Controlled Discovery and Localization of Astronomical Point Sources via Bayesian Linear Programming (BLiP)

Lucas Janson

Harvard University Department of Statistics

International CHASC AstroStatistics Centre, May 17, 2022

Coauthor

Asher Spector (First-year PhD student at Stanford Statistics)

Outline

- Motivation
- Problem statement
- Methodological contribution: **Bayesian Linear Programming (BLiP)**
- Simulations
- Application to genetic fine-mapping
- Application to astronomical point-source detection

Astronomical point source detection

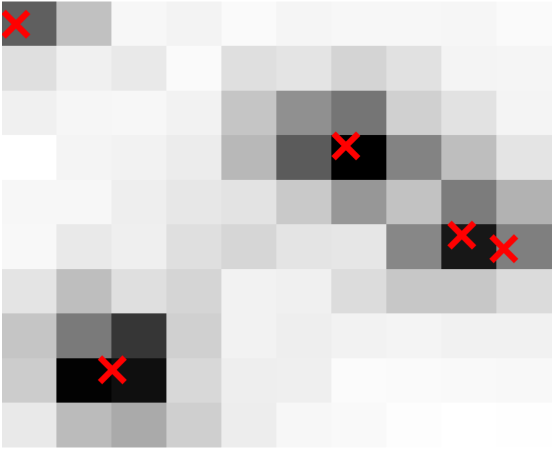


Figure: Cartoon of partial point source data

A simple approach

$\mathcal{L}$ = space of potential signal locations

Pre-specify partition of $\mathcal{L}$ into disjoint regions $G_1, \ldots, G_m$ and do selection

A simple approach

$\mathcal{L}$ = space of potential signal locations
Pre-specify partition of $\mathcal{L}$ into disjoint regions $G_1, \ldots, G_m$ and do selection

- Outputted regions disjoint

A simple approach

$\mathcal{L}$ = space of potential signal locations
Pre-specify partition of $\mathcal{L}$ into disjoint regions $G_1, \ldots, G_m$ and do selection

- Outputted regions disjoint
- Can control false discoveries via many existing methods (e.g., multiple testing on region p-values or Bayesian selection on region PIPs)

A simple approach

$\mathcal{L}$ = space of potential signal locations

Pre-specify partition of $\mathcal{L}$ into disjoint regions $G_1, \ldots, G_m$ and do selection

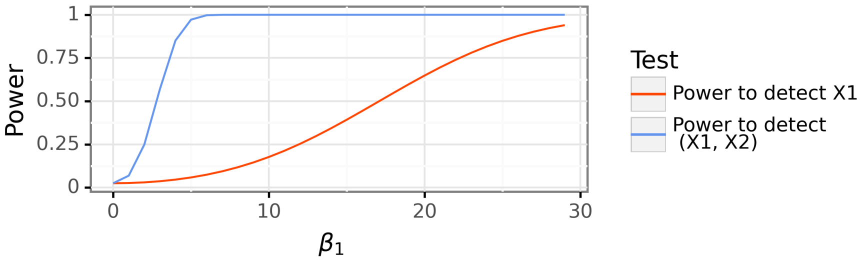
- Outputted regions disjoint
- Can control false discoveries via many existing methods (e.g., multiple testing on region p-values or Bayesian selection on region PIPs)
- But how to pre-specify partition?

A simple approach

$\mathcal{L}$ = space of potential signal locations
Pre-specify partition of $\mathcal{L}$ into disjoint regions $G_1, \ldots, G_m$ and do selection

- Outputted regions disjoint
- Can control false discoveries via many existing methods (e.g., multiple testing on region p-values or Bayesian selection on region PIPs)
- But how to pre-specify partition?

Toy example: Corr$(X_1, X_2) = 0.99$, $\quad Y = X_1\beta_1 + \varepsilon$; split (X_1, X_2) or no?



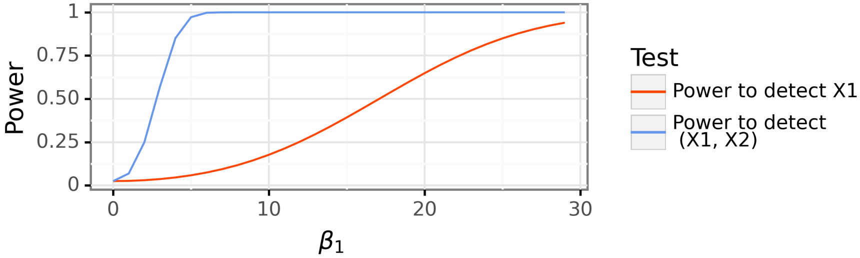
A simple approach

$\mathcal{L}$ = space of potential signal locations

Pre-specify partition of $\mathcal{L}$ into disjoint regions $G_1, \ldots, G_m$ and do selection

- Outputted regions disjoint
- Can control false discoveries via many existing methods (e.g., multiple testing on region p-values or Bayesian selection on region PIPs)
- But how to pre-specify partition?

Toy example: Corr$(X_1, X_2) = 0.99$, $\quad Y = X_1\beta_1 + \varepsilon$; split (X_1, X_2) or no?



