A unified treatment of multiple testing with prior knowledge

Aaditya Ramdas

(+ Rina F. Barber, Martin J. Wainwright, Michael I. Jordan)

A unified treatment of multiple testing with prior knowledge

Aaditya Ramdas

(+ Rina F. Barber, Martin J. Wainwright, Michael I. Jordan)

A unified treatment of multiple testing with prior knowledge

Aaditya Ramdas

(+ Rina F. Barber, Martin J. Wainwright, Michael I. Jordan)

A unified treatment of multiple testing with prior knowledge

Aaditya Ramdas

(+ Rina F. Barber, Martin J. Wainwright, Michael I. Jordan)

Starting point : the **p-filter algorithm** from "Multilayer FDR control for grouped hypotheses".

Barber-Ramdas '16

Starting point : the **p-filter algorithm** from "Multilayer FDR control for grouped hypotheses".

- **Input**: N p-values (one per hypothesis)
  M arbitrary partitions of these hypotheses
  M target FDR levels

Barber-Ramdas '16

Starting point : the **p-filter algorithm** from
"Multilayer FDR control for grouped hypotheses".

- **Input**: N p-values (one per hypothesis)
  M arbitrary partitions of these hypotheses
  M target FDR levels

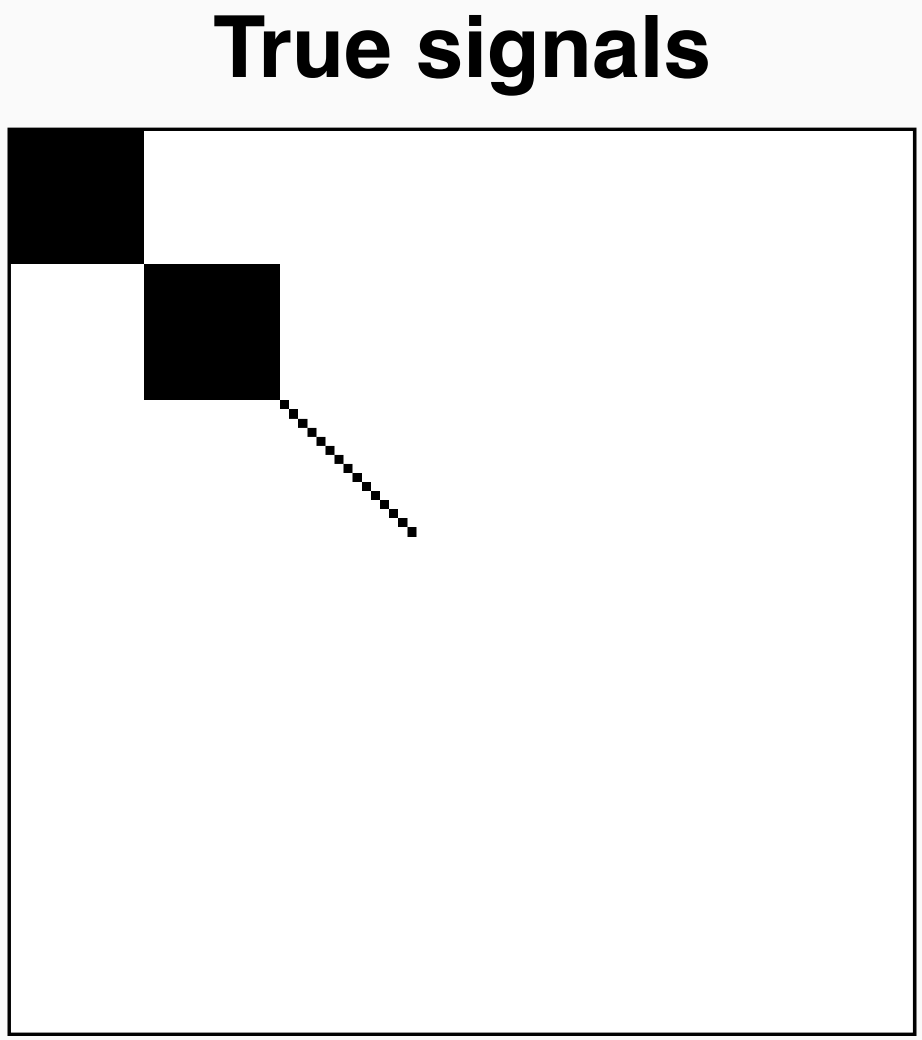
- **Output**: A set of rejected hypotheses and groups
  that are "internally consistent"
  such that group FDR is simultaneously
  controlled for all partitions.

Barber-Ramdas '16

Starting point : the **p-filter algorithm** from "Multilayer FDR control for grouped hypotheses".

- **Input**: N p-values (one per hypothesis)
  M arbitrary partitions of these hypotheses
  M target FDR levels
- **Output**: A set of rejected hypotheses and groups
  that are "internally consistent"
  such that group FDR is simultaneously
  controlled for all partitions.
- **Special cases**: Benjamini-Hochberg procedure
  Simes test for the global null

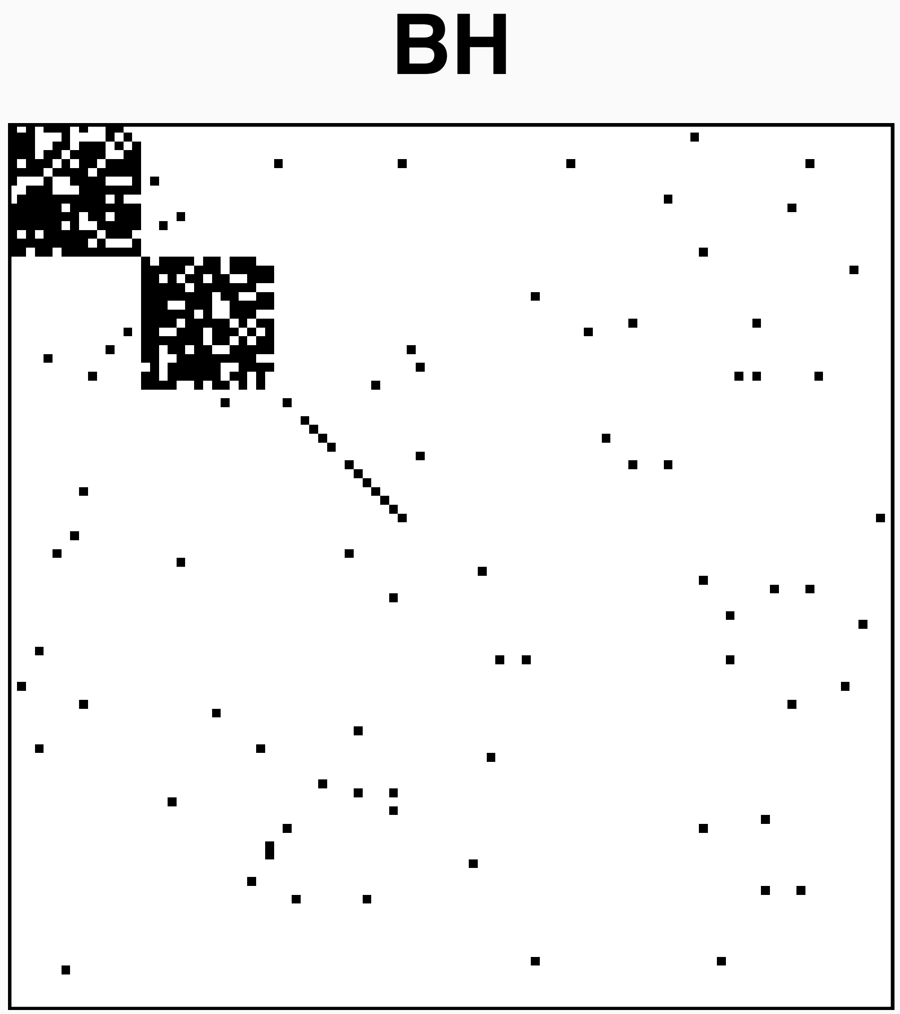
Barber-Ramdas '16



10,000 hypotheses in a100 x 100 grid.

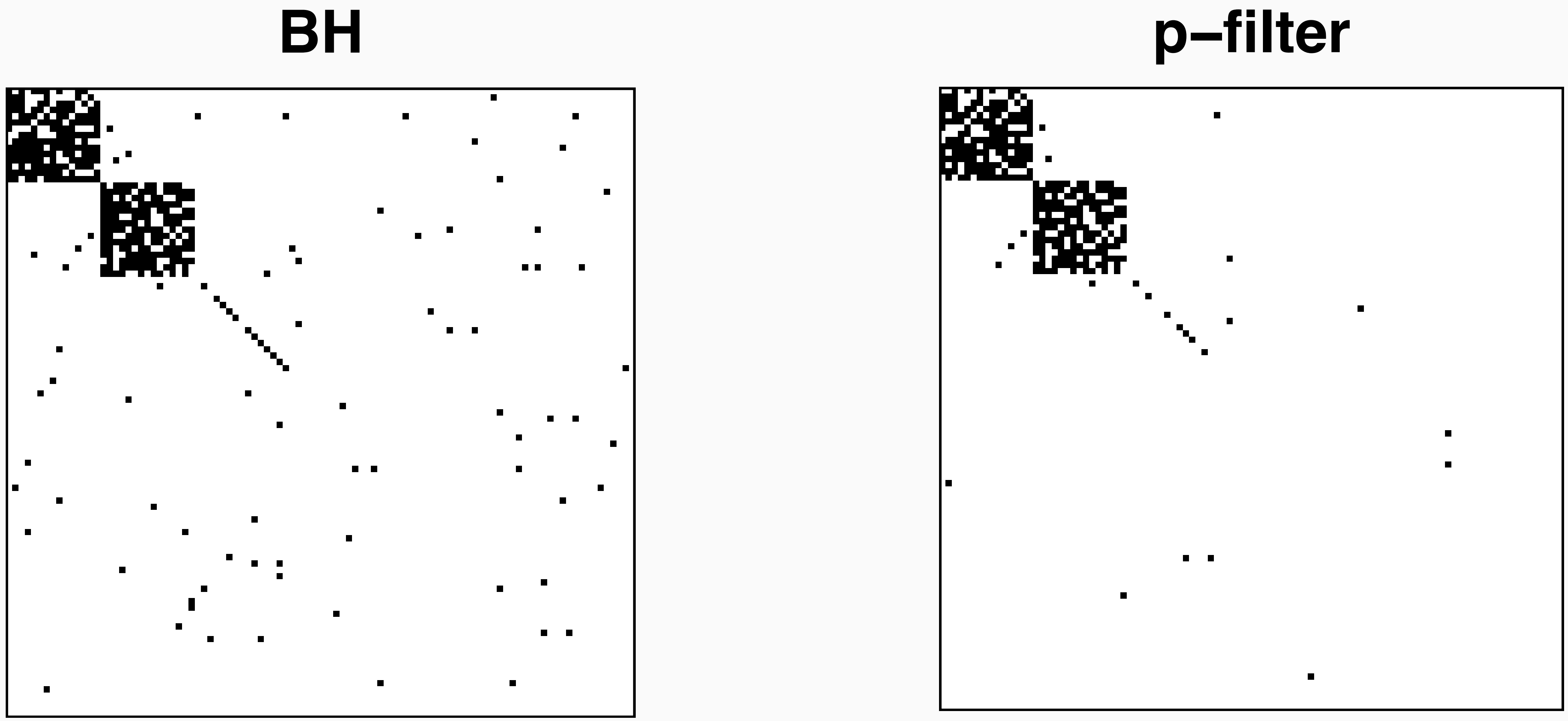
White (nulls) ~ N (0 , 1)
Black (non-nulls) ~ N(m , 1) for some m > 0

Barber-Ramdas '16



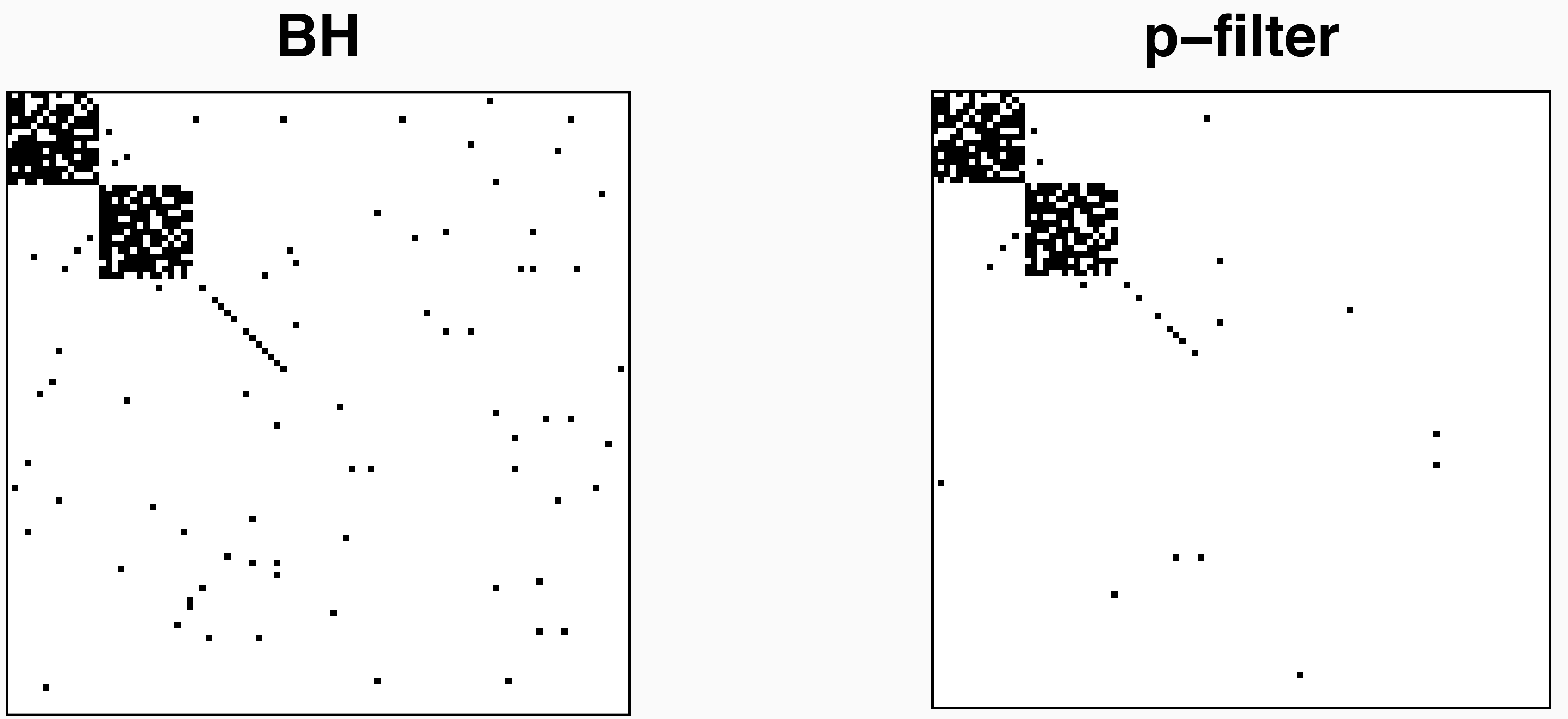
Target FDR = 0.2

Barber-Ramdas '16



Target FDR = 0.2

Barber-Ramdas '16



Target FDR = 0.2

3 partitions (rows, columns, entries)
Each target FDR = 0.2.

