Exploring Permutation P-value

Jean Fan

7/15/2020

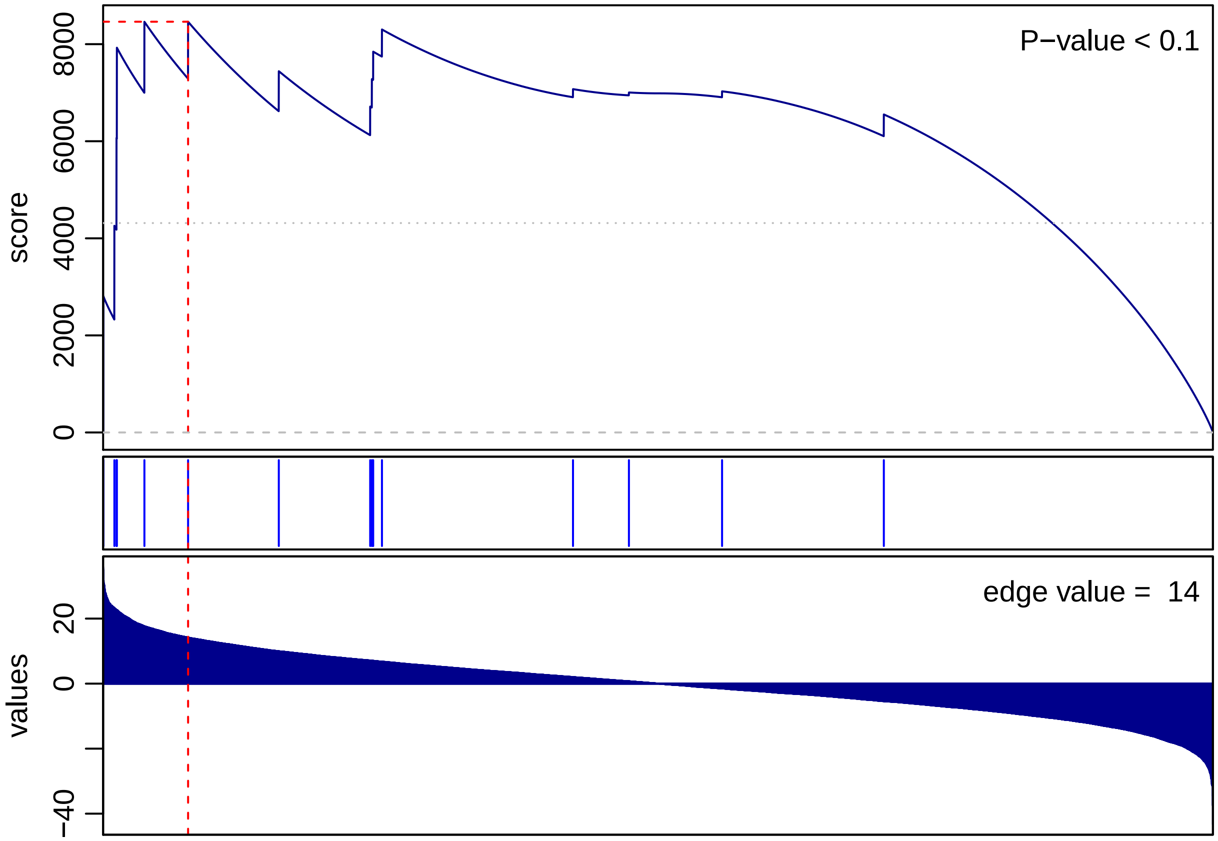
In this tutorial, we will explore what it means to use derive p-values by permutation. `liger` and other many gene set enrichment analysis procedures derive p-value through permutation. So it is important to understand what p-values mean when they are derived from 100, 1000, 10,000 or more permutations.

