# Package 'LFSPRO'

September 18, 2015

**Type** Package

**Title** TP53 mutation carrier estimation

**Version** 1.0.4

**Date** 2015-09-18

**Author** Gang Peng, Wenyi Wang

**Maintainer** Gang Peng <michael.gang.peng@gmail.com>

**Description** TP53 mutation is a main cause of Li-Fraumeni syndrome. This package is designed to estimate the probability that the counselee is a TP53 mutation carrier on the basis of his/her family cancer history. We also included LFS classic and Chompret criteria in the package.

**License** GPL-3

**LazyData** true

## R topics documented:

LFSPRO-package *TP53 mutation carrier estimation*

## Description

TP53 mutation is a main cause of Li-Fraumeni syndrome. This package is designed to estimate the probability that the counselee is a TP53 mutation carrier on the basis of his/her family cancer history. We also included LFS classic and Chompret criteria in the package.

## Details

| | |
|---|---|
| Package: | LFSPRO |
| Type: | Package |
| Version: | 1.0.4 |
| Date: | 2015-09-18 |
| License: | GPL-3 |

## Author(s)

Gang Peng, Wenyi Wang

Maintainer: Gang Peng <michael.gang.peng@gmail.com>

## References

Peng, G., Bojadzieva, J., Ballinger, M., Thomas, D., Strong, L. and Wang, W. Estimating TP53 Mutation Carrier Probability in Families with Li-Fraumeni Syndrome Using LFSpro. **Submitted**.

Chen, S., Wang, W., Broman, K. and Parmigiani, G. (2004) BayesMendel: An R Environment for Mendelian Risk Prediction. *Statistical Application in Genetics and Molecular Biology*, **3(1):** Article 21.

## See Also

lfspro, LFSClassic, LFSChompret

## Examples

```
options(stringsAsFactors = FALSE)
fam.id <- c("fam1","fam2","fam2","fam2","fam2")
id <- c(0,0,2,100,200)
counselee.id <- data.frame(fam.id, id)
LFSClassic(fam.data, cancer.data, counselee.id)
LFSChompret(fam.data, cancer.data, counselee.id)

allef.g <- list(c(0.9999,0.0001))
mRate.g <- 5e-4

lfspro(fam.data, cancer.data, LFSpenet.2010, counselee.id, allef.g, 1,mRate.g)
```

`calLK` *Calculate the likelihood Pr(D|G).*

## Description

Calculate the likelihood (Pr(D|G), probability of healthy status D given the genotype G) for each individual in the family.

## Usage

```
calLK(fam.cancer.data, penetrance.all)
```

## Arguments

| | |
|---|---|
| `fam.cancer.data` | Data including family information and cancer information. See fam.cancer.data for details. |
| `penetrance.all` | The penetrance of three kinds of genotype (wild type, one copy TP53 mutation and two copy TP53 mutation) for male and female in the population. See LFSpenet.2010 for details. |

## Value

Return a n*3 matrix. The likelihood of three kinds of genotype for each individual in the family. n denotes the number of individuals in the family.

## Author(s)

Gang Peng

## See Also

`lkNoneAffect`, `lfsproC`, `peelingRC`

## Examples

```
fam.cancer.data <- CombineData(fam.data, cancer.data)
calLK(fam.cancer.data[[1]], LFSpenet.2010)
```

`cancer.data` *Cancer information data*

## Description

Cancer information data storing the cancer type and diagnosed date

## Usage

```
cancer.data
```

## Format

A data frame with 4 variables.

`fam.id` Family id

`id` Individual id

`cancer.type` Cancer type. See LFSpro.cancer.type for details

`diag.age` The age when the individual was diagnosed with cancer

## See Also

`LFSpro.cancer.type`, `cancer.type.all`, `lfs.cut`, `invasive.cut`

## Examples

```
cancer.data
```

---

`cancer.type.all` *Predefined cancer types in LFSPRO.*

---

## Description

We classified the cancers into 11 groups according to NCCCN Guidelines Version 1.2012 Li-Fraumeni Syndrome criteria. sts: soft tissue sarcoma; ost: osterosarcoma; brain: brain tumorl breast: breast cancer; acc: adrenocortical carcinoma; leukemia: leukemia; lung: lung bronchoalveolar cancer; choroid: choroid plexus carcinoma; other.lfs: other LFS spectrum cancers; non.lfs: non LFS spectrum invasive cancers; benign: benign tumors