Solution: **adaptively selected regions**

Informal problem statement

Don't want to narrow potential discovery regions until *after* seeing data

Informal problem statement

Don't want to narrow potential discovery regions until *after* seeing data

Want method that looks at the data and outputs regions $G_1, \ldots, G_R$ so as to:

Informal problem statement

Don't want to narrow potential discovery regions until *after* seeing data

Want method that looks at the data and outputs regions $G_1, \ldots, G_R$ so as to:

$$\begin{aligned} \max \quad & \mathbb{E}\left[\text{Power}(G_1, \ldots, G_R)\right] \\ \text{s.t.} \quad & \text{FDR} := \mathbb{E}\left[\frac{\#\{G_r \text{ containing no signal}\}}{\max(1, R)}\right] \leq q, \\ & G_1, \ldots, G_R \subset \mathcal{L} \text{ are disjoint.} \end{aligned}$$

Informal problem statement

Don't want to narrow potential discovery regions until *after* seeing data

Want method that looks at the data and outputs regions $G_1, \ldots, G_R$ so as to:

$$\begin{aligned} \max \quad & \mathbb{E}\left[\text{Power}(G_1, \ldots, G_R)\right] \\ \text{s.t.} \quad & \text{FDR} := \mathbb{E}\left[\frac{\#\{G_r \text{ containing no signal}\}}{\max(1, R)}\right] \leq q, \\ & G_1, \ldots, G_R \subset \mathcal{L} \text{ are disjoint.} \end{aligned}$$

What does high Power() look like?

Informal problem statement

Don't want to narrow potential discovery regions until *after* seeing data

Want method that looks at the data and outputs regions $G_1, \ldots, G_R$ so as to:

$$\begin{aligned} \max \quad & \mathbb{E}\left[\text{Power}(G_1, \ldots, G_R)\right] \\ \text{s.t.} \quad & \text{FDR} := \mathbb{E}\left[\frac{\#\{G_r \text{ containing no signal}\}}{\max(1, R)}\right] \leq q, \\ & G_1, \ldots, G_R \subset \mathcal{L} \text{ are disjoint.} \end{aligned}$$

What does high Power() look like?

- As many (true) discovered regions G_r as possible

Informal problem statement

Don't want to narrow potential discovery regions until *after* seeing data

Want method that looks at the data and outputs regions $G_1, \ldots, G_R$ so as to:

$$\begin{aligned} \max \quad & \mathbb{E}\left[\text{Power}(G_1, \ldots, G_R)\right] \\ \text{s.t.} \quad & \text{FDR} := \mathbb{E}\left[\frac{\#\{G_r \text{ containing no signal}\}}{\max(1, R)}\right] \leq q, \\ & G_1, \ldots, G_R \subset \mathcal{L} \text{ are disjoint.} \end{aligned}$$

What does high Power() look like?

- As many (true) discovered regions G_r as possible
- Discovered regions G_r should be as small as possible

Informal problem statement

Don't want to narrow potential discovery regions until *after* seeing data

Want method that looks at the data and outputs regions $G_1, \ldots, G_R$ so as to:

$$\begin{aligned} \max \quad & \mathbb{E}\left[\text{Power}(G_1, \ldots, G_R)\right] \\ \text{s.t.} \quad & \text{FDR} := \mathbb{E}\left[\frac{\#\{G_r \text{ containing no signal}\}}{\max(1, R)}\right] \leq q, \\ & G_1, \ldots, G_R \subset \mathcal{L} \text{ are disjoint.} \end{aligned}$$

What does high Power() look like?

- As many (true) discovered regions G_r as possible
- Discovered regions G_r should be as small as possible

Existing work: no formalization of what "power" means, so cannot optimize it

Valuing discovered regions

Define a weighting function $w(G)$ that measures value of discovering a group

Valuing discovered regions

Define a weighting function $w(G)$ that measures value of discovering a group

- Should penalize larger groups

Valuing discovered regions

Define a weighting function $w(G)$ that measures value of discovering a group

- Should penalize larger groups
- A canonical choice is inverse-size weighting: $w(G) = 1/|G|$

Valuing discovered regions

Define a weighting function $w(G)$ that measures value of discovering a group

- Should penalize larger groups
- A canonical choice is inverse-size weighting: $w(G) = 1/|G|$
- If G are circles on a sky survey, $w(G) = 1/\text{radius}(G)$ natural

Valuing discovered regions

Define a weighting function $w(G)$ that measures value of discovering a group

- Should penalize larger groups
- A canonical choice is inverse-size weighting: $w(G) = 1/|G|$
- If G are circles on a sky survey, $w(G) = 1/\text{radius}(G)$ natural
- If want to precisely know the *number* of sources in each G:

Valuing discovered regions

Define a weighting function $w(G)$ that measures value of discovering a group

- Should penalize larger groups
- A canonical choice is inverse-size weighting: $w(G) = 1/|G|$
- If G are circles on a sky survey, $w(G) = 1/\text{radius}(G)$ natural
- If want to precisely know the *number* of sources in each G:
  - Pair each G with a $J \subset \mathbb{N}$ representing possible numbers of sources in G

Valuing discovered regions

Define a weighting function $w(G)$ that measures value of discovering a group

- Should penalize larger groups
- A canonical choice is inverse-size weighting: $w(G) = 1/|G|$
- If G are circles on a sky survey, $w(G) = 1/\text{radius}(G)$ natural
- If want to precisely know the *number* of sources in each G:
  - Pair each G with a $J \subset \mathbb{N}$ representing possible numbers of sources in G
  - Set $w(G, J) = 1/|J|$ (we call this the "separation-based" weight function)

Optimizing resolution-adjusted power

Sum weights of true rejections to get Power():

$$\text{Power}(G_1, \ldots, G_R) = \sum_{r=1}^{R} I_{G_r} w(G_r),$$

where I_G is the indicator that G contains a signal (i.e., is a true discovery)

Optimizing resolution-adjusted power

Sum weights of true rejections to get Power():

$$\text{Power}(G_1, \ldots, G_R) = \sum_{r=1}^{R} I_{G_r} w(G_r),$$

where I_G is the indicator that G contains a signal (i.e., is a true discovery)

