Statistical Inference Relief (STIR) feature selection

Trang T. Le[1], Ryan J. Urbanowicz[1], Jason H. Moore[1], Brett A. McKinney[2]

[1]Institute of Biomedical Informatics, Department of Biostatistics, Epidemiology and Informatics, University of Pennsylvania, Philadelphia, PA
[2]Tandy School of Computer Science, Department of Mathematics, University of Tulsa, Tulsa, OK

DEPARTMENT of BIOSTATISTICS EPIDEMIOLOGY & INFORMATICS

Introduction

Identifying relevant features in high-dimensional data can be challenging when their effect on a phenotype may be obscured by a complex **interaction architecture**. Using **nearest-neighbors,** Relief-based algorithms account for statistical interactions when selecting features. However, without a parameterized model, it is difficult to determine the **statistical significance** of Relief-based attribute estimates. Thus, a **statistical inferential** formalism is needed to avoid imposing arbitrary thresholds to select the most important features.

Methods

STatistical Inference Relief (STIR) We re-conceptualize the Relief-based algorithm to create a new family of STIR estimators that

- retains the ability to identify interactions;
- while incorporating sample variance of the nearest neighbor distances into the attribute importance estimation. This variance permits the calculation of statistical significance of features and adjustment for multiple testing of Relief-based scores (Eq. (*)).

The reformulated version allows for algorithm optimization by precomputing miss and hit matrices and using a vectorized diff function. Pseudo-code for STIR works similarly (**Fig. 1**).

Performance evaluation On **simulated** and **real-world RNA-Seq** data:

- STIR with multiSURF (an adaptive neighborhood method)
- permutation test
- *t*-test

Reformulation **Modification**

$$\overline{M}_a = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{k_{M_i}}\sum_{j_i=1}^{k_{M_i}}\text{diff}(a,(R_i,M_{j_i})) \qquad W_R[a,M,H] = \overline{M}_a - \overline{H}_a$$

$$\overline{H}_a = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{k_{H_i}}\sum_{j_i=1}^{k_{H_i}}\text{diff}(a,(R_i,H_{j_i})) \qquad W_{\text{STIR}}[a,M,H] = \frac{\overline{M}_a - \overline{H}_a}{S_p[M,H]\sqrt{1/|M|+1/|H|}} \quad (*)$$

Figure 1. Comparison of the pseudo-code of the original ReliefF algorithm as implemented in ReBATE (Algorithm 1, left) versus the reformulated version of ReliefF (Algorithm 2, right).

Algorithm 1: Original ReliefF algorithm

```
m ← number of training instances
p ← number of attributes
k ← number of nearest hits or misses
pre-process dataset X
pre-compute distance matrix D (Eq. 1)
initialize all feature weights W[a] := 0

for i := 1 to m do
    for j := 1 to m do
        identify k nearest hits and k nearest misses
    end

    for all hits and misses do
        # attribute weight update
        for a := 1 to p do
            W[a] := W[a] − diff(a,R_i,H)/(m·k) + diff(a,R_i,M)/(m·k)
        end
    end
end

return vector W of feature scores
```

Algorithm 2: Reformulated ReliefF algorithm

```
m ← number of training instances
p ← number of attributes
k ← number of nearest hits or misses
pre-process dataset X
pre-compute distance matrix D (Eq. 1)
initialize all feature weights W[a] := 0
pre-compute miss matrix M and hit matrix H (Sec. 2.1)

for a := 1 to p do
    # compute diff vectors then sum:
    M_a = diff(a, (X[M[,1], a], X[M[,2], a]))
    H_a = diff(a, (X[H[,1], a], X[H[,2], a]))
    W[a] := 1/(m·k) (∑ M_a − ∑ H_a)
end

return vector W of feature scores
```

Funding This work was supported by NIH LM010098 and LM012601(to JHM) and GM121312 and GM103456 (to BAM).

Availability Code and data of significant genes are available at available at http://insilico.utulsa.edu/software/STIR.