## Usage

```
cancer.type.all
```

## Format

"sts" "ost" "brain" "breast" "acc" "leukemia" "lung" "choroid" "other.lfs" "non.lfs" "benign"

## See Also

`LFSpro.cancer.type`, `cancer.data`, `lfs.cut`, `invasive.cut`

## Examples

```
cancer.type.all
```

CombineData *Combine the family information data and cancer information data*

## Description

Combine the family information data and cancer information data into a single combined data. The data is organized family by family.

## Usage

```
CombineData(fam.data, cancer.data)
```

## Arguments

| | |
|---|---|
| fam.data | Data frame storing family information. See fam.data for details. |
| cancer.data | Data frame storing cancer information. See cancer.data for details. |

## Value

A list. Each component in the list stores the family and cancer information for a family. See fam.cancer.data for details.

## Author(s)

Gang Peng

## See Also

lfspro,lfsproC

## Examples

```
# convert cancer type to specific number and check the cancer type
num.cancer <- nrow(cancer.data)
cancer.type.num <- rep(-1, num.cancer)
for(i in 1:num.cancer){
  tmp <- LFSpro.cancer.type[cancer.data$cancer.type[i]]
  if(is.na(tmp)){
    print(paste("Cannot find cancer ", cancer.data$cancer.type[i],
                " in the LFSpro predefined cancer type", sep = ""))
    print("LFSpro predefined cancer types are: ")
    print(cancer.type.all)
    print("Please check the input cancer information data.")

    num.counselee <- nrow(counselee.id)
    pp <- rep(-1, num.counselee)

    rlt <- data.frame(cbind(counselee.id, pp),check.names = FALSE)
    colnames(rlt) <- c("fam.id", "id", "pp")
    return(rlt)
  }
  cancer.type.num[i] <- tmp
}
```

```
cancer.data$cancer.type <- cancer.type.num

CombineData(fam.data, cancer.data)
```

---

`fam.cancer.data` *Combined data with family and cancer information*

---

## Description

Combined data with family and cancer information. It is used in function LFSproC.

## Usage

```
fam.cancer.data
```

## Format

List. Each component of the list stores the family and cancer information of one family. Each component is also a list with the following variables:

`fam.id` family id

`id` individual id

`fid` father id

`mid` mother id

`gender` gender, 0: female, 1:male

`age` If the individual is dead, it is death age, otherwise it is current age. We use last contact date to calculate current age.

`cancer.info` Cancer informatice. List. Each component of the list stores cancer information for each individual in the family. For each component, if there there is no cancer for the individual, it is a 0 by 0 data frame. Otherwise, it is a data frame with cancer.type (coded number) and diag.age(the age at which the cancer was diagnosed).

## Examples

```
fam.cancer.data
```

---

`fam.data` *Family information data*

---

## Description

Family information data used in function LFSpro.

## Usage

```
fam.data
```

## Format

A data frame containing 9 variables.

`fam.id` family id

`id` individual id

`fid` father id

`mid` mother id

`gender` gender. 0: female, 1: male

`age` If the individual was alive at the last contact date, use the age at that time. otherwise, use the age at death.

## Details

Family id could include characters, but id, father id and mother id must be interger larger than 0.

## See Also

`cancer.data`

## Examples

```
fam.data
```

---

# `firstDegreeRelative` *First degree relatives*

---

## Description

Find out the first degree relatives of a given individuals in a family.

## Usage

```
firstDegreeRelative(pedigree, ii)
```

## Arguments

| | |
|---|---|
| `pedigree` | A data frame including ID, fID and mID. These IDs should be a positive integer. |
| `ii` | Individual *index* (not ID here). If you want to get the first degree relatives of the second individual in the family, set ii as 2. |

## Value

The *index* of individuals who are the first degree relatives to the given individual.