Then the power of a Bayesian method that discovers $G_1, \ldots, G_R$ is

$$\mathbb{E}[\text{Power}(G_1, \ldots, G_R) \mid \text{Data}] \; = \; \mathbb{E}\left[\sum_{r=1}^{R} I_{G_r} w(G_r) \,\middle|\, \text{Data}\right] \; = \; \sum_{G \subseteq \mathcal{L}} p_G w(G) x_G,$$

- $x_G \in \{0, 1\}$ is indicator that G is one of the method's discoveries
- $p_G = \mathbb{E}[I_G \mid \text{Data}]$ is *posterior inclusion probability* (PIP)

Posterior optmization

Optimal Bayesian method would solve:

$$
\begin{aligned}
\max_{\{x_G\}_{G \subseteq \mathcal{L}}} \quad & \text{Power} = \sum_G p_G w(G) x_G \\
\text{s.t.} \quad & \text{FDR} := \mathbb{E}\left[\frac{\#\{\text{false discoveries}\}}{\#\{\text{discoveries}\}} \,\middle|\, \text{Data} \right] = \frac{\sum_G (1 - p_G) x_G}{\sum_G x_G} \leq q \\
& \sum_{G \ni \ell} x_G \leq 1 \quad \forall \ell \qquad \text{(all discoveries are disjoint)} \\
& x_G \in \{0, 1\} \quad \forall G.
\end{aligned}
$$

Posterior optmization

Optimal Bayesian method would solve:

$$\max_{\{x_G\}_{G\subseteq\mathcal{L}}} \quad \text{Power} = \sum_G p_G w(G) x_G$$

$$\text{s.t.} \quad \text{FDR} := \mathbb{E}\left[\frac{\#\{\text{false discoveries}\}}{\#\{\text{discoveries}\}} \,\middle|\, \text{Data}\right] = \frac{\sum_G (1-p_G)x_G}{\sum_G x_G} \le q$$

$$\sum_{G\ni\ell} x_G \le 1 \quad \forall \ell \qquad \text{(all discoveries are disjoint)}$$

$$x_G \in \{0,1\} \quad \forall G.$$

- Objective and disjointness constraints are linear (good!)

Posterior optmization

Optimal Bayesian method would solve:

$$
\begin{aligned}
\max_{\{x_G\}_{G \subseteq \mathcal{L}}} \quad & \text{Power} = \sum_G p_G w(G) x_G \\
\text{s.t.} \quad & \text{FDR} := \mathbb{E}\left[\frac{\#\{\text{false discoveries}\}}{\#\{\text{discoveries}\}} \,\middle|\, \text{Data} \right] = \frac{\sum_G (1 - p_G) x_G}{\sum_G x_G} \leq q \\
& \sum_{G \ni \ell} x_G \leq 1 \quad \forall \ell \qquad \text{(all discoveries are disjoint)} \\
& x_G \in \{0, 1\} \quad \forall G.
\end{aligned}
$$

- Objective and disjointness constraints are linear (good!)
- FDR constraint not obviously convex (maybe bad)

Posterior optmization

Optimal Bayesian method would solve:

$$
\begin{aligned}
\max_{\{x_G\}_{G\subseteq\mathcal{L}}} \quad & \text{Power} = \sum_G p_G w(G) x_G \\
\text{s.t.} \quad & \text{FDR} := \mathbb{E}\left[\frac{\#\{\text{false discoveries}\}}{\#\{\text{discoveries}\}} \,\middle|\, \text{Data}\right] = \frac{\sum_G (1-p_G)x_G}{\sum_G x_G} \le q \\
& \sum_{G\ni\ell} x_G \le 1 \quad \forall \ell \qquad \text{(all discoveries are disjoint)} \\
& x_G \in \{0,1\} \quad \forall G.
\end{aligned}
$$

- Objective and disjointness constraints are linear (good!)
- FDR constraint not obviously convex (maybe bad)
- Integer constraint definitely not convex (bad!)

Posterior optmization

Optimal Bayesian method would solve:

$$
\begin{aligned}
\max_{\{x_G\}_{G \subseteq \mathcal{L}}} \quad & \text{Power} = \sum_G p_G w(G) x_G \\
\text{s.t.} \quad & \text{FDR} := \mathbb{E}\left[\frac{\#\{\text{false discoveries}\}}{\#\{\text{discoveries}\}} \,\middle|\, \text{Data} \right] = \frac{\sum_G (1 - p_G) x_G}{\sum_G x_G} \le q \\
& \sum_{G \ni \ell} x_G \le 1 \quad \forall \ell \qquad \text{(all discoveries are disjoint)} \\
& x_G \in \{0, 1\} \quad \forall G.
\end{aligned}
$$

- Objective and disjointness constraints are linear (good!)
- FDR constraint not obviously convex (maybe bad)
- Integer constraint definitely not convex (bad!)
- $G \subseteq \mathcal{L}$ is too many to search over (bad!)

Posterior optmization

Optimal Bayesian method would solve:

$$\max_{\{x_G\}_{G\subseteq\mathcal{L}}} \quad \text{Power} = \sum_G p_G w(G) x_G$$

$$\text{s.t.} \quad \text{FDR} := \mathbb{E}\left[\frac{\#\{\text{false discoveries}\}}{\#\{\text{discoveries}\}} \,\middle|\, \text{Data}\right] = \frac{\sum_G (1-p_G)x_G}{\sum_G x_G} \leq q$$

$$\sum_{G\ni\ell} x_G \leq 1 \quad \forall \ell \qquad \text{(all discoveries are disjoint)}$$

$$x_G \in \{0,1\} \quad \forall G.$$

- Objective and disjointness constraints are linear (good!)
- FDR constraint not obviously convex (maybe bad)
- Integer constraint definitely not convex (bad!)
- $G \subseteq \mathcal{L}$ is too many to search over (bad!)

