Package 'SignatureEstimation'

April 26, 2017

Type Package

Title decompose a tumor sample and estimate the signature exposures with bootstrap or simulated annealing distribution

Version 1.0.0

Author Xiaoqing Huang, Damian Wojtowicz

Maintainer Xiaoqing Huang <hxq.1001@gmail.com>

Description Given a tumor sample profile with the frequencies or the proportions of 96 mutational context types, the package can decompose the tumor catalogue into the known signatures with certain intensities of signature exposures, by choosing decompose method quadratic programming or simulated annealing, and also provide the bootstrap distribution or simulated annealing distribution of the exposures

Depends R (>= 3.3.1)

License GPL (>= 2)

Imports quadprog, GenSA

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1.9000

R topics documented:

bootstrapSigExposures 2
decomposeQP 2
decomposeSA 3
findSigExposures 3
suboptimalSigExposures 4

Index **5**

bootstrapSigExposures *bootstrapSigExposures Function This function allows to obtain the bootstrap distribution of the signature exposures of a certain tumor sample*

Description

bootstrapSigExposures Function This function allows to obtain the bootstrap distribution of the signature exposures of a certain tumor sample

Usage

```
bootstrapSigExposures(m, P, R, mutation.count = NULL,
  decomposition.method = decomposeQP, ...)
```

Arguments

m	observed turmor profile vector for a patient/sample, 96 by 1. It can be mutation counts, or mutation probabilities.
P	signature profile matrix, 96 by N (N = # signatures, COSMIC: N=30).
R	The number of bootstrap replicates.
mutation.count	if m is a vector of counts, then mutation.count equals the summation of all the counts. If m is probabilities, then mutation.count has to be specified.
decompostion.method	which method is selected to get the optimal solution: decomposeQP or decomposeSA

Examples

```
bootstrapSigExposures(tumorBRCA[,1], signaturesCOSMIC, 100, 2000, decomposeQP)
sigsBRCA = c(1,2,3,5,6,8,13,17,18,20,26,30)
bootstrapSigExposures(tumorBRCA[,1], signaturesCOSMIC[,sigsBRCA], 10, 1000, decomposeQP)
```

decomposeQP *decomposeQP Function*

Description

This function allows to get the optimal solution by using dual method to solve the quadratic programming problem.

Usage

```
decomposeQP(m, P, ...)
```

Arguments

m	observed turmor profile vector for a single patient/sample, 96 by 1. m is normalized.
P	signature profile matrix, 96 by N(N = # signatures, COSMIC: N=30)
control	some control parameter that can be passed into the solve.QP function

Examples

```
decomposeQP(tumorBRCA[,1], signaturesCOSMIC)
```

---

decomposeSA *decomposeSA Function*

Description

This function allows to get the optimal solution by using simulated annealing to solve the optimization problem.

Usage

```
decomposeSA(m, P, control = list())
```

Arguments

m	observed turmor profile vector for a single patient/sample, 96 by 1. m is normalized.
P	signature profile matrix, 96 by N(N = # signatures, COSMIC: N=30)
control	some control parameter that can be passed into the GenSA function

Examples

```
decomposeSA(tumorBRCA[,1], signaturesCOSMIC)
```

---

findSigExposures *findSigExposures Function wrapper function This function allows to obtain the optimal solution by specifying quadratic programming or simulated annealing to solve the optimization problem.*

Description

findSigExposures Function wrapper function This function allows to obtain the optimal solution by specifying quadratic programming or simulated annealing to solve the optimization problem.

Usage

```
findSigExposures(M, P, decomposition.method = decomposeQP, ...)
```

Arguments

M	observed turmor profile matrix for all the patient/sample, 96 by G. G is the number of patients. Each column can be mutation counts, or mutation probabilities. Each column will be normalized to sum up to 1.
P	signature profile matrix, 96 by N(N = # signatures, COSMIC: N=30)
decompostion.method	which method is selected to get the optimal solution: decomposeQP or decomposeSA

Examples

```
E1 = findSigExposures(tumorBRCA, signaturesCOSMIC, decomposeQP)
sigsBRCA = c(1,2,3,5,6,8,13,17,18,20,26,30)
E2 = findSigExposures(tumorBRCA, signaturesCOSMIC[, sigsBRCA], decomposeQP)
E3 = findSigExposures(tumorBRCA[, 1:10], signaturesCOSMIC, decomposeSA, list(maxit=1000, temperature=100))
E4 = findSigExposures(tumorBRCA[, 1:10], signaturesCOSMIC[, sigsBRCA], decomposeSA, list(maxit=2000))
E5 = findSigExposures(round(tumorBRCA*10000), signaturesCOSMIC, decomposeQP)
```

---

suboptimalSigExposures

suboptimalSigExposures Function This function allows to obtain the simulated annealing distribution of the signature exposures of a certain tumor sample

---

Description

suboptimalSigExposures Function This function allows to obtain the simulated annealing distribution of the signature exposures of a certain tumor sample

Usage

```
suboptimalSigExposures(m, P, R, optimal.error = NULL,
  suboptimal.factor = 1.05, control = list())
```

Arguments

m	observed turmor profile vector for a patient/sample, 96 by 1. It can be mutation counts, or mutation probabilities.
P	signature profile matrix, 96 by N(N = # signatures, COSMIC: N=30)
R	The number of replicates/trials of simulated annealing.
optimal.error	if it is NULL, then use SA method to obtain. Or you can provide one.
suboptimal.factor	suboptimal error.
control	some control parameter that can be passed into the function
mutation.count	if m is a vector of counts, then mutation.count equals the summation of all the counts. If m is probabilities, then mutation.count has to be specified.

Examples

```
suboptimalSigExposures(tumorBRCA[,1], signaturesCOSMIC, 100, optimal.error = NULL, 1.05)
```

Index

∗Topic **'simulated**
    suboptimalSigExposures, 4
∗Topic **QP**
    findSigExposures, 3
∗Topic **SA**
    findSigExposures, 3
∗Topic **annealing'**
    suboptimalSigExposures, 4
∗Topic **annealing**
    decomposeSA, 3
∗Topic **bootstrap**
    bootstrapSigExposures, 2
∗Topic **method:**
    findSigExposures, 3
∗Topic **optimal**
    findSigExposures, 3
∗Topic **or**
    findSigExposures, 3
∗Topic **programming**
    decomposeQP, 2
∗Topic **quadratic**
    decomposeQP, 2
∗Topic **simulated**
    decomposeSA, 3
∗Topic **suboptimal**
    suboptimalSigExposures, 4

bootstrapSigExposures, 2

decomposeQP, 2
decomposeSA, 3

findSigExposures, 3

suboptimalSigExposures, 4