## Author(s)

Gang Peng

## See Also

`secondDegreeRelative`, `LFSClassic`, `LFSChompret`

## Examples

```
fam.cancer.data <- CombineData(fam.data, cancer.data)
lfsData <- reformatForClassicChompret(fam.cancer.data[[1]])
pedigree <- data.frame(lfsData$ID,lfsData$fID,lfsData$mID,stringsAsFactors=FALSE)
names(pedigree) <- c("ID","fID","mID")
firstDegreeRelative(pedigree,2)
```

---

# invasive.cut *Cutoff for malignant cancer*

---

## Description

We coded diffrent cancers into numbers. If the code for a tumor is less than invasive.cut, it is malignant cancer otherwise it is benign tumor.

## Usage

```
invasive.cut
```

## Format

The format is: num 100

## Examples

```
invasive.cut
```

---

# lfs.cut *Cutoff for Li-Fraumeni Syndrome spectrum cancer*

---

## Description

We coded diffrent cancers into numbers. If the code for a tumor is less than lfs.cut, it is Li-Fraumeni Syndrome spectrum cancer.

## Usage

```
lfs.cut
```

## Format

The format is: num 50

## Examples

```
lfs.cut
```

`LFSChompret` *The Chompret criteria for Li-Fraumeni syndrome*

## Description

Use LFS Chompret criteria to find out the TP53 mutation carriers in the family.

## Usage

```
LFSChompret(fam.data, cancer.data, counselee.id)
```

## Arguments

| | |
|---|---|
| `fam.data` | Family information data. See fam.data for details. |
| `cancer.data` | Cancer information data. See cancer.data for details. |
| `counselee.id` | Data frame including two variables: fam.id (family id of counselees) and id (individual id of counselees). |

## Value

A data frame of three columns:fam.id, id and result. The result column is a vector of boolean variables for all the counselees indicating whether they are TP53 mutation carriers. TRUE: carrier. FALSE: non-carrier.

## Author(s)

Gang Peng

## References

Chompret, A., et al. (2001). Sensitivity and predictive value of criteria for p53 germline mutation screening. *J Med Genet*, **38(1)**: 43-47.

Tinat, J., et al. (2009). 2009 version of the Chompret criteria for Li Fraumeni syndrome. *J Clin Oncol*, **27(26)**: e108-109; author reply e110.

## See Also

`LFSClassic`, `lfspro`

## Examples

```
options(stringsAsFactors = FALSE)
fam.id <- c("fam1","fam2","fam2","fam2","fam2")
id <- c(0,0,2,100,200)
counselee.id <- data.frame(fam.id, id)
LFSChompret(fam.data, cancer.data, counselee.id)
```

`LFSClassic` *The Classic criteria for Li-Fraumeni syndrome*

## Description

Use LFS Classic criteria to find out the TP53 mutation carriers in the family.

## Usage

```
LFSClassic(fam.data, cancer.data, counselee.id)
```

## Arguments

| | |
|---|---|
| `fam.data` | Family information data. See fam.data for details. |
| `cancer.data` | Cancer information data. See cancer.data for details. |
| `counselee.id` | Data frame including two variables: fam.id (family id of counselees) and id (individual id of counselees). |

## Value

A data frame of three columns:fam.id, id and result. The result column is a vector of boolean variables for all the counselees indicating whether they are TP53 mutation carriers. TRUE: carrier. FALSE: non-carrier.

## Author(s)

Gang Peng

## References

Li, F. P., et al. (1988). A cancer family syndrome in twenty-four kindreds. *Cancer Res*, **48(18)**: 5358-5362.

## See Also

`LFSChompret`, `lfspro`

## Examples

```
options(stringsAsFactors = FALSE)
fam.id <- c("fam1","fam2","fam2","fam2","fam2")
id <- c(0,0,2,100,200)
counselee.id <- data.frame(fam.id, id)
LFSClassic(fam.data, cancer.data, counselee.id)
```

## LFSpenet.2010 *Penetrance for TP53 mutation*

### Description

Penetrance table of three kinds of genotype of TP53 gene at age from 1 to 110 for both male and female.

### Usage

```
LFSpenet.2010
```

### Format

List. Two component of list indicates the penetrance of male and female (fMX: male, fFX: female). For each component, it is a 110*3 matrix containing the penetrance from age 1 to age 110 for three kinds of genotype: P530 (wild type), P531 (one allele mutation) and P532 (two allele mutation).