Barber-Ramdas '16

Several past works have looked at incorporating various forms of prior knowledge to improve power and precision of multiple testing methods.

Several past works have looked at incorporating various forms of prior knowledge to improve power and precision of multiple testing methods.

- Prior weights (likelihood of being non-null)

Several past works have looked at incorporating various forms of prior knowledge to improve power and precision of multiple testing methods.

- Prior weights (likelihood of being non-null)
- Penalty weights (differing importances)

Several past works have looked at incorporating various forms of prior knowledge to improve power and precision of multiple testing methods.

- Prior weights (likelihood of being non-null)
- Penalty weights (differing importances)
- Null-proportion adaptivity under independence

Several past works have looked at incorporating various forms of prior knowledge to improve power and precision of multiple testing methods.

- Prior weights (likelihood of being non-null)
- Penalty weights (differing importances)
- Null-proportion adaptivity under independence
- Guarding against arbitrary dependence

Several past works have looked at incorporating various forms of prior knowledge to improve power and precision of multiple testing methods.

- Prior weights (likelihood of being non-null)
- Penalty weights (differing importances)
- Null-proportion adaptivity under independence
- Guarding against arbitrary dependence
- Group structure (possibly several partitions)

Several past works have looked at incorporating various forms of prior knowledge to improve power and precision of multiple testing methods.

- Prior weights (likelihood of being non-null)
- Penalty weights (differing importances)
- Null-proportion adaptivity under independence
- Guarding against arbitrary dependence
- Group structure (possibly several partitions)

We integrate all of the above in one framework.

Several past works have looked at incorporating various forms of prior knowledge to improve power and precision of multiple testing methods.

- Prior weights (likelihood of being non-null)
- Penalty weights (differing importances)
- Null-proportion adaptivity under independence
- Guarding against arbitrary dependence
- Group structure (possibly several partitions)

We integrate all of the above in one framework.

(essentially "p-filter on steroids")

Benjamini-Hochberg controls FDR under independence and positive dependence.

$$\widehat{k} := \max\left\{k : P_{(k)} \leq \frac{\alpha \cdot k}{n}\right\}$$

$P_{(k)} = k\text{-th smallest value among the set of } P_i.$

$$\mathrm{FDR} := \mathbb{E}\left[\frac{\sum_{i \in \mathcal{H}_0} \mathbf{1}(i \in \mathcal{R})}{\sum_{i} \mathbf{1}(i \in \mathcal{R})}\right]$$

Benjamini-Hochberg '95

Prior-weighted BH controls FDR
under independence and positive dependence.

$$\widehat{k} := \max\left\{k : Q_{(k)} \leq \frac{\alpha \cdot k}{n}\right\}$$

$Q_{(k)}$ = k-th smallest value among the set of $Q_i := P_i / w_i$.

Genovese-Wasserman-Roeder '06

Penalty-weighted BH controls weighted-FDR under independence and positive dependence.

$$\widehat{k} := \max\left\{k : P_{(k)} \leq \frac{\alpha \cdot U_{(k)}}{n}\right\}$$

$U_{(k)}$ = sum of penalty weights of smallest k p-values.

$$\mathrm{FDR}_u := \mathbb{E}\left[\frac{\sum_{i \in \mathcal{H}_0} u_i \mathbf{1}(i \in \mathcal{R})}{\sum_i u_i \mathbf{1}(i \in \mathcal{R})}\right]$$

Benjamini-Yekutieli '97

Prior+penalty-weighted BH controls weighted-FDR under independence and positive dependence.

$$\widehat{k} := \max\left\{k : Q_{(k)} \leq \frac{\alpha \cdot U_{(k)}}{n}\right\}$$

$Q_{(k)}$ = k-th smallest value among the set of $Q_i := P_i / w_i$.

$U_{(k)}$ = sum of penalty weights of smallest k p-values.

Blanchard-Roquain '08

Null-proportion adaptivity for BH controls FDR only under independence.

$$\widehat{k} := \max\left\{k : P_{(k)} \leq \frac{\alpha \cdot k}{\widehat{\pi}_0 \cdot n}\right\}$$

$$\widehat{\pi}_0 = \frac{1 + \sum_i \mathbf{1}\{P_i > \lambda\}}{n(1-\lambda)}$$

$$\mathbb{E}\left[\frac{1}{\widehat{\pi}_0}\right] \leq \frac{n}{|\mathcal{H}_0|}$$

Schweder-Spjotvoll '72, Storey et al. '03, Benj-Kriegier-Yeku '06

Null-proportion adaptivity for Prior+penalty-weighted BH controls weighted-FDR only under independence.

$$\widehat{k} := \max\left\{k : Q_{(k)} \le \frac{\alpha \cdot U_{(k)}}{\widehat{\pi}_0 \cdot n}\right\}$$

$$\widehat{\pi}_0 = \frac{\max_j u_j w_j + \sum_i u_i w_i \cdot \mathbf{1}\{P_i > \lambda\}}{n(1-\lambda)}$$

$$\mathbb{E}\left[\frac{1}{\widehat{\pi}_0}\right] \le \frac{n}{\sum_{i \in \mathcal{H}^0} u_i w_i}$$

New!

Reshaped BH controls FDR under arbitrary dependence.

$$\widehat{k} := \max\left\{k : P_{(k)} \leq \frac{\alpha \cdot \beta(k)}{n}\right\}$$

$$\beta(u) = \int_0^u x \; d\nu(x) \quad \text{where} \quad \nu([0,n]) = 1$$

Benjamini-Yekutieli '01, Blanchard-Roquain '08, Sarkar '08

Reshaped prior+penalty-weighted BH controls weighted-FDR under arbitrary dependence.

$$\widehat{k} := \max\left\{ k : Q_{(k)} \leq \frac{\alpha \cdot \beta(U_{(k)})}{n} \right\}$$

$$\beta(u) = \int_0^u x \; d\nu(x) \quad \text{where} \quad \nu([0, n]) = 1$$

New!

LEMMA 1 (Super-uniformity lemma). *Let* $f : [0,1]^n \mapsto [0,\infty)$ *be arbitrary, and let index* $i \in \mathcal{H}^0$ *refer to some null hypothesis.*

New! Generalization of Blanchard-Roquain '08.

LEMMA 1 (Super-uniformity lemma). *Let* $f : [0,1]^n \mapsto [0,\infty)$ *be arbitrary, and let index* $i \in \mathcal{H}^0$ *refer to some null hypothesis.*

(b) *If f is nonincreasing, then under positive dependence and super-uniformity, we have*

$$\mathbb{E}\left[\frac{\mathbf{1}\{P_i \le f(P)\}}{f(P)}\right] \le 1.$$

New! Generalization of Blanchard-Roquain '08.

LEMMA 1 (Super-uniformity lemma). *Let $f : [0,1]^n \mapsto [0,\infty)$ be arbitrary, and let index $i \in \mathcal{H}^0$ refer to some null hypothesis.*

(b) *If f is nonincreasing, then under positive dependence and super-uniformity, we have*

$$\mathbb{E}\left[\frac{\mathbf{1}\{P_i \leq f(P)\}}{f(P)}\right] \leq 1.$$

(c) *For any constant $c > 0$ and any non-negative functions $f, \widetilde{f}$, under arbitrary dependence and super-uniformity, we have*

$$\mathbb{E}\left[\frac{\mathbf{1}\left\{P_i \leq c\beta_1(f(P))\beta_2(\widetilde{f}(P))\right\}}{cf(P)\widetilde{f}(P)}\right] \leq 1 \qquad \textit{for all pairs } \beta_1, \beta_2 \in \beta(\mathcal{V}).$$

New! Generalization of Blanchard-Roquain '08.

LEMMA 1 (Super-uniformity lemma). *Let $f : [0,1]^n \mapsto [0,\infty)$ be arbitrary, and let index $i \in \mathcal{H}^0$ refer to some null hypothesis.*

(a) If f is nonincreasing, has range $[0,1]$, and satisfies the LOOP condition (3.6), then under independence and uniformity, we have

$$\mathbb{E}\left[\frac{\mathbf{1}\{P_i \leq f(P)\}}{f(P)} \,\middle|\, P^{-i}\right] = 1, \quad \textit{and hence} \quad \mathbb{E}\left[\frac{\mathbf{1}\{P_i \leq f(P)\}}{f(P)}\right] = 1.$$

(b) If f is nonincreasing, then under positive dependence and super-uniformity, we have

$$\mathbb{E}\left[\frac{\mathbf{1}\{P_i \leq f(P)\}}{f(P)}\right] \leq 1.$$

(c) For any constant $c > 0$ and any non-negative functions $f, \widetilde{f}$, under arbitrary dependence and super-uniformity, we have

$$\mathbb{E}\left[\frac{\mathbf{1}\left\{P_i \leq c\beta_1(f(P))\beta_2(\widetilde{f}(P))\right\}}{cf(P)\widetilde{f}(P)}\right] \leq 1 \qquad \textit{for all pairs } \beta_1, \beta_2 \in \beta(\mathcal{V}).$$

New! Generalization of Blanchard-Roquain '08.

LEMMA 1 (Super-uniformity lemma). *Let $f : [0,1]^n \mapsto [0,\infty)$ be arbitrary, and let index $i \in \mathcal{H}^0$ refer to some null hypothesis.*

(a) If f is nonincreasing, has range $[0,1]$, and satisfies the LOOP condition (3.6), then under independence and uniformity, we have

$$\mathbb{E}\left[\frac{\mathbf{1}\{P_i \le f(P)\}}{f(P)} \,\middle|\, P^{-i}\right] = 1, \quad \textit{and hence} \quad \mathbb{E}\left[\frac{\mathbf{1}\{P_i \le f(P)\}}{f(P)}\right] = 1.$$

(b) If f is nonincreasing, then under positive dependence and super-uniformity, we have

$$\boxed{\mathbb{E}\left[\frac{\mathbf{1}\{P_i \le f(P)\}}{f(P)}\right] \le 1.}$$

(c) For any constant $c > 0$ and any non-negative functions $f, \widetilde{f}$, under arbitrary dependence and super-uniformity, we have

$$\mathbb{E}\left[\frac{\mathbf{1}\left\{P_i \le c\beta_1(f(P))\beta_2(\widetilde{f}(P))\right\}}{cf(P)\widetilde{f}(P)}\right] \le 1 \qquad \textit{for all pairs } \beta_1, \beta_2 \in \beta(\mathcal{V}).$$

New! Generalization of Blanchard-Roquain '08.

The Simes test controls type-I error for the global null under positive dependence.

$$\text{Simes}(P) := \min_{1 \leq k \leq n} \frac{P_{(k)} \cdot n}{k}$$

We test if $\text{Simes}(P) \leq \alpha$, which happens iff $\widehat{k}_{BH} \geq 1$.

Simes '86

The Simes test controls type-I error for the global null under positive dependence.

$$\mathrm{Simes}(P) := \min_{1 \le k \le n} \frac{P_{(k)} \cdot n}{k}$$

We test if $\mathrm{Simes}(P) \le \alpha$, which happens iff $\widehat{k}_{BH} \ge 1$.

The Simes p-value really is a p-value for the global null.
It is uniformly distributed if P is entirely null.

Simes '86

The reshaped prior-weighted Simes test controls type-I error for the global null under arbitrary dependence.

$$\mathrm{Simes}(P) := \min_{1 \le k \le n} \frac{Q_{(k)} \cdot n}{\beta(k)}$$

We test if $\mathrm{Simes}(P) \le \alpha$, which happens iff $\widehat{k}_{BH} \ge 1$.

New!

The reshaped prior-weighted Simes test controls type-I error for the global null under arbitrary dependence.

$$\text{Simes}(P) := \min_{1 \le k \le n} \frac{Q_{(k)} \cdot n}{\beta(k)}$$

We test if $\text{Simes}(P) \le \alpha$, which happens iff $\widehat{k}_{BH} \ge 1$.

The Simes p-value really is a p-value for the global null.
It is uniformly distributed if P is entirely null.

New!

Suppose we want to control the
penalty-weighted group-FDR for a single partition :

$$\underbrace{\overbrace{P_1}^{\text{weight } w_1^{(1)}}, \dots, \overbrace{P_{n_1}}^{\text{weight } w_{n_1}^{(1)}}}_{\text{Group } A_1, \text{ weights } u_1^{(2)}, w_1^{(2)}}, \dots, \underbrace{\overbrace{P_{n_1+\dots+n_{G-1}+1}}^{\text{weight } w_{n_1+\dots+n_{G-1}+1}^{(1)}}, \dots, \overbrace{P_n}^{\text{weight } w_n^{(1)}}}_{\text{Group } A_G, \text{ weights } u_G^{(2)}, w_G^{(2)}}$$

New!

