DANA-FARBER CANCER INSTITUTE

HARVARD T.H. CHAN SCHOOL OF PUBLIC HEALTH

Reproducible and replicable comparisons of methods controlling false discoveries in computational biology

Patrick Kimes, PhD
Postdoctoral Fellow
Dana-Farber Cancer Institute
Harvard TH Chan School of Public Health

BIRS/CMO
November 7, 2018

DANA-FARBER CANCER INSTITUTE

HARVARD T.H. CHAN SCHOOL OF PUBLIC HEALTH

Reproducible and replicable comparisons of methods controlling false discoveries in computational biology

Patrick Kimes, PhD
Postdoctoral Fellow
Dana-Farber Cancer Institute
Harvard TH Chan School of Public Health

Kimes PK* and Reyes A*. (2018). Reproducible and replicable comparisons using *SummarizedBenchmark*. Bioinformatics.

BIRS/CMO
November 7, 2018

DANA-FARBER CANCER INSTITUTE

HARVARD T.H. CHAN SCHOOL OF PUBLIC HEALTH

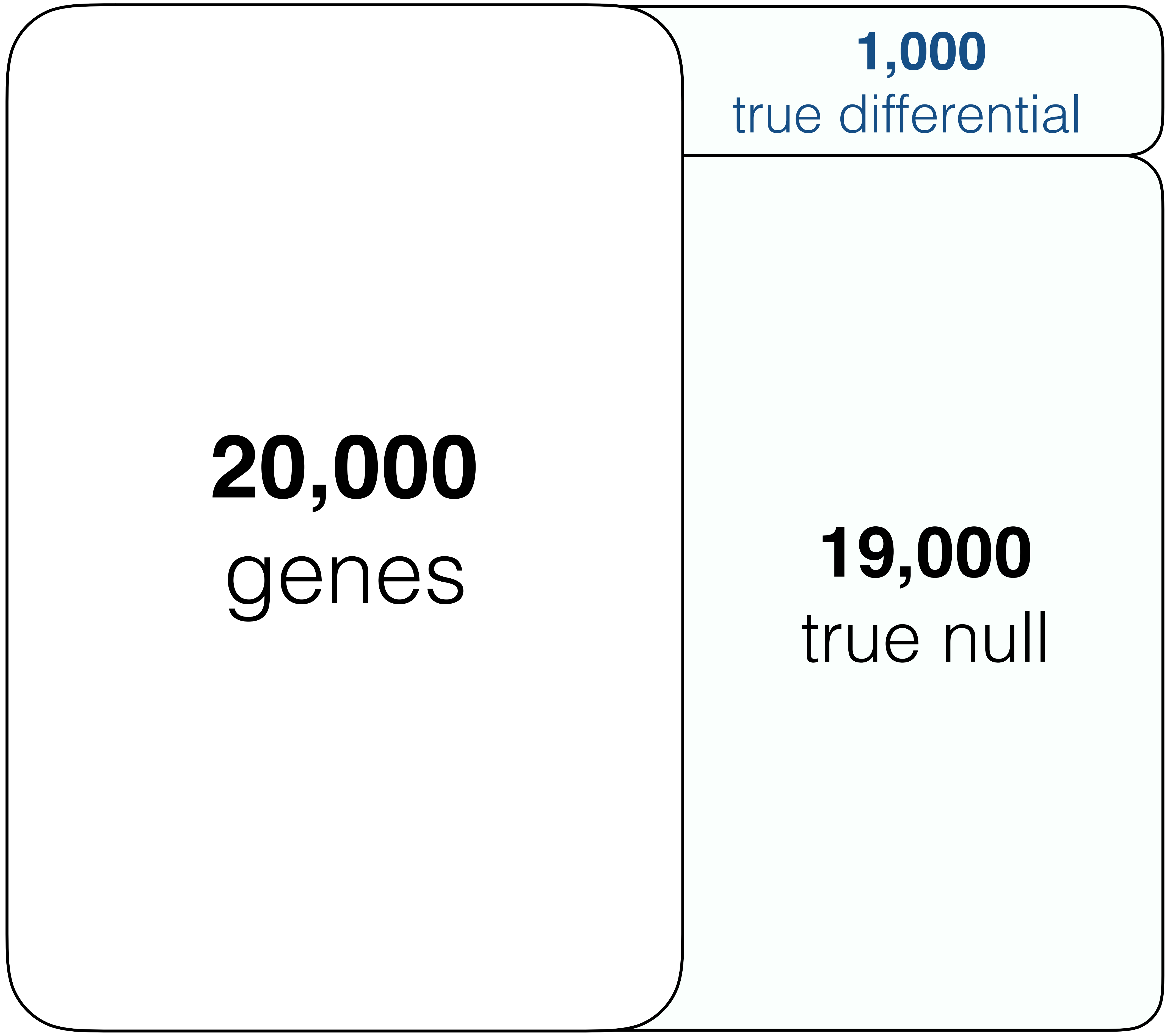
Reproducible and replicable comparisons of methods controlling false discoveries in computational biology

Patrick Kimes, PhD
Postdoctoral Fellow
Dana-Farber Cancer Institute
Harvard TH Chan School of Public Health

Korthauer K*, Kimes PK*, …, Hicks SC. (2018). A practical guide to methods controlling false discoveries in computational biology. bioRxiv.