Linearizing the FDR constraint

$$\text{FDR} = \frac{\sum_G (1 - p_G) x_G}{\sum_G x_G} \leq q \qquad \Leftrightarrow \qquad \sum_G (1 - p_G - q) x_G \leq 0$$

Linearizing the FDR constraint

$$\text{FDR} = \frac{\sum_G (1-p_G)x_G}{\sum_G x_G} \le q \qquad \Leftrightarrow \qquad \sum_G (1-p_G-q)x_G \le 0$$

$$\begin{aligned}
\max_{\{x_G\}_{G\subseteq\mathcal{L}}} \quad & \sum_G p_G w(G) x_G && \text{(Power)} \\
\text{s.t.} \quad & \sum_G (1-p_G-q)x_G \le 0 && \text{(FDR)} \\
& \sum_{G\ni\ell} x_G \le 1 \quad \forall \ell && \text{(all discoveries are disjoint)} \\
& x_G \in \{0,1\} \quad \forall G.
\end{aligned}$$

Linearizing the FDR constraint

$$\text{FDR} = \frac{\sum_G (1-p_G)x_G}{\sum_G x_G} \leq q \qquad \Leftrightarrow \qquad \sum_G (1-p_G-q)x_G \leq 0$$

$$\begin{aligned}
\max_{\{x_G\}_{G\subseteq\mathcal{L}}} \quad & \sum_G p_G w(G) x_G && \text{(Power)} \\
\text{s.t.} \quad & \sum_G (1-p_G-q)x_G \leq 0 && \text{(FDR)} \\
& \sum_{G\ni\ell} x_G \leq 1 \quad \forall \ell && \text{(all discoveries are disjoint)} \\
& x_G \in \{0,1\} \quad \forall G.
\end{aligned}$$

Mixed-integer linear program (**MILP**) non-convex; fast solvers for small problems

Convexifying the integer constraint

$$
\begin{aligned}
\max_{\{x_G\}_{G\subseteq\mathcal{L}}} \quad & \sum_G p_G w(G) x_G \qquad \text{(Power)} \\
\text{s.t.} \quad & \sum_G (1 - p_G - q) x_G \le 0 \qquad \text{(FDR)} \\
& \sum_{G \ni \ell} x_G \le 1 \quad \forall \ell \qquad \text{(all discoveries are disjoint)} \\
& x_G \in [0,1] \quad \forall G.
\end{aligned}
$$

Convexifying the integer constraint

$$\begin{aligned} \max_{\{x_G\}_{G\subseteq\mathcal{L}}} \quad & \sum_G p_G w(G) x_G \qquad \text{(Power)} \\ \text{s.t.} \quad & \sum_G (1 - p_G - q) x_G \leq 0 \qquad \text{(FDR)} \\ & \sum_{G\ni\ell} x_G \leq 1 \quad \forall \ell \qquad \text{(all discoveries are disjoint)} \\ & x_G \in [0,1] \quad \forall G. \end{aligned}$$

Now a **linear program (LP)** $\Rightarrow$ optimal solution $\{x_G^\star\}$ easy for large problems

- If $x_G^\star \in \{0,1\}\ \forall G \Rightarrow$ solution to MILP

Convexifying the integer constraint

$$\begin{aligned}
\max_{\{x_G\}_{G \subseteq \mathcal{L}}} \quad & \sum_G p_G w(G) x_G \qquad \text{(Power)} \\
\text{s.t.} \quad & \sum_G (1 - p_G - q) x_G \leq 0 \qquad \text{(FDR)} \\
& \sum_{G \ni \ell} x_G \leq 1 \quad \forall \ell \qquad \text{(all discoveries are disjoint)} \\
& x_G \in [0,1] \quad \forall G.
\end{aligned}$$

Now a **linear program (LP)** $\Rightarrow$ optimal solution $\{x_G^\star\}$ easy for large problems

- If $x_G^\star \in \{0,1\} \; \forall G \Rightarrow$ solution to MILP
- Else, fix $\{x_G^\star \in \{0,1\}\}$ and solve MILP over rest of x_G to get $\{x_G^{\star\star}\}$

Convexifying the integer constraint

$$\begin{aligned}
\max_{\{x_G\}_{G \subseteq \mathcal{L}}} \quad & \sum_G p_G w(G) x_G \qquad \text{(Power)} \\
\text{s.t.} \quad & \sum_G (1 - p_G - q) x_G \le 0 \qquad \text{(FDR)} \\
& \sum_{G \ni \ell} x_G \le 1 \quad \forall \ell \qquad \text{(all discoveries are disjoint)} \\
& x_G \in [0, 1] \quad \forall G.
\end{aligned}$$

Now a **linear program (LP)** $\Rightarrow$ optimal solution $\{x_G^\star\}$ easy for large problems

- If $x_G^\star \in \{0, 1\}\ \forall G \Rightarrow$ solution to MILP
- Else, fix $\{x_G^\star \in \{0, 1\}\}$ and solve MILP over rest of x_G to get $\{x_G^{\star\star}\}$

Empirically, Very few $x_G^\star \notin \{0, 1\}$, and $\{x_G^{\star\star}\}$ very nearly MILP-optimal

