dBUMscore

December 16, 2014

---

dBUMscore   *Function to transform p-values into scores according to the fitted beta-uniform mixture model and/or after controlling false discovery rate*

Description

dBUMscore is supposed to take as input a vector of p-values, which are transformed into scores according to the fitted beta-uniform mixture model. Also if the FDR threshold is given, it is used to make sure that p-values below this are considered significant and thus scored positively. Instead, those p-values above the given FDR are considered insigificant and thus scored negatively.

Usage

```
dBUMscore(fit, method = c("pdf", "cdf"), fdr = NULL, scatter.bum = T)
```

Arguments

fit   an object of class "BUM"

method   the method used for the transformation. It can be either "pdf" for the method based on the probability density function of the fitted model, or "cdf" for the method based on the cumulative distribution function of the fitted model

fdr   the given FDR threshold. By default, it is set to NULL, meaning there is no constraint. If given, those p-values with the FDR below this are considered significant and thus scored positively. Instead, those p-values with the FDR above this given FDR are considered insigificant and thus scored negatively

scatter.bum   logical to indicate whether the scatter graph of scores against p-values should be drawn. Also indicated is the p-value (called tau) corresponding to the given FDR threshold (if any)

Value

- scores: a vector of scores

Note

The transformation from the input p-value x to the score $S(x)$ is based on the fitted beta-uniform mixture model with two parameters λ and a: $f(x|\lambda, a) = \lambda + (1 - \lambda) * a * x^{a-1}$. Specifically, it considers the log-likelihood ratio between the signal and noise component of the model. The probability density function (pdf) of the signal component and the noise component are $(1 - \lambda) * a * (x^{a-1} - 1)$ and $\lambda + (1 - \lambda) * a$, respectively. Accordingly, the cumulative distribution function (cdf) of the signal component and the noise component are $\int_0^x (1 - \lambda) * a * (x^{a-1} - 1) \, dx$ and $\int_0^x \lambda + (1 - \lambda) * a \, dx$. In order to take into account the significance of the p-value, the fdr threshold is also used for down-weighting the score. According to how to measure both components, there are two methods implemented for deriving the score $S(x)$:

- The method "pdf": $S(x) = log_2 \frac{(1-\lambda)*a*(x^{a-1}-1)}{\lambda+(1-\lambda)*a} - log_2 \frac{(1-\lambda)*a*(\tau^{a-1}-1)}{\lambda+(1-\lambda)*a} = log_2\left(\frac{x^{a-1}-1}{\tau^{a-1}-1}\right)$.
  For the purpose of down-weighting scores, it must ensure $log_2 \frac{(1-\lambda)*a*(\tau^{a-1}-1)}{\lambda+(1-\lambda)*a} \geq 0$, that is, the constraint via $\tau \leq \left(\frac{\lambda+2*a*(1-\lambda)}{a*(1-\lambda)}\right)^{\frac{1}{a-1}}$

- The method "cdf": $S(x) = log_2 \frac{\int_0^x (1-\lambda)*a*(x^{a-1}-1)\,dx}{\int_0^x \lambda+(1-\lambda)*a\,dx} - log_2 \frac{\int_0^\tau (1-\lambda)*a*(\tau^{a-1}-1)\,dx}{\int_0^\tau \lambda+(1-\lambda)*a\,dx} = log_2 \frac{(1-\lambda)*(x^{a-1}-a)}{\lambda+(1-\lambda)*a} - log_2 \frac{(1-\lambda)*(\tau^{a-1}-a)}{\lambda+(1-\lambda)*a} = log_2\left(\frac{x^{a-1}-a}{\tau^{a-1}-a}\right)$. For the purpose of down-weighting scores, it must ensure $log_2 \frac{(1-\lambda)*(\tau^{a-1}-a)}{\lambda+(1-\lambda)*a} \geq 0$, that is, the constraint via $\tau \leq \left(\frac{\lambda+2*a*(1-\lambda)}{1-\lambda}\right)^{\frac{1}{a-1}}$

- Where $\tau = \left[\frac{\lambda+(1-\lambda)*a-fdr*\lambda}{fdr*(1-\lambda)}\right]^{\frac{1}{a-1}}$, i.e. the p-value corresponding to the exact fdr threshold. It can be deduced from the definition of the false discovery rate: $fdr \doteq \frac{\int_0^\tau \lambda+(1-\lambda)*a\,dx}{\int_0^\tau \lambda+(1-\lambda)*a*x^{a-1}\,dx}$. Notably, if the calculated τ exceeds the contraint, it will be reset to the maximum end of that constraint

See Also

dBUMfit

Examples

```
# 1) generate an vector consisting of random values from beta distribution
x <- rbeta(1000, shape1=0.5, shape2=1)

# 2) fit a p-value distribution under beta-uniform mixture model
fit <- dBUMfit(x)

# 3) calculate the scores according to the fitted BUM and fdr=0.01
# using "pdf" method
scores <- dBUMscore(fit, method="pdf", fdr=0.01)
# using "cdf" method
scores <- dBUMscore(fit, method="cdf", fdr=0.01)
```