BIRS/CMO
November 7, 2018

The problem of multiple hypothesis testing

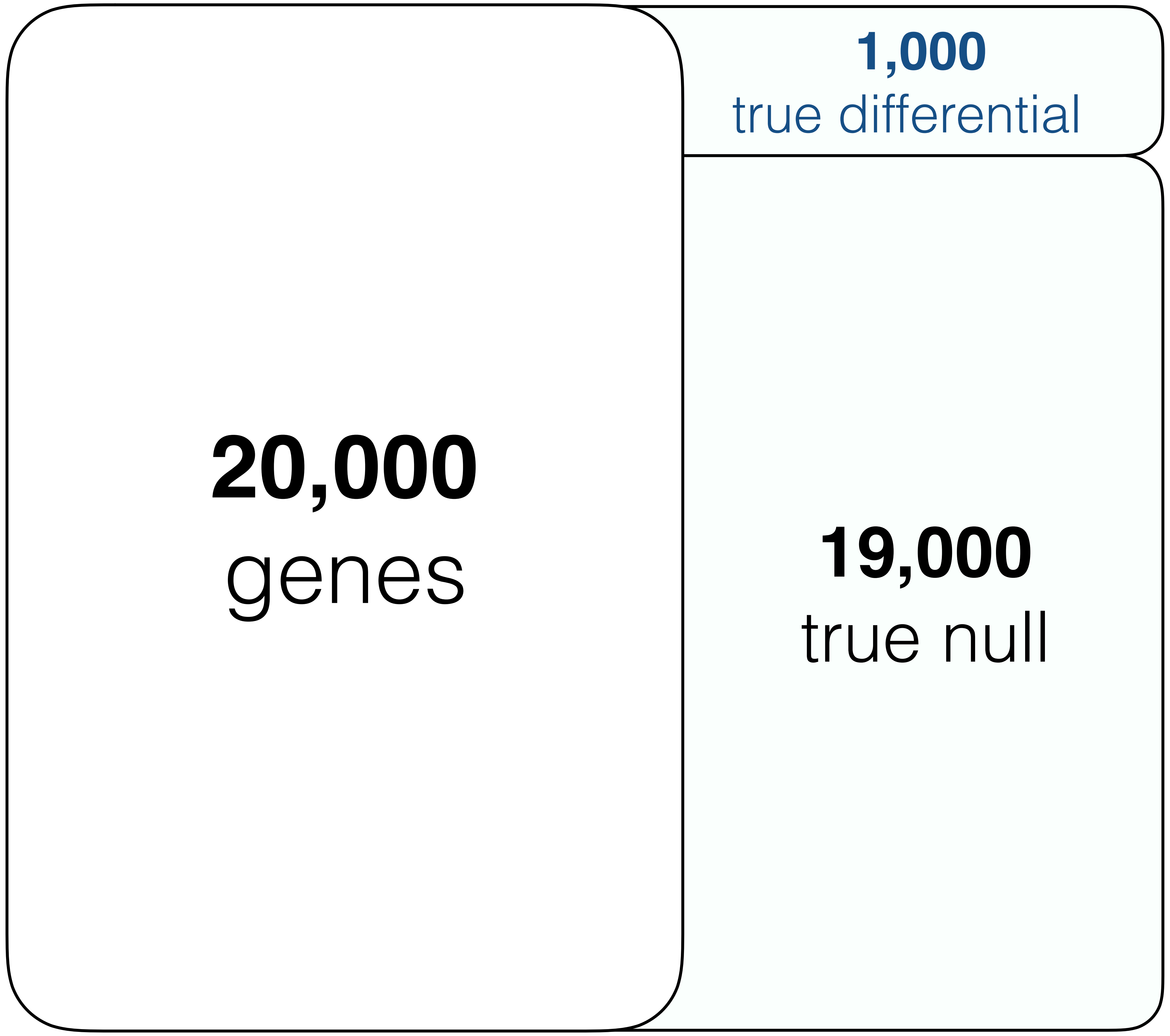


20,000
genes

1,000
true differential

19,000
true null

The problem of multiple hypothesis testing

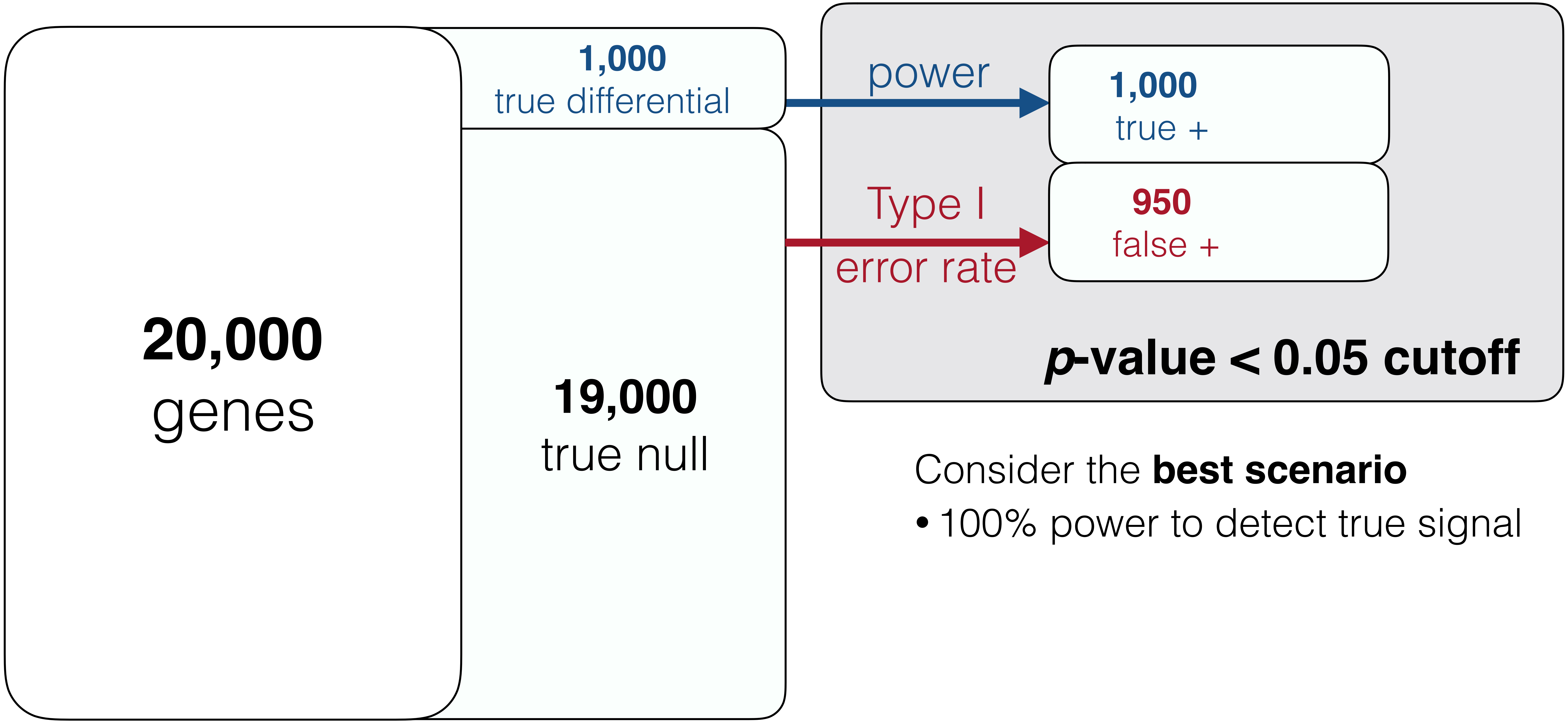


***p*-value < 0.05 cutoff**

Consider the **best scenario**

- 100% power to detect true signal

The problem of multiple hypothesis testing



20,000
genes

1,000
true differential

19,000
true null

power

Type I
error rate

1,000
true +

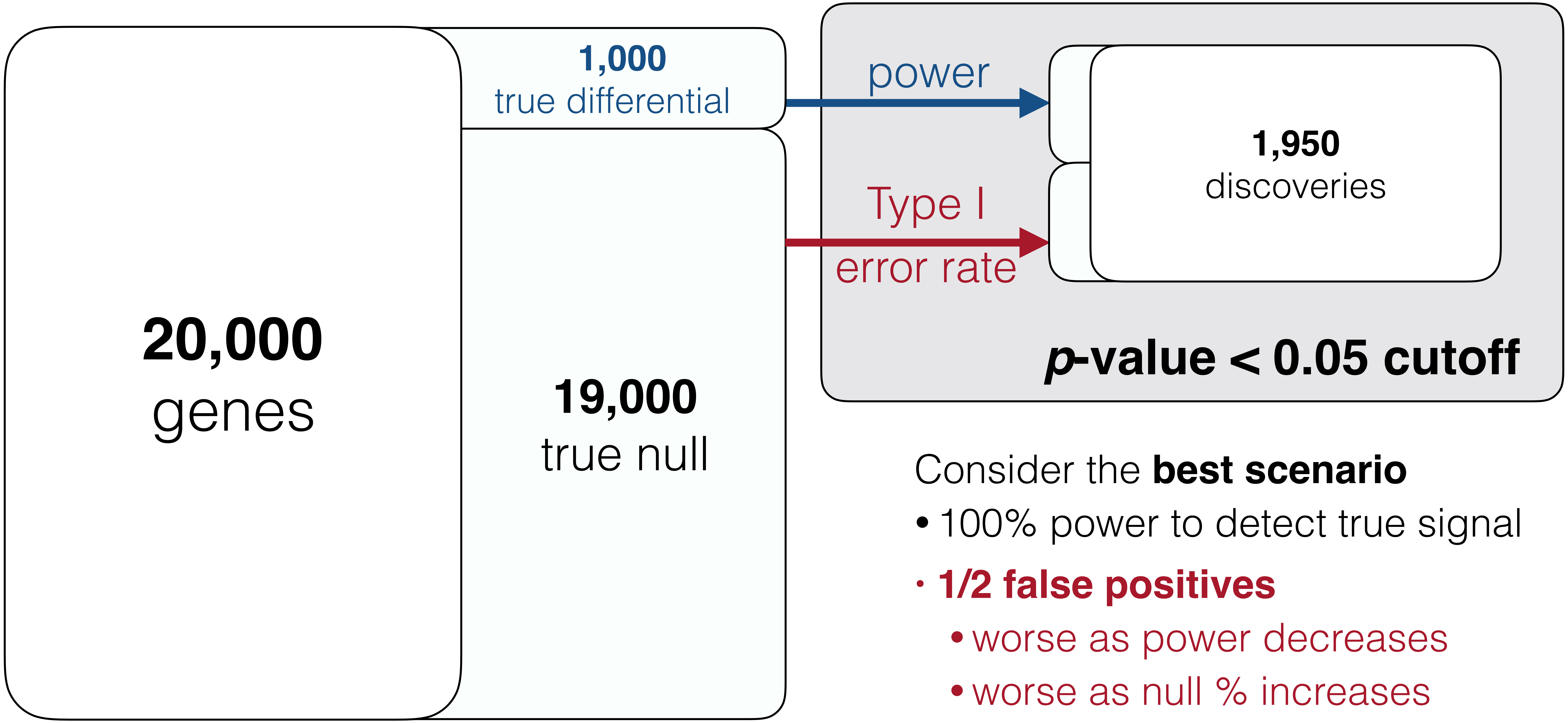
950
false +

***p*-value < 0.05 cutoff**

Consider the **best scenario**

- 100% power to detect true signal

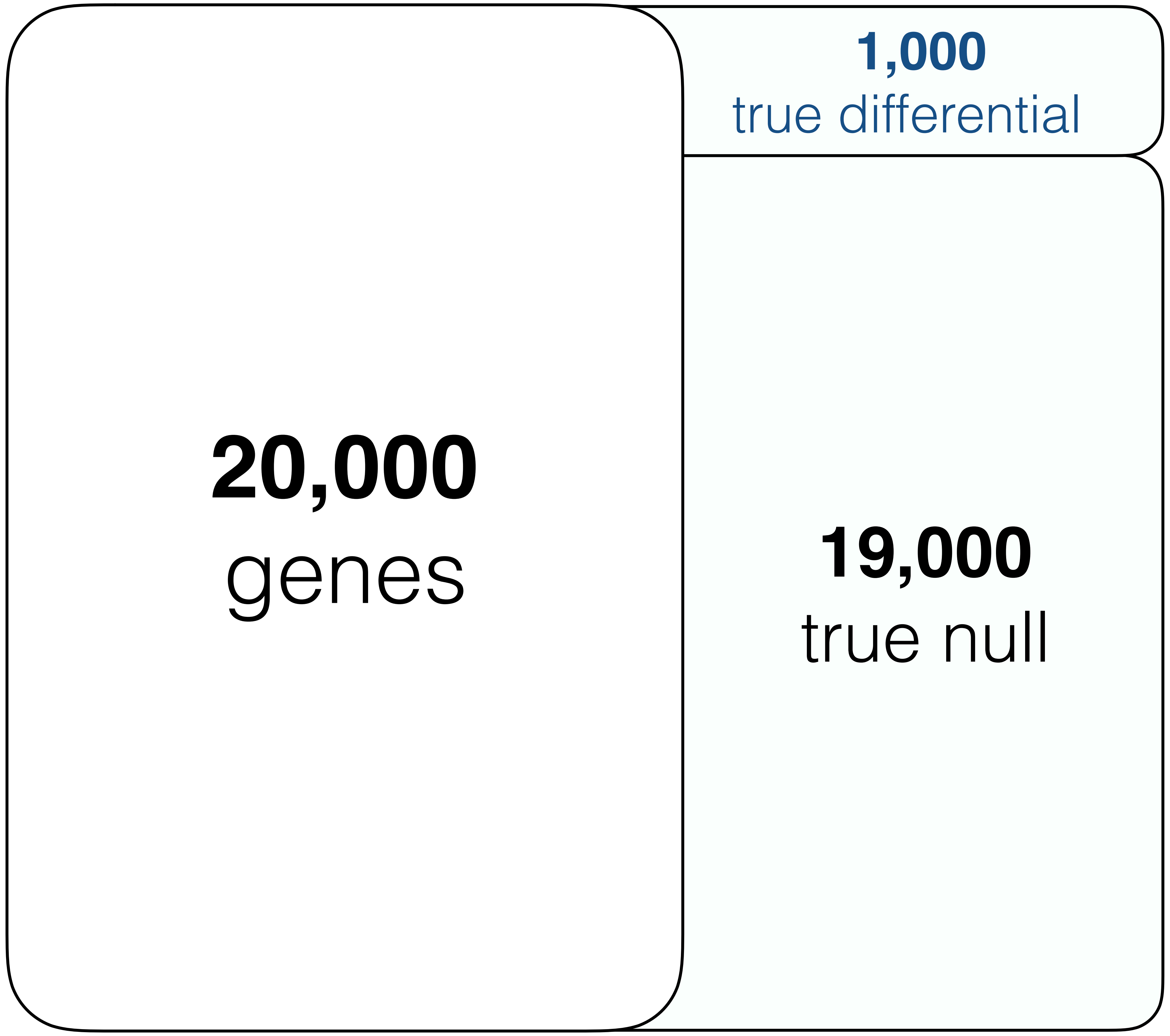
The problem of multiple hypothesis testing



Consider the **best scenario**

- 100% power to detect true signal
- **1/2 false positives**
  - worse as power decreases
  - worse as null % increases

Controlling false positives



20,000
genes

1,000
true differential

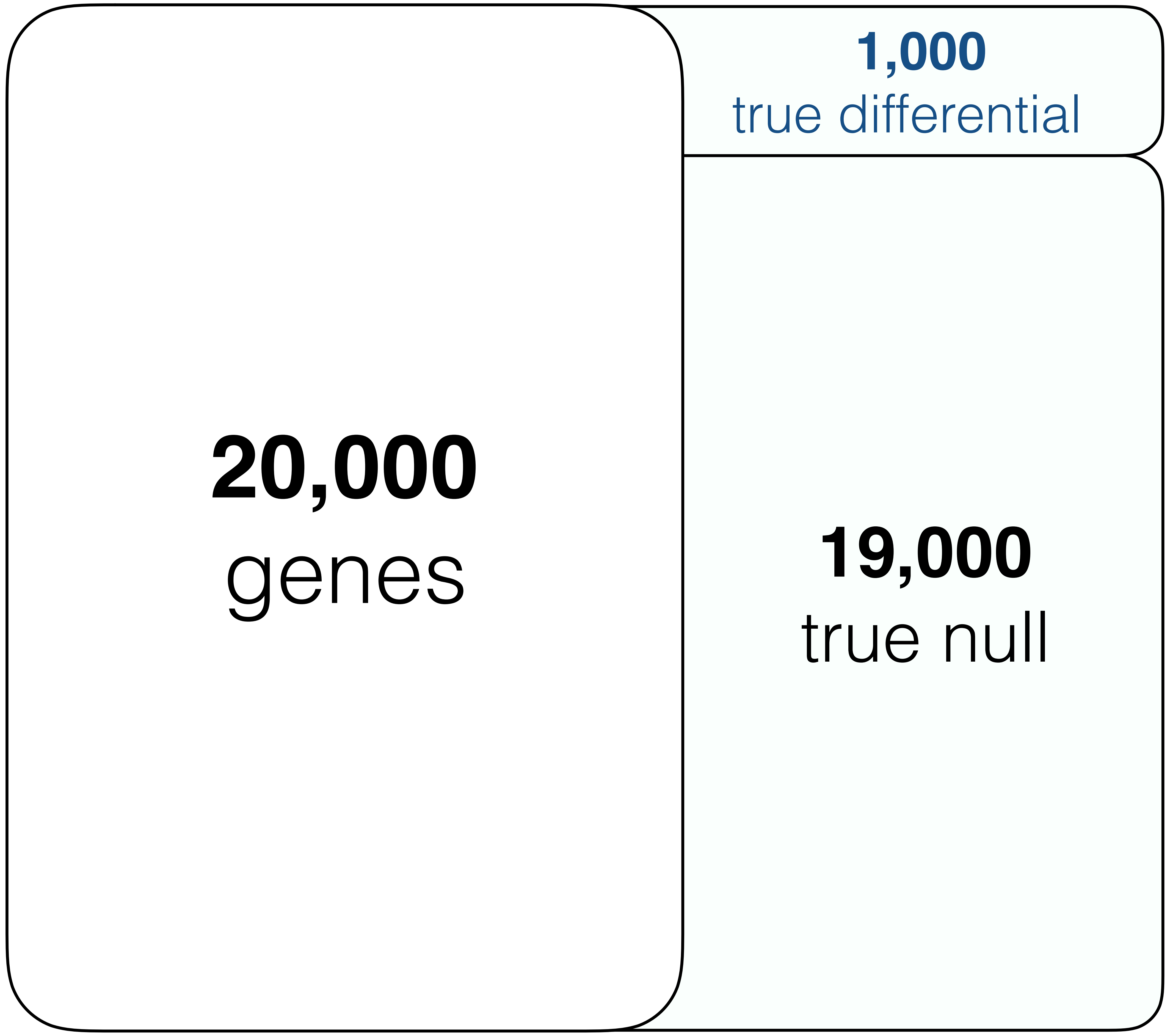
19,000
true null

Family-wise Error Rate (FWER)

- *Bonferroni correction*

$$P(\text{ at least 1 false positive }) < \alpha$$

Controlling false positives



Family-wise Error Rate (FWER)

- *Bonferroni correction*

$$\mathbf{P(}\text{ at least 1 false positive }\mathbf{)} < \alpha$$

False Discovery Rate (FDR)

- *Benjamini-Hochberg (BH) procedure*
- *Storey's q-value*

$$\mathbf{E}\left(\frac{\#\text{ false positives}}{\#\text{ total positives}}\right) < \alpha$$

Moving beyond BH and Storey’s *q*-value

BH and *q*-value

- all tests treated equal

Moving beyond BH and Storey's *q*-value

BH and *q*-value

- all tests treated equal

Reality

- all tests **not** equal
  - **eQTL** *cis* vs. *trans*
  - **RNA-seq** read depth

Moving beyond BH and Storey's *q*-value

BH and *q*-value

- all tests treated equal

Reality

- all tests **not** equal
  - **eQTL** *cis* vs. *trans*
  - **RNA-seq** read depth

Covariate-aware methods

- model differences in tests via covariates
- recent explosion of methods

Moving beyond BH and Storey's *q*-value

BH and *q*-value

- all tests treated equal

Reality

- all tests **not** equal
  - **eQTL** *cis* vs. *trans*
  - **RNA-seq** read depth

Covariate-aware methods

- model differences in tests via covariates
- recent explosion of methods

1995	BH procedure
2001	Storey's *q*-value
2009	conditional local FDR (**LFDR**)
2015	FDR regression (**FDRreg**)
2016	Independent Hypothesis Weighting (**IHW**)
2017	Adaptive Shrinkage (**ASH**)
2018	Boca-Leek (**BL**)
	Adaptive *p*-value Thresholding (**AdaPT**)

Understanding covariate-aware methods for FDR control

consider the two-groups model

probability of
test being null

$$p_i \sim \pi_0 f_0 + (1-\pi_0) f_1$$

distribution
under **null**
(uniform)

distribution
under **alternative**

Understanding covariate-aware methods for FDR control

consider the two-groups model

classic methods

$$p_i \sim \pi_0 f_0 + (1\text{-}\pi_0) f_1$$

- BH procedure
- Storey's *q*-value

Understanding covariate-aware methods for FDR control

consider the two-groups model

classic methods

$$p_i \sim \pi_0 f_0 + (1\text{-}\pi_0) f_1$$

- BH procedure
- Storey's *q*-value

covariate-aware methods

$$p_i \,\big|\, x_i \sim \pi_{0(x_i)} f_0 + (1\text{-}\pi_{0(x_i)}) f_{1(x_i)}$$

Understanding covariate-aware methods for FDR control

consider the two-groups model

classic methods

$$p_i \sim \pi_0 f_0 + (1\text{-}\pi_0) f_1$$

- BH procedure
- Storey's *q*-value

covariate-aware methods

$$p_i \,\Big|\, x_i \sim \pi_{0(x_i)} f_0 + (1\text{-}\pi_{0(x_i)}) f_{1(x_i)}$$

eQTL
cis/trans

RNA-seq
read depth

Understanding covariate-aware methods for FDR control

consider the two-groups model

classic methods

$$p_i \sim \pi_0 f_0 + (1\text{-}\pi_0) f_1$$

- BH procedure
- Storey’s *q*-value

covariate-aware methods

$$p_i \,\big|\, x_i \sim \pi_0(x_i) f_0 + (1\text{-}\pi_0(x_i)) f_1(x_i)$$

eQTL
cis/trans

RNA-seq
read depth

• IHW	●	●
• BL	●	
• LFDR	●	●
• AdaPT	●	●
• FDRreg*	●	
• ASH*		●

Understanding covariate-aware methods for FDR control

consider the two-groups model

classic methods

$$p_i \sim \pi_0 f_0 + (1\text{-}\pi_0) f_1$$

- BH procedure
- Storey's *q*-value

covariate-aware methods

$$p_i \,\middle|\, x_i \sim \pi_0(x_i) f_0 + (1\text{-}\pi_0(x_i)) f_1(x_i)$$

● eQTL *cis/trans*

● RNA-seq read depth

- IHW ● ●
- BL ●
- LFDR ● ●
- AdaPT ● ●
- FDRreg* ●
- ASH* ●

Understanding covariate-aware methods for FDR control

consider the two-groups model

classic methods

$$p_i \sim \pi_0 f_0 + (1\text{-}\pi_0) f_1$$

- BH procedure
- Storey's *q*-value

covariate-aware methods

$$\cancel{p_i \mid x_i} \sim \pi_0(x_i) f_0 + (1\text{-}\pi_0(x_i)) f_1(x_i)$$

$$z_i \mid x_i$$

• IHW	●	●
• BL	●	
• LFDR	●	●
• AdaPT	●	●
• FDRreg*	●	
• ASH*		●

Understanding covariate-aware methods for FDR control

consider the two-groups model

classic methods

$$p_i \sim \pi_0 f_0 + (1\text{-}\pi_0) f_1$$

- BH procedure
- Storey's *q*-value

covariate-aware methods

$$\cancel{p_i \,|\, x_i} \sim \pi_{0(x_i)} f_0 + (1\text{-}\pi_{0(x_i)}) f_{1(x_i)}$$