BLiP is verifiably nearly optimal

An exact upper-bound for the suboptimality of $\{x_G^{\star\star}\}$ is

$$\sum_G p_G w(G) x_G^\star - \sum_G p_G w(G) x_G^{\star\star},$$

since $\{x_G^\star\}$ optimizes LP which relaxes MILP

BLiP is verifiably nearly optimal

An exact upper-bound for the suboptimality of $\{x_G^{\star\star}\}$ is

$$\sum_G p_G w(G) x_G^\star - \sum_G p_G w(G) x_G^{\star\star},$$

since $\{x_G^\star\}$ optimizes LP which relaxes MILP

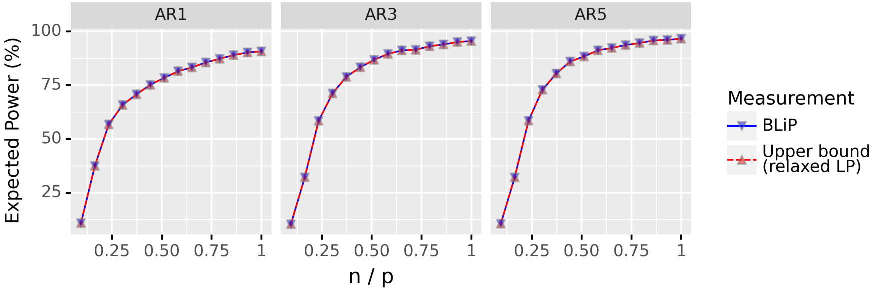


Figure: Expected power (objective function) of $\{x_G^{\star\star}\}$ (BLiP) vs. $\{x_G^\star\}$ (Upper bound). Optimization dimension $\geq 50,000$.

Reducing the search space

Can rapidly solve big LPs...

Reducing the search space

Can rapidly solve big LPs... but not $2^{\mathcal{L}}$ big...

Reducing the search space

Can rapidly solve big LPs... but not $2^{\mathcal{L}}$ big... restrict to set of regions $\mathcal{G} \subset 2^{\mathcal{L}}$:

$$
\begin{aligned}
\max_{\{x_G\}_{G \in \mathcal{G}}} \quad & \sum_G p_G w(G) x_G \qquad \text{(Power)} \\
\text{s.t.} \quad & \sum_G (1 - p_G - q) x_G \leq 0 \qquad \text{(FDR)} \\
& \sum_{G \ni \ell} x_G \leq 1 \quad \forall \ell \qquad \text{(all discoveries are disjoint)} \\
& x_G \in [0, 1] \quad \forall G.
\end{aligned}
$$

Reducing the search space

Can rapidly solve big LPs... but not $2^{\mathcal{L}}$ big... restrict to set of regions $\mathcal{G} \subset 2^{\mathcal{L}}$:

$$\begin{aligned} \max_{\{x_G\}_{G \in \mathcal{G}}} \quad & \sum_G p_G w(G) x_G \qquad \text{(Power)} \\ \text{s.t.} \quad & \sum_G (1 - p_G - q) x_G \leq 0 \qquad \text{(FDR)} \\ & \sum_{G \ni \ell} x_G \leq 1 \quad \forall \ell \qquad \text{(all discoveries are disjoint)} \\ & x_G \in [0,1] \quad \forall G. \end{aligned}$$

$\mathcal{G}$ can be as large as computationally feasible, like $|\mathcal{G}| \sim 10^8$

Reducing the search space

Can rapidly solve big LPs... but not $2^{\mathcal{L}}$ big... restrict to set of regions $\mathcal{G} \subset 2^{\mathcal{L}}$:

$$
\begin{aligned}
\max_{\{x_G\}_{G\in\mathcal{G}}} \quad & \sum_G p_G w(G) x_G \qquad \text{(Power)} \\
\text{s.t.} \quad & \sum_G (1 - p_G - q) x_G \le 0 \qquad \text{(FDR)} \\
& \sum_{G \ni \ell} x_G \le 1 \quad \forall \ell \qquad \text{(all discoveries are disjoint)} \\
& x_G \in [0, 1] \quad \forall G.
\end{aligned}
$$

$\mathcal{G}$ can be as large as computationally feasible, like $|\mathcal{G}| \sim 10^8$

- E.g., (fine-mapping) all contiguous groups of size ≤ 25

Reducing the search space

Can rapidly solve big LPs... but not $2^{\mathcal{L}}$ big... restrict to set of regions $\mathcal{G} \subset 2^{\mathcal{L}}$:

$$\begin{aligned}
\max_{\{x_G\}_{G\in\mathcal{G}}} \quad & \sum_G p_G w(G) x_G && \text{(Power)} \\
\text{s.t.} \quad & \sum_G (1 - p_G - q) x_G \le 0 && \text{(FDR)} \\
& \sum_{G \ni \ell} x_G \le 1 \quad \forall \ell && \text{(all discoveries are disjoint)} \\
& x_G \in [0,1] \quad \forall G.
\end{aligned}$$

$\mathcal{G}$ can be as large as computationally feasible, like $|\mathcal{G}| \sim 10^8$

- E.g., (fine-mapping) all contiguous groups of size ≤ 25
- E.g., (point-source detection) circles of varying radii and centers

Reducing the search space

Can rapidly solve big LPs... but not $2^{\mathcal{L}}$ big... restrict to set of regions $\mathcal{G} \subset 2^{\mathcal{L}}$:

$$
\begin{aligned}
\max_{\{x_G\}_{G \in \mathcal{G}}} \quad & \sum_G p_G w(G) x_G \qquad \text{(Power)} \\
\text{s.t.} \quad & \sum_G (1 - p_G - q) x_G \leq 0 \qquad \text{(FDR)} \\
& \sum_{G \ni \ell} x_G \leq 1 \quad \forall \ell \qquad \text{(all discoveries are disjoint)} \\
& x_G \in [0,1] \quad \forall G.
\end{aligned}
$$

$\mathcal{G}$ can be as large as computationally feasible, like $|\mathcal{G}| \sim 10^8$

- E.g., (fine-mapping) all contiguous groups of size ≤ 25
- E.g., (point-source detection) circles of varying radii and centers
- E.g., (variable selection) hierarchical clustering(s)