Suppose we want to control the
penalty-weighted group-FDR for a single partition :

$$\underbrace{\overbrace{P_1}^{\text{weight } w_1^{(1)}}, \ldots, \overbrace{P_{n_1}}^{\text{weight } w_{n_1}^{(1)}}}_{\boxed{\text{Group } A_1}\text{, weights } u_1^{(2)}, w_1^{(2)}}, \ldots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weight } w_{n_1+\cdots+n_{G-1}+1}^{(1)}}, \ldots, \overbrace{P_n}^{\text{weight } w_n^{(1)}}}_{\text{Group } A_G\text{, weights } u_G^{(2)}, w_G^{(2)}};$$

New!

Suppose we want to control the penalty-weighted group-FDR for a single partition :

$$\underbrace{\overbrace{P_1}^{\text{weight } w_1^{(1)}}, \ldots, \overbrace{P_{n_1}}^{\text{weight } w_{n_1}^{(1)}}}_{\boxed{\text{Group } A_1}\text{, weights } u_1^{(2)}, w_1^{(2)}}, \ldots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weight } w_{n_1+\cdots+n_{G-1}+1}^{(1)}}, \ldots, \overbrace{P_n}^{\text{weight } w_n^{(1)}}}_{\boxed{\text{Group } A_G}\text{, weights } u_G^{(2)}, w_G^{(2)}};$$

New!

Suppose we want to control the
penalty-weighted group-FDR for a single partition :

$$\underbrace{\overbrace{P_1}^{\text{weight } w_1^{(1)}}, \ldots, \overbrace{P_{n_1}}^{\text{weight } w_{n_1}^{(1)}}}_{\boxed{\text{Group } A_1}\text{, weights } \boxed{u_1^{(2)}}, w_1^{(2)}}, \ldots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weight } w_{n_1+\cdots+n_{G-1}+1}^{(1)}}, \ldots, \overbrace{P_n}^{\text{weight } w_n^{(1)}}}_{\boxed{\text{Group } A_G}\text{, weights } u_G^{(2)}, w_G^{(2)}};$$

New!

Suppose we want to control the penalty-weighted group-FDR for a single partition :

$$\underbrace{\overbrace{P_1}^{\text{weight } w_1^{(1)}}, \ldots, \overbrace{P_{n_1}}^{\text{weight } w_{n_1}^{(1)}}}_{\boxed{\text{Group } A_1}\text{, weights } \boxed{u_1^{(2)}}, w_1^{(2)}}, \ldots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weight } w_{n_1+\cdots+n_{G-1}+1}^{(1)}}, \ldots, \overbrace{P_n}^{\text{weight } w_n^{(1)}}}_{\boxed{\text{Group } A_G}\text{, weights } \boxed{u_G^{(2)}}, w_G^{(2)}};$$

New!

Suppose we want to control the penalty-weighted group-FDR for a single partition :

$$\underbrace{\overbrace{P_1}^{\text{weight } w_1^{(1)}}, \ldots, \overbrace{P_{n_1}}^{\text{weight } w_{n_1}^{(1)}}}_{\text{Group } A_1,\ \text{weights } u_1^{(2)}, w_1^{(2)}}, \ldots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weight } w_{n_1+\cdots+n_{G-1}+1}^{(1)}}, \ldots, \overbrace{P_n}^{\text{weight } w_n^{(1)}}}_{\text{Group } A_G,\ \text{weights } u_G^{(2)}, w_G^{(2)}};$$

New!

Suppose we want to control the
penalty-weighted group-FDR for a single partition :

$$\underbrace{\overbrace{P_1}^{\text{weight } w_1^{(1)}}, \ldots, \overbrace{P_{n_1}}^{\text{weight } w_{n_1}^{(1)}}}_{\boxed{\text{Group } A_1}, \text{ weights } \boxed{u_1^{(2)}}, \boxed{w_1^{(2)}}}, \ldots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weight } w_{n_1+\cdots+n_{G-1}+1}^{(1)}}, \ldots, \overbrace{P_n}^{\text{weight } w_n^{(1)}}}_{\boxed{\text{Group } A_G}, \text{ weights } \boxed{u_G^{(2)}}, \boxed{w_G^{(2)}}};$$

New!

Suppose we want to control the
penalty-weighted group-FDR for a single partition :

$$\underbrace{\overbrace{P_1}^{\text{weight } w_1^{(1)}}, \ldots, \overbrace{P_{n_1}}^{\text{weight } w_{n_1}^{(1)}}}_{\text{Group } A_1, \text{ weights } u_1^{(2)}, w_1^{(2)}}, \ldots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weight } w_{n_1+\cdots+n_{G-1}+1}^{(1)}}, \ldots, \overbrace{P_n}^{\text{weight } w_n^{(1)}}}_{\text{Group } A_G, \text{ weights } u_G^{(2)}, w_G^{(2)}};$$

New!

Suppose we want to control the penalty-weighted group-FDR for a single partition :

$$\underbrace{\overbrace{P_1}^{\text{weight } w_1^{(1)}}, \ldots, \overbrace{P_{n_1}}^{\text{weight } w_{n_1}^{(1)}}}_{\text{Group } A_1,\ \text{weights } u_1^{(2)}, w_1^{(2)}}, \ldots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weight } w^{(1)}_{n_1+\cdots+n_{G-1}+1}}, \ldots, \overbrace{P_n}^{\text{weight } w_n^{(1)}}}_{\text{Group } A_G,\ \text{weights } u_G^{(2)}, w_G^{(2)}};$$

New!

Suppose we want to control the
penalty-weighted group-FDR for a single partition :

$$\underbrace{\overbrace{P_1}^{\text{weight } w_1^{(1)}}, \ldots, \overbrace{P_{n_1}}^{\text{weight } w_{n_1}^{(1)}}}_{\text{Group } A_1,\ \text{weights } u_1^{(2)}, w_1^{(2)}}, \ldots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weight } w^{(1)}_{n_1+\cdots+n_{G-1}+1}}, \ldots, \overbrace{P_n}^{\text{weight } w_n^{(1)}}}_{\text{Group } A_G,\ \text{weights } u_G^{(2)}, w_G^{(2)}};$$

New!

Suppose we want to control the
penalty-weighted group-FDR for a single partition :

$$\underbrace{\overbrace{P_1}^{\text{weight } w_1^{(1)}}, \ldots, \overbrace{P_{n_1}}^{\text{weight } w_{n_1}^{(1)}}}_{\text{Group } A_1\text{, weights } u_1^{(2)}, w_1^{(2)}}, \ldots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weight } w_{n_1+\cdots+n_{G-1}+1}^{(1)}}, \ldots, \overbrace{P_n}^{\text{weight } w_n^{(1)}}}_{\text{Group } A_G\text{, weights } u_G^{(2)}, w_G^{(2)}};$$

New!

Suppose we want to control the penalty-weighted group-FDR for a single partition :

$$\underbrace{\overbrace{P_1}^{\text{weight } w_1^{(1)}}, \ldots, \overbrace{P_{n_1}}^{\text{weight } w_{n_1}^{(1)}}}_{\text{Group } A_1,\ \text{weights } u_1^{(2)}, w_1^{(2)}}, \ldots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weight } w_{n_1+\cdots+n_{G-1}+1}^{(1)}}, \ldots, \overbrace{P_n}^{\text{weight } w_n^{(1)}}}_{\text{Group } A_G,\ \text{weights } u_G^{(2)}, w_G^{(2)}};$$

$$\mathrm{gFDR}_{u^{(2)}} := \mathbb{E}\left[\frac{\sum_{g \in \mathcal{H}_0^G} u_g^{(2)} \mathbf{1}(g \in \mathcal{R}_G)}{\sum_g u_g^{(2)} \mathbf{1}(g \in \mathcal{R}_G)}\right]$$

New!

Suppose we want to control the
penalty-weighted group-FDR for a single partition :

$$\underbrace{\overbrace{P_1}^{\text{weight } w_1^{(1)}}, \dots, \overbrace{P_{n_1}}^{\text{weight } w_{n_1}^{(1)}}}_{\text{Group } A_1,\ \text{weights } u_1^{(2)}, w_1^{(2)}}, \dots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weight } w_{n_1+\cdots+n_{G-1}+1}^{(1)}}, \dots, \overbrace{P_n}^{\text{weight } w_n^{(1)}}}_{\text{Group } A_G,\ \text{weights } u_G^{(2)}, w_G^{(2)}};$$

$$\mathrm{gFDR}_{u^{(2)}} := \mathbb{E}\left[\frac{\sum_{g \in \mathcal{H}_0^G} u_g^{(2)} \mathbf{1}(g \in \mathcal{R}_G)}{\sum_g u_g^{(2)} \mathbf{1}(g \in \mathcal{R}_G)}\right]$$

Solution : Calculate Simes p-value of each group, and then run BH on the Simes p-values.

New!

When we have **multiple partitions**, we would like to have "internally consistent" rejections, (also known as consonance+coherence) :

Barber-Ramdas '16

When we have **multiple partitions**, we would like to have "<u>internally consistent</u>" rejections, (also known as consonance+coherence) :

For every partition,

- Each rejected group should contain at least one rejected element.
- Each rejected element should be contained in some rejected group.

Barber-Ramdas '16

Eg: let's see what happens if we have **two partitions**, the finest one (each element is a group), and a coarser one, we want to control elementwise FDR and group FDR :

Barber-Ramdas '16

Eg: let's see what happens if we have **two partitions**, the finest one (each element is a group), and a coarser one, we want to control elementwise FDR and group FDR :

$$\alpha_{\text{indiv}} = 0.1$$
$$\alpha_{\text{group}} = 0.2$$

						Simes p−value
Group 1	0.03	0.01	0.18	0.04	0.08	0.05
Group 2	0.05	0.11	0.06	0.01	0.89	0.05
Group 3	0.14	0.12	0.58	0.11	0.11	0.18
Group 4	0.88	0.24	0.09	0.66	0.45	0.45

Barber-Ramdas '16

Running BH+Simes on each partition separately might lead to "contradictory" findings.

Barber-Ramdas '16

Running BH+Simes on each partition separately might lead to "contradictory" findings.

$$\alpha_{\text{indiv}} = 0.1$$
$$\alpha_{\text{group}} = 0.2$$

Barber-Ramdas '16

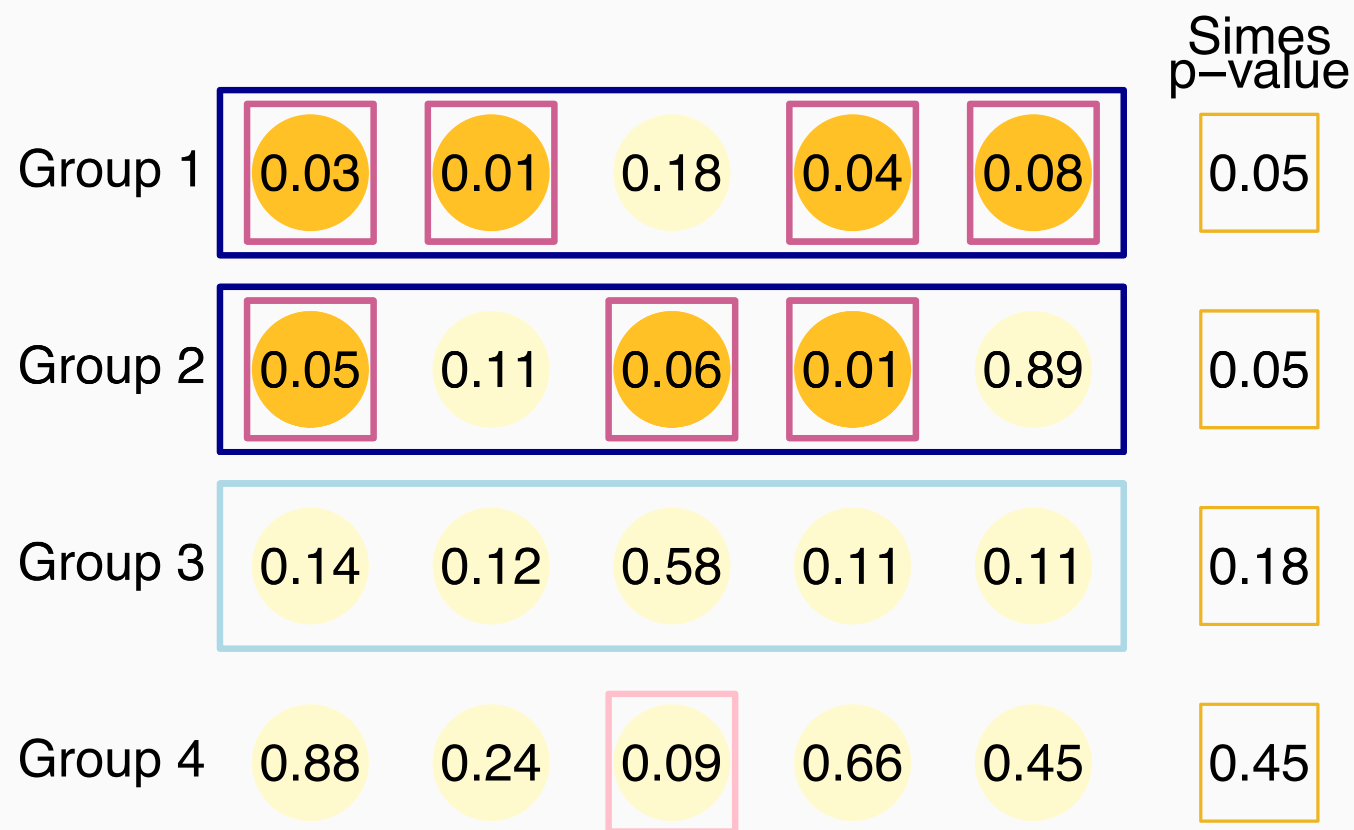
Naively intersecting the two sets of findings may control *neither* elementwise *nor* group *FDR.*

Barber-Ramdas '16

Naively intersecting the two sets of findings may control *neither* elementwise *nor* group *FDR*.

$$\alpha_{\text{indiv}} = 0.1$$

$$\alpha_{\text{group}} = 0.2$$



Barber-Ramdas '16

$$\underbrace{\overbrace{P_1}^{u_1^{(1)},w_1^{(1)}},\ldots,\overbrace{P_{n_1}}^{u_{n_1}^{(1)},w_{n_1}^{(1)}}}_{\text{Group 1, weights } u_1^{(2)},w_1^{(2)}},\ldots,\underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weights ...}},\ldots,\overbrace{P_{n-\ell_2}}^{\text{weights ...}}}_{\text{Group } G\text{, weights } u_G^{(2)},w_G^{(2)}} \Bigg| \underbrace{\overbrace{P_{n-\ell_2+1}}^{\text{weights ...}},\ldots,\overbrace{P_n}^{u_n^{(1)},w_n^{(1)}}}_{\text{Leftover group } L^{(2)}}$$

New!