$$\hat{\beta}_i \,|\, \hat{s}_i$$

- IHW 🔴 🔵
- BL 🔴
- LFDR 🔴 🔵
- AdaPT 🔴 🔵
- FDRreg* 🔴
- ASH* 🔵

Benchmarking for practical recommendations

- BH procedure
- Storey's *q*-value

- IHW
- BL
- AdaPT
- LFDR
- FDRreg
- ASH

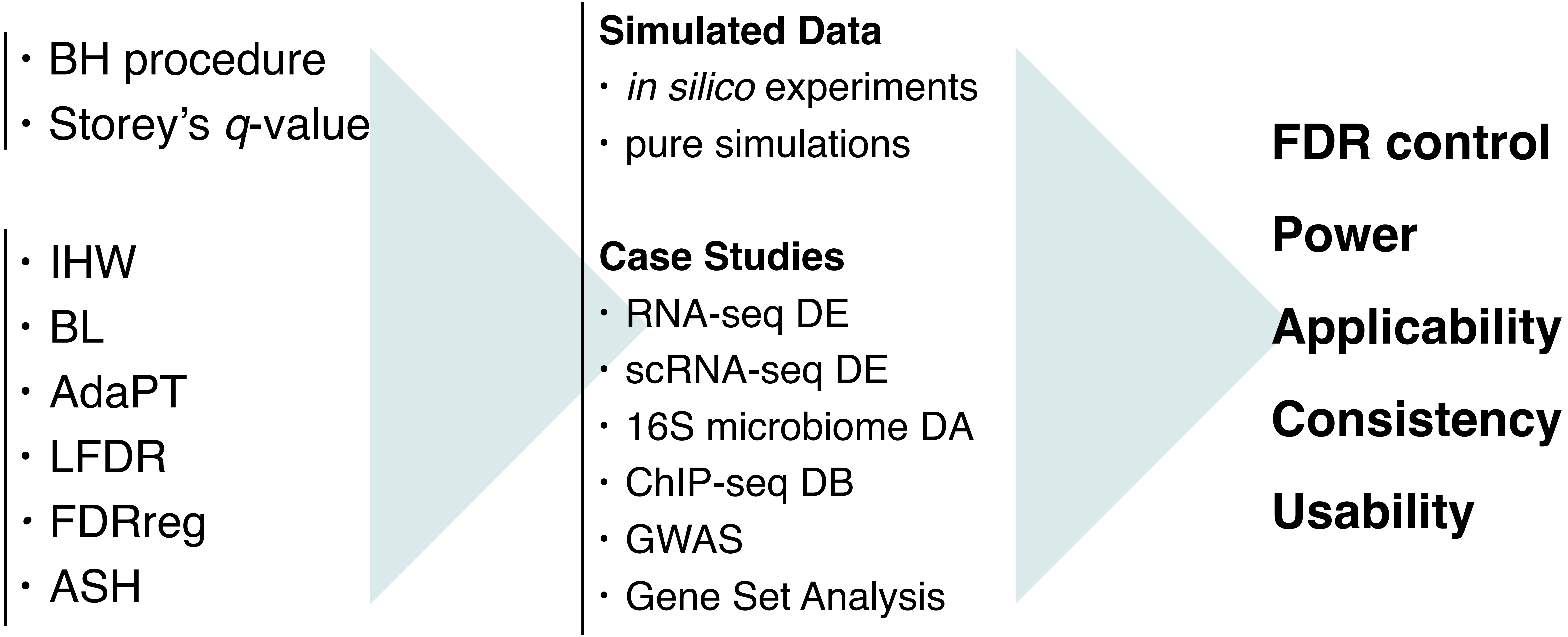
Simulated Data

- *in silico* experiments
- pure simulations

Case Studies

- RNA-seq DE
- scRNA-seq DE
- 16S microbiome DA
- ChIP-seq DB
- GWAS
- Gene Set Analysis

Benchmarking for practical recommendations



- BH procedure
- Storey's *q*-value

- IHW
- BL
- AdaPT
- LFDR
- FDRreg
- ASH

Simulated Data

- *in silico* experiments
- pure simulations

Case Studies

- RNA-seq DE
- scRNA-seq DE
- 16S microbiome DA
- ChIP-seq DB
- GWAS
- Gene Set Analysis

FDR control

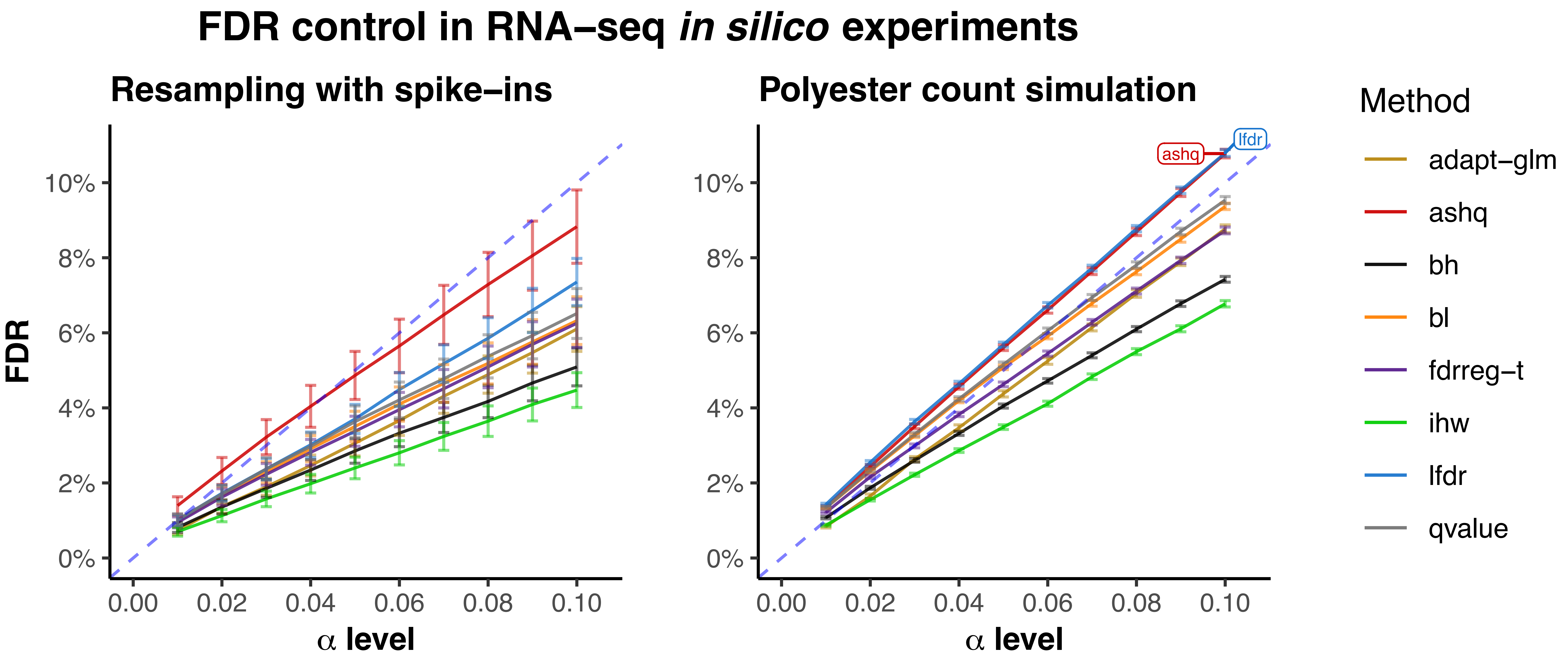
Power

Applicability

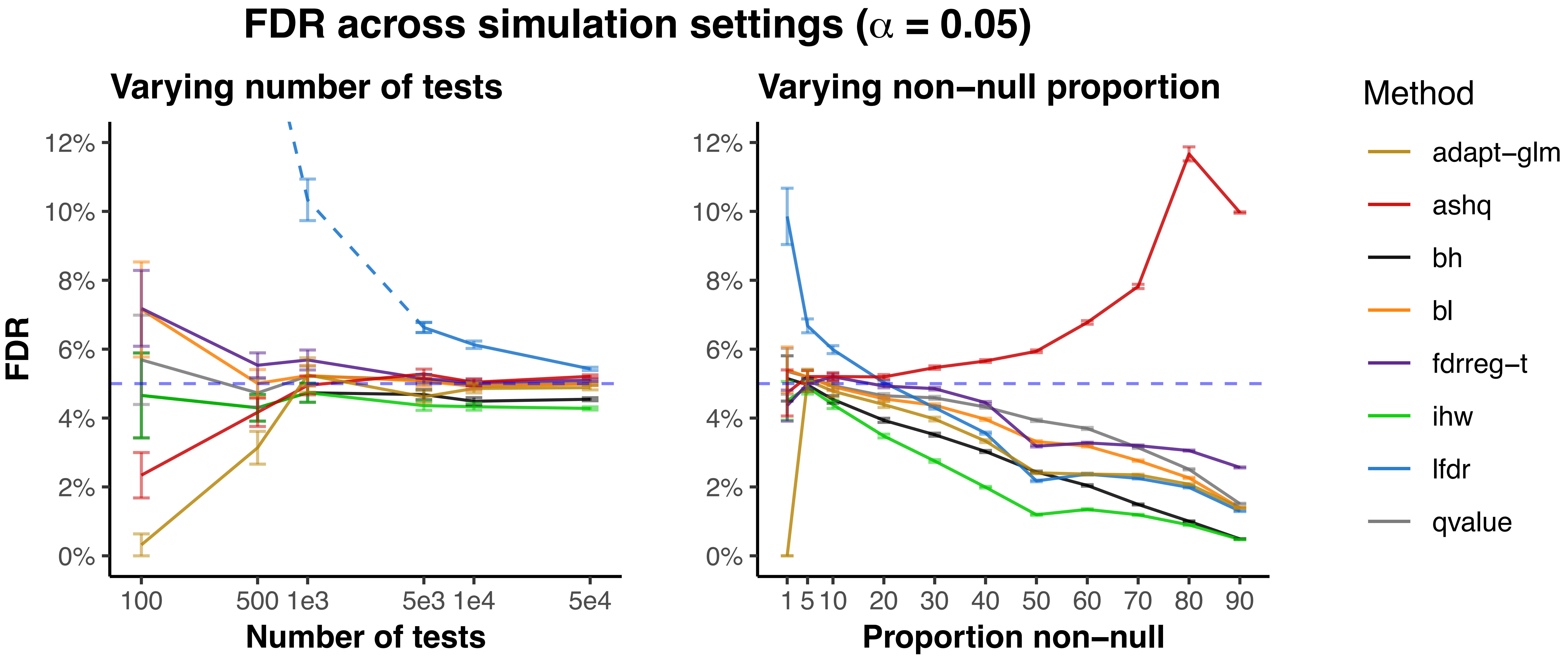
Consistency

Usability

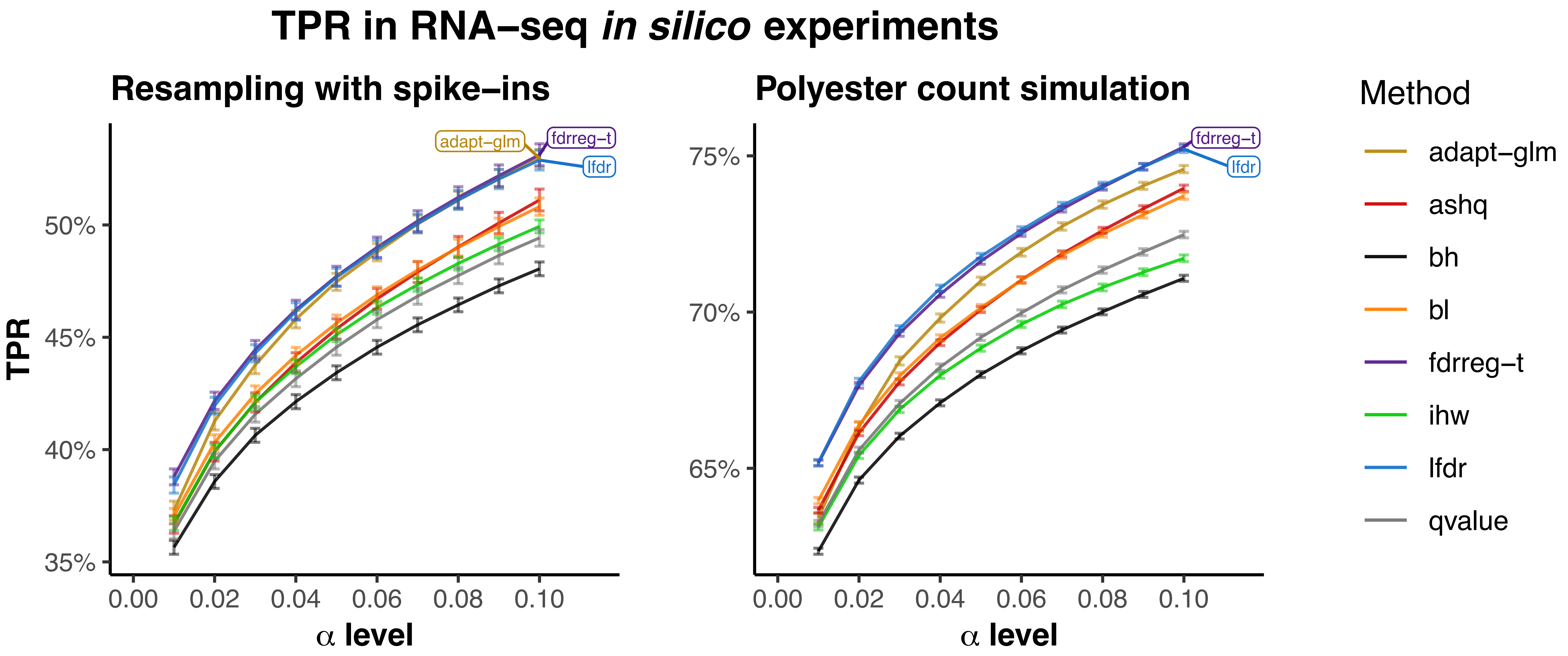
Most covariate-aware methods control FDR