To demonstrate this, we will first simulate a significantly enriched gene set.

```
library(liger)
# load gene set
data("org.Hs.GO2Symbol.list")
# get universe
universe <- unique(unlist(org.Hs.GO2Symbol.list))
# get a gene set
gs <- org.Hs.GO2Symbol.list[[1]]
# fake dummy example where everything in gene set is perfectly enriched
vals <- rnorm(length(universe), 0, 10)
names(vals) <- universe
set.seed(0)
vals[gs] <- rnorm(length(gs), 10, 10)
```

We will test this gene set of enrichment with only 10 permutations to generate the null distribution.

```
set.seed(0)
gsea(values=vals, geneset=gs, plot=TRUE, n.rand=10)
```

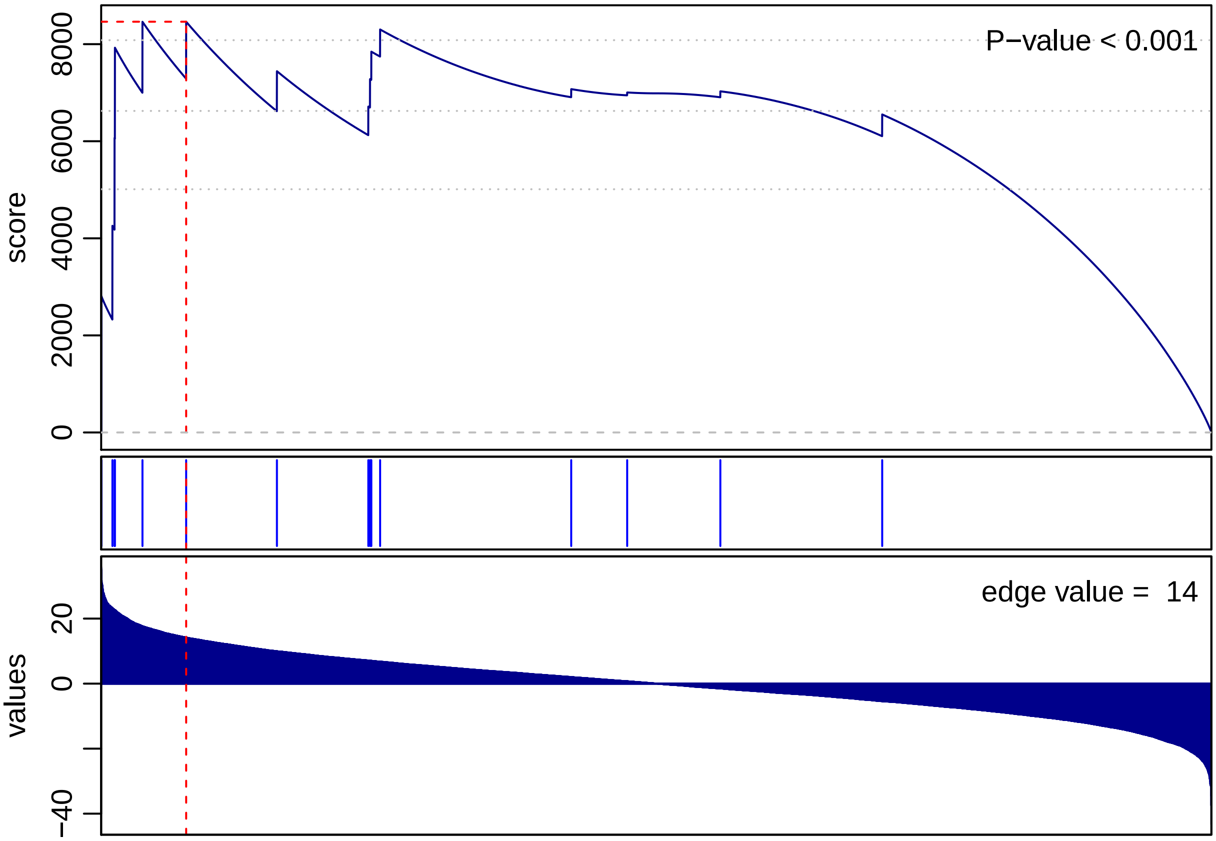


```
## [1] 0.1
```

Note that the returned p-value is 0.1. This is larger than our typicaly significance threshold of 0.05, so we may be led to believe that this gene set is not significant. But wait! What this really means is that the true p-value is < 0.1. But due to using only 10 permutations, 1/10 or 0.1 is the most precisely we can estimate the p-value.