### References

Wu, C. C., et al. (2010). Effects of measured susceptibility genes on cancer risk in family studies. *Hum Genet* **127(3)**: 349-357.

### Examples

```
LFSpenet.2010
```

## lfspro *Estimating TP53 mutation probability for families with Li-Fraumeni Syndrome*

### Description

We use Mendelian risk prediction model to estimate the probability for the counselee as a TP53 mutation carrier on the basis of his/her family cancer history.

### Usage

```
lfspro(fam.data, cancer.data, penetrance.all, counselee.id, allef, nloci, mRate)
```

### Arguments

| | |
|---|---|
| `fam.data` | Family information data. See fam.data for details. |
| `cancer.data` | Cancer information data. See cancer.data for details. |
| `penetrance.all` | Penetrance data. See LFSpenet.2010 for details. |
| `counselee.id` | Data frame including two variables: fam.id (family id of counselees) and id (individual id of counselees). |

| | |
|---|---|
| allef | List. Allele frequency for each locus/gene. If there is only one gene and two alleles in the gene (allele frequency is 0.1 and 0.9), allef = list(c(0.1,0.9)), If there are two genes,two alleles (allele frequency is 0.1 and 0.9) for gene 1 and three alleles (allele frequncy is 0.2, 0.2 and 0.6) for gene 2, allef = list(c(0.1,0.9),c(0.2,0.2,0.6)). We suggeste to use list(c(1-maf,maf)). maf = 0.0001. |
| nloci | Number of loci/genes in the model. It is set to be 1 here. |
| mRate | Mutation rate. We suggest to use 5e-4. |

## Value

The TP53 mutation carrier probability for each counselee. It is a data frame with 3 variables: fam.id (family id), id (individual id) and pp(posterior probability that the counselee is a TP53 mutation carrier).

## Author(s)

Gang Peng, Wenyi Wang

## References

Peng, G., Bojadzieva, J., Ballinger, M., Thomas, D., Strong, L. and Wang, W. Estimating TP53 Mutation Carrier Probability in Families with Li-Fraumeni Syndrome Using LFSPRO. **Submitted**.

Chen, S., Wang, W., Broman, K. and Parmigiani, G. (2004) BayesMendel: An R Environment for Mendelian Risk Prediction. *Statistical Application in Genetics and Molecular Biology*, **3(1):** Article 21.

## See Also

LFSClassic, LFSChompret, lfsproC

## Examples

```
allef.g <- list(c(0.9999,0.0001))
mRate.g <- 5e-4
options(stringsAsFactors = FALSE)
fam.id <- c("fam1","fam2","fam2","fam2","fam2")
id <- c(0,0,2,100,200)
counselee.id <- data.frame(fam.id, id)
lfspro(fam.data, cancer.data, LFSpenet.2010, counselee.id, allef.g, 1,mRate.g)
```

---

# LFSpro.cancer.type *Predefined cancer types and the corresponding number in LFSpro*

---

## Description

We classified the cancers into 11 groups according to NCCCN Guidelines Version 1.2012 Li-Fraumeni Syndrome criteria. And then we coded these different groups of cancer into different number.

sts(soft tissue sarcoma): 1

ost(osterosarcoma): 2

brain(brain tumor): 3

breast(breast cancer): 4

acc(adrenocortical carcinoma): 5

leukemia: 6

lung(lung bronchoalveolar cancer): 7

choroid(choroid plexus carcinoma): 8

other.lfs(other LFS spectrum cancers): 20

non.lfs(non LFS spectrum invasive cancers): 50

benign(benign tumors): 100

## Usage

```
LFSpro.cancer.type
```

## Format

A vector with names of different cancer types.

sts ost brain breast acc leukemia lung choroid other.lfs non.lfs benign

1 2 3 4 5 6 7 8 20 50 100

## Examples

```
LFSpro.cancer.type
```

---

`lfsproC` *Estimating TP53 mutation probability for families with Li-Fraumeni Syndrome for combined family and cancer data*

---

## Description

We use Mendelian risk prediction model to estimate the probability for the counselee as a TP53 mutation carrier on the basis of his/her family cancer history.