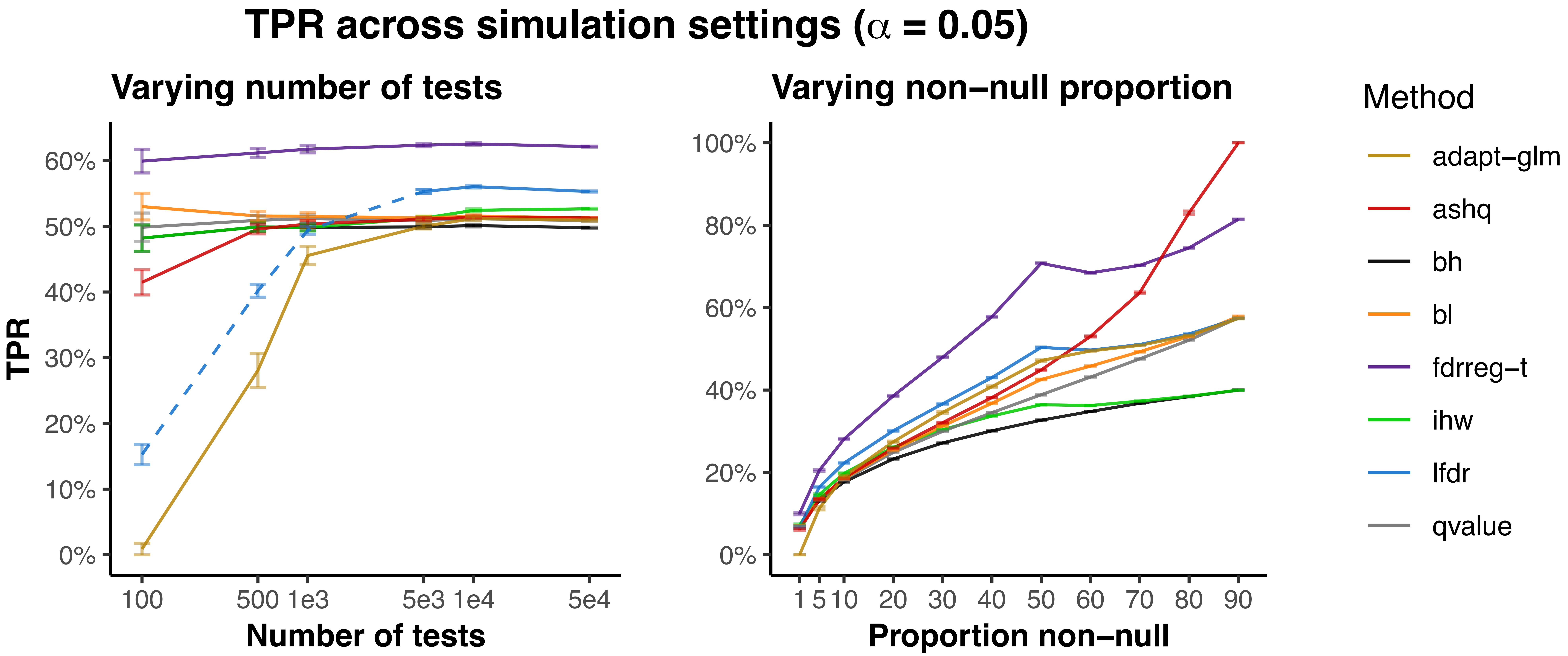
Some methods were sensitive to number of tests, null proportion



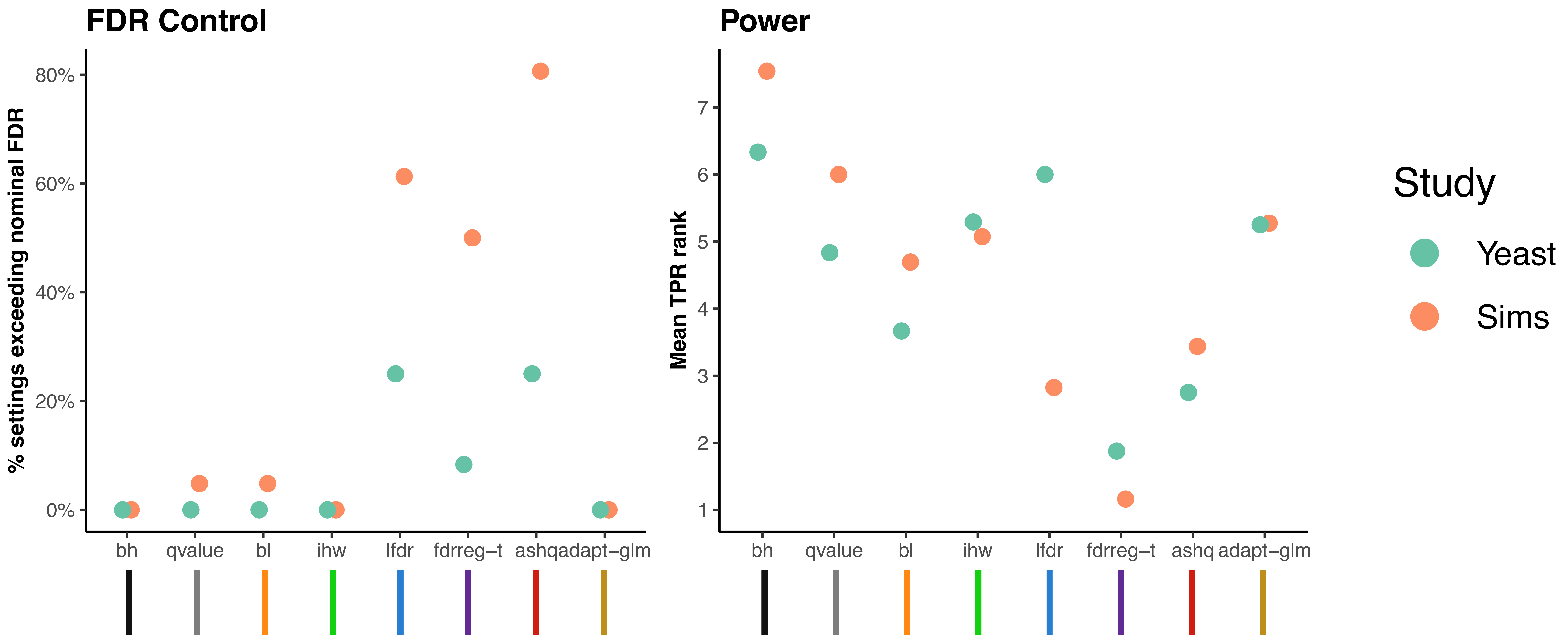
Covariate-aware methods were modestly more powerful



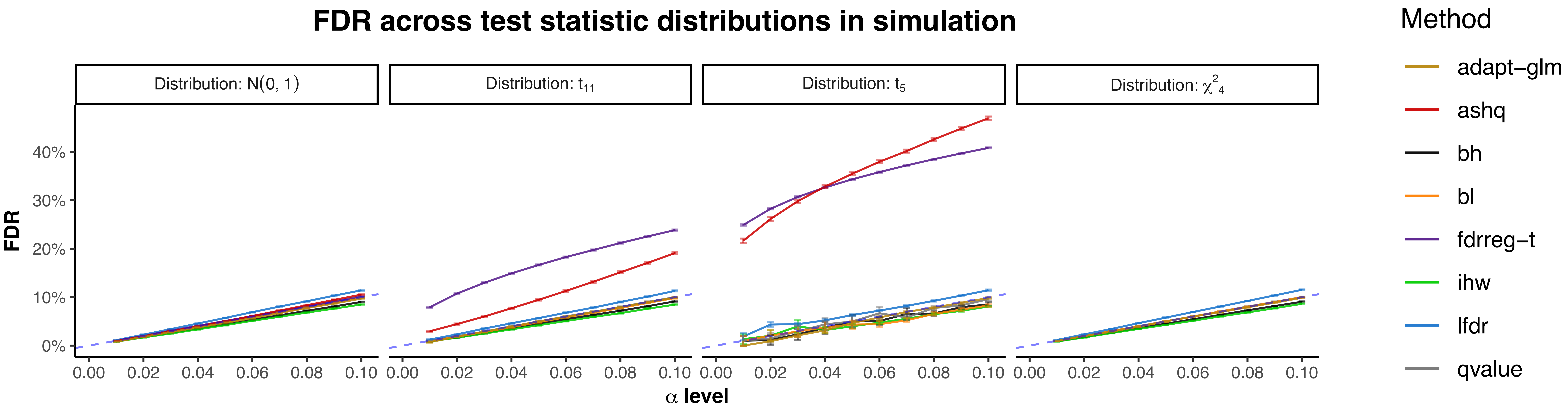
Some methods were sensitive to number of tests, null proportion



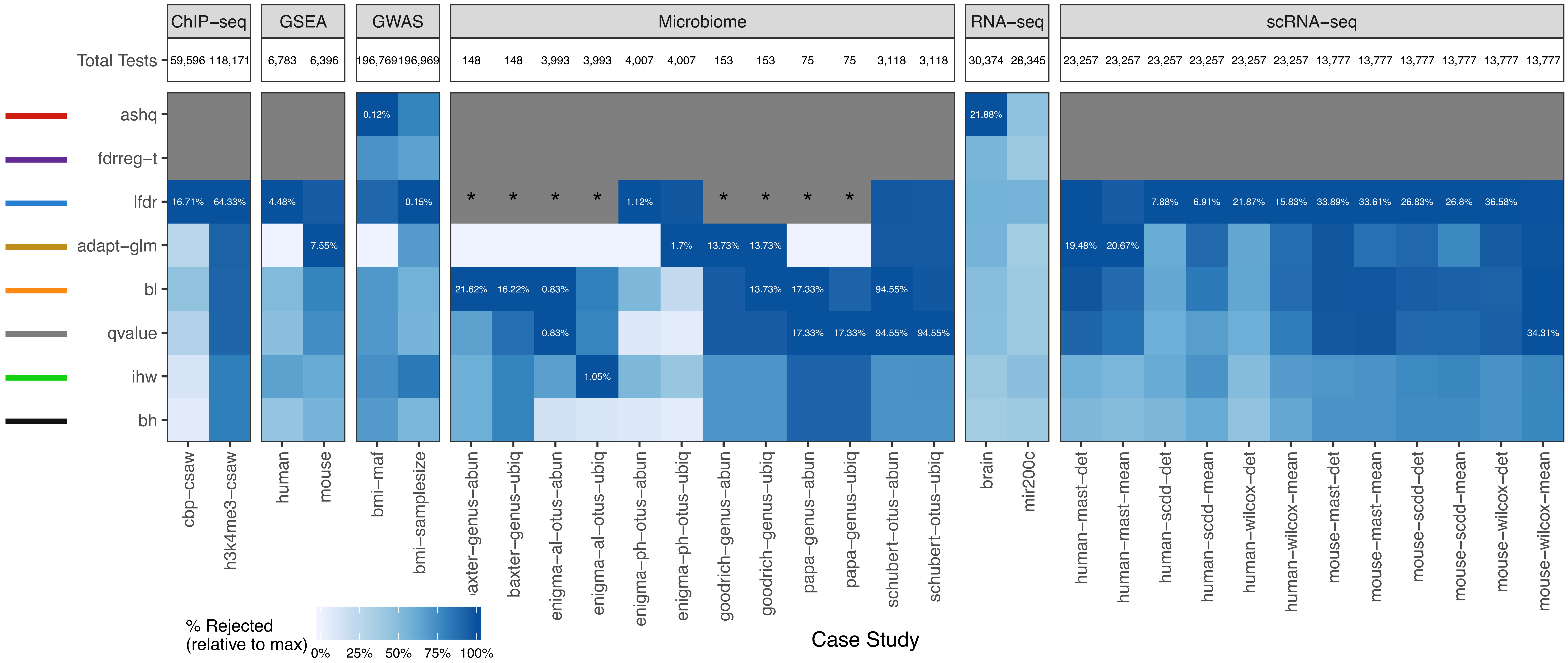
Summarizing FDR control and power across simulations



Some methods were sensitive to the test statistic



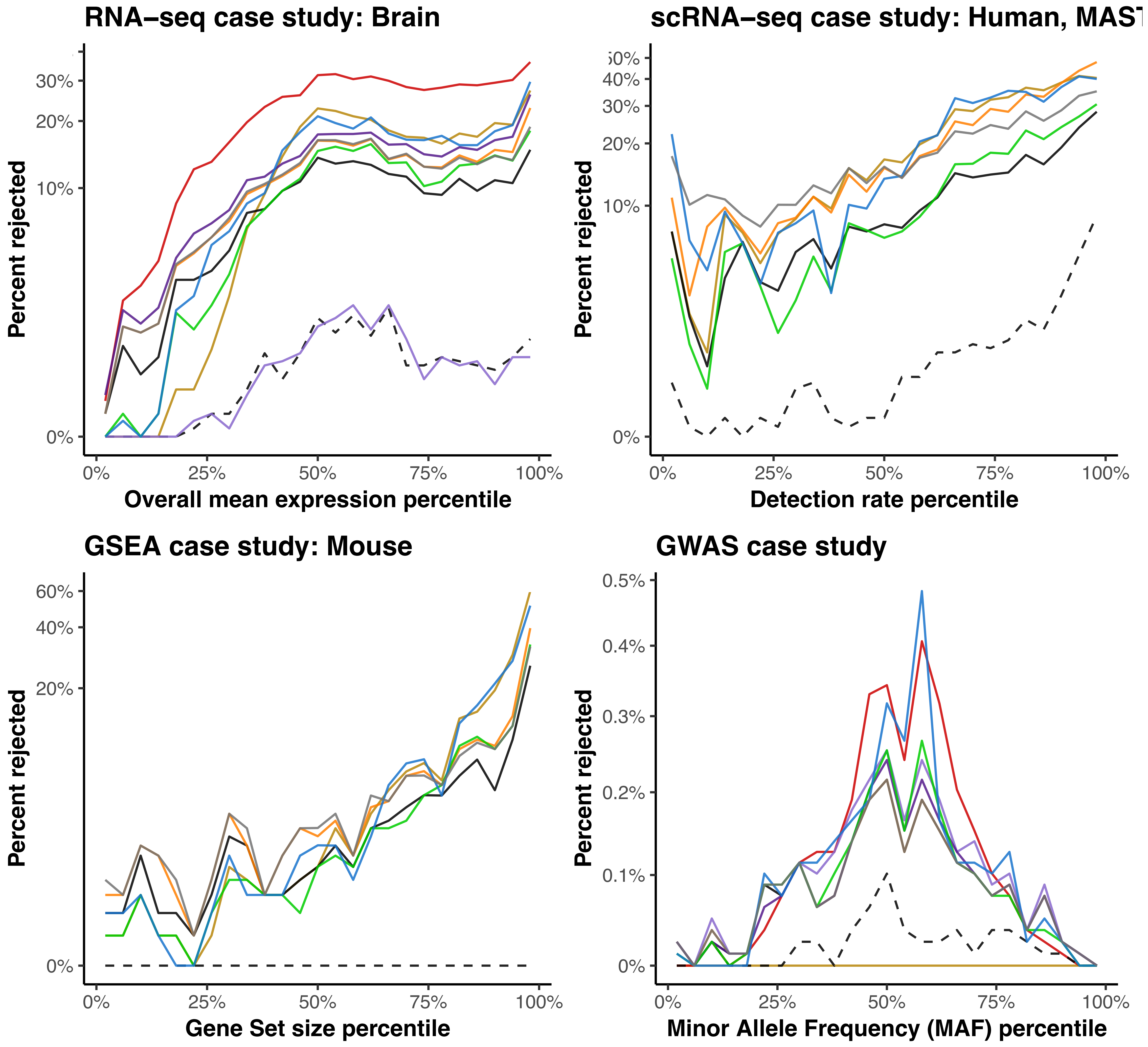
Not all methods could be applied to all case studies