Our unifying framework can handle

$$\underbrace{\overbrace{P_1}^{u_1^{(1)},w_1^{(1)}} \;, \ldots, \; \overbrace{P_{n_1}}^{u_{n_1}^{(1)},w_{n_1}^{(1)}}}_{\text{Group 1, weights } u_1^{(2)},w_1^{(2)}} \;, \ldots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weights ...}}, \ldots, \; \overbrace{P_{n-\ell_2}}^{\text{weights ...}}}_{\text{Group } G\text{, weights } u_G^{(2)},w_G^{(2)}} \;\Big|\; \underbrace{\overbrace{P_{n-\ell_2+1}}^{\text{weights ...}}, \ldots, \; \overbrace{P_n}^{u_n^{(1)},w_n^{(1)}}}_{\text{Leftover group } L^{(2)}}$$

New!

Our unifying framework can handle

- many arbitrary, incomplete partitions,

$$\underbrace{\overbrace{P_1}^{u_1^{(1)},w_1^{(1)}}\ ,\dots,\ \overbrace{P_{n_1}}^{u_{n_1}^{(1)},w_{n_1}^{(1)}}}_{\text{Group 1, weights } u_1^{(2)},w_1^{(2)}},\dots,\underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weights ...}},\dots,\ \overbrace{P_{n-\ell_2}}^{\text{weights ...}}}_{\text{Group } G\text{, weights } u_G^{(2)},w_G^{(2)}}\ \Big|\ \underbrace{\overbrace{P_{n-\ell_2+1}}^{\text{weights ...}},\dots,\ \overbrace{P_n}^{u_n^{(1)},w_n^{(1)}}}_{\text{Leftover group } L^{(2)}}$$

New!

Our unifying framework can handle

- many arbitrary, incomplete partitions,
- consisting of possibly-overlapping groups

$$\underbrace{\overbrace{P_1}^{u_1^{(1)},w_1^{(1)}},\ldots,\overbrace{P_{n_1}}^{u_{n_1}^{(1)},w_{n_1}^{(1)}}}_{\text{Group 1, weights } u_1^{(2)},w_1^{(2)}},\ldots,\underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weights ...}},\ldots,\overbrace{P_{n-\ell_2}}^{\text{weights ...}}}_{\text{Group } G\text{, weights } u_G^{(2)},w_G^{(2)}} \Big| \underbrace{\overbrace{P_{n-\ell_2+1}}^{\text{weights ...}},\ldots,\overbrace{P_n}^{u_n^{(1)},w_n^{(1)}}}_{\text{Leftover group } L^{(2)}}$$

New!

Our unifying framework can handle

- many arbitrary, incomplete partitions,
- consisting of possibly-overlapping groups
- with group-level prior+penalty weights, along with

$$\underbrace{\overbrace{P_1}^{u_1^{(1)},w_1^{(1)}}, \dots, \overbrace{P_{n_1}}^{u_{n_1}^{(1)},w_{n_1}^{(1)}}}_{\text{Group 1, weights } u_1^{(2)},w_1^{(2)}}, \dots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weights ...}}, \dots, \overbrace{P_{n-\ell_2}}^{\text{weights ...}}}_{\text{Group } G\text{, weights } u_G^{(2)},w_G^{(2)}} \;\Big|\; \underbrace{\overbrace{P_{n-\ell_2+1}}^{\text{weights ...}}, \dots, \overbrace{P_n}^{u_n^{(1)},w_n^{(1)}}}_{\text{Leftover group } L^{(2)}}$$

New!

Our unifying framework can handle

- many arbitrary, incomplete partitions,
- consisting of possibly-overlapping groups
- with group-level prior+penalty weights, along with
- group-level adaptivity* or reshaping**.

$$\underbrace{\overbrace{P_1}^{u_1^{(1)},w_1^{(1)}}, \ldots, \overbrace{P_{n_1}}^{u_{n_1}^{(1)},w_{n_1}^{(1)}}}_{\text{Group 1, weights } u_1^{(2)},w_1^{(2)}}, \ldots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weights ...}}, \ldots, \overbrace{P_{n-\ell_2}}^{\text{weights ...}}}_{\text{Group } G\text{, weights } u_G^{(2)},w_G^{(2)}} \;\Big|\; \underbrace{\overbrace{P_{n-\ell_2+1}}^{\text{weights ...}}, \ldots, \overbrace{P_n}^{u_n^{(1)},w_n^{(1)}}}_{\text{Leftover group } L^{(2)}}$$

New!

Our unifying framework can handle

- many arbitrary, incomplete partitions,
- consisting of possibly-overlapping groups
- with group-level prior+penalty weights, along with
- group-level adaptivity* or reshaping**.

$$\underbrace{\overbrace{P_1}^{u_1^{(1)},w_1^{(1)}}\ ,\dots,\ \overbrace{P_{n_1}}^{u_{n_1}^{(1)},w_{n_1}^{(1)}}}_{\text{Group 1, weights } u_1^{(2)},w_1^{(2)}}\ ,\dots,\underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weights ...}},\dots,\ \overbrace{P_{n-\ell_2}}^{\text{weights ...}}}_{\text{Group } G\text{, weights } u_G^{(2)},w_G^{(2)}}\ \Bigg|\ \boxed{\underbrace{\overbrace{P_{n-\ell_2+1}}^{\text{weights ...}},\dots,\ \overbrace{P_n}^{u_n^{(1)},w_n^{(1)}}}_{\text{Leftover group } L^{(2)}}}$$

Using the p-filter algorithm results in

New!

Our unifying framework can handle

- many arbitrary, incomplete partitions,
- consisting of possibly-overlapping groups
- with group-level prior+penalty weights, along with
- group-level adaptivity* or reshaping**.

$$\underbrace{\overbrace{P_1}^{u_1^{(1)},w_1^{(1)}},\dots,\overbrace{P_{n_1}}^{u_{n_1}^{(1)},w_{n_1}^{(1)}}}_{\text{Group 1, weights } u_1^{(2)},w_1^{(2)}},\dots,\underbrace{\overbrace{P_{n_1+\dots+n_{G-1}+1}}^{\text{weights ...}},\dots,\overbrace{P_{n-\ell_2}}^{\text{weights ...}}}_{\text{Group } G\text{, weights } u_G^{(2)},w_G^{(2)}} \Bigg| \underbrace{\overbrace{P_{n-\ell_2+1}}^{\text{weights ...}},\dots,\overbrace{P_n}^{u_n^{(1)},w_n^{(1)}}}_{\text{Leftover group } L^{(2)}}$$

Using the p-filter algorithm results in
- an internally consistent set of rejections,

New!

Our unifying framework can handle

- many arbitrary, incomplete partitions,
- consisting of possibly-overlapping groups
- with group-level prior+penalty weights, along with
- group-level adaptivity* or reshaping**.

$$\underbrace{\overbrace{P_1}^{u_1^{(1)},w_1^{(1)}}\;, \ldots ,\; \overbrace{P_{n_1}}^{u_{n_1}^{(1)},w_{n_1}^{(1)}}}_{\text{Group 1, weights } u_1^{(2)},w_1^{(2)}}, \ldots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weights ...}}, \ldots, \overbrace{P_{n-\ell_2}}^{\text{weights ...}}}_{\text{Group } G\text{, weights } u_G^{(2)},w_G^{(2)}} \;\Big|\; \underbrace{\overbrace{P_{n-\ell_2+1}}^{\text{weights ...}}, \ldots, \overbrace{P_n}^{u_n^{(1)},w_n^{(1)}}}_{\text{Leftover group } L^{(2)}}$$

Using the p-filter algorithm results in
- an internally consistent set of rejections,
- *simultaneously* controls weighted FDR for all partitions,

New!

Our unifying framework can handle

- many arbitrary, incomplete partitions,
- consisting of possibly-overlapping groups
- with group-level prior+penalty weights, along with
- group-level adaptivity* or reshaping**.

$$\underbrace{\overbrace{P_1}^{u_1^{(1)},w_1^{(1)}}, \ldots, \overbrace{P_{n_1}}^{u_{n_1}^{(1)},w_{n_1}^{(1)}}}_{\text{Group 1, weights } u_1^{(2)},w_1^{(2)}}, \ldots, \underbrace{\overbrace{P_{n_1+\cdots+n_{G-1}+1}}^{\text{weights } \ldots}, \ldots, \overbrace{P_{n-\ell_2}}^{\text{weights } \ldots}}_{\text{Group } G\text{, weights } u_G^{(2)},w_G^{(2)}} \Bigg| \underbrace{\overbrace{P_{n-\ell_2+1}}^{\text{weights } \ldots}, \ldots, \overbrace{P_n}^{u_n^{(1)},w_n^{(1)}}}_{\text{Leftover group } L^{(2)}}$$

Using the p-filter algorithm results in

- an internally consistent set of rejections,
- *simultaneously* controls weighted FDR for all partitions,
- under independence*, positive or arbitrary** dependence.

New!

Algorithm 1 The `p-filter` for multi-layer FDR control

Input: A vector of p-values $P \in [0,1]^n$;
M possibly incomplete partitions of possibly overlapping groups;
M target FDR levels $\alpha_1, \dots, \alpha_M$;
M sets of prior weights and/or penalty weights, one pair of weights for each group in each partition;
M thresholds for adaptive null proportion estimation $\lambda_1, \dots, \lambda_M$.

Initialize: Set $k_m = G_m$, and $\widehat{\pi}_m$ as in definition (7.2).

repeat

for $m = 1, \dots, M$ **do**

Update the mth vector: defining $\widehat{\mathcal{S}}_m(\vec{k})$ as in equation (7.4), let

$$k_m \leftarrow \max\left\{k'_m \in [0, G_m] : \sum_{g \in \widehat{\mathcal{S}}_m(k_1,\dots,k_{m-1},k'_m,k_{m+1},\dots,k_M)} u_g^{(m)} \geq k'_m\right\} \tag{7.8}$$

end for

until the vectors $k_1, \dots, k_M$ are all unchanged for one full cycle.

Output: Adaptive vectors $\widehat{k}_1 = k_1, \dots, \widehat{k}_M = k_M$.

Algorithm 1 The `p-filter` for multi-layer FDR control

Input: A vector of p-values $P \in [0,1]^n$;
M possibly incomplete partitions of possibly overlapping groups;
M target FDR levels $\alpha_1, \dots, \alpha_M$;
M sets of prior weights and/or penalty weights, one pair of weights for each group in each partition;
M thresholds for adaptive null proportion estimation $\lambda_1, \dots, \lambda_M$.

Initialize: Set $k_m = G_m$, and $\widehat{\pi}_m$ as in definition (7.2).

repeat

for $m = 1, \dots, M$ **do**

Update the mth vector: defining $\widehat{\mathcal{S}}_m(\vec{k})$ as in equation (7.4), let

$$k_m \leftarrow \max\left\{k'_m \in [0, G_m] : \sum_{g \in \widehat{\mathcal{S}}_m(k_1, \dots, k_{m-1}, k'_m, k_{m+1}, \dots, k_M)} u_g^{(m)} \geq k'_m\right\} \tag{7.8}$$

end for

until the vectors $k_1, \dots, k_M$ are all unchanged for one full cycle.

Output: Adaptive vectors $\widehat{k}_1 = k_1, \dots, \widehat{k}_M = k_M$.

Algorithm 1 The `p-filter` for multi-layer FDR control

Input: A vector of p-values $P \in [0,1]^n$;

M possibly incomplete partitions of possibly overlapping groups;

M target FDR levels $\alpha_1, \dots, \alpha_M$;

M sets of prior weights and/or penalty weights, one pair of weights for each group in each partition;

M thresholds for adaptive null proportion estimation $\lambda_1, \dots, \lambda_M$.

Initialize: Set $k_m = G_m$, and $\widehat{\pi}_m$ as in definition (7.2).

repeat

for $m = 1, \dots, M$ **do**

Update the mth vector: defining $\widehat{\mathcal{S}}_m(\vec{k})$ as in equation (7.4), let

$$k_m \leftarrow \max \left\{ k'_m \in [0, G_m] : \sum_{g \in \widehat{\mathcal{S}}_m(k_1, \dots, k_{m-1}, k'_m, k_{m+1}, \dots, k_M)} u_g^{(m)} \geq k'_m \right\} \tag{7.8}$$

end for

until the vectors $k_1, \dots, k_M$ are all unchanged for one full cycle.

Output: Adaptive vectors $\widehat{k}_1 = k_1, \dots, \widehat{k}_M = k_M$.

Algorithm 1 The `p-filter` for multi-layer FDR control

Input: A vector of p-values $P \in [0,1]^n$;

M possibly incomplete partitions of possibly overlapping groups;

M target FDR levels $\alpha_1, \ldots, \alpha_M$;

M sets of prior weights and/or penalty weights, one pair of weights for each group in each partition;

M thresholds for adaptive null proportion estimation $\lambda_1, \ldots, \lambda_M$.