## Usage

```
lfsproC(fam.cancer.data, penetrance.all, counselee.id, allef, nloci, mRate)
```

## Arguments

| | |
|---|---|
| `fam.cancer.data` | Combined family and cancer information data for *ONE FAMILY ONLY*. See fam.cancer.data for details. |
| `penetrance.all` | Penetrance data. See LFSpenet.2010 for details. |
| `counselee.id` | Individual id for the counselee. If you want to estimate multiple samples at the same time, just set counselee.id as a vector of IDs for all the samples you want to estimate. |

| | |
|---|---|
| allef | List. Allele frequency for each locus/gene. If there is only one gene and two alleles in the gene (allele frequency is 0.1 and 0.9), allef = list(c(0.1,0.9)), If there are two genes,two alleles (allele frequency is 0.1 and 0.9) for gene 1 and three alleles (allele frequncy is 0.2, 0.2 and 0.6) for gene 2, allef = list(c(0.1,0.9),c(0.2,0.2,0.6)). We suggeste to use list(c(1-maf,maf)). maf = 0.0001. |
| nloci | Number of loci/genes in the model. It is set to be 1 here. |
| mRate | Mutation rate. We suggest to use 5e-4. |

## Value

The TP53 mutation carrier probability for each counselee.

## Author(s)

Gang Peng, Wenyi Wang

## References

Peng, G., Bojadzieva, J., Ballinger, M., Thomas, D., Strong, L. and Wang, W. Estimating TP53 Mutation Carrier Probability in Families with Li-Fraumeni Syndrome Using LFSPRO. **Submitted**.

Chen, S., Wang, W., Broman, K. and Parmigiani, G. (2004) BayesMendel: An R Environment for Mendelian Risk Prediction. *Statistical Application in Genetics and Molecular Biology*, **3(1):** Article 21.

## See Also

LFSClassic, LFSChompret, lfspro, peelingRC

## Examples

```
# convert cancer type to specific number and check the cancer type
num.cancer <- nrow(cancer.data)
cancer.type.num <- rep(-1, num.cancer)
for(i in 1:num.cancer){
  tmp <- LFSpro.cancer.type[cancer.data$cancer.type[i]]
  if(is.na(tmp)){
    print(paste("Cannot find cancer ", cancer.data$cancer.type[i],
                " in the LFSpro predefined cancer type", sep = ""))
    print("LFSpro predefined cancer types are: ")
    print(cancer.type.all)
    print("Please check the input cancer information data.")

    num.counselee <- nrow(counselee.id)
    pp <- rep(-1, num.counselee)

    rlt <- data.frame(cbind(counselee.id, pp),check.names = FALSE)
    colnames(rlt) <- c("fam.id", "id", "pp")
    return(rlt)
  }
  cancer.type.num[i] <- tmp
}

cancer.data$cancer.type <- cancer.type.num

allef.g <- list(c(0.9999,0.0001))
```

```
mRate.g <- 5e-4
fam.cancer.data <- CombineData(fam.data, cancer.data)
lfsproC(fam.cancer.data[[1]], LFSpenet.2010, 0, allef.g, 1,mRate.g)
```

---

`lkNoneAffect` *Likelihood for non-affected individuals*

---

## Description

Calculate the likelihood for non-affected individuals.

## Usage

```
lkNoneAffect(penetrance, age)
```

## Arguments

| | |
|---|---|
| `penetrance` | The penetrance matrix for only male or female. See LFSpenet.2010 for details. |
| `age` | The age when the individual is still healthy. |

## Value

The likelihood (Pr(D|G)) for the none affected individual at 'age' years old. G can be TP530, TP531 and PT532.

## Author(s)

Gang Peng

## See Also

`calLK`

## Examples

```
lkNoneAffect(LFSpenet.2010$fFX, 50)
```

---

`peelingRC` *Peeling interface in R.*

---

## Description

Peeling (Elston-Stewart algorithm) is the key function in LFSpro. We implmented it in C++ to make it fast. peelingRC is used to link the peeling algorithm in C++ version to R.