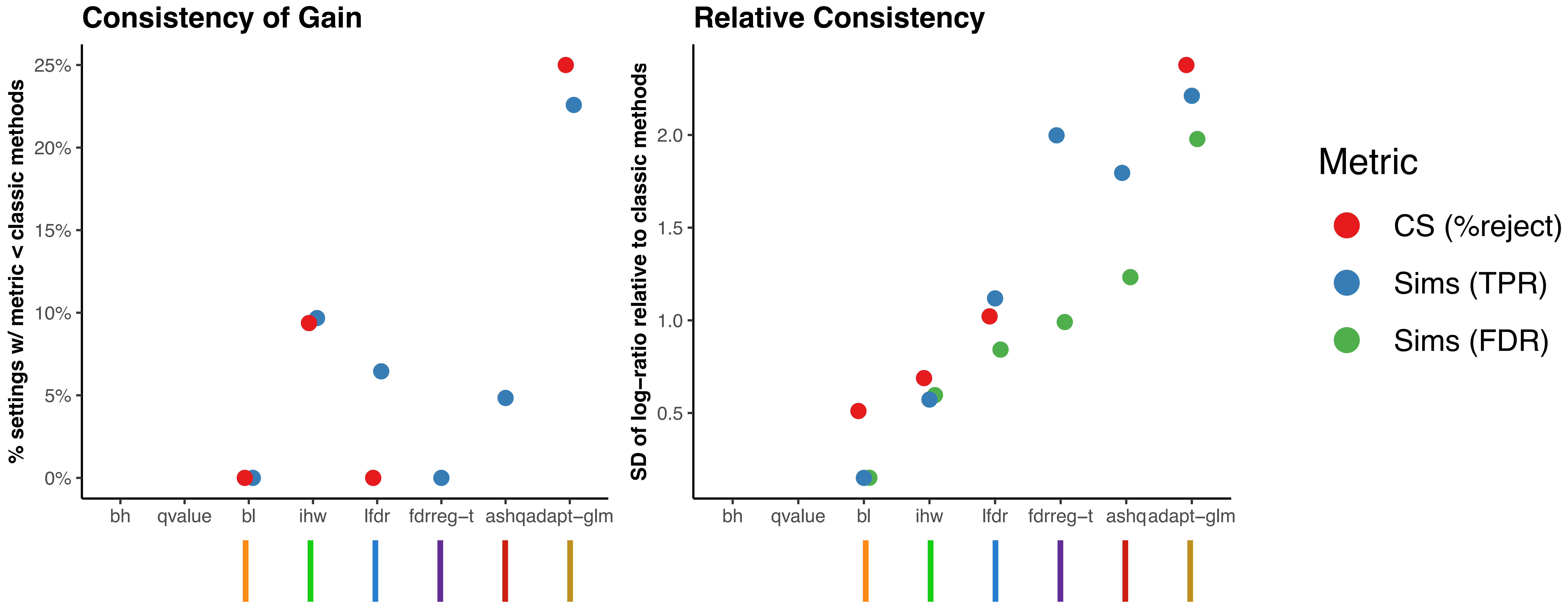
Number of rejections in case studies

Informative covariates in the case studies

Case Study	Covariate
Bulk RNA-seq	**mean gene expression**
Single Cell RNA-seq	mean non-zero gene expression, **detection rate**
Microbiome	mean non-zero abundance, ubiquity
ChIP-seq	mean read depth, window size
GWAS	**minor allele frequency**, sample size
Gene Set Analysis	**gene set size**

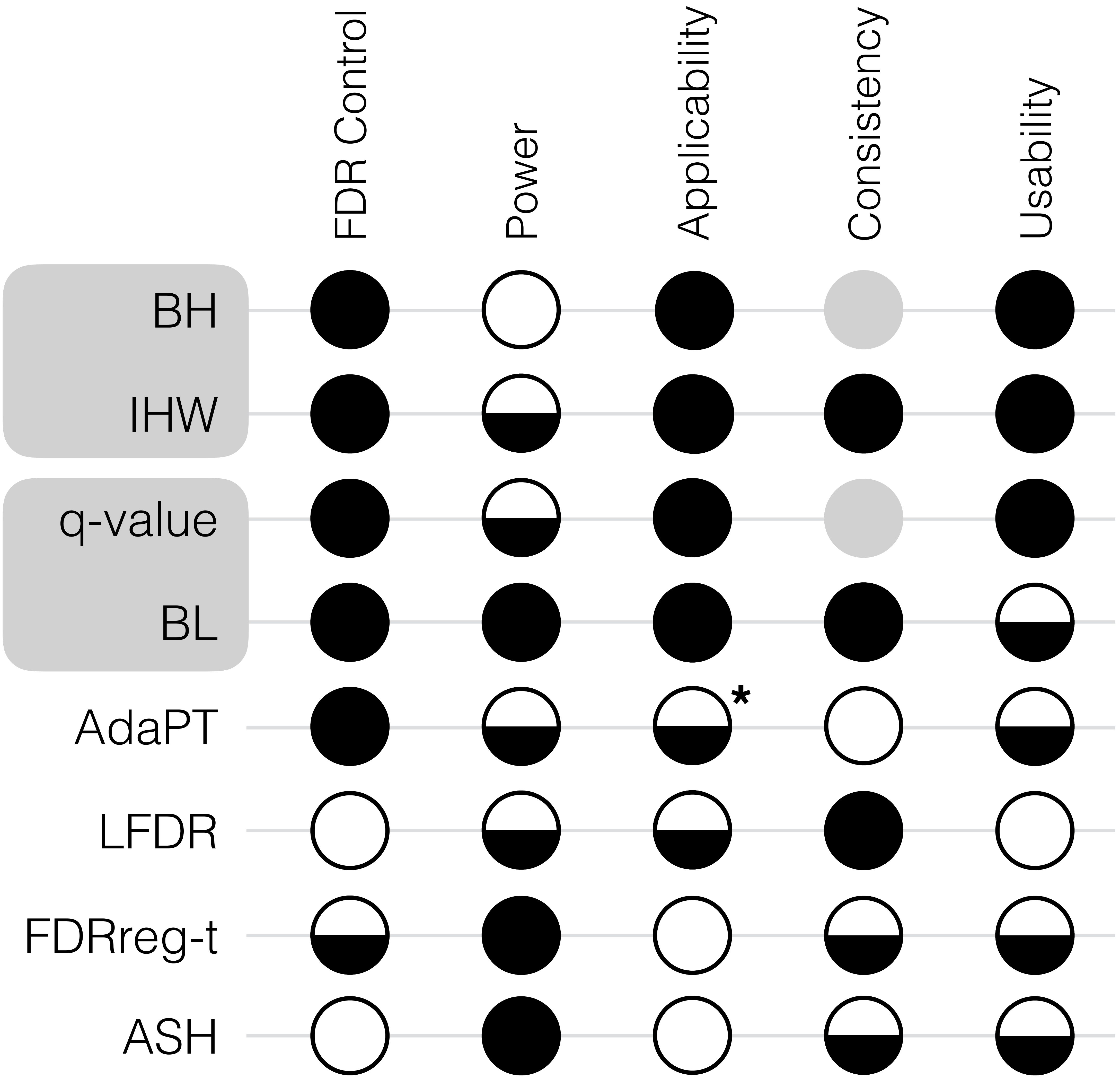


Gains relative to classic methods varied across methods



Takeaways

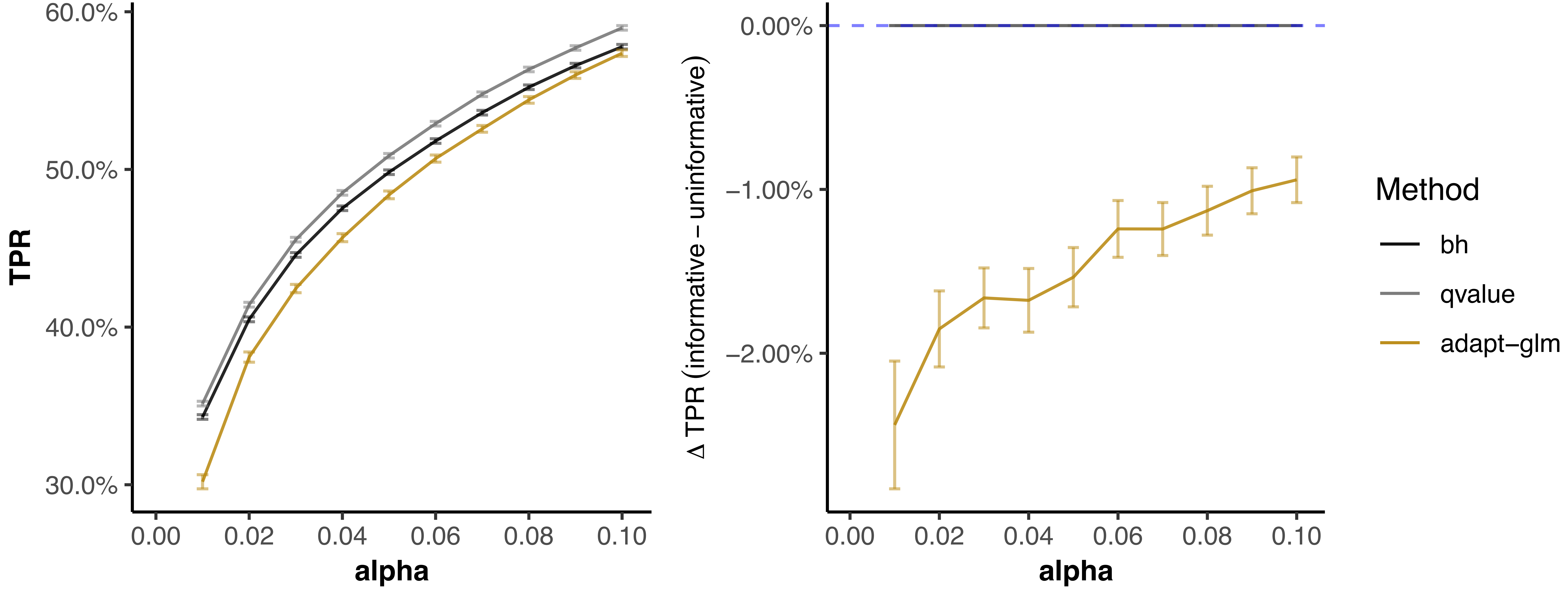
- Many covariate-aware methods provide consistent FDR control (IHW, BL, AdaPT)
- These covariate-aware methods typically provide modest gains in power
- Not all methods could be applied to all simulations and case studies (FDRreg, ASH)
- Some methods showed highly variable performance across simulations and case studies (AdaPT)
- **Not all R packages are created equal**



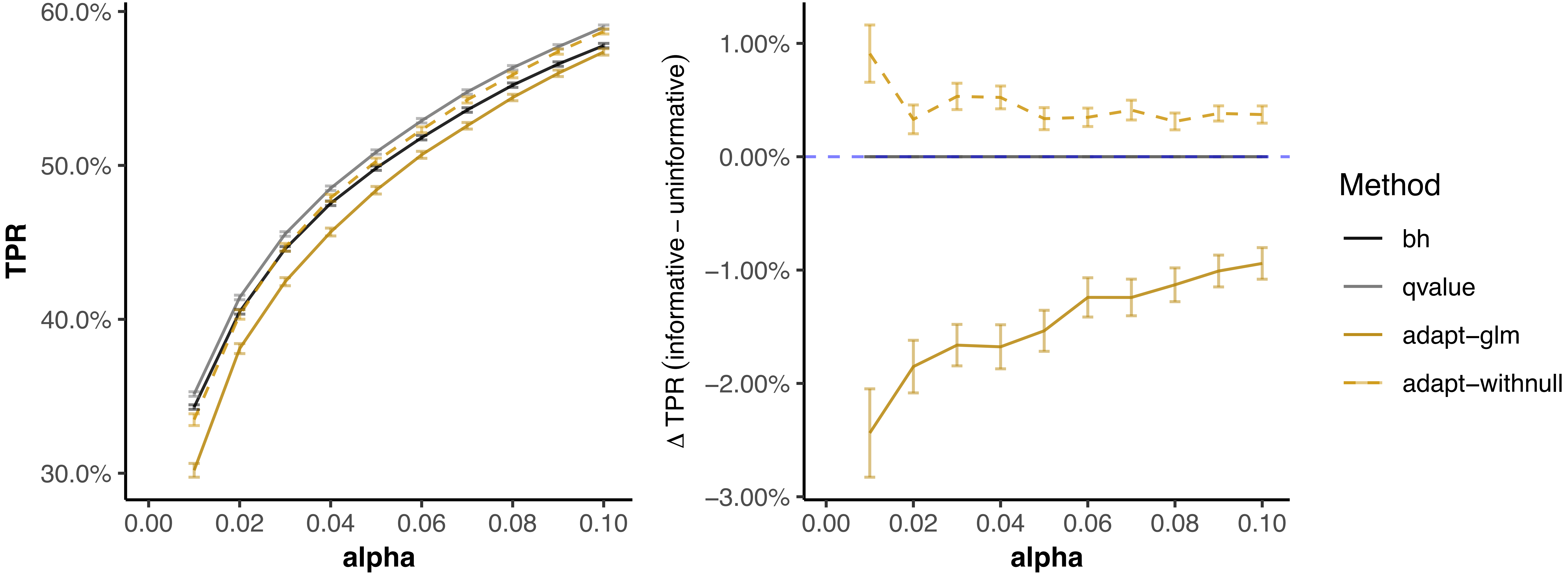
	FDR Control	Power	Applicability	Consistency	Usability
BH	●	○	●	◌	●
IHW	●	◒	●	●	●
q-value	●	◒	●	◌	●
BL	●	●	●	●	◒
AdaPT	●	◒	◒*	○	◒
LFDR	○	◒	◒	●	○
FDRreg-t	◒	●	○	◒	◒
ASH	○	●	○	◒	◒

Korthauer K*, Kimes PK*, …, Hicks SC. (2018). A practical guide to methods controlling false discoveries in computational biology. bioRxiv.