Initialize: Set $k_m = G_m$, and $\widehat{\pi}_m$ as in definition (7.2).

repeat

  for $m = 1, \ldots, M$ **do**

    Update the mth vector: defining $\widehat{\mathcal{S}}_m(\vec{k})$ as in equation (7.4), let

$$k_m \leftarrow \max\left\{ k'_m \in [0, G_m] : \sum_{g \in \widehat{\mathcal{S}}_m(k_1, \ldots, k_{m-1}, k'_m, k_{m+1}, \ldots, k_M)} u_g^{(m)} \geq k'_m \right\} \tag{7.8}$$

  end for

until the vectors $k_1, \ldots, k_M$ are all unchanged for one full cycle.

Output: Adaptive vectors $\widehat{k}_1 = k_1, \ldots, \widehat{k}_M = k_M$.

Algorithm 1 The `p-filter` for multi-layer FDR control

Input: A vector of p-values $P \in [0,1]^n$;

M possibly incomplete partitions of possibly overlapping groups;

M target FDR levels $\alpha_1, \dots, \alpha_M$;

M sets of prior weights and/or penalty weights, one pair of weights for each group in each partition;

M thresholds for adaptive null proportion estimation $\lambda_1, \dots, \lambda_M$.

Initialize: Set $k_m = G_m$, and $\widehat{\pi}_m$ as in definition (7.2).

repeat

for $m = 1, \dots, M$ **do**

Update the mth vector: defining $\widehat{\mathcal{S}}_m(\vec{k})$ as in equation (7.4), let

$$k_m \leftarrow \max\left\{k'_m \in [0, G_m] : \sum_{g \in \widehat{\mathcal{S}}_m(k_1, \dots, k_{m-1}, k'_m, k_{m+1}, \dots, k_M)} u_g^{(m)} \geq k'_m\right\} \tag{7.8}$$

end for

until the vectors $k_1, \dots, k_M$ are all unchanged for one full cycle.

Output: Adaptive vectors $\widehat{k}_1 = k_1, \dots, \widehat{k}_M = k_M$.

Algorithm 1 The `p-filter` for multi-layer FDR control

Input: A vector of p-values $P \in [0,1]^n$;

M possibly incomplete partitions of possibly overlapping groups;

M target FDR levels $\alpha_1, \dots, \alpha_M$;

M sets of prior weights and/or penalty weights, one pair of weights for each group in each partition;

M thresholds for adaptive null proportion estimation $\lambda_1, \dots, \lambda_M$.

Initialize: Set $k_m = G_m$, and $\widehat{\pi}_m$ as in definition (7.2).

repeat

for $m = 1, \dots, M$ **do**

Update the mth vector: defining $\widehat{\mathcal{S}}_m(\vec{k})$ as in equation (7.4), let

$$k_m \leftarrow \max\left\{k'_m \in [0, G_m] : \sum_{g \in \widehat{\mathcal{S}}_m(k_1,\dots,k_{m-1},k'_m,k_{m+1},\dots,k_M)} u_g^{(m)} \geq k'_m \right\} \tag{7.8}$$

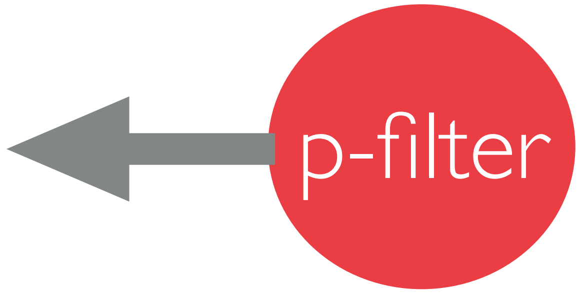
end for

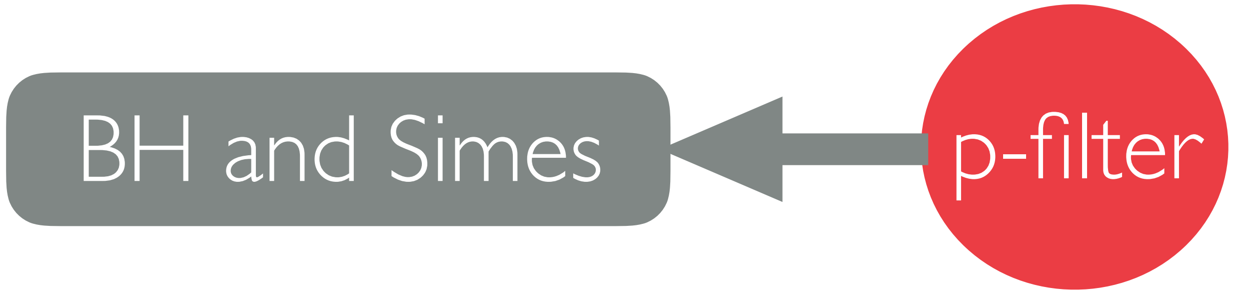
until the vectors $k_1, \dots, k_M$ are all unchanged for one full cycle.

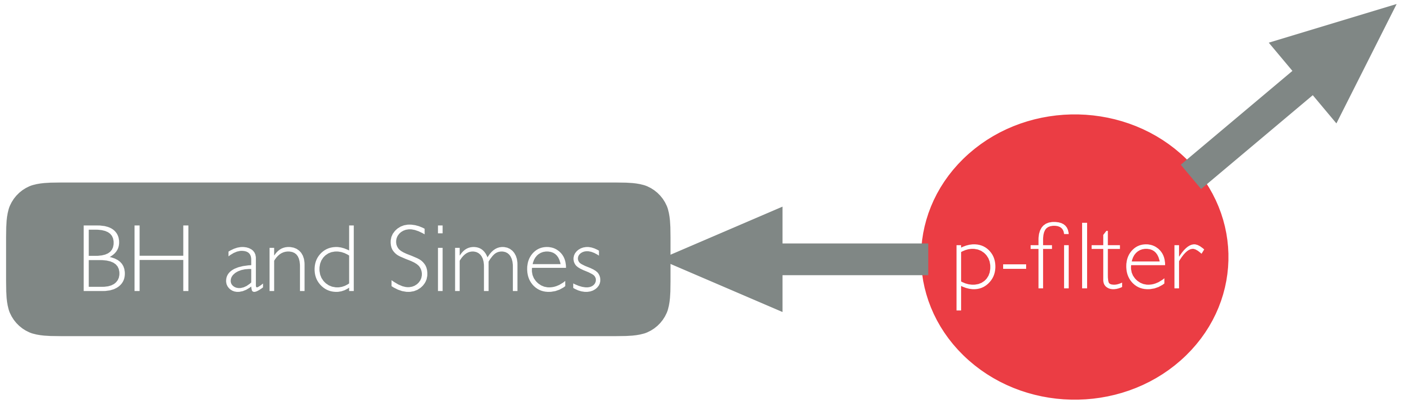
Output: Adaptive vectors $\widehat{k}_1 = k_1, \dots, \widehat{k}_M = k_M$.

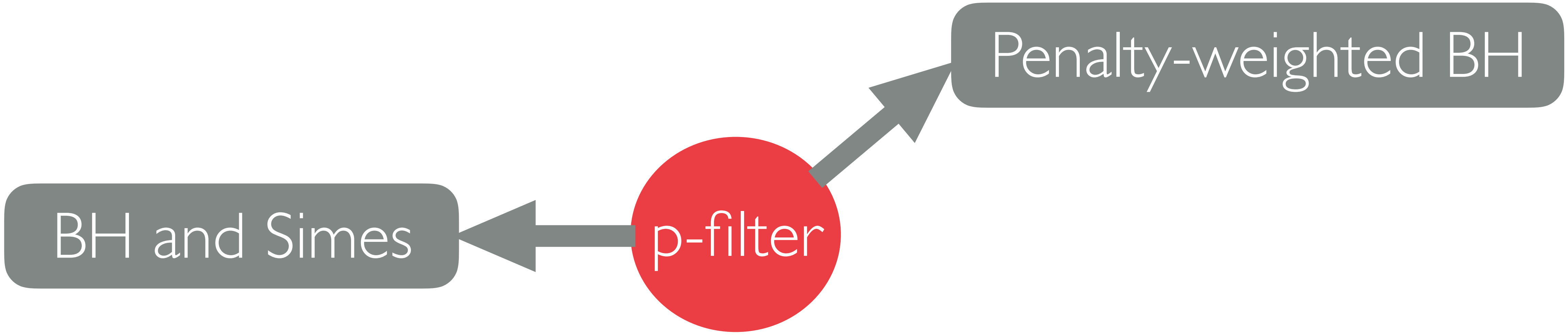
Theorem : The algorithm returns a set of rejected hypotheses and groups that are internally consistent, and weighted group FDR is simultaneously controlled for all partitions, under independence, positive dependence or arbitrary dependence.

