the solution for optimizing the number of discoveries in a multistage proteomic study. The solution comprises of the significance levels of the paired t test at stage I/II, the group Hotelling T test at stage I/II and sample size at stage II/III that maximizes numbers of proteins with true effects from the three-stage proteomic study. It uses simulation annealing method that bases on the Metropolis function to determine if a solution shall be accepted at each step. The input parameters are the mean difference and its standard deviation for each protein between the matched case-control from the stage I discovery, the group that each protein is assigned, the stage I sample size and a vector to define the technical artifact correction. A nested simulation annealing method is used to construct multiple sub-searching spaces from the global solution space.

Usage

```
optim.two.stage.group(budget, protein, n1, artifact, iter.number, assaycost2.function, assaycost3.fu
```

Arguments

`budget` The budget of the three-stage proteomic study for the stage II verification and stage III clinical validation. It dose not include stage I cost

`protein` The protein dataset which need to have four variables: proteinid,beta,sigma,group. proteinid is the numerical id for each protein, beta is the mean difference, sigma is the standard deviation of the difference, group is the group that the protein is being assigned. The protein dataset need to have at least 2 proteins.

`n1` The stage I sample size

`artifact` The technical artifact correction factor for each protein

`iter.number` The number of iterations for the nested-simulation annealing

`assaycost2.function`
The assay cost function of number of proteins (p) and number of patients(n) at stage II

`assaycost3.function`
The assay cost function of number of proteins (p) at stage III

`recruit` The recruitment cost for a patient. It is assumed to be the same at each stage. The default value is 100.

`s` The slake term of the total budget which is a small amount of dollars that transforms the inequality constraint to an equal constraint. The default value is 1000 dollars.

`a1.t.min` The minimal value of stage I t test p value for the solution: the default value is 0.01.

`a1.t.max` The maximal value of the stage I t test p value for the solution: the default value is 0.25.

`a1.f.min` The minimal value of the stage I F test p value for the solution: the default value is 0.01.

`a1.f.max` The maximal value of the stage I F test p value for the solution: the default value is 0.25.

`a1.step` The interval of the p values for t and F test of stage I: the default value is 0.025.

`a2.t.min` The minimal value of the stage II t test p value for the solution: the default value is 0.01.

`a2.t.max` The maximal value of the stage II t test p value for the solution: the default value is 0.05.

`a2.f.min` The minimal value of the stage II F test p value for the solution: the default value is 0.05.

`a2.f.max` The maximal value of the stage II F test p value for the solution: the default value is 0.05.

`a2.step` The interval of the p value for stage II: the default value is 0.025.

`n2.min` The minimal value of sample size at stage II: the default value is 100.

`n2.max` The maximal value of sample size at stage II: the default value is 1000.

`n2.step` The interval of sample size at stage II: the default value is 100.

Details

The solutions space is constructed by p values of t test and Hotelling T test at stage I and stage II, the sample sizes at stage II and III. The ranges of these design parameters are set at the default value and divided into grids by using the .step. The protein dataset has the design information for each single protein. In this current version, it assumes the study is a matched case-control or a paired design study. The mean difference and its standard deviation of each protein is derived from the pilot and other prior information, the group information for each protein comes from the pathway analysis or the biologist's expert knowledge. Proteins in the dataset is arranged in the order of the group and the protein ID. The protein ID will be useful to track the protein selection at each stage when using the power.group.cost() with the optimize value = FALSE. The sample size of stage I is assumed to be known and is a conditional parameter for the optimal design parameters. The verification (stage II) sample size is set to be between 100-1000,which is a recommended range in the National Cancer Institute guideline. However, it can have different ranges in different studies(i.e. 10-100). The validation (stage III) sample size is conditional on the total budget. The current algorithm used the slake term to convert an inequality problem to an equality problem. In such setting, the stage III sample size at each iteration is derived from the approximated budget given the stage I/II design parameters of the current solution. In another words, it is a solution that assumes we use up all the available fund. To identify any other optimal solutions that with a smaller budget, a serial of slack terms can be applied. The ranges of significant levels for paired t-test, Hotelling T test at stage I and stage II are also user-defined. The ranges of all the design parameters will determine the running time of the program.