Reducing the search space

Can rapidly solve big LPs... but not $2^{\mathcal{L}}$ big... restrict to set of regions $\mathcal{G} \subset 2^{\mathcal{L}}$:

$$
\begin{aligned}
\max_{\{x_G\}_{G \in \mathcal{G}}} \quad & \sum_G p_G w(G) x_G && \text{(Power)} \\
\text{s.t.} \quad & \sum_G (1 - p_G - q) x_G \leq 0 && \text{(FDR)} \\
& \sum_{G \ni \ell} x_G \leq 1 \quad \forall \ell && \text{(all discoveries are disjoint)} \\
& x_G \in [0, 1] \quad \forall G.
\end{aligned}
$$

$\mathcal{G}$ can be as large as computationally feasible, like $|\mathcal{G}| \sim 10^8$

- E.g., (fine-mapping) all contiguous groups of size ≤ 25
- E.g., (point-source detection) circles of varying radii and centers
- E.g., (variable selection) hierarchical clustering(s)

Can also solve problem as if $|\mathcal{G}|$ were much bigger via *adaptive pruning*

Putting it all together: BLiP

Bayesian Linear Programming (BLiP)!

Putting it all together: BLiP

Bayesian Linear Programming (BLiP)!

Just needs posterior inclusion probabilities p_G as input

- From any Bayesian algorithm for computing/approximating the posterior,
- E.g., MCMC (average over posterior samples whether G contains a signal)
- E.g., variational inference

Putting it all together: BLiP

Bayesian Linear Programming (BLiP)!

Just needs posterior inclusion probabilities p_G as input

- From any Bayesian algorithm for computing/approximating the posterior,
- E.g., MCMC (average over posterior samples whether G contains a signal)
- E.g., variational inference

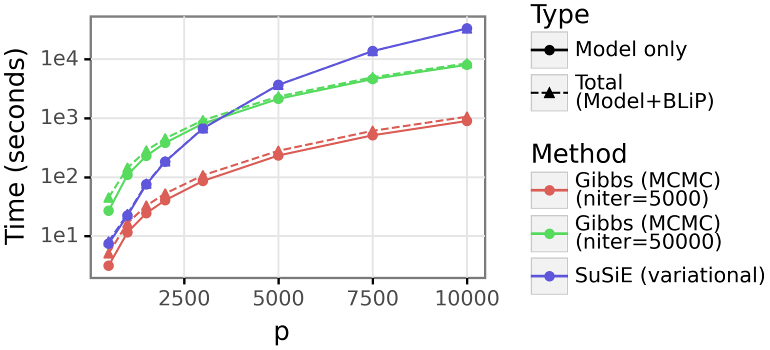


Figure: p denotes dimension of linear model being fit, with $n = p/2$

Comparison with alternatives

- Bayesian (variational): DAP-G (Lee et al., 2018), SuSiE (Wang et al., 2020)

Comparison with alternatives

- Bayesian (variational): DAP-G (Lee et al., 2018), SuSiE (Wang et al., 2020)
- Frequentist: Yekutieli (Yekutieli, 2008), FBH (Katsevich et al., 2021)

Comparison with alternatives

- Bayesian (variational): DAP-G (Lee et al., 2018), SuSiE (Wang et al., 2020)
- Frequentist: Yekutieli (Yekutieli, 2008), FBH (Katsevich et al., 2021)
- Any Bayesian method + BLiP (LSS = linear spike & slab via Gibbs)

Comparison with alternatives

- Bayesian (variational): DAP-G (Lee et al., 2018), SuSiE (Wang et al., 2020)
- Frequentist: Yekutieli (Yekutieli, 2008), FBH (Katsevich et al., 2021)
- Any Bayesian method + BLiP (LSS = linear spike & slab via Gibbs)

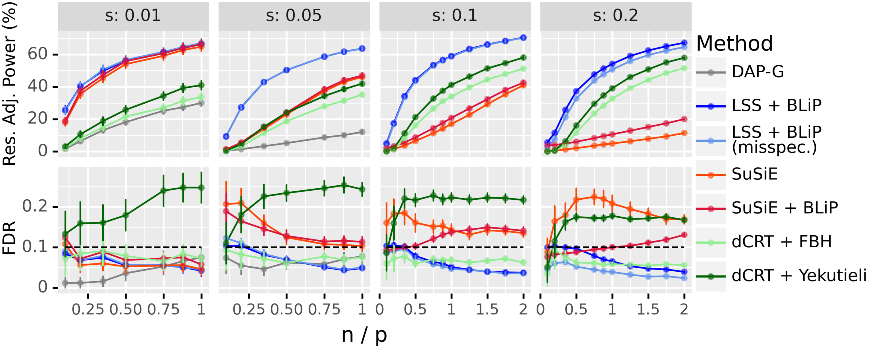


Figure: Linear model w/ autocorrelated X, sparsity s, sample size n, and dimension p

