Knockoffs.jl: Variable Selection with Knockoffs in julia

Benjamin B. Chu, Chiara Sabatti

Department of Biomedical Data Sciences, Stanford University

Contact: bbchu@stanford.edu, sabatti@stanford.edu
Software page: https://github.com/biona001/Knockoffs.jl
Documentation: https://biona001.github.io/Knockoffs.jl/dev

docs latest CI passing codecov 23%

Introduction

- Do you need to perform variable selection?
- Does your features/covariates exhibit ***arbitrary*** correlation structure?
- And you still want to control the **False Discovery Rate (FDR)**?

Knockoffs.jl is a Julia package for generating statistical knockoffs that will control the FDR when performing variable selection, even when variables are arbitrarily correlated.

As the name suggests, the knockoff filter operates by manufacturing knockoff variables that are cheap — their construction does not require collecting any new data — and are designed to mimic the correlation structure found within the original variables. The knockoffs serve as negative controls and they allow one to identify the truly important predictors, while controlling the false discovery rate (FDR) — the expected fraction of false discoveries among all discoveries.

Two Goals of Knockoffs

1. Instead of controlling FWER*, the knockoff procedure controls the FDR

$$FDR = E\left(\frac{\#\text{false positives}}{\#\text{ total discoveries}}\right)$$

This significantly improves power.

2. Knockoff based inference tests *conditional hypotheses.* If G is a variable or a group of variables, we test

$$\mathcal{H}_0 = Y \perp\!\!\!\perp X_G \mid X_{-G}$$

where X_{-G} means all variables except G. Conditioning on X_{-G} removes variables only marginally associated with the response due to high correlation. In genetics, this helps us prioritize causal associations.

*FWER: Family-wise error rate – this is what Bonferroni correction controls

The Knockoff-Filter Procedure

1. For each sample $X \in \mathbb{R}^p$, generate *knockoffs* $\tilde{X} \in \mathbb{R}^p$ s.t.
   - $Y \perp\!\!\!\perp \tilde{X} \mid X$
   - $(X, \tilde{X}) \overset{d}{=} (X, \tilde{X})_{\text{swap}(S)}\ \forall S$. E.g. If $S = \{2\}$, then $(X_1, X_2, X_3, \tilde{X}_1, \tilde{X}_2, \tilde{X}_3) \overset{d}{=} (X_1, \tilde{X}_2, X_3, \tilde{X}_1, X_2, \tilde{X}_3)$
2. Compute feature importance statistic on matrix $[\mathbf{X}\ \tilde{\mathbf{X}}]$
3. For all G, compute knockoff scores

$$\boxed{W_G = ImportanceScore(G) - ImportanceScore(\tilde{G})}$$

4. Choose all G such that $W_G \geq \tau$, where τ depends on FDR q

$$\tau = \min\left\{t > 0 : \frac{1 + \#\{G : W_G \leq -t\}}{\#\{j : W_G \geq t\} \vee 1} \leq q\right\}$$

Package Feature & Comparisons

Currently, Knockoffs.jl supports Gaussian covariates (fixed-X or model-X) and covariates that can be modeled by a hidden Markov model (HMM) commonly used in genetic studies.

	Language	Fixed-X knockoffs	Model-X knockoffs	HMM knockoffs	Sequential knockoffs	Supported data formats
Knockoffs.jl	Julia	SDP and equi	SDP and equi	Single-SNP fastPHASE and SHAPEIT	Single only	Numeric matrix, binary PLINK
Knockoff-filter	Matlab and R	SDP, equi, ASDP	SDP, equi, ASDP			Numeric Matrix
SNPKNOCK and SNPKNOCK2	R/C++			SHAPEIT and fastPHASE (single and group SNPs)		Binary PLINK and BGEN
KnockoffScreen	R				Single and multiple	VCF, BGEN
knockpy	Python	MVR, MAXENT, SDP, equi, CI	MVR, MAXENT, SDP, equi, CI			Numeric Matrix

Development Roadmap for Knockoffs.jl: Support for BGEN/VCF/PGEN inputs, MVR fixed/model-X knockoffs, more efficient sequential knockoffs, grouped HMM knockoffs, linear-time model-X knockoffs

Examples of using Knockoffs.jl

Generate fixed-X knockoffs with equicorrelated construction

```
julia> Xk = fixed_knockoffs(X, :equi)
```

Generate exact model-X SDP knockoffs (i.e. known mean and variance)

```
julia> Xk = modelX_gaussian_knockoffs(X, :sdp, μ, Σ)
```

Generate 2nd order model-X SDP knockoffs (mean and covariance estimated)

```
julia> Xk = modelX_gaussian_knockoffs(X, :sdp)
```

Generate hidden Markov model (HMM) knockoffs by running fastPHASE on unphased genotype data stored in binary PLINK format

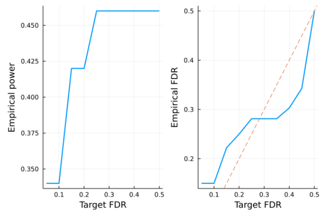
```
julia> Xk = hmm_knockoff("plinkfile",
            plink_outfile="fastphase.knockoffs")
```

Generate multiple knockoffs as in KnockoffScreen by sliding window

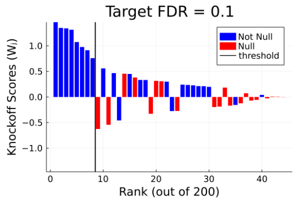
```
julia> Xk = full_knockoffscreen("plinkfile",
            windowsize=50)
```

Example Results Visualized

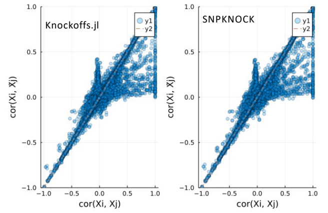
Compare Power vs FDR for simulation studies



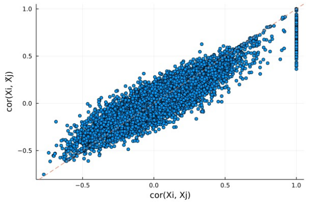
Display knockoff statistics for simulation studies



Comparing HMM knockoffs on 1000 samples from UK Biobank Chr10 data



Comparing $cor(X_j, \tilde{X}_j)$ for all j in HMM knockoffs


cor(Xi, X̃j)
cor(Xi, Xj)