Value

solution.matrix

`comp1`	The number of true positives detected from the third stage
`comp2`	The stage I T test p value decision threshold
`comp3`	The stage I F test p value decision threshold
`comp4`	The stage II T test p value decision threshold
`comp5`	The stage II F test p value decision threshold
`comp6`	The sample size at stage II
`comp7`	The radius of the search subspace at the final iteration
`comp8`	from the number 8 component to last component are all solution space boundary parameters

Author(s)

Irene Suilan Zeng, Thomas Lumley

References

Babak Abbasi, S. T. A. N., Mehrzad Abdi Khalife, Yasser Faize (2011). "A hyprid Variable neighborhood search and simulated annealing algorithm to estimate the three parameters of the Weibull distribution." Expert Systems with Application 38: 700-708. Belisle, C. J. P. (1992). "Convergence theorems for a class of simulated annealing algorithm on Rd " Journal of Applied Probability 29: 885-895. Jorge Nocedal, S. J. W. (1999). Numerical Optimization. New York, Springer.

See Also

optim.two.stage.single(), optim.two.stage.appr(), power.group.cost()

Examples

```
assaycost2=function(n,p){280*p+1015*n}
assaycost3=function(p){200*p}
protein<-data.frame(proteinid=c(100,101,103,104,105),beta=c(2.4,2.6,0.5,2.6,0.7),sigma=c(0.6,0.7,0.3,0.7,0.4),
optim.two.stage.group(budget=500000,protein=protein,n1=60,artifact=rep(1,5),iter.number=1,assaycost2.function=
```

`optim.two.stage.single` *Optimization of the number of discoveries from a multistage clinical proteomic study*

Description

This function provides the optimal solution to maximize the number of discoveries of a three-stages proteomic study. The solution comprises of the significance levels of the paired t test at stage I/II and the sample sizes at stage II/III that maximizes the expected numbers of proteins with true effects from the multistage studies. It uses simulation annealing method that bases on the Metropolis function to determine if a solution shall be accepted at each step. The input parameters are the mean difference and its standard deviation in the relative/absolute quantity for each protein between the paired samples, the stage I sample size and a vector to define the technical artifact correction. A nested simulation annealing method is used to construct the multiple searching sub-space of the global solution space.

Usage

```
optim.two.stage.single(budget, protein, n1, artifact, iter.number, assaycost2.function, assaycost3.f
```

Arguments

`budget` The budget of the three-stage proteomic study for the stage II verification and stage III clinical validation: It does not include stage I cost.

`protein` The protein dataset: it needs to have four variables: proteinid,beta,sigma,group. proteinid is the numerical id for each protein, beta is the mean difference, sigma is the standard deviation of the difference, group is the group that the protein is being assigned. The protein dataset needs to have at least 2 proteins.

`n1` The stage I sample size

`artifact` The technical artifact correction factor for each protein

`iter.number` The number of iterations for the nested-simulation annealing

`assaycost2.function`
The assay cost function of number of proteins (p) and number of patients(n) at stage II.

`assaycost3.function`
The assay cost function of number of proteins (p) at stage III.

`recruit` The recruitment cost for a patient assuming it is the same at each stage: the default value is 100.

`s` The slake term of the total budget which is a small amount of dollars that transforms the inequality constraint to an equal constraint. The default value is 1000 dollars.

`a1.t.min` The minimal value of stage I t test p value for the solution: the default value is 0.01.

`a1.t.max` The maximal value of the stage I t test p value for the solution: the default value is 0.25.

`a1.step` The interval of the p values for t test of stage I: the default value is 0.025.

`a2.t.min` The minimal value of the stage II t test p value for the solution: the default value is 0.01.

`a2.t.max` The maximal value of the stage II t test p value for the solution: the default value is 0.05.

`a2.step` The interval of the p value for stage II: the default value is 0.025.

`n2.min` The minimal value of sample size at stage II: the default value is 100.

`n2.max` The maximal value of sample size at stage II: the default value is 1000.

`n2.step` The interval of sample size at stage II: the default value is 100.

Details

The solutions space is constructed by p values of t test at stage I and stage II, the sample sizes at stage II and III. The ranges of these design parameters are set at the default value and divided into grids by using the .step. The protein dataset has the design information for each single protein, in this current version, it assumes the study is a matched case-control or a paired sample study. The mean difference and its standard deviation of each protein can be derived from the pilot and other prior information. Proteins in the dataset are arranged in the order of the protein ID. The protein ID will be useful in tracking the protein selection at each stage when use the power.single.cost() with the optimize value = FALSE. The sample size of stage I is assumed to be known and is a conditional parameter for the optimal design parameters. The verification (stage II) sample size is set to be between 100-1000,which is a recommended range in the National Cancer Institute annual report(2007). However, it can have different ranges in different studies (i.e. 10-100). The validation(stage III) sample size is conditional on the total budget. The current algorithm used the slack term to convert an inequality problem to an equality problem. In such setting, the stage III sample size is derived from the approximated budget given the stage I/II design parameters. In another words, it is a solution that assumes we use up all the available fund. To identify any other optimal solutions with a smaller budget, a serial of slack terms can be applied. The ranges of significant levels for paired t-test, F test at stage I and stage II are also user-defined. The ranges of all the design parameters will determine the running time of the program.