Other error rates

BLiP idea works for other error rates: local FDR, PFER, FWER

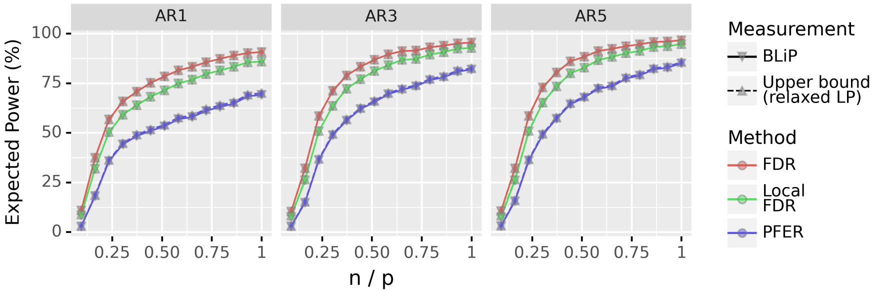


Figure: BLiP's solution indistinguishable from upper-bound for optimal solution

Change point detection

BLiP applies out of the box to change point detection

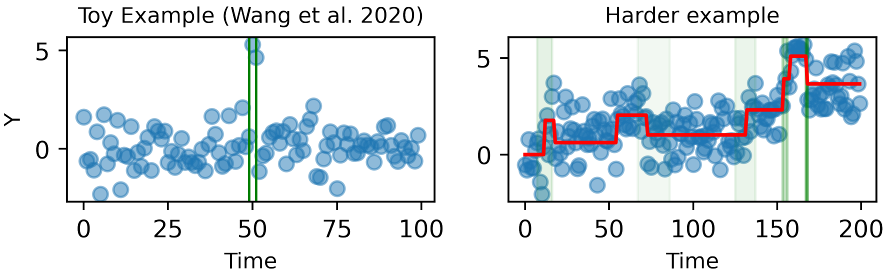


Figure: Green bands denote LSS+BLiP's outputted regions; left is example SuSiE fails on due to variational approximation

Point-source detection

100 x 100 pixel sub-image of Messier 2 star cluster from Sloan Digital Sky Survey

- Ground truth available from much more powerful Hubble Space Telescope

Point-source detection

100 x 100 pixel sub-image of Messier 2 star cluster from Sloan Digital Sky Survey

- Ground truth available from much more powerful Hubble Space Telescope
- StarNet (Liu et al., 2021): variational approx.'s MAPs + 0.5-pixel slack

Point-source detection

100 x 100 pixel sub-image of Messier 2 star cluster from Sloan Digital Sky Survey

- Ground truth available from much more powerful Hubble Space Telescope
- StarNet (Liu et al., 2021): variational approx.'s MAPs + 0.5-pixel slack
- *continuous* space of locations $\mathcal{L}$: BLiP takes < 10 min for 15 FDRs

Point-source detection

100 x 100 pixel sub-image of Messier 2 star cluster from Sloan Digital Sky Survey

- Ground truth available from much more powerful Hubble Space Telescope
- StarNet (Liu et al., 2021): variational approx.'s MAPs + 0.5-pixel slack
- *continuous* space of locations $\mathcal{L}$: BLiP takes < 10 min for 15 FDRs

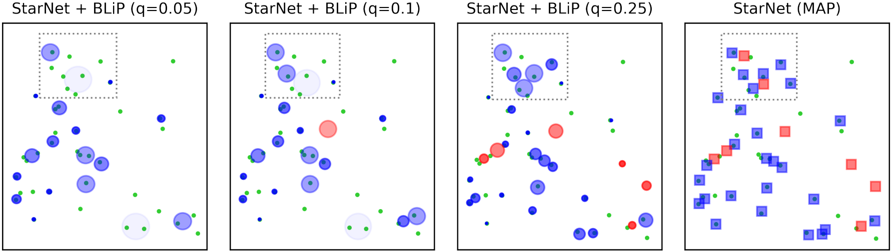
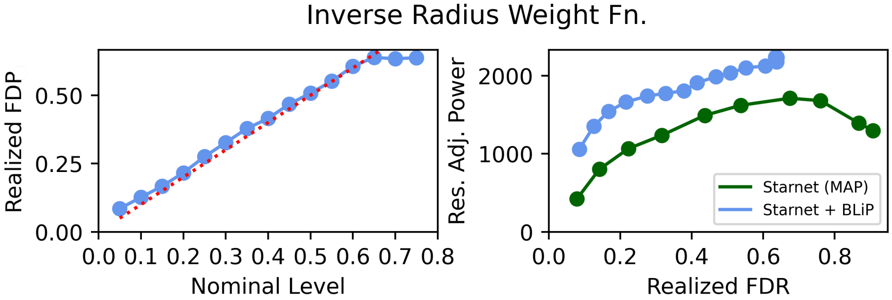
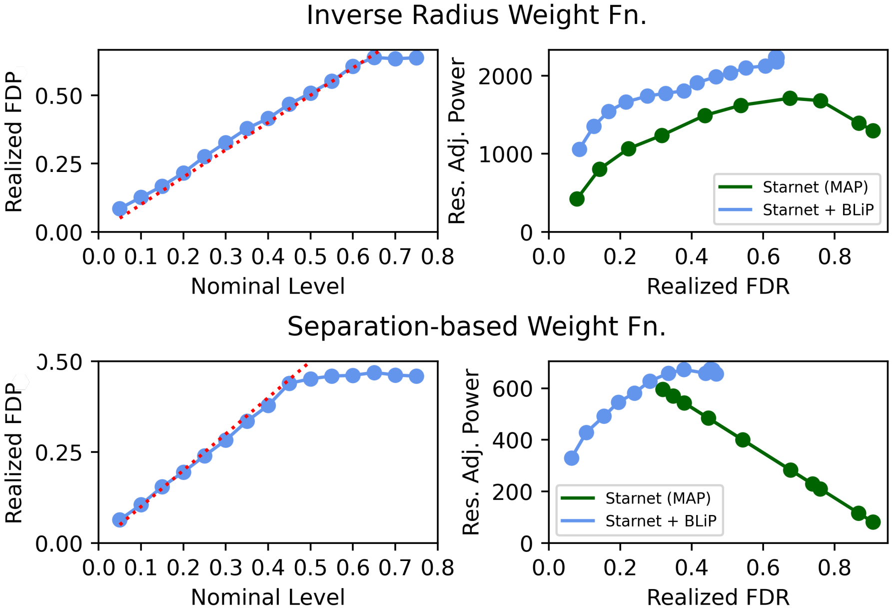