Benchmarking as a social exercise



Benchmarking as a social exercise



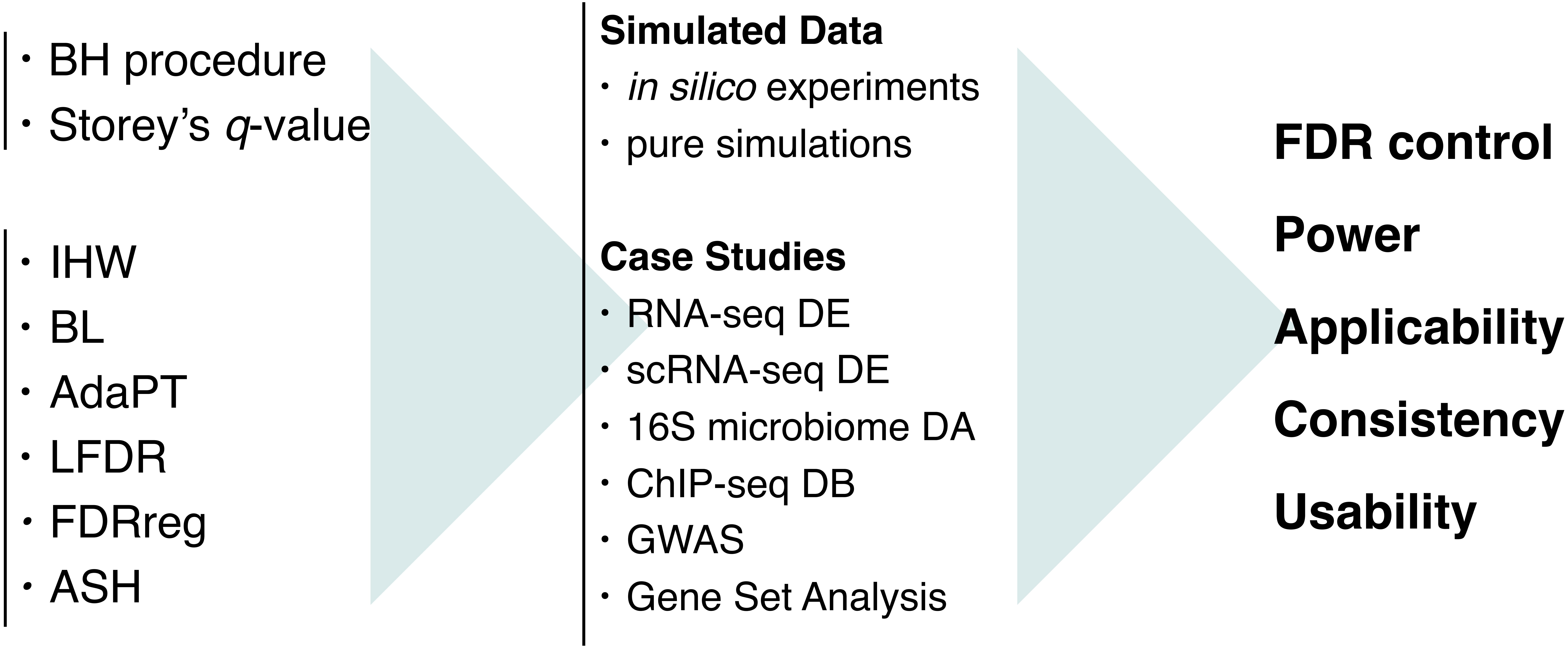
Reproducible and replicable comparisons of methods controlling false discoveries in computational biology

Patrick Kimes, PhD
Postdoctoral Fellow
Dana-Farber Cancer Institute
Harvard TH Chan School of Public Health

Kimes PK* and Reyes A*. (2018). Reproducible and replicable comparisons using *SummarizedBenchmark*. Bioinformatics.

BIRS/CMO
November 7, 2018

Recall the FDR benchmark setup



- BH procedure
- Storey's *q*-value

- IHW
- BL
- AdaPT
- LFDR
- FDRreg
- ASH

Simulated Data

- *in silico* experiments
- pure simulations

Case Studies

- RNA-seq DE
- scRNA-seq DE
- 16S microbiome DA
- ChIP-seq DB
- GWAS
- Gene Set Analysis

FDR control

Power

Applicability

Consistency

Usability

How the FDR benchmarking project started

```
fdr_methods <- function(dat) {

    ## keep adjusted p-values
    adj_pset <- list()

    ## Bonferroni
    adj_p <- p.adjust(dat$pval, "bonferroni")
    adj_pset$bonf <- adj_p

    ## BH
    adj_p <- p.adjust(dat$pval, "BH")
    adj_pset$bh <- adj_p

    ## qvalue (Storey)
    adj_p <- qvalue::qvalue(p=dat$pval)$qvalues
    adj_pset$qvalue <- adj_p

    . . .

    return(adj_pset)
}
```

dat → **fdr_methods()** → **adj_pset**

```
head(tdat, n = 3)
    H test_statistic effect_size         pval        SE
  1 1      -3.247964   -1.708222 4.465398e-03 0.5259363
  2 1      -2.453800   -1.039939 2.454995e-02 0.4238076
  3 1      -4.684693   -1.895645 1.845383e-04 0.4046467


p_table <- fdr_methods(tdat)
p_table
$bonf
 [1] 0.19489 1.00000 0.00103 1.00000 1.00000
 [6] 1.00000 0.05170 0.11135 1.00000 0.68348
 . . .

saveRDS(p_table, file = “my_p_table.rds”)
```

typical questions

how do we organize **data + results**?

what **parameters** did we use?

which **package version** did you use?

...

Problems with benchmarking computational methods

- simulation results are unstructured
    - *SummarizedBenchmark* class
- simulation code is unstructured
    - *BenchDesign* class
- code and results are disconnected
    - SummarizedBenchmark

```
head(tdat, n = 3)
    H test_statistic effect_size          pval        SE
 1  1      -3.247964   -1.708222 4.465398e-03 0.5259363
 2  1      -2.453800   -1.039939 2.454995e-02 0.4238076
 3  1      -4.684693   -1.895645 1.845383e-04 0.4046467


p_table <- fdr_methods(tdat)
p_table
$bonf
 [1] 0.19489 1.00000 0.00103 1.00000 1.00000
 [6] 1.00000 0.05170 0.11135 1.00000 0.68348
 . . .

saveRDS(p_table, file = “my_p_table.rds”)
```

typical questions

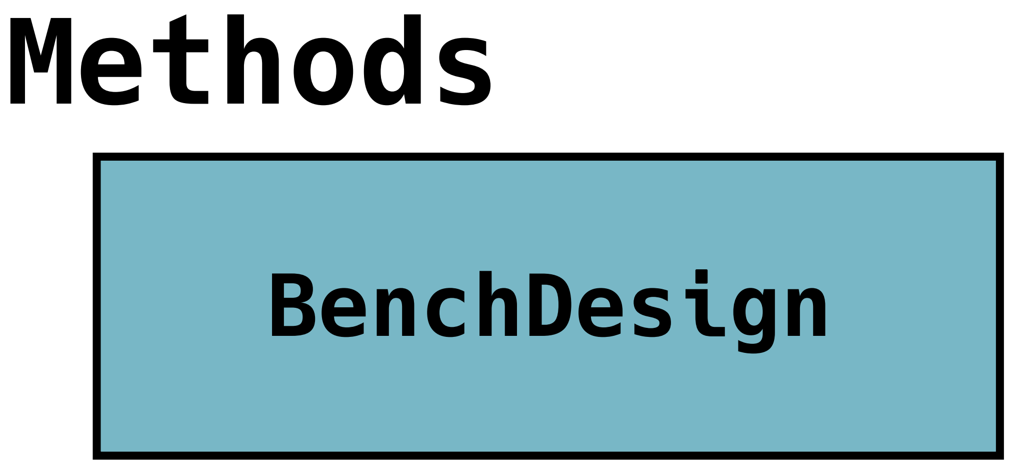
how do we organize **data + results**?

what **parameters** did we use?

which **package version** did you use?

...

SummarizedBenchmark framework


Methods

BenchDesign

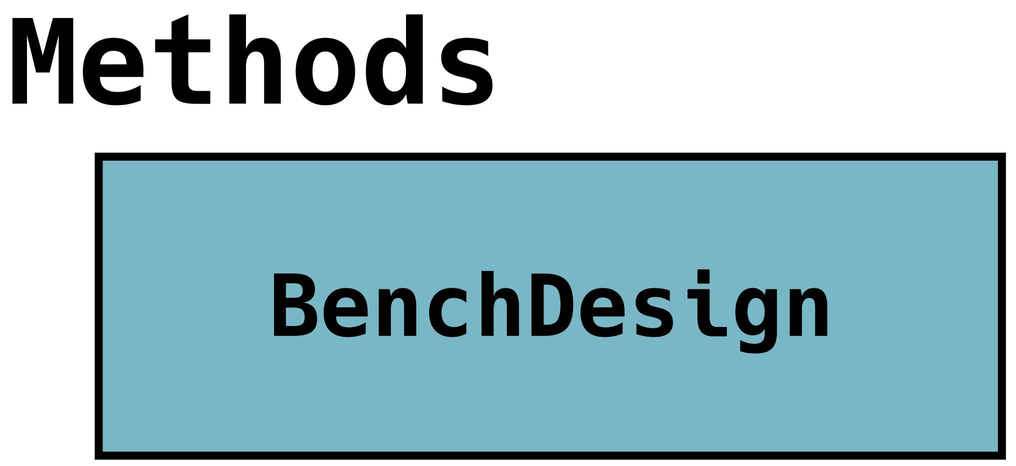
BenchDesign class

- collection of methods
  - function
  - map: data → function parameters

```
## regular function call
p.adjust(p = data$pval, method = “BH”)

## BenchDesign format
BDMethod(
  x = p.adjust,
  params = quos(p = pval, method = “BH”)
)
```

SummarizedBenchmark framework

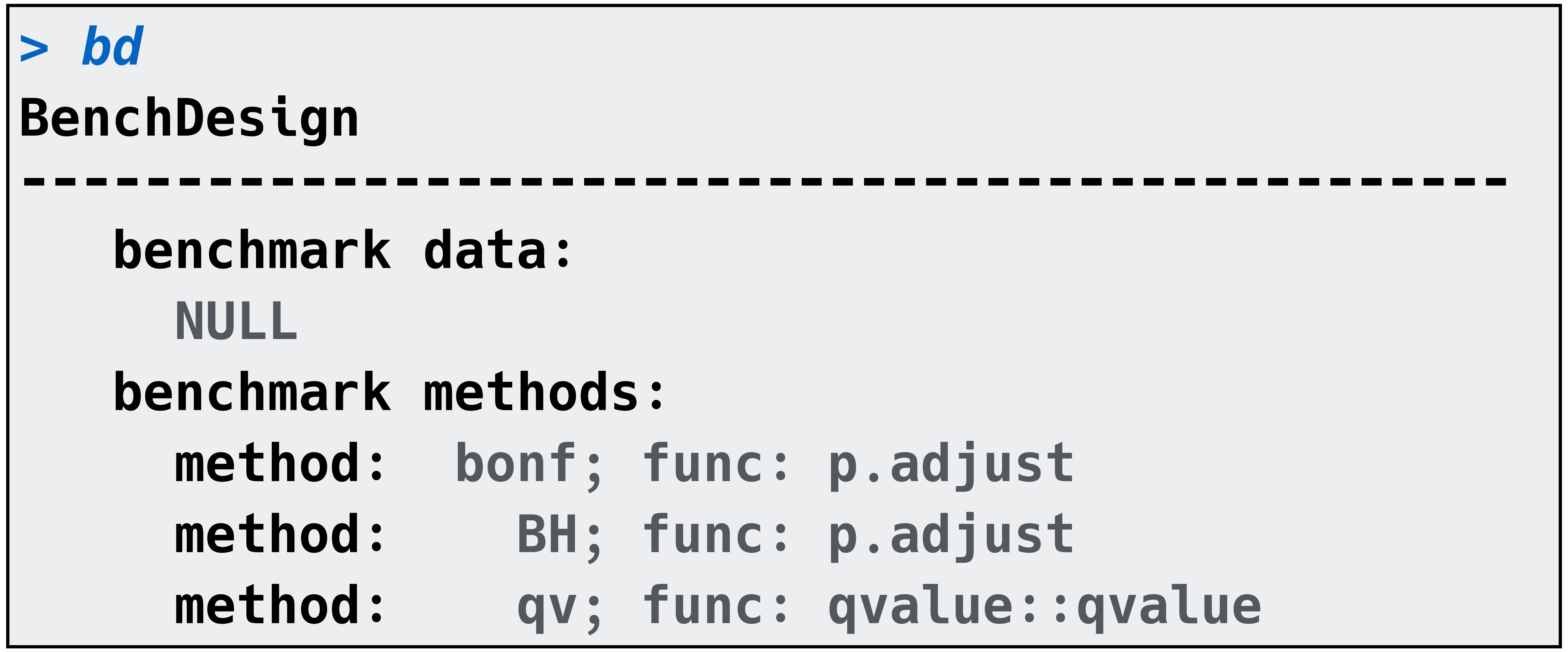


Methods

BenchDesign

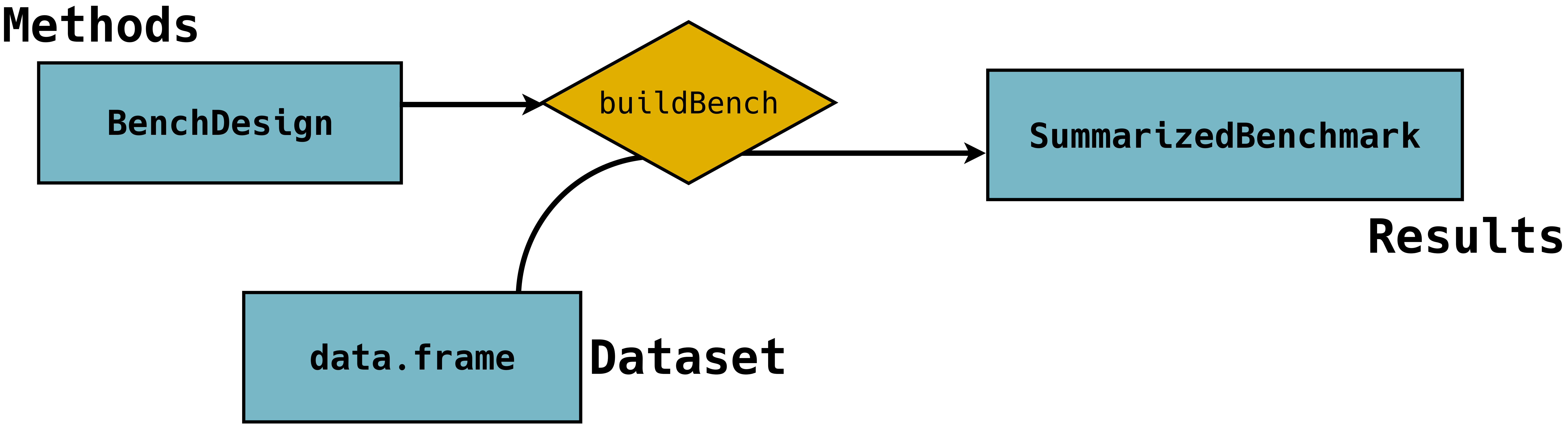
BenchDesign class

- collection of methods
  - function
  - map: data → function parameters



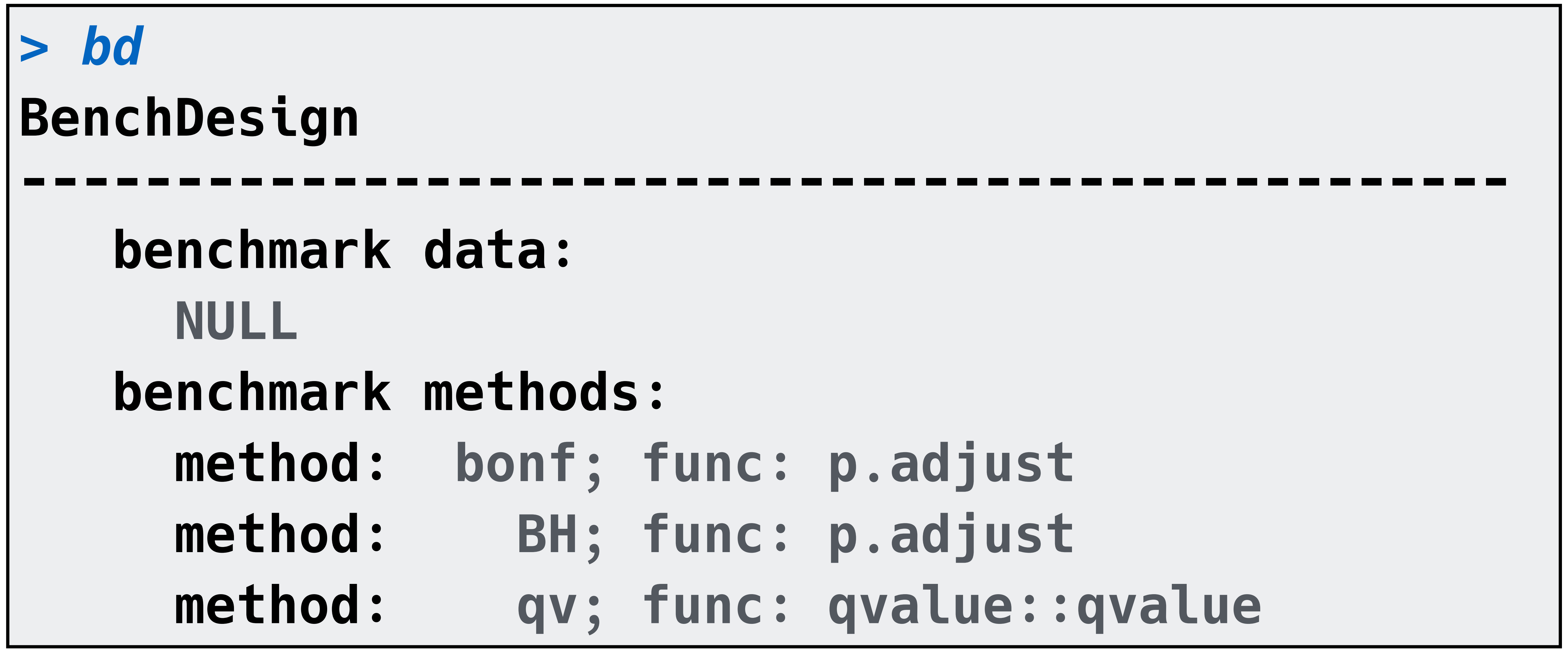
```
> bd
BenchDesign
-------------------------------------------------
   benchmark data:
     NULL
   benchmark methods:
     method:  bonf; func: p.adjust
     method:    BH; func: p.adjust
     method:    qv; func: qvalue::qvalue
```