So let's try 1000 permutations then.

```
set.seed(0)
gsea(values=vals, geneset=gs, plot=TRUE, n.rand=1e3)
```

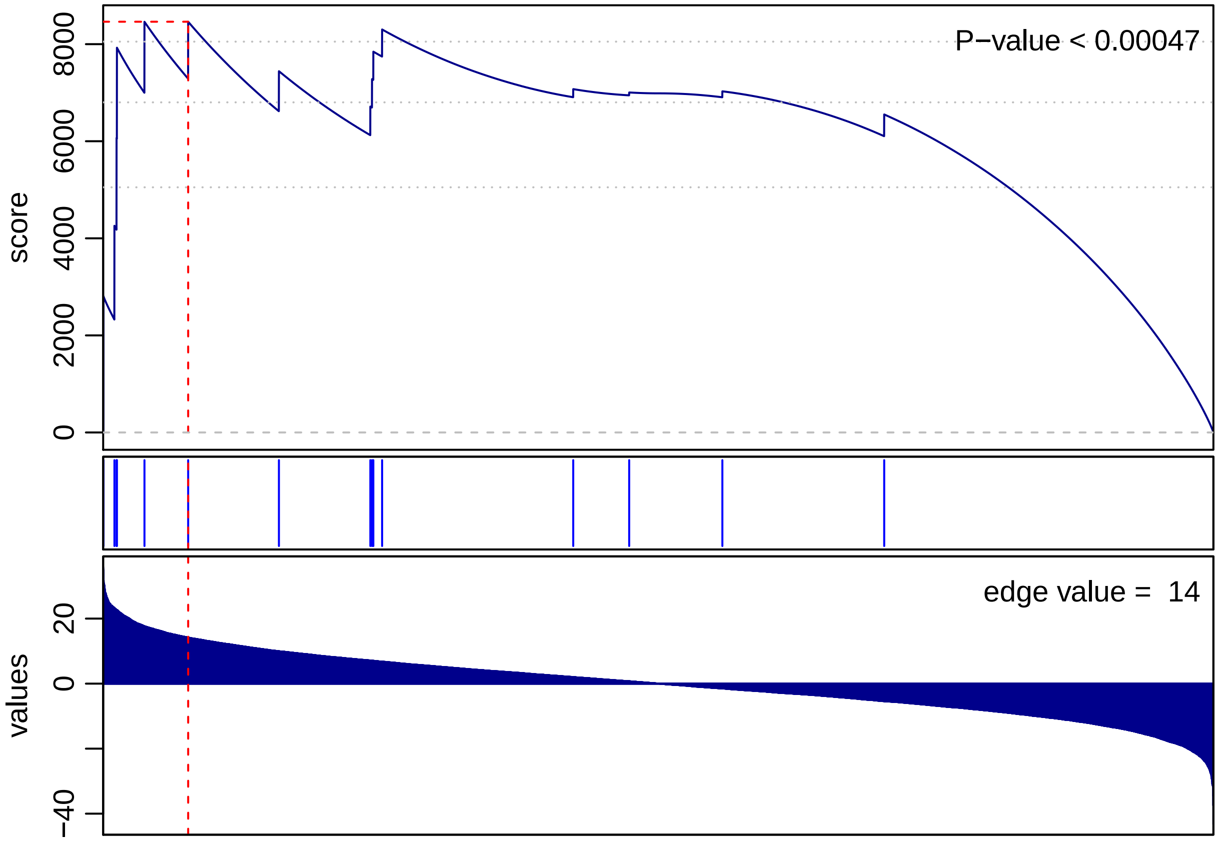


```
## [1] 0.001
```

Now the returned p-value is 0.001. But again, what this really means is that the true p-value is < 0.001 but due to using only 1000 permutations, 1/1000 or 0.001 is the most precisely we can estimate the p-value.

So let's try 1e5 permutations then.

```
set.seed(0)
gsea(values=vals, geneset=gs, plot=TRUE, n.rand=1e5)
```



```
## [1] 0.00047
```

As you can see, the more permutations we use, the more precisely we can estimate how 'strange' is our observed enrichment score compared to what we might expect to see by chance. We can always derive a permutation p-value but the precision of our p-value will be limited by the number of permutations we use.

Try it out for yourself

- What p-value would you expect returned if we use only 20 permutations?
- What happens if we use 1e6 permutations?

R Session Info

```
sessionInfo()
```

```
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
```

```
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] liger_1.0
##
## loaded via a namespace (and not attached):
##  [1] compiler_3.6.0  magrittr_1.5    tools_3.6.0     htmltools_0.4.0
##  [5] yaml_2.2.1      Rcpp_1.0.4.6    stringi_1.4.6   rmarkdown_2.2
##  [9] knitr_1.28      stringr_1.4.0   xfun_0.14       digest_0.6.25
## [13] rlang_0.4.6     evaluate_0.14
```