Figure: 20 x 20 pixel sub-image; green dots = ground truth, red regions = false discoveries, blue regions = true discoveries

Point-source detection (contd)



Point-source detection (contd)

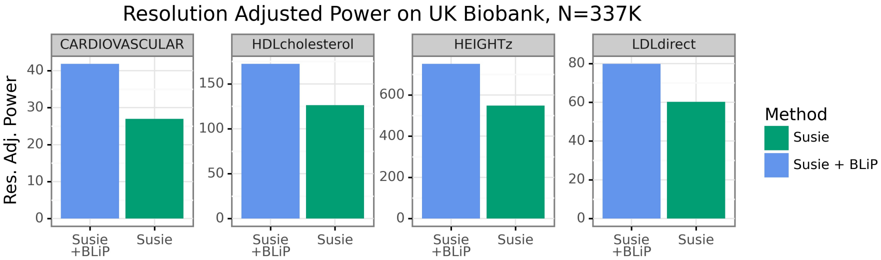


Fine-mapping

UK Biobank data: $n \approx 337,000$, $p \approx 19,000,000$; BLiP takes < 1 min per trait

Fine-mapping

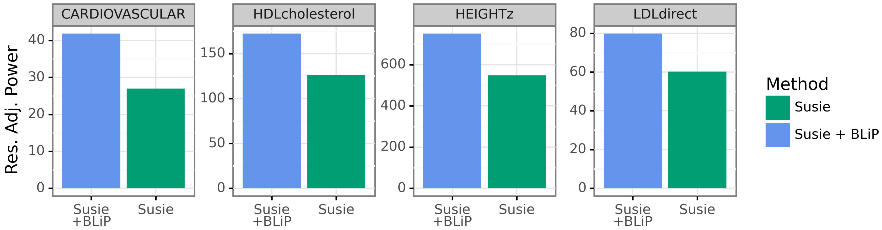
UK Biobank data: $n \approx 337,000$, $p \approx 19,000,000$; BLiP takes <1 min per trait



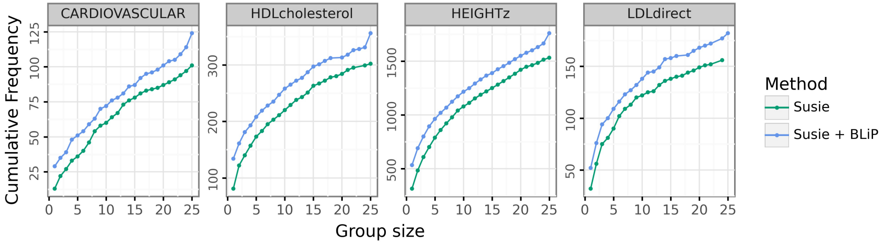
Fine-mapping

UK Biobank data: $n \approx 337,000$, $p \approx 19,000,000$; BLiP takes <1 min per trait

Resolution Adjusted Power on UK Biobank, N=337K



Cumulative Frequency of Discovered Group Sizes



Conclusion

BLiP is a powerful, principled, efficient, and flexible method for resolution-adaptive signal discovery

- Provable error control and verifiable near-optimality

Conclusion

BLiP is a powerful, principled, efficient, and flexible method for resolution-adaptive signal discovery

- Provable error control and verifiable near-optimality
- Substantial power gains in minutes on fine-mapping and point-source detection

Conclusion

BLiP is a powerful, principled, efficient, and flexible method for resolution-adaptive signal discovery

- Provable error control and verifiable near-optimality
- Substantial power gains in minutes on fine-mapping and point-source detection
- Software packages `pyblip` (Python) and `blipr` (R)

Conclusion

BLiP is a powerful, principled, efficient, and flexible method for resolution-adaptive signal discovery

- Provable error control and verifiable near-optimality
- Substantial power gains in minutes on fine-mapping and point-source detection
- Software packages `pyblip` (Python) and `blipr` (R)
- Potential for other signal discovery problems with spatial structure

Conclusion

BLiP is a powerful, principled, efficient, and flexible method for resolution-adaptive signal discovery

- Provable error control and verifiable near-optimality
- Substantial power gains in minutes on fine-mapping and point-source detection
- Software packages `pyblip` (Python) and `blipr` (R)
- Potential for other signal discovery problems with spatial structure

paper available at: `https://arxiv.org/abs/2203.17208`
all code posted at: `https://github.com/amspector100`

Conclusion

BLiP is a powerful, principled, efficient, and flexible method for resolution-adaptive signal discovery

- Provable error control and verifiable near-optimality
- Substantial power gains in minutes on fine-mapping and point-source detection
- Software packages `pyblip` (Python) and `blipr` (R)
- Potential for other signal discovery problems with spatial structure

paper available at: `https://arxiv.org/abs/2203.17208`
all code posted at: `https://github.com/amspector100`

Thank you!
`http://lucasjanson.fas.harvard.edu`
ljanson@fas.harvard.edu

References

Katsevich, E., Sabatti, C., and Bogomolov, M. (2021). Filtering the rejection set while preserving false discovery rate control. *Journal of the American Statistical Association*, 0(0):1–12.

Lee, Y., Luca, F., Pique-Regi, R., and Wen, X. (2018). Bayesian multi-snp genetic association analysis: Control of fdr and use of summary statistics. *bioRxiv*.

Liu, R., McAuliffe, J. D., and Regier, J. (2021). Variational inference for deblending crowded starfields.

Wang, G., Sarkar, A., Carbonetto, P., and Stephens, M. (2020). A simple new approach to variable selection in regression, with application to genetic fine mapping. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(5):1273–1300.

Yekutieli, D. (2008). Hierarchical false discovery rate-controlling methodology. *Journal of the American Statistical Association*, 103(481):309–316.