SummarizedBenchmark framework



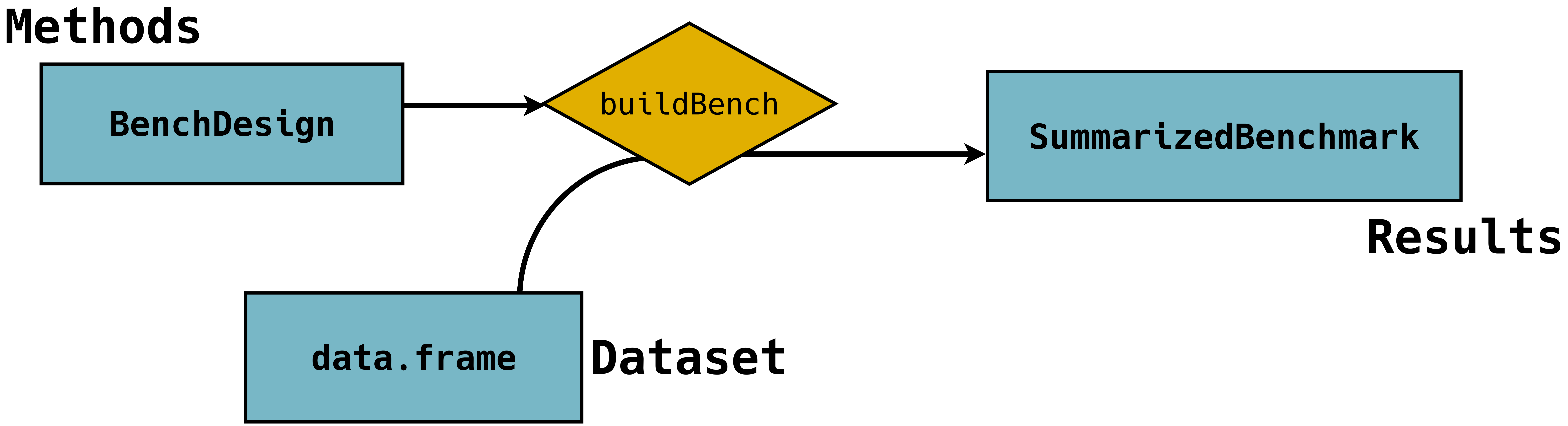
BenchDesign class

- collection of methods
  - function
  - map: data → function parameters



```
> bd
BenchDesign
------------------------------------------------
   benchmark data:
     NULL
   benchmark methods:
     method:  bonf; func: p.adjust
     method:    BH; func: p.adjust
     method:    qv; func: qvalue::qvalue
```

SummarizedBenchmark framework



Methods

BenchDesign

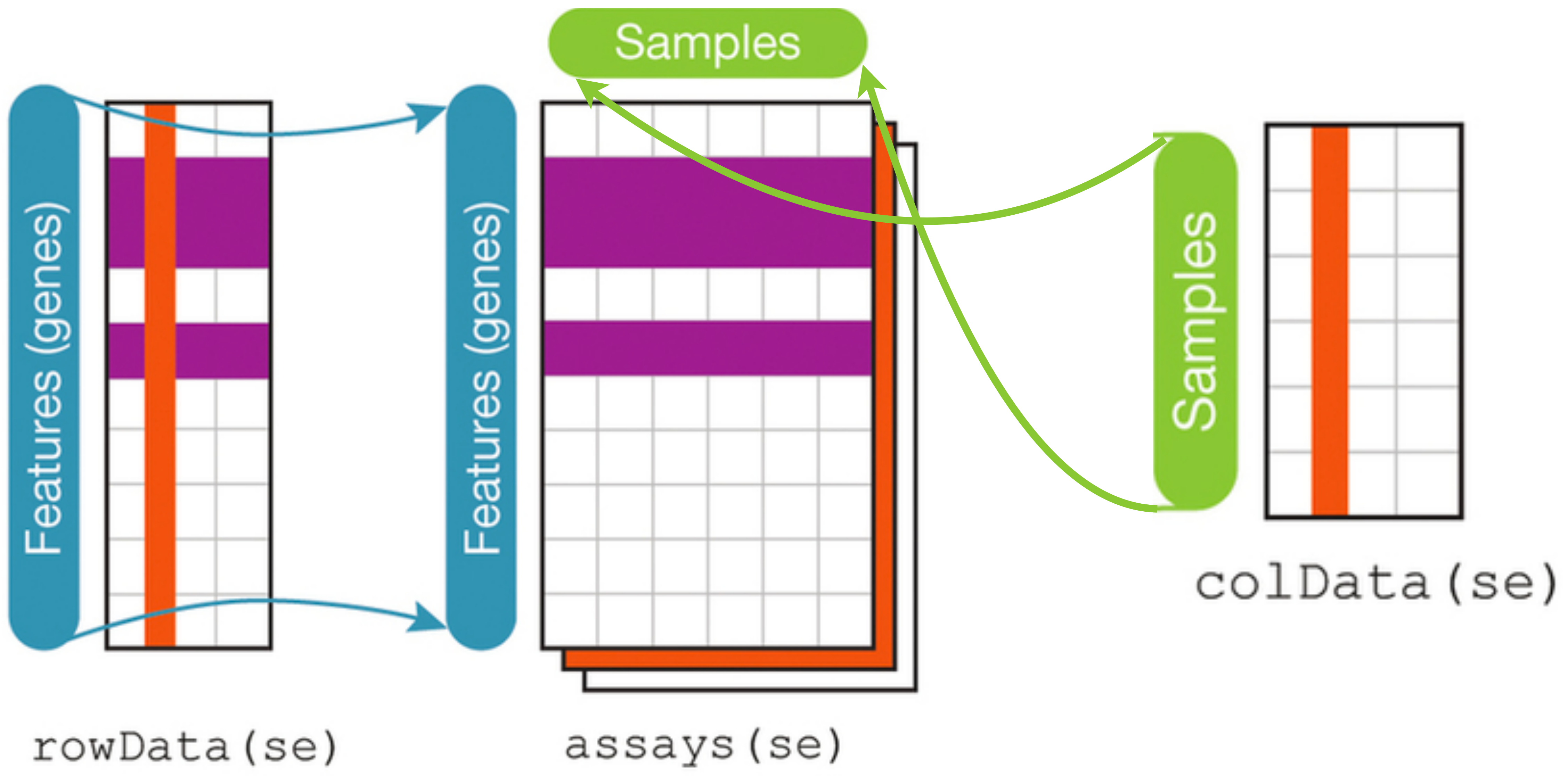
buildBench

SummarizedBenchmark

Results

data.frame

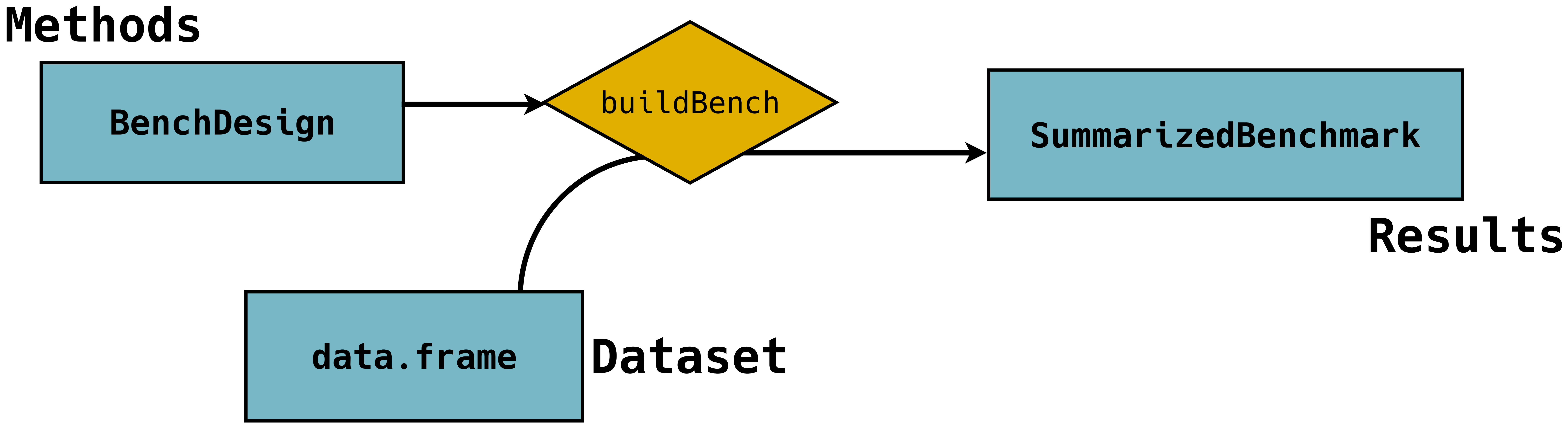
Dataset



SummarizedBenchmark class

- RangedSummarizedExperiment class
  - structure tying **row/col data + results**
  - slot for performance metrics