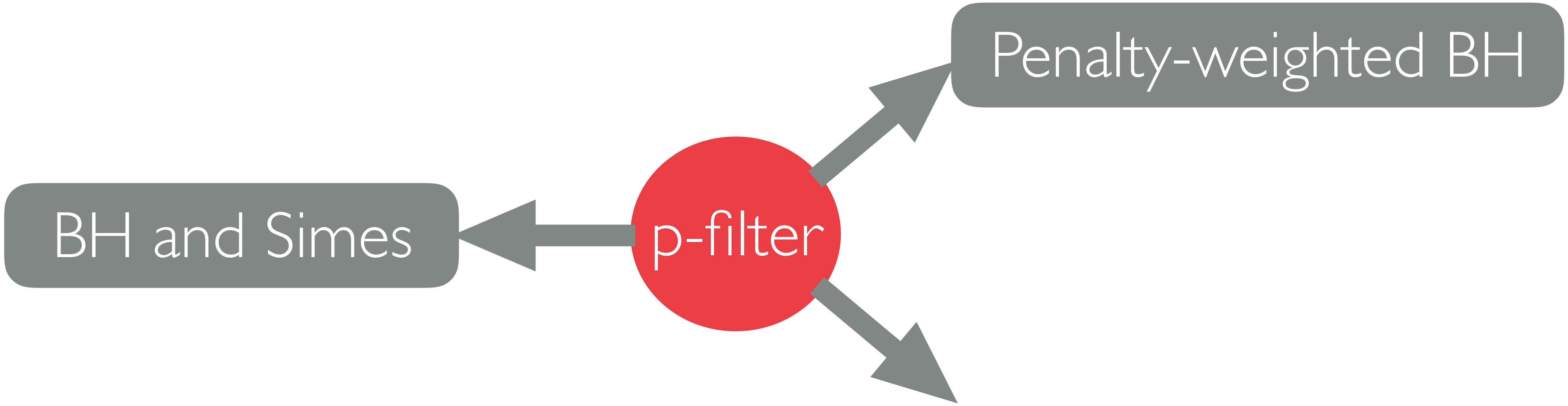

p-filter

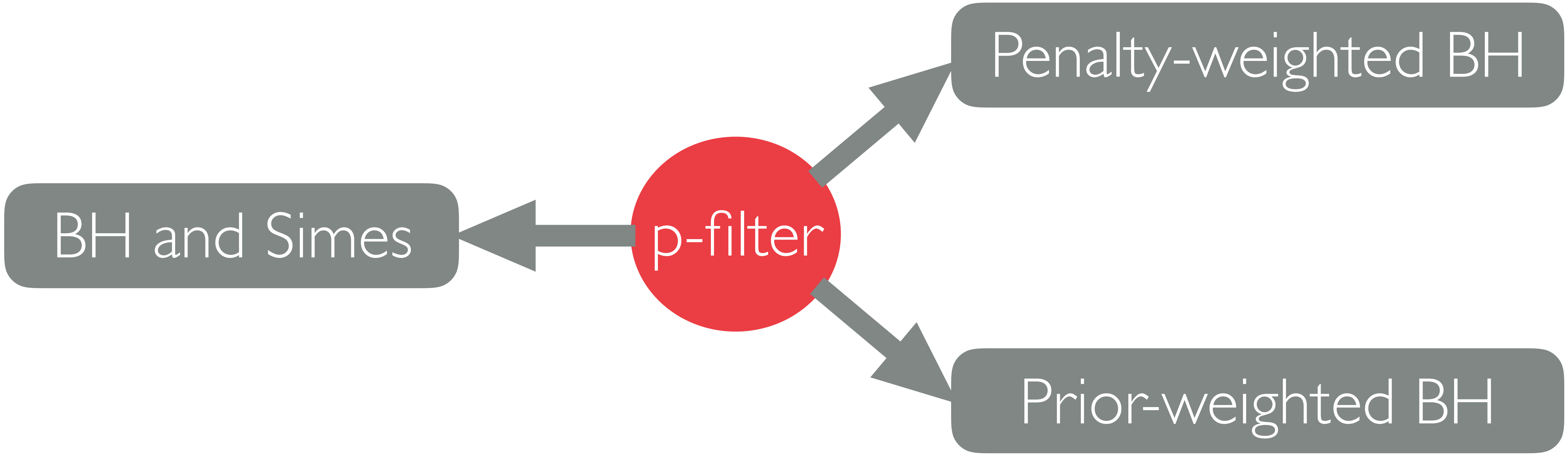

p-filter


BH and Simes
p-filter

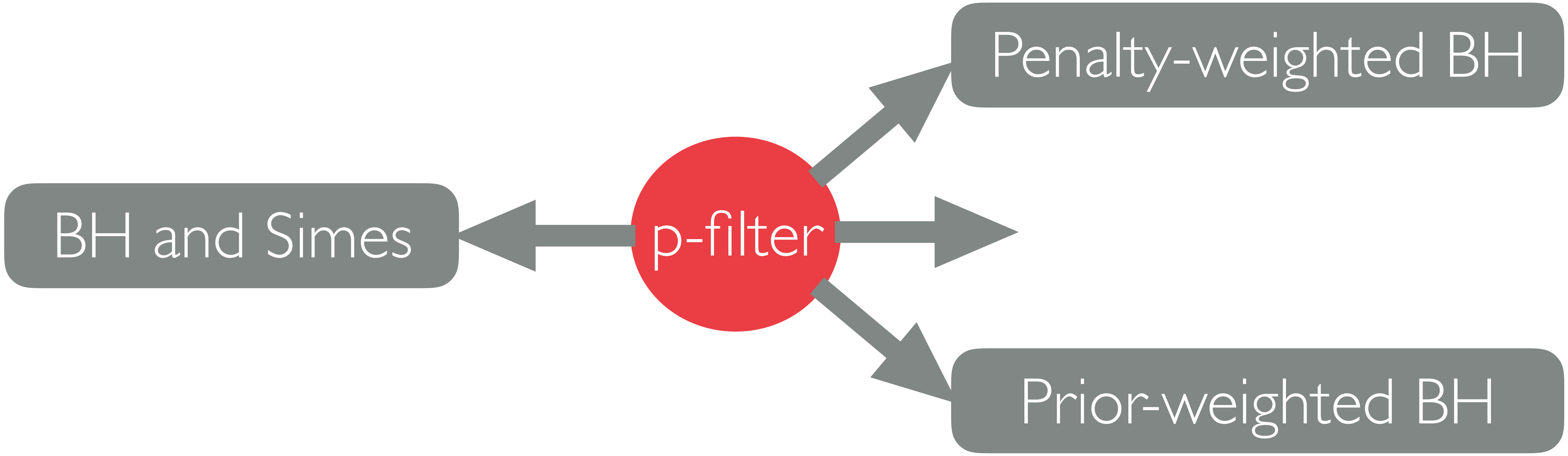

BH and Simes
p-filter

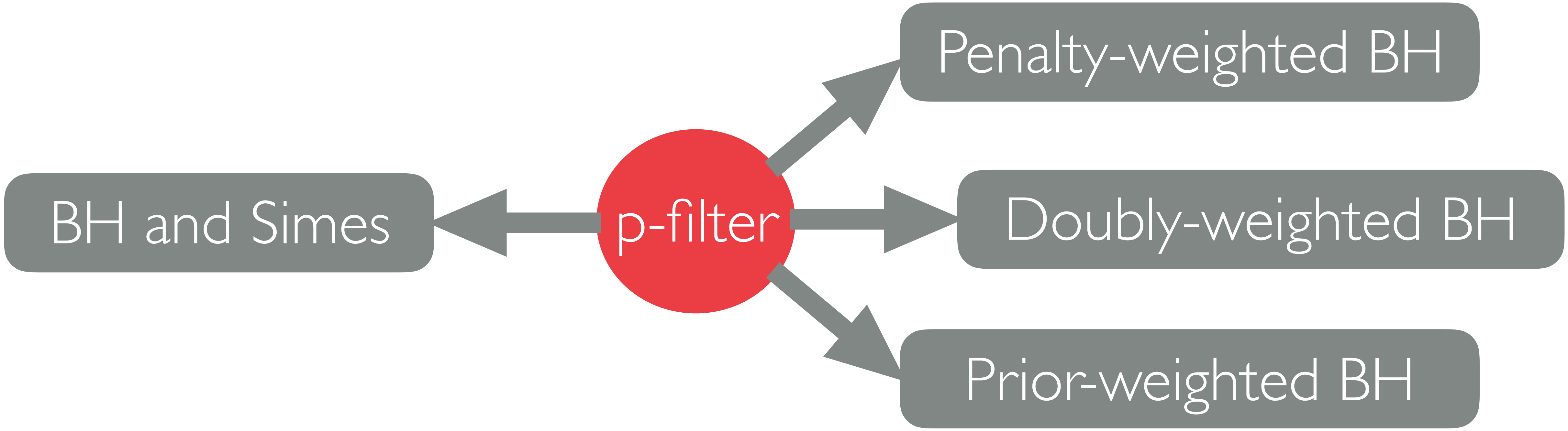

Penalty-weighted BH
BH and Simes
p-filter


Penalty-weighted BH
BH and Simes
p-filter

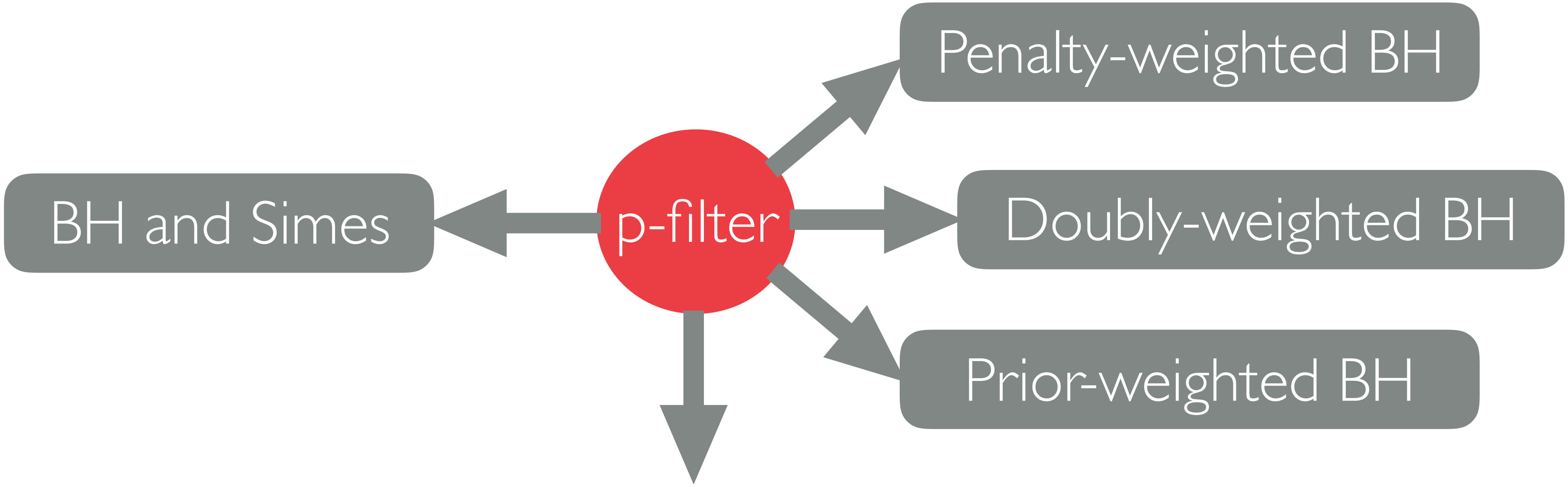


Penalty-weighted BH
BH and Simes
p-filter
Prior-weighted BH

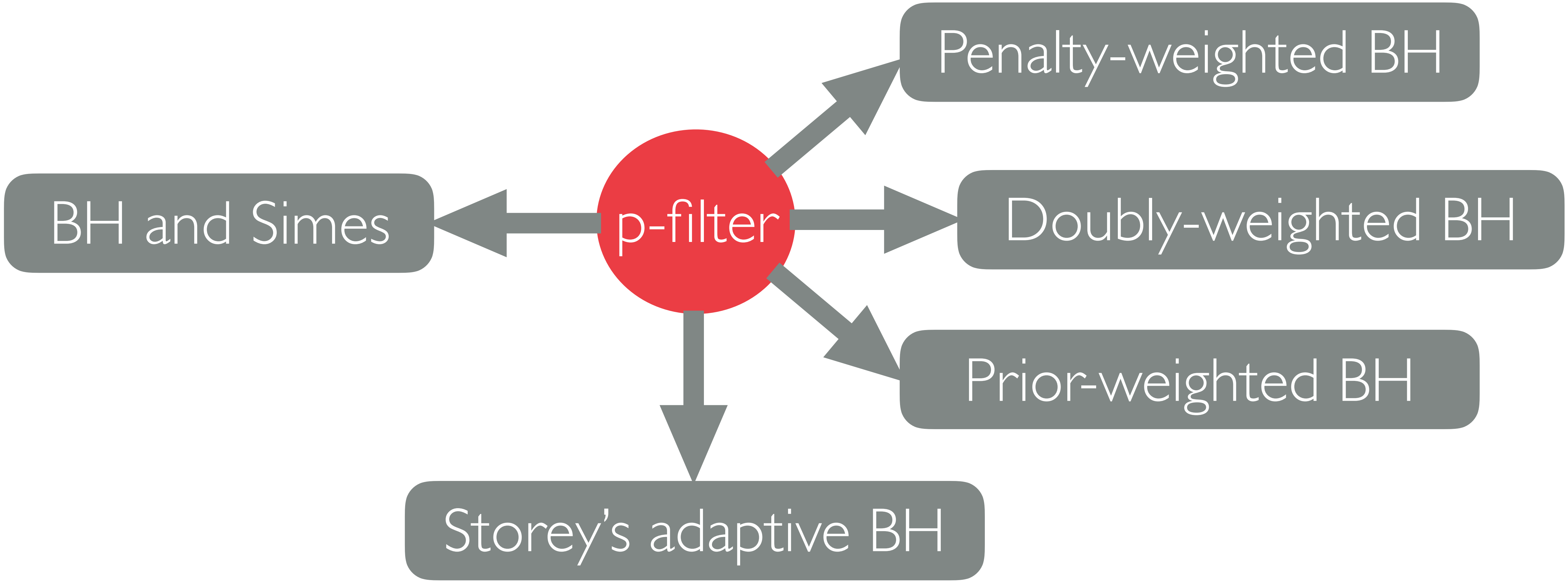

Penalty-weighted BH
BH and Simes
p-filter
Prior-weighted BH

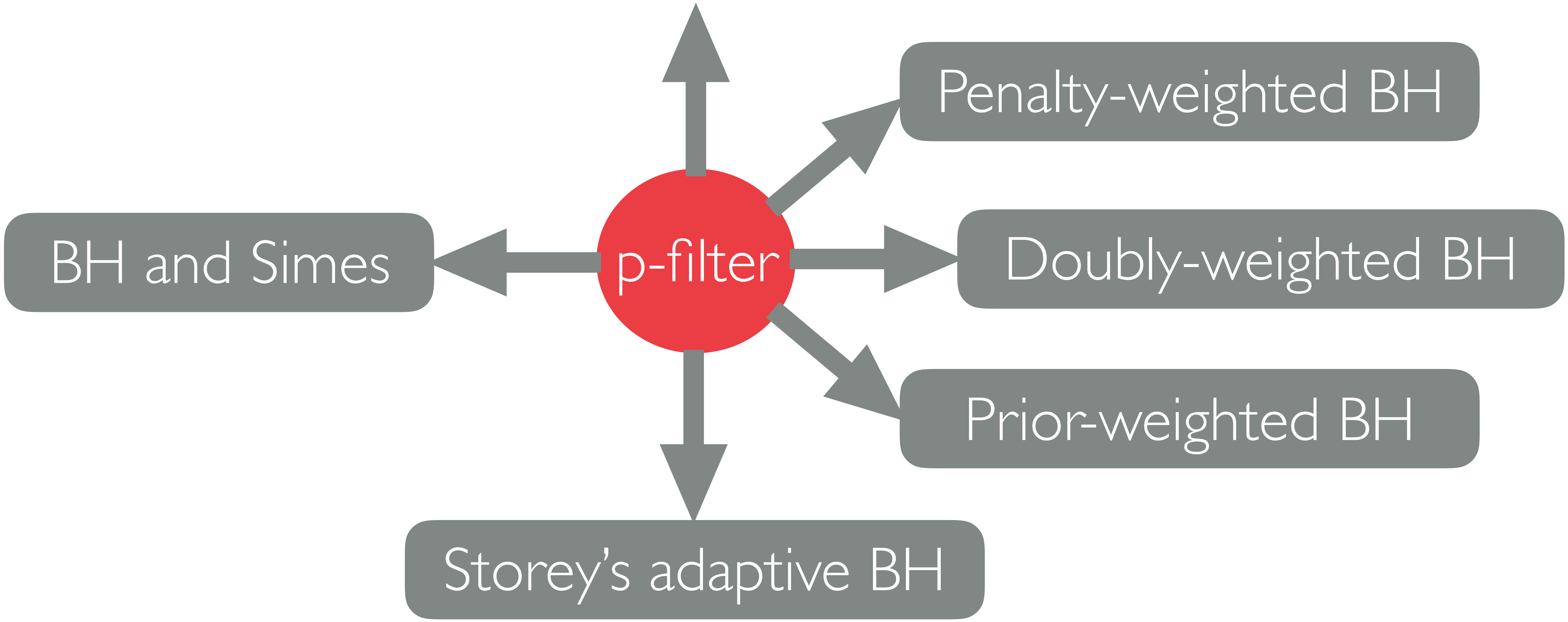


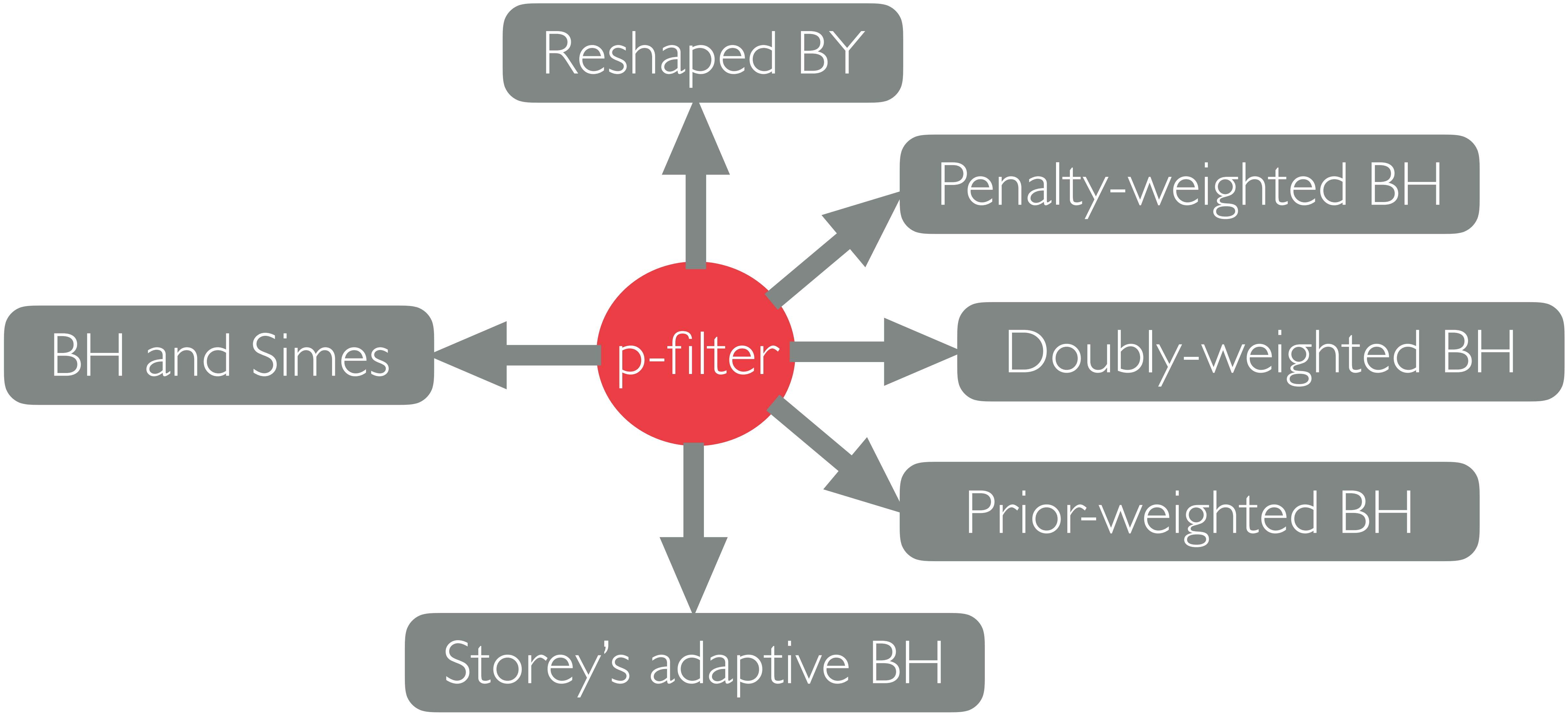