Value

solution.matrix

`comp1` The number of detectable proteins from the third stage

`comp2` The stage I T test p value decision threshold

`comp3` The stage II T test p value decision threshold

`comp4` The sample size at stage II

`comp5` The radius of the search subspace at the final iteration

`comp6` from the number 5 component to the last component are all the solution boundary parameters

Author(s)

Irene Suilan Zeng, Thomas Lumley

References

Babak Abbasi, S. T. A. N., Mehrzad Abdi Khalife, Yasser Faize (2011). "A hyprid Variable neighborhood search and simulated annealing algorithm to estimate the three parameters of the Weibull distribution." Expert Systems with Application 38: 700-708. Belisle, C. J. P. (1992). "Convergence theorems for a class of simulated annealing algorithm on Rd " Journal of Applied Probability 29: 885-895. Jorge Nocedal, S. J. W. (1999). Numerical Optimization. New York, Springer.

See Also

optim.two.stage.appr(), optim.two.stage.group(), power.single.cost()

Examples

```
## An example of 5 proteins from an immunology proteomic study
assaycost2=function(n,p){280*p+1015*n}
assaycost3=function(p){200*p}
protein<-data.frame(proteinid=c(100,101,103,104,105),beta=c(2.4,2.6,0.5,2.6,0.7),sigma=c(0.6,0.7,0.3,0.7,0.4))
optim.two.stage.single(budget=500000,artifact=rep(1,5),protein=protein,n1=30,iter.number=1,assaycost2.function
```

`ots.env` *assign the current working environment*

Description

assign the current working environment

`power` *The power function used in the optim.two.stage.single function*

Description

The power function used in the optim.two.stage.single() function. It uses the Monte Carlo simulation to estimate the expected number of detectable proteins with the true effects, and utilize the grouping information.

Usage

```
power(initial, optimize = TRUE)
```

Arguments

`initial` The current solution selected from the solution space

`optimize` A logical variable to control the output of the function: When it is true , it will output the expected number of true effects.

`power.appr` *The power function used in the optim.two.stage.appr function*

Description

The power function used in the optim.two.stage.appr() function. It uses the analytical approximation to estimate the expected number of detectable proteins with the true effects, and use the grouping information.

Usage

```
power.appr(initial)
```

Arguments

`initial` The current solution selected from the solution space

`power.group.cost` *Derive the averaged estimated costs of stage II and III and the stage III sample size from the 1000 Monte Carlo simulated functions of a three-stage proteomis study, given a solution of the design parameters*

Description

The power.group.cost estimates the average costs at stage II and III and derives the stage III sample size with a user defined solution from the optim.two.stage.group(), or optim.two.stage.appr(). To obtain the cost and stage III sample size, the parameter "optimize" needs to be set to the default value "FALSE". Otherwise, it will give the expected number of proteins with true effect instead.

Usage

```
power.group.cost(initial, protein, n1, artifact, budget, s, assaycost2.function, assaycost3.function,
```

Arguments

`initial` A vector of the solution from the optim.two.stage.group(), it comprises of stage I/II t test and F test p value thresholds, and stage II sample size

`protein` The protein dataset that needs to have four variables: proteinid, beta, sigma, and group. proteinid is the numerical id for each protein, beta is the mean difference, sigma is the standard deviation of the difference, group is the group that the protein is being assigned. The protein dataset needs to have at least 2 proteins.

`n1` The stage I sample size

`artifact` The technical artifact correction factor for each protein

`budget` The budget of the three-stage proteomic study for the stage II verification and stage III clinical validation: It does not include stage I cost

`s` The slake term of the total budget which is a small amount of dollars that transfers the inequality constraint to an equal constraint. The default value is 1000 dollars.

`assaycost2.function`
The assay cost function of number of proteins (p) and number of patients(n) at stage II

`assaycost3.function`
The assay cost function of number of proteins (p) ONLY at stage III

`recruit` The recruitment cost for a patient. It is assumed to be the same at each stage. The default value is 100.

`optimize` A logical variable: when the default value is FALSE, it will output stage III sample size, the costs for stage II and III. When the value sets to TRUE, it will output the expected number of proteins with true effect for the defined solution.