SummarizedBenchmark framework



Methods

BenchDesign

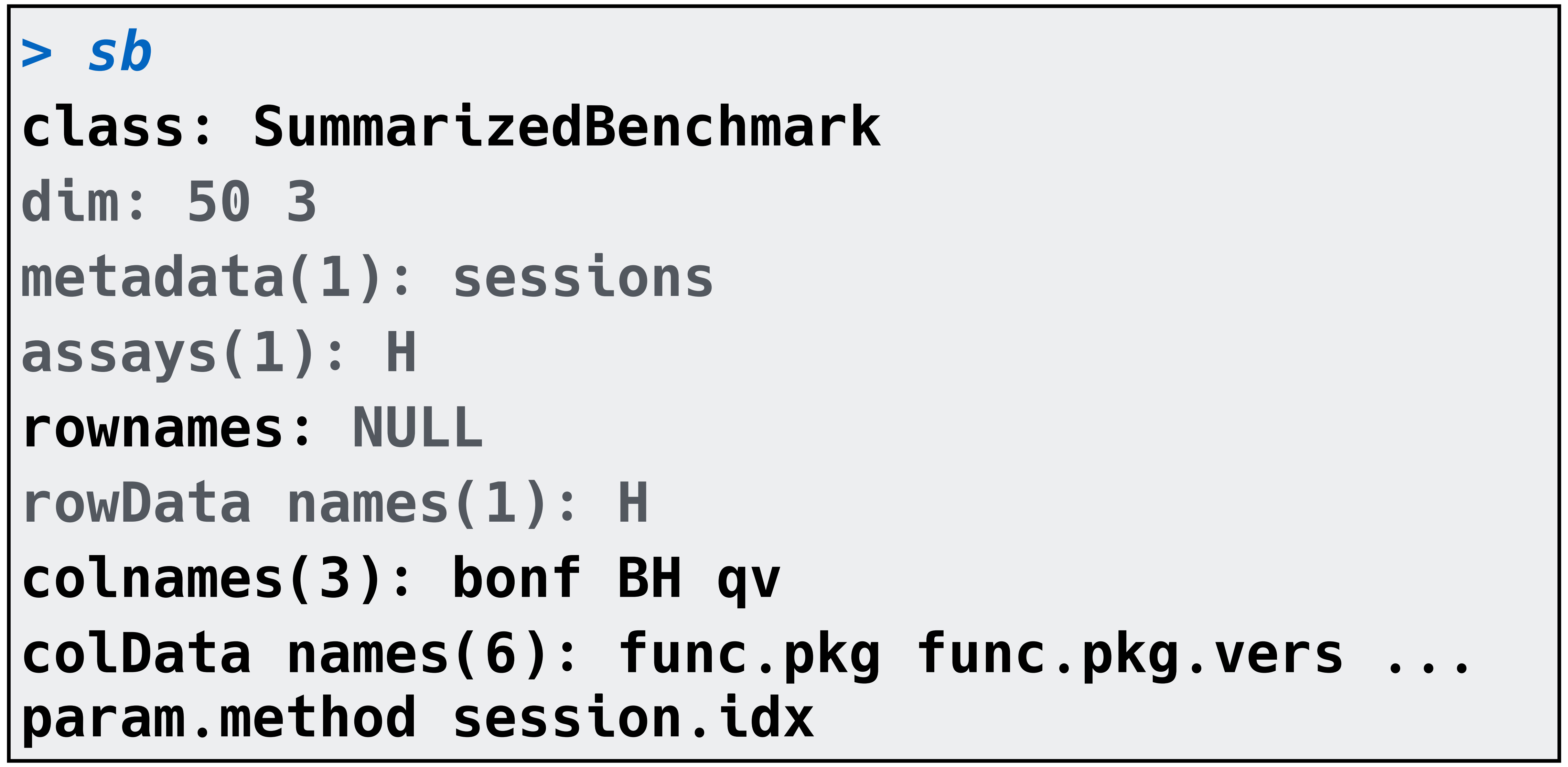
buildBench

SummarizedBenchmark

Results

data.frame

Dataset

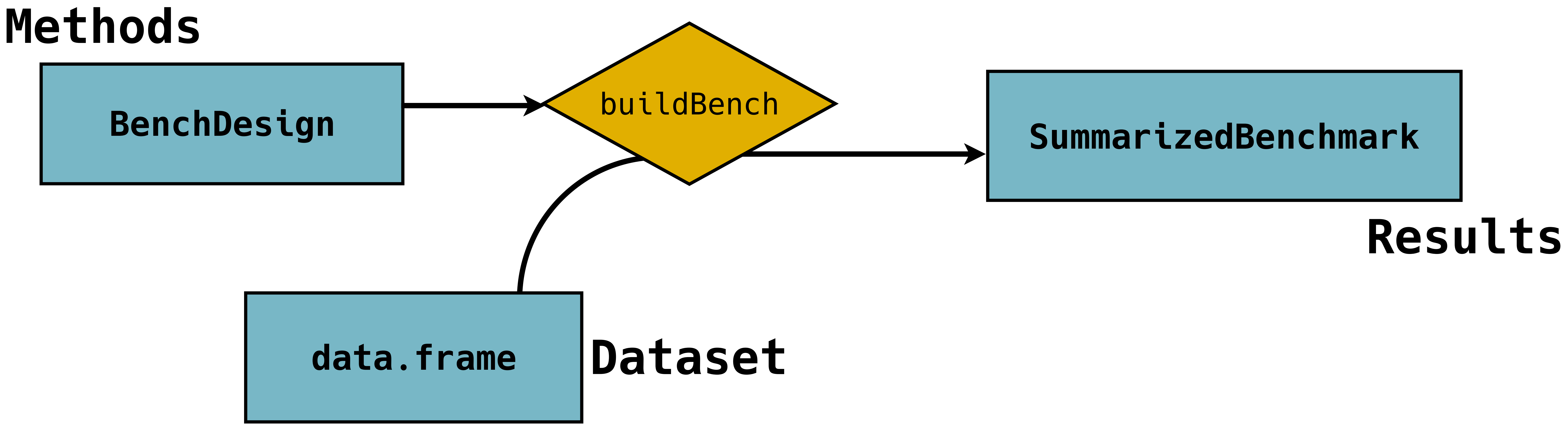


```
> sb
class: SummarizedBenchmark
dim: 50 3
metadata(1): sessions
assays(1): H
rownames: NULL
rowData names(1): H
colnames(3): bonf BH qv
colData names(6): func.pkg func.pkg.vers ...
param.method session.idx
```

SummarizedBenchmark class

- RangedSummarizedExperiment class
  - structure tying row/col data + results
  - slot for performance metrics

SummarizedBenchmark framework



Methods

BenchDesign

buildBench

SummarizedBenchmark

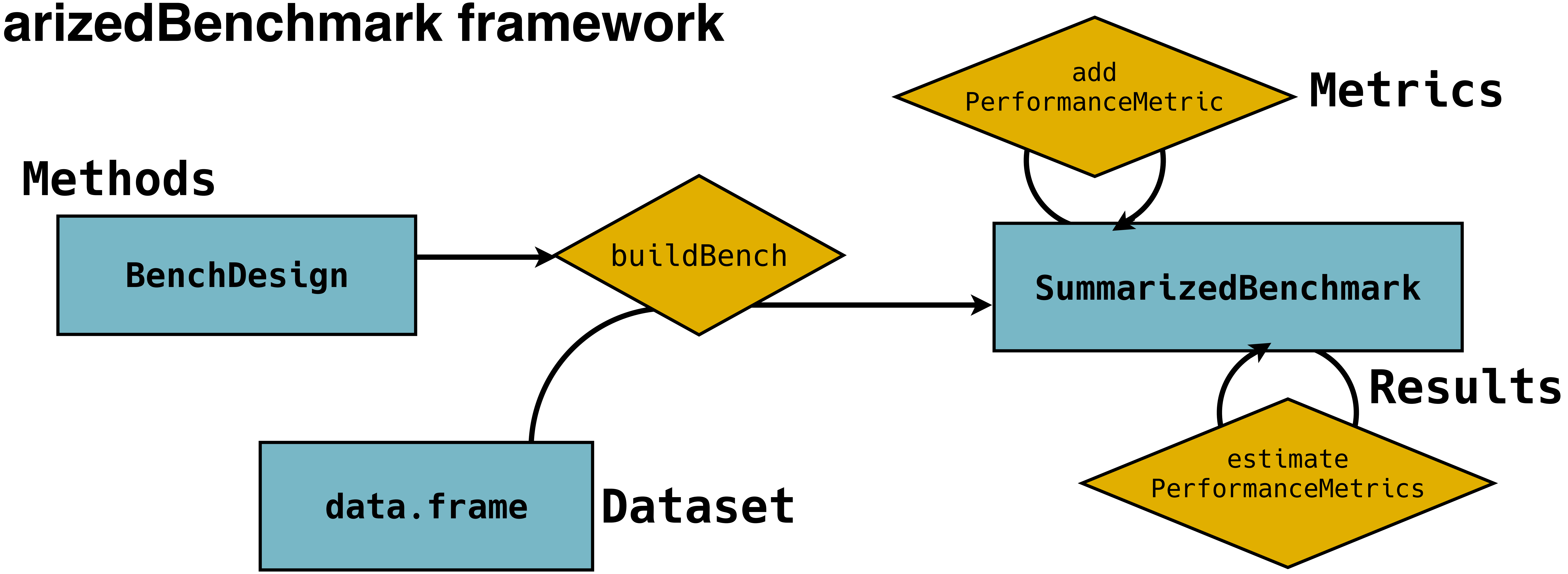
Results

data.frame

Dataset

```
> colData(sb)
 DataFrame with 3 rows and 5 columns
         func.pkg func.pkg.vers func.pkg.manual     param.p param.method
      <character>   <character>       <logical> <character>  <character>
 bonf       stats         3.5.0           FALSE        pval "bonferroni"
 BH         stats         3.5.0           FALSE        pval         "BH"
 qv        qvalue        2.12.0           FALSE        pval           NA
```

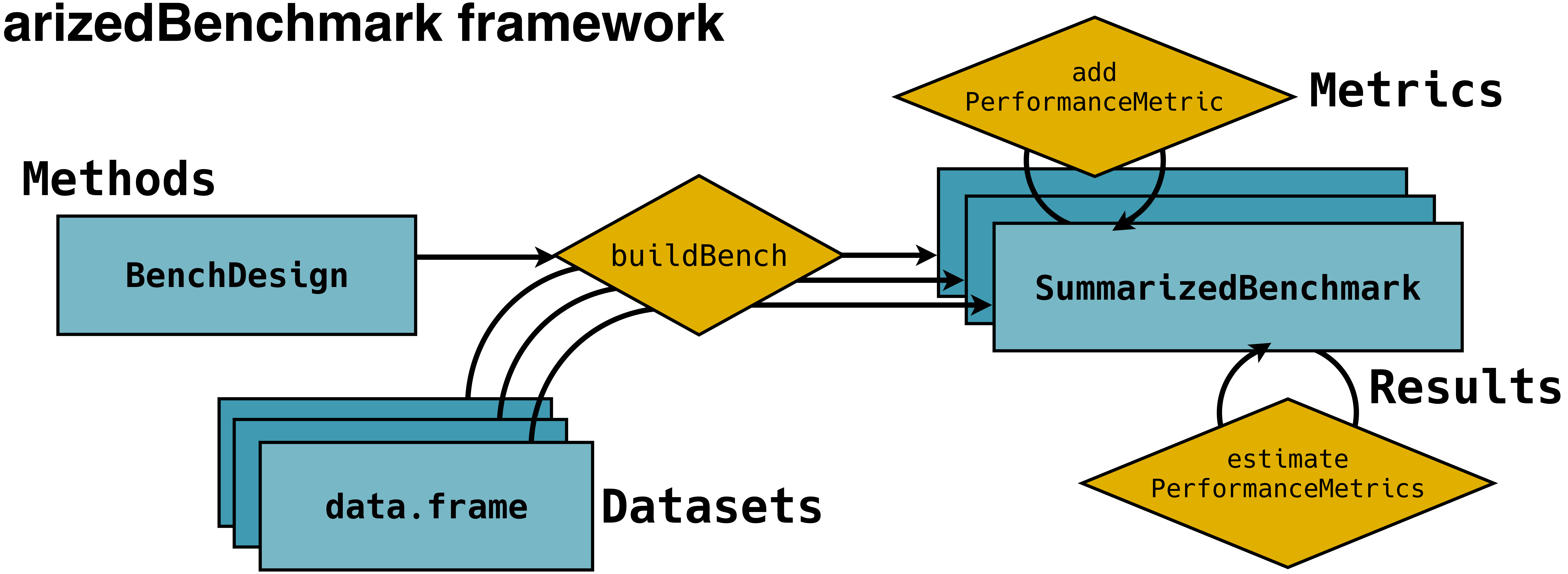
SummarizedBenchmark framework



performanceMetrics

- map: results (+ metadata) → metrics
- e.g. FDR, TPR, #rejections

SummarizedBenchmark framework



Additional Features

- iterative benchmarking
- error handling
- parallelization

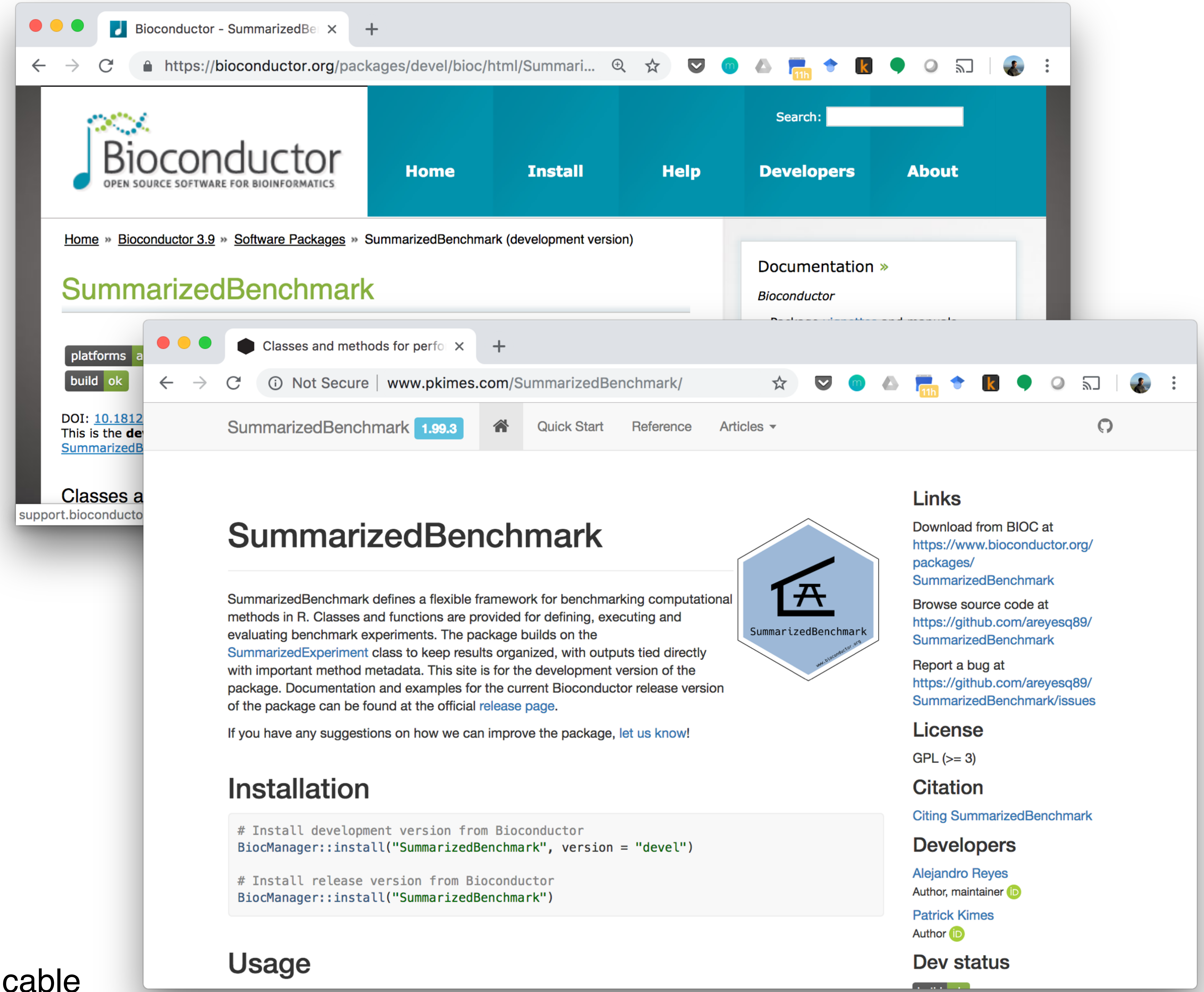
Ongoing Work

- handling larger pipelines
- additional default metrics

SummarizedBenchmark

Proposed framework

- methods
- data
- results
- summaries


Bioconductor - SummarizedBe
https://bioconductor.org/packages/devel/bioc/html/Summari...
Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS
Search:
Home
Install
Help
Developers
About
Documentation »
Bioconductor
platforms a
build ok
support.bioconducto
Classes and methods for perfo
Not Secure
www.pkimes.com/SummarizedBenchmark/
SummarizedBenchmark 1.99.3
Quick Start
Reference
Articles
SummarizedBenchmark

Home » Bioconductor 3.9 » Software Packages » SummarizedBenchmark (development version)

SummarizedBenchmark

DOI: 10.1812
This is the de
SummarizedB

Classes a

SummarizedBenchmark

SummarizedBenchmark defines a flexible framework for benchmarking computational methods in R. Classes and functions are provided for defining, executing and evaluating benchmark experiments. The package builds on the SummarizedExperiment class to keep results organized, with outputs tied directly with important method metadata. This site is for the development version of the package. Documentation and examples for the current Bioconductor release version of the package can be found at the official release page.

If you have any suggestions on how we can improve the package, let us know!

Installation

```
# Install development version from Bioconductor
BiocManager::install("SummarizedBenchmark", version = "devel")

# Install release version from Bioconductor
BiocManager::install("SummarizedBenchmark")
```

Usage

Links

Download from BIOC at https://www.bioconductor.org/packages/SummarizedBenchmark

Browse source code at https://github.com/areyesq89/SummarizedBenchmark

Report a bug at https://github.com/areyesq89/SummarizedBenchmark/issues

License

GPL (>= 3)

Citation

Citing SummarizedBenchmark

Developers

Alejandro Reyes
Author, maintainer

Patrick Kimes
Author

Dev status

Kimes PK* and Reyes A*. (2018). Reproducible and replicable comparisons using *SummarizedBenchmark*. Bioinformatics.

DANA-FARBER CANCER INSTITUTE

HARVARD T.H. CHAN SCHOOL OF PUBLIC HEALTH

Acknowledgements

FDR Benchmarking

- Keegan Korthauer*
- Stephanie Hicks
- Claire Duvallet
- Ayshwarya Subramanian
- Alejandro Reyes
- Chinmay Shukla
- Mingxiang Teng

SummarizedBenchmark

- Alejandro Reyes*

Rafael Irizarry