Penalty-weighted BH
BH and Simes
p-filter
Doubly-weighted BH
Prior-weighted BH



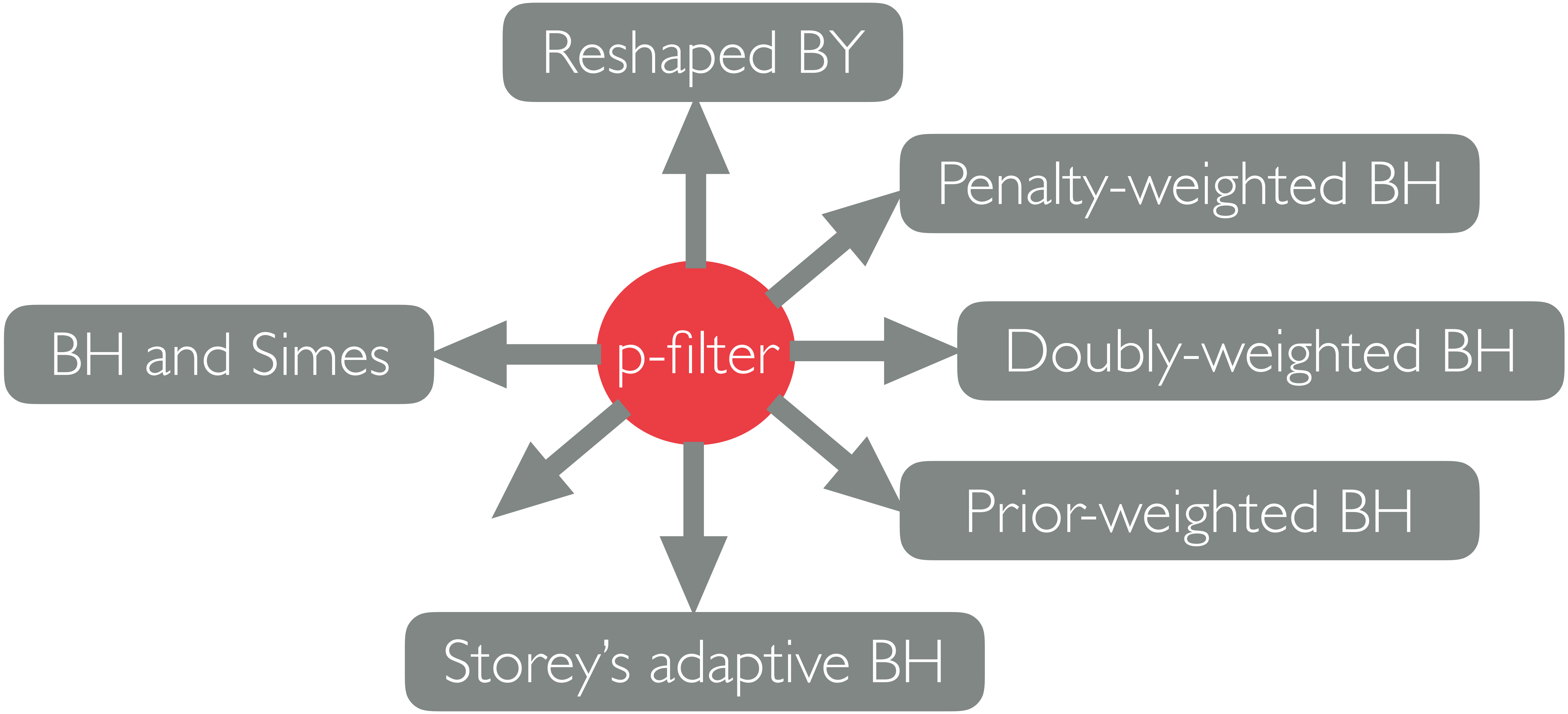
Penalty-weighted BH
BH and Simes
p-filter
Doubly-weighted BH
Prior-weighted BH


p-filter
BH and Simes
Penalty-weighted BH
Doubly-weighted BH
Prior-weighted BH
Storey's adaptive BH

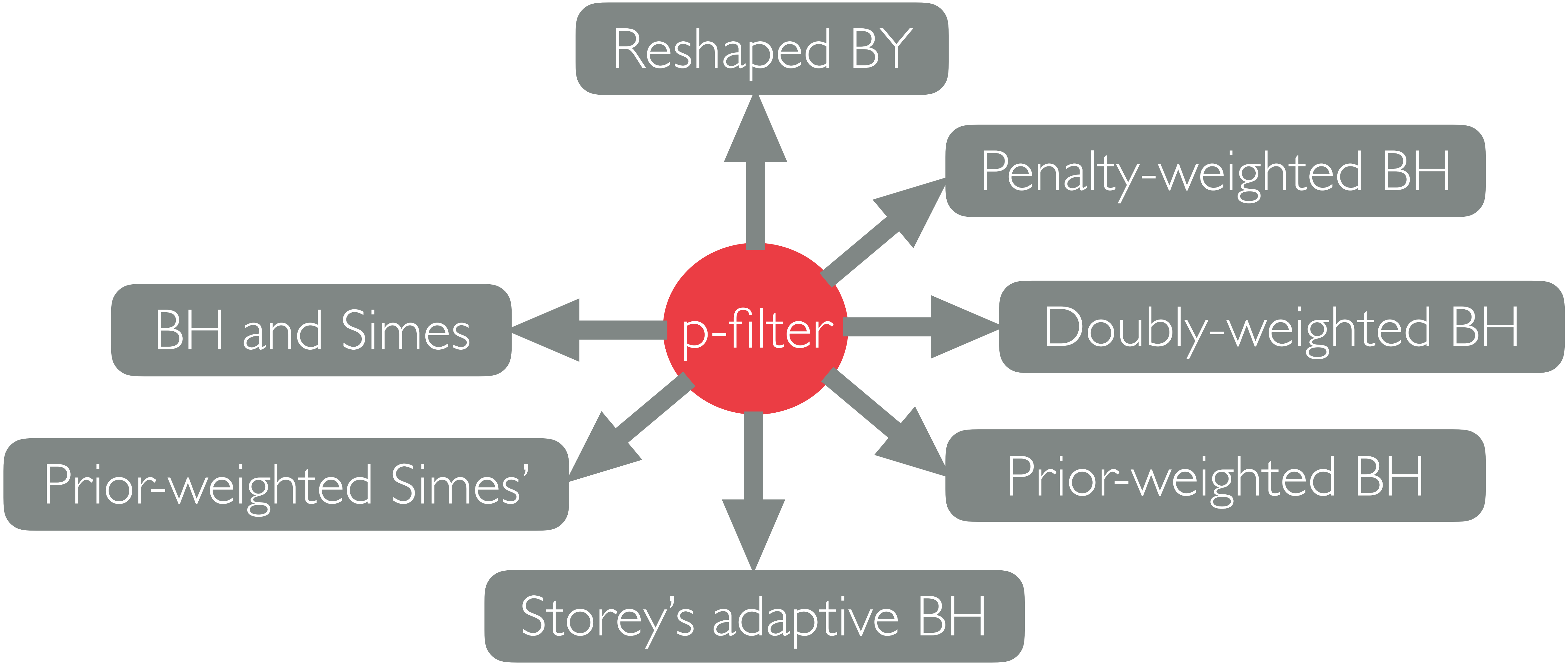

Penalty-weighted BH
BH and Simes
p-filter
Doubly-weighted BH
Prior-weighted BH
Storey's adaptive BH

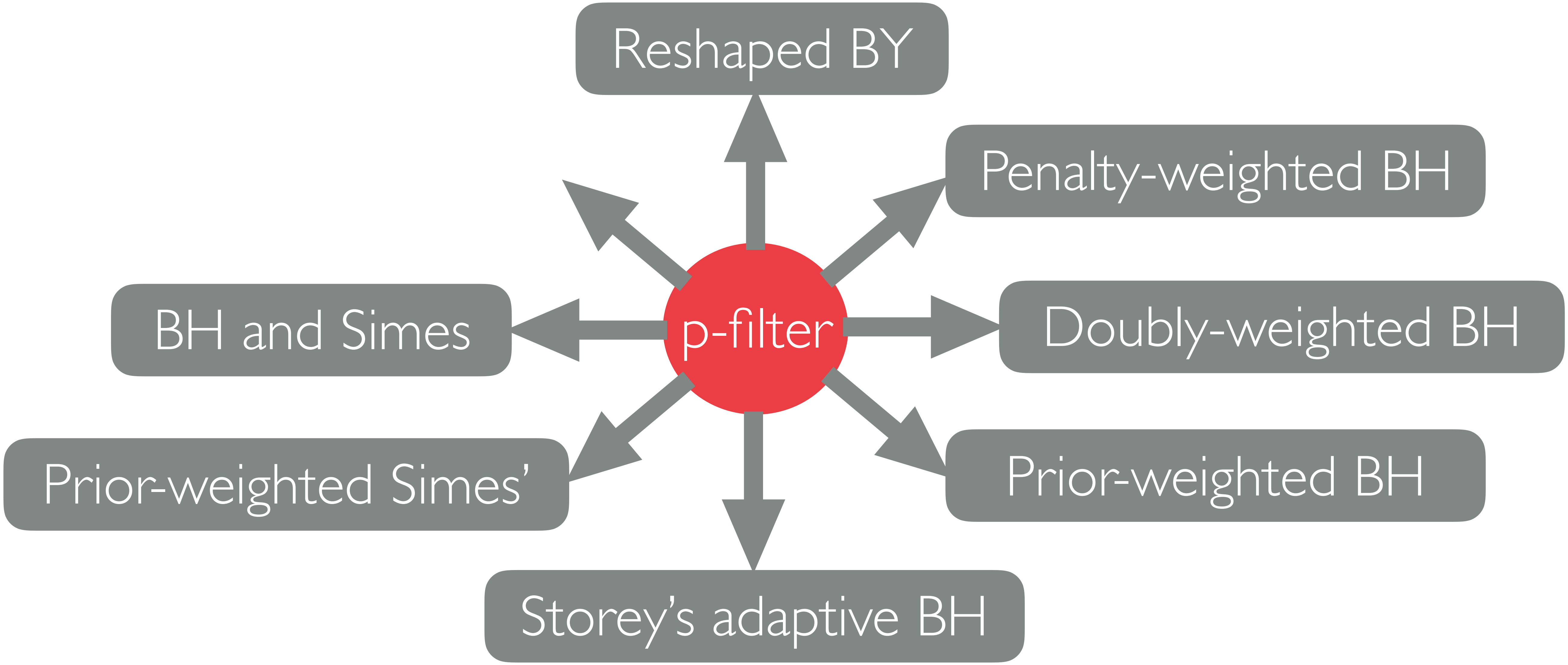


Reshaped BY
Penalty-weighted BH
BH and Simes
p-filter
Doubly-weighted BH
Prior-weighted BH
Storey's adaptive BH