Value

`mean.n3` The averaged stage III sample size

`mean.stage2.cost`
The averaged stage II cost

`mean.stage3.cost`
The averaged stage III cost

Note

This function is an additional function for optim.two.stage.group(), it use the same Monte carlo simulated objective function as that in optim.two.stage.group(). Users need to define the same cost functions as that used in the optim.two.stage.group(), and a solution. The solution vector can be the optimal solution from optim.two.stage.appr(), or optim.two.stage.group(). It can also be an arbitrary one in the solution space.

Author(s)

Irene Suilan Zeng

See Also

power.single.cost(), optim.two.stage.single(), optim.two.stage.appr()

Examples

```
assaycost2=function(n,p){280*p+1015*n}
assaycost3=function(p){200*p}
protein<-data.frame(proteinid=c(100,101,103,104,105),beta=c(2.4,2.6,0.5,2.6,0.7),sigma=c(0.6,0.7,0.3,0.7,0.4),
initial=c(0.01,0.18,0.01,0.05,100)
power.group.cost(initial,protein=protein,artifact=rep(1,5),n1=30,budget=500000,s=1000,assaycost2.function=assa
```

`power.single.cost` *Derive the averaged estimated costs of stage II and III and the stage III sample size from the 1000 Monte Carlo simulated functions of a three-stage proteomic study, given a solution of the design parameters*

Description

The power.single.cost estimated the average costs at stage II and III and derive the stage III sample size with a user defined solution from the optim.two.stage.single() To obtain the cost and stage III sample size, the parameter "optimize" need to set to the default value "FALSE". Otherwise, it will give the expected number of proteins with true effect instead.

Usage

```
power.single.cost(initial, protein, n1, artifact, budget, s, assaycost2.function, assaycost3.functio
```

Arguments

`initial` , A vector of the solution from the optim.two.stage.single(), it comprises of stage I/II t test p value thresholds, and stage II sample size

`protein` The protein dataset that needs to have three variables: proteinid, beta, sigma. proteinid is the numerical id for each protein, beta is the mean difference, sigma is the standard deviation of the difference. The protein dataset need to have at least 2 proteins.

`n1` The stage I sample size

`artifact` The technical artifact correction factor for each protein

`budget` The budget of the three-stage proteomic study for the stage II verification and stage III clinical validation: it does not include stage I cost

`s` The slake term of the total budget which is a small amount of dollars that transfers the inequality constraint to an equal constraint. The default value is 1000 dollars.

`assaycost2.function`
The assay cost function of number of proteins (p) and number of patients(n) at stage II.

`assaycost3.function`
The assay cost function of number of proteins (p) ONLY at stage III.

`recruit` The recruitment cost for a patient assuming it is the same at each stage. The default value is 100.

`optimize` A logical variable, the default value is FALSE when it will output stage III sample size, the costs for stage II and III. When the value sets to TRUE, it will output the expected number of detected proteins with true effect for the defined solution.

Value

`mean.n3` The averaged stage III sample size

`mean.stage2.cost`
The averaged stage II cost

`mean.stage3.cost`
The averaged stage III cost

Note

This function is an additional function for optim.two.stage.single(), it use the same Monte Carlo simulating objective function as that in optim.two.stage.single(). Users need to define the same cost functions as that used in the optim.two.stage.single(), and a solution. The solution vector can be the optimal solution from optim.two.stage.single(), or an arbitrary one in the solution space.

Author(s)

Irene S.L. Zeng

See Also

power.group.cost(), optim.two.stage.single(), optim.two.stage.group(),optim.two.stage.appr()

Examples

```
assaycost2=function(n,p){280*p+1015*n}
assaycost3=function(p){200*p}
protein<-data.frame(proteinid=c(100,101,103,104,105),beta=c(2.4,2.6,0.5,2.6,0.7),sigma=c(0.6,0.7,0.3,0.7,0.4),
initial=c(0.01,0.01,100)
power.single.cost(initial,protein=protein,artifact=rep(1,5),n1=30,budget=500000,s=1000,assaycost2.function=ass
```

`power.t` *The power function used in the optim.two.stage.single function*

Description

The power function used in the optim.two.stage.single() function. It uses the Monte Carlo simulation to estimate the expected number of detectable proteins with the true effects.

Usage

```
power.t(initial, optimize = TRUE)
```

Arguments

`initial`	The current solution selected from the solution space
`optimize`	A logical variable to control the output of the function: When it is true, it outputs the expected number of true effects.

`Ttest` *The function to calculate the p values for each protein in the optim.two.stage.single function*

Description

The function to calculate the p values for each protein in the optim.two.stage.single function. It inputs the protein sample, number of samples and number of proteins.

Usage

```
Ttest(sample, n, m)
```

Arguments

`sample`	The protein sample
`n`	The number of biological samples
`m`	The number of the proteins

Index