## Usage

```
peelingRC(allef, LIK, ped, counselee.id, nloci = 1, mRate = 0)
```

## Arguments

| | |
|---|---|
| allef | List. Allele frequency for each locus/gene. If there is only one gene and two alleles in the gene (allele frequency is 0.1 and 0.9), allef = list(c(0.1,0.9)), If there are two genes,two alleles (allele frequency is 0.1 and 0.9) for gene 1 and three alleles (allele frequncy is 0.2, 0.2 and 0.6) for gene 2. allef = list(c(0.1,0.9),c(0.2,0.2,0.6)) |
| LIK | Matrix, likelihood, Pr(D\|G), for three kinds of genotype for all the individuals in the family. D: healthy status. G: genotype. |
| ped | Pedigree structure. A data frame with four varaibles: ID(individual id), Gender (gender, 0: female, 1: male), FatherID(father id) and MotherID(mother id). |
| counselee.id | Individual id for the counselee. If you want to estimate multiple samples at the same time, just set counselee.id as a vector of IDs for all the samples you want to estimate. |
| nloci | Number of loci/genes in the model. |
| mRate | Mutation rate. |

## Details

One family a time.

## Value

The posterior probability (Pr(G|D)) for each counselee

## Author(s)

Gang Peng

## References

Elston, R. C., Stewart, J. (1971) A general model for the genetic analysis of pedigree data. *Hum Hered.*, **21**, 523-542.

## See Also

lfsproC

## Examples

```
fam.cancer.data <- CombineData(fam.data, cancer.data)
allef <- list(c(0.9999,0.0001))
FamData <- fam.cancer.data[[1]]
lik <- calLK(FamData, LFSpenet.2010)

#################################
# convert data
#################################
counselee.id <- FamData$id[1]
id <- as.integer(FamData$id)
fid <- as.integer(FamData$fid)
mid <- as.integer(FamData$mid)
counselee.id <- as.integer(counselee.id)

if(min(id)==0){
```

```
  id <- id+1
  fid <- fid+1
  mid <- mid+1
  counselee.id <- counselee.id+1
}
fid[is.na(fid)] <- 0
mid[is.na(mid)] <- 0

ped <- data.frame(ID=id, Gender = FamData$gender,
                  FatherID = fid,
                  MotherID = mid,
                  stringsAsFactors=FALSE)

peelingRC(allef, lik, ped, counselee.id, 1, 5e-4)
```

---

`reformatForClassicChompret` *Reformat input data of LFSpro for evaluation using the Classic or the Chompret criteria*

---

## Description

Reformat input data of LFSpro for evaluation using the Classic or the Chompret criteria

## Usage

```
reformatForClassicChompret(fam.cancer.data)
```

## Arguments

`fam.cancer.data` Combined family and cancer information data for *ONE FAMILY ONLY*. See fam.cancer.data for details.

## Value

Data with format used in LFSClassic and LFSChompret. See `LFSClassic` and `LFSChompret` for details.

## Note

One family a time.

## Author(s)

Gang Peng

## See Also

`LFSClassic`, `LFSChompret`, `lfsproC`

## Examples

```
fam.cancer.data <- CombineData(fam.data, cancer.data)
reformatForClassicChompret(fam.cancer.data[[1]])
```

`secondDegreeRelative` *Second degree relative*

## Description

Find out the second degree relatives of a given individuals in a family.

## Usage

```
secondDegreeRelative(pedigree, ii)
```

## Arguments

| | |
|---|---|
| `pedigree` | A data frame including ID, fID and mID. These IDs should be a positive integer. |
| `ii` | Individual *index* (not ID here). If you want to get the second degree relatives of the second individual in the family, set ii as 2. |

## Value

The *index* of individuals who are the second degree relatives to the given individual.

## Author(s)

Gang Peng

## See Also

`firstDegreeRelative`, `LFSClassic`, `LFSChompret`

## Examples

```
fam.cancer.data <- CombineData(fam.data, cancer.data)
lfsData <- reformatForClassicChompret(fam.cancer.data[[1]])
pedigree <- data.frame(lfsData$ID,lfsData$fID,lfsData$mID,stringsAsFactors=FALSE)
names(pedigree) <- c("ID","fID","mID")
secondDegreeRelative(pedigree,2)
```

# Index