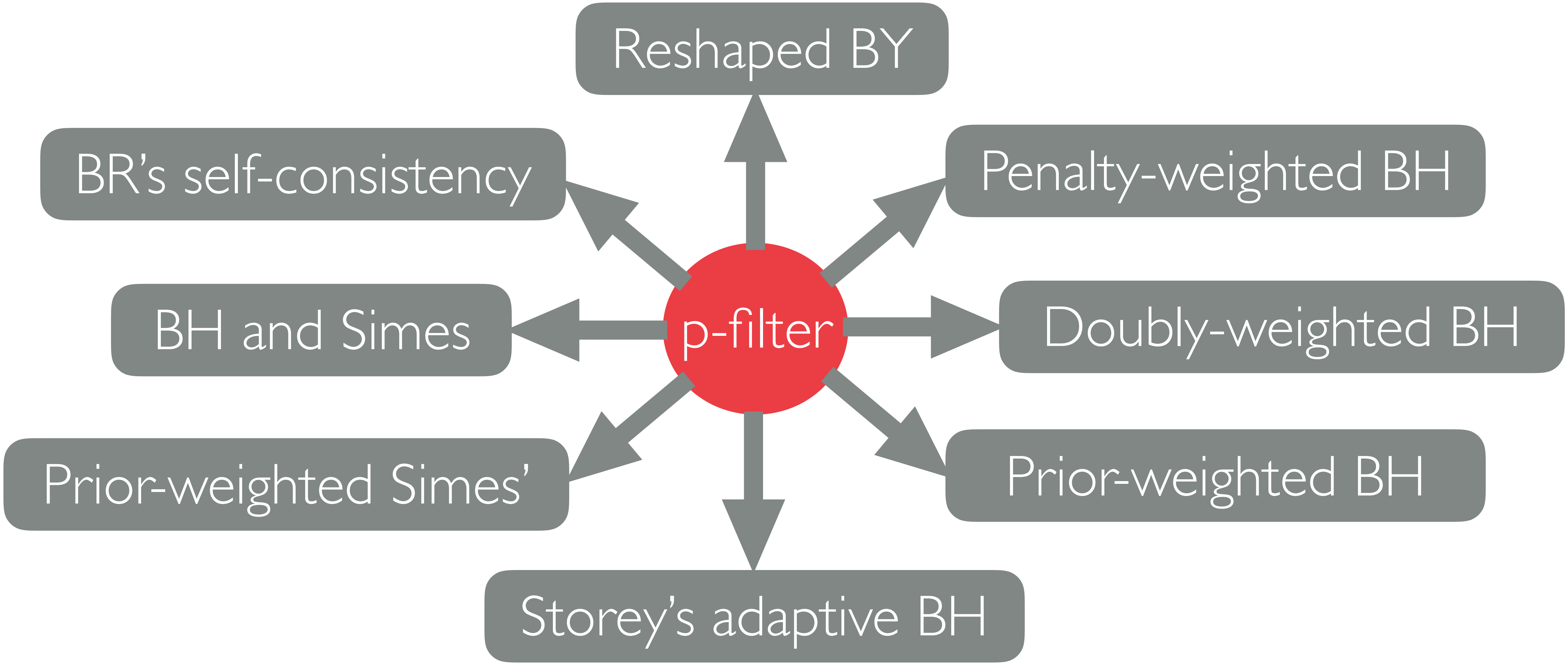


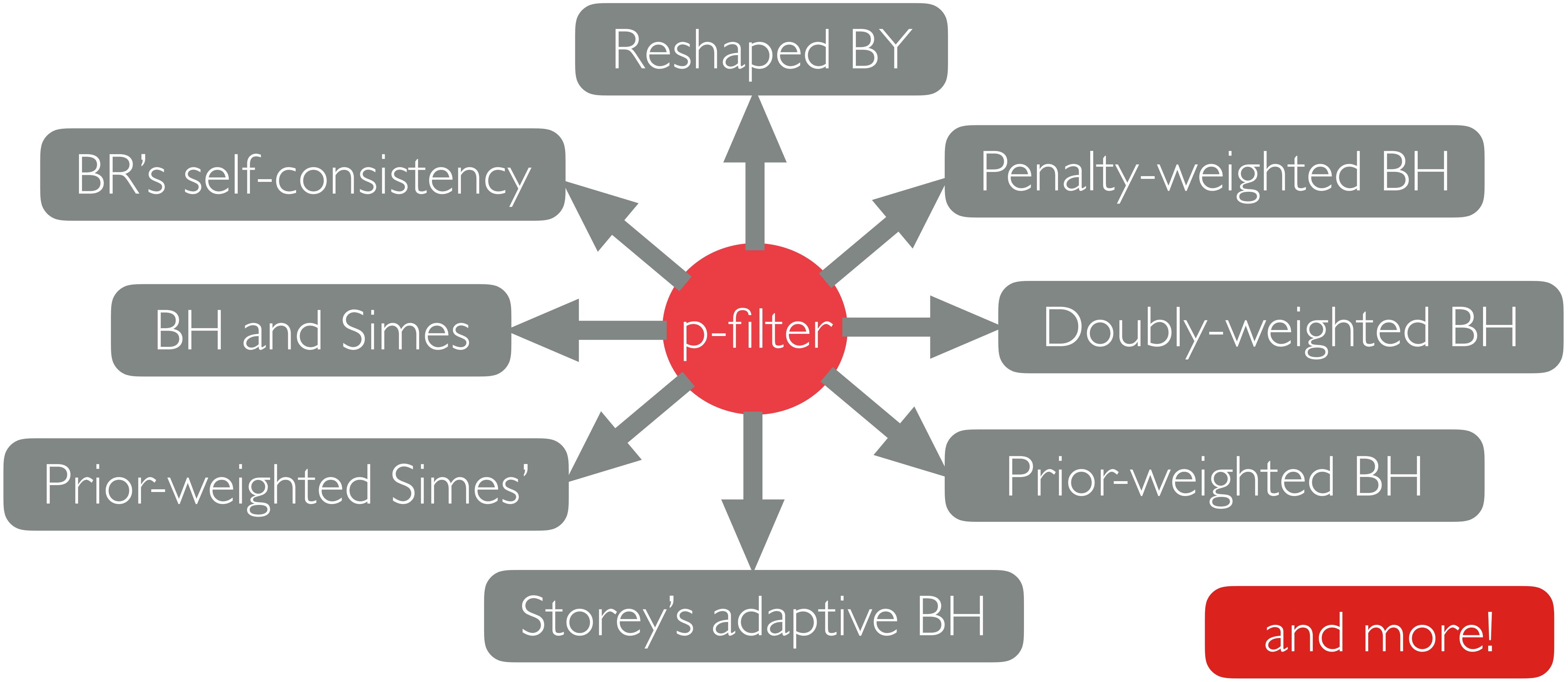
Reshaped BY
Penalty-weighted BH
BH and Simes
p-filter
Doubly-weighted BH
Prior-weighted BH
Storey's adaptive BH

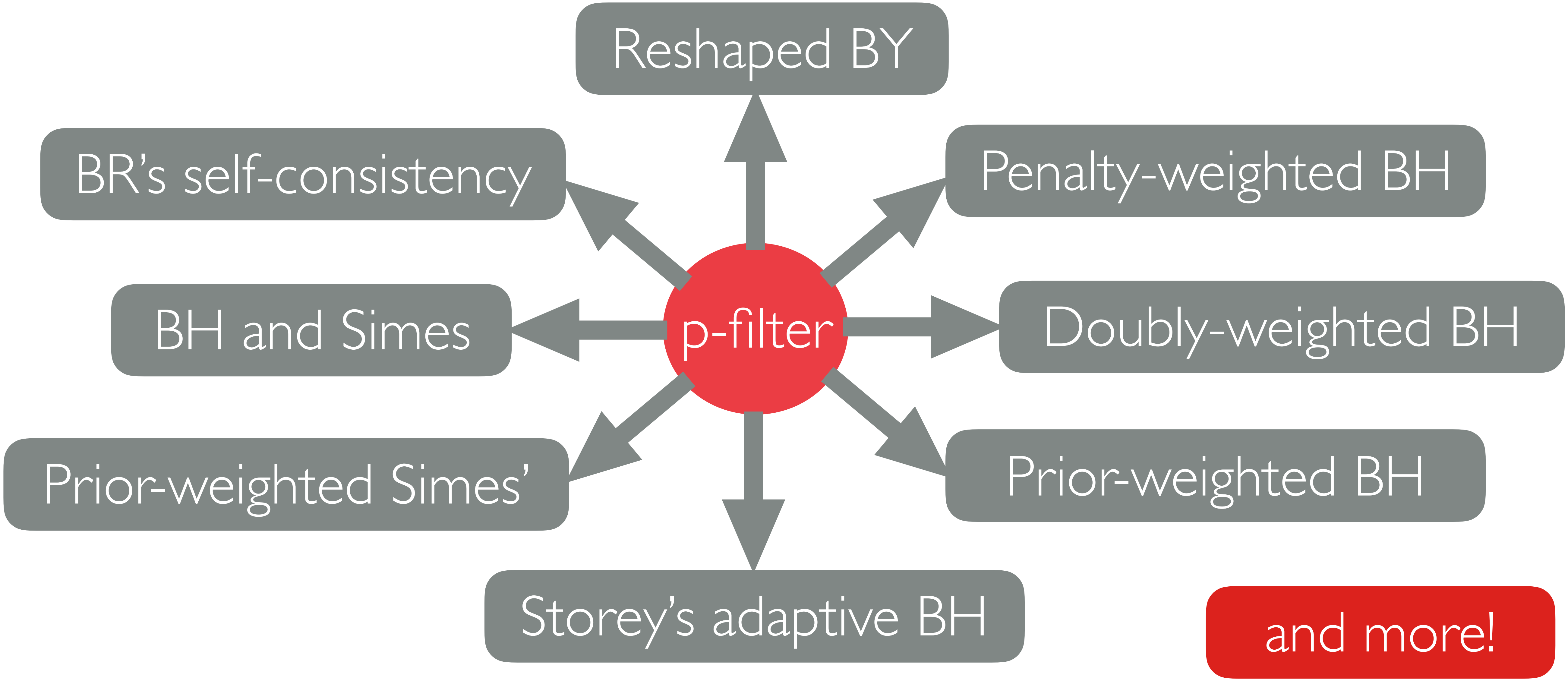

Reshaped BY
Penalty-weighted BH
BH and Simes
p-filter
Doubly-weighted BH
Prior-weighted Simes'
Prior-weighted BH
Storey's adaptive BH



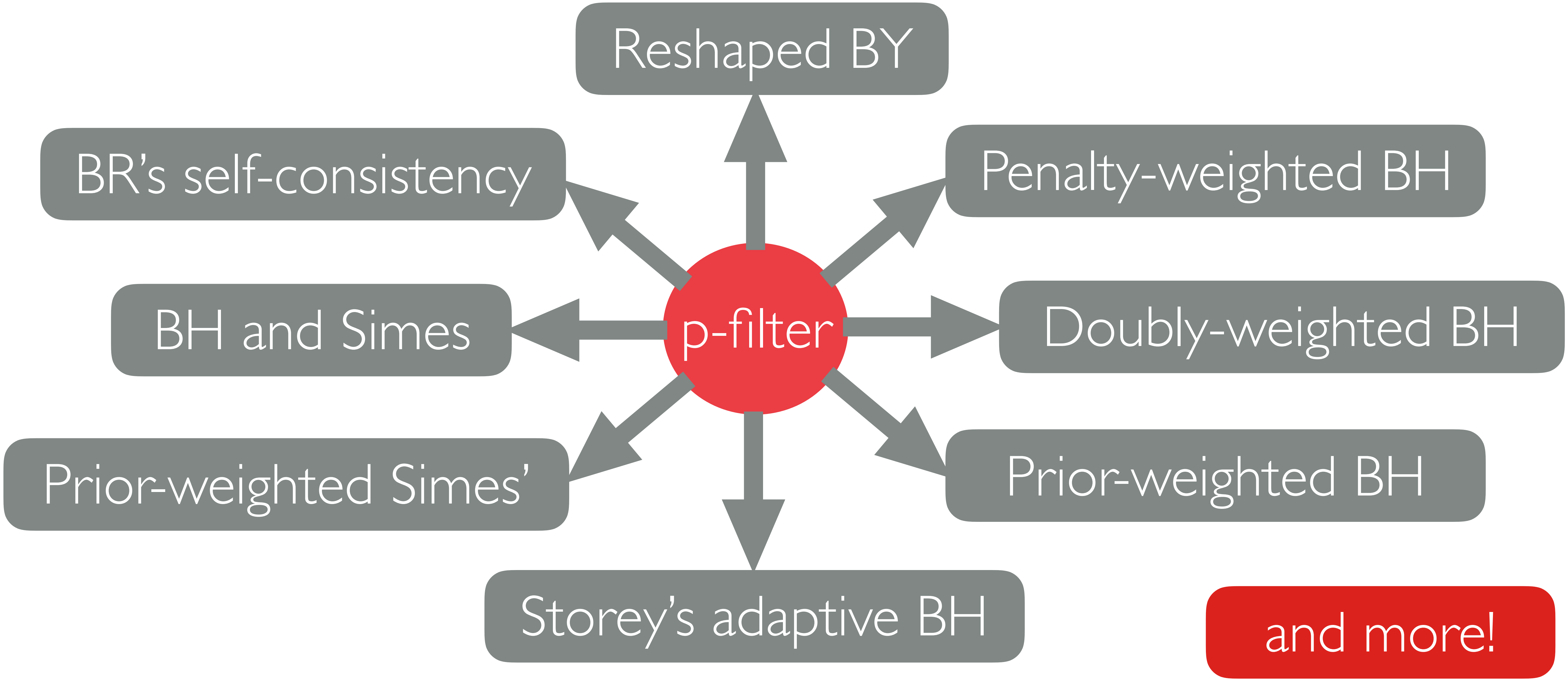
Reshaped BY
Penalty-weighted BH
BH and Simes
p-filter
Doubly-weighted BH
Prior-weighted Simes'
Prior-weighted BH
Storey's adaptive BH


Reshaped BY
BR's self-consistency
Penalty-weighted BH
BH and Simes
p-filter
Doubly-weighted BH
Prior-weighted Simes'
Prior-weighted BH
Storey's adaptive BH


Reshaped BY
BR's self-consistency
Penalty-weighted BH
BH and Simes
p-filter
Doubly-weighted BH
Prior-weighted Simes'
Prior-weighted BH
Storey's adaptive BH
and more!



A unified treatment of multiple testing
with prior knowledge (on Arxiv)



A unified treatment of multiple testing with prior knowledge (on Arxiv)

Aaditya Ramdas
Rina F. Barber
Martin J. Wainwright
Michael I. Jordan