```r
library(knitr)
opts_chunk$set(tidy=TRUE, cache=TRUE, warning=FALSE, eval=FALSE)
```

fFDR Package Vignette (Version ??)

David G. Robinson

June 30, 2014

1 Introduction

This is a vignette for the `fFDR` package, which performs functional false discovery rate control.

2 Example Usage

Simulated data can be found in the `simtests` dataset.

```
library(fFDR)
data(simtests)

head(simtests)
```

This dataset contains ?? one-sample t-tests of different means and sample sizes. The means (while provided in the table) can be assumed to be unknown, while the sample sizes are known. Thus, we need to adjust the p-values from the t-tests based on the information we have about the sample sizes.

To do this, we use the `fqvalue` function, which takes at least two arguments: the p-values and the values it should be controlling for, which in this case is the sample size.

```
fq = fqvalue(simtests$pvalue, simtests$sample.size)
```

This produces a `data.table` with 6 columns, and the same number of rows as the number of pvalues given:

```
fq
```

The first two, `pvalue` and `X`, are the original inputs, and `qX` is the quantile of `X` that was used in the functional FDR computations. `pi0` is the estimate of π_0 for each hypothesis, which varies depending on `X`.

The `fqvalue` is the functional q-value. This is designed so that rejecting the null hypothesis for all tests with q-value less than q should lead to a false

```r
compareQvalue(fq)
```

Figure 1: Comparison between the traditional q-value and the functional q-value, colored based on the quantile of the X distribution. Notably, values with high X have functional q-values that are lower than their traditional q-values, while values with low X have higher functional q-values.

discovery rate of q. The cumulative density is a value used in the computation of the functional q-value: it represents the estimated probability that a p-value at the given X_i would be less than p_i.

2.1 Analysis

The `fq$fqvalue` column shows us the computed q-values:

```r
sum(fq$fqvalue < 0.05)
sum(fq$fqvalue < 0.01)
```

We can also perform a comparison to a traditional q-value, in this case using the `qvalue` package, and see how many we find:

```r
library(qvalue)

q = qvalue(fq$pvalue)
sum(q$qvalues < 0.05)
sum(q$qvalues < 0.01)
```

3 Plotting

The `fFDR` package provides functions for plotting various useful graphs based on the output of`fqvalue`.

One of `fFDR`'s functionalities is to calculate how π_0 varies based on X. Plotting this π_0 based on the quantile of X can show how the

```r
plotPi0(fq, horizontal.line = TRUE)
```

Another question is to what extent the functional q-value differs from the traditional q-value, which does not take X into account.

Another way of comparing traditional q-values to functional q-values is to scatter the p-values against the quantiles of the X distribution.

```r
plot(fq)
```

Figure 2: The vertical lines represent the q-value significance thresholds using traditional methods, in which the q-value depends only on the p-value and not X. The colors of the points represent the significance thresholds of the functional q-value.

```r
plot(fq, doublelog = TRUE)
```

Figure 3: Equivalent to Figure 2, but places the p-value on a $\log(-\log(p))$ scale, which more clearly shows the effects on very low p-values.