Results

Figure 2. STIR versus permutation-test multiSURF and univariate *t*-test to detect functional attributes.

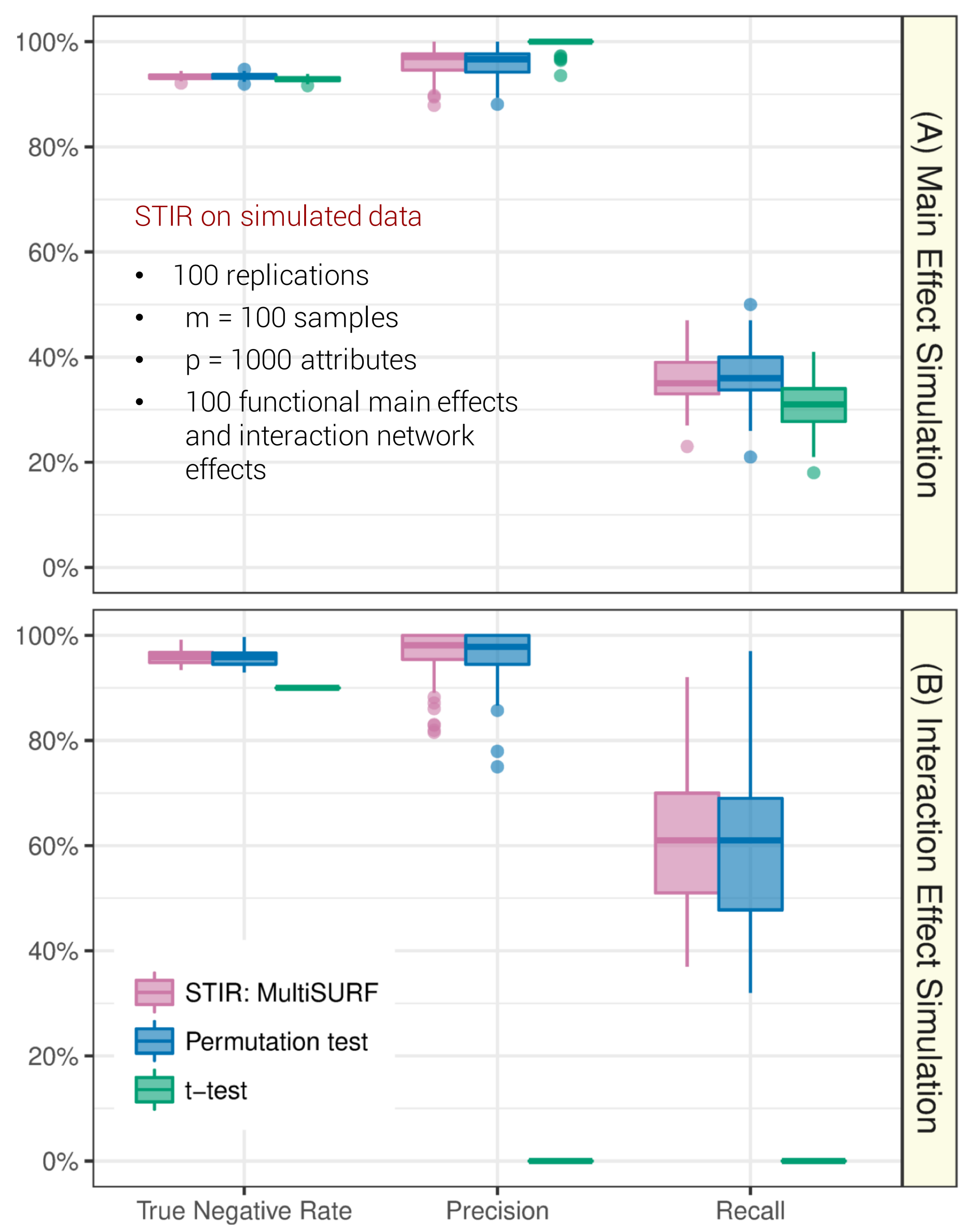


Figure 3. The effect of *k* on the performance of STIR to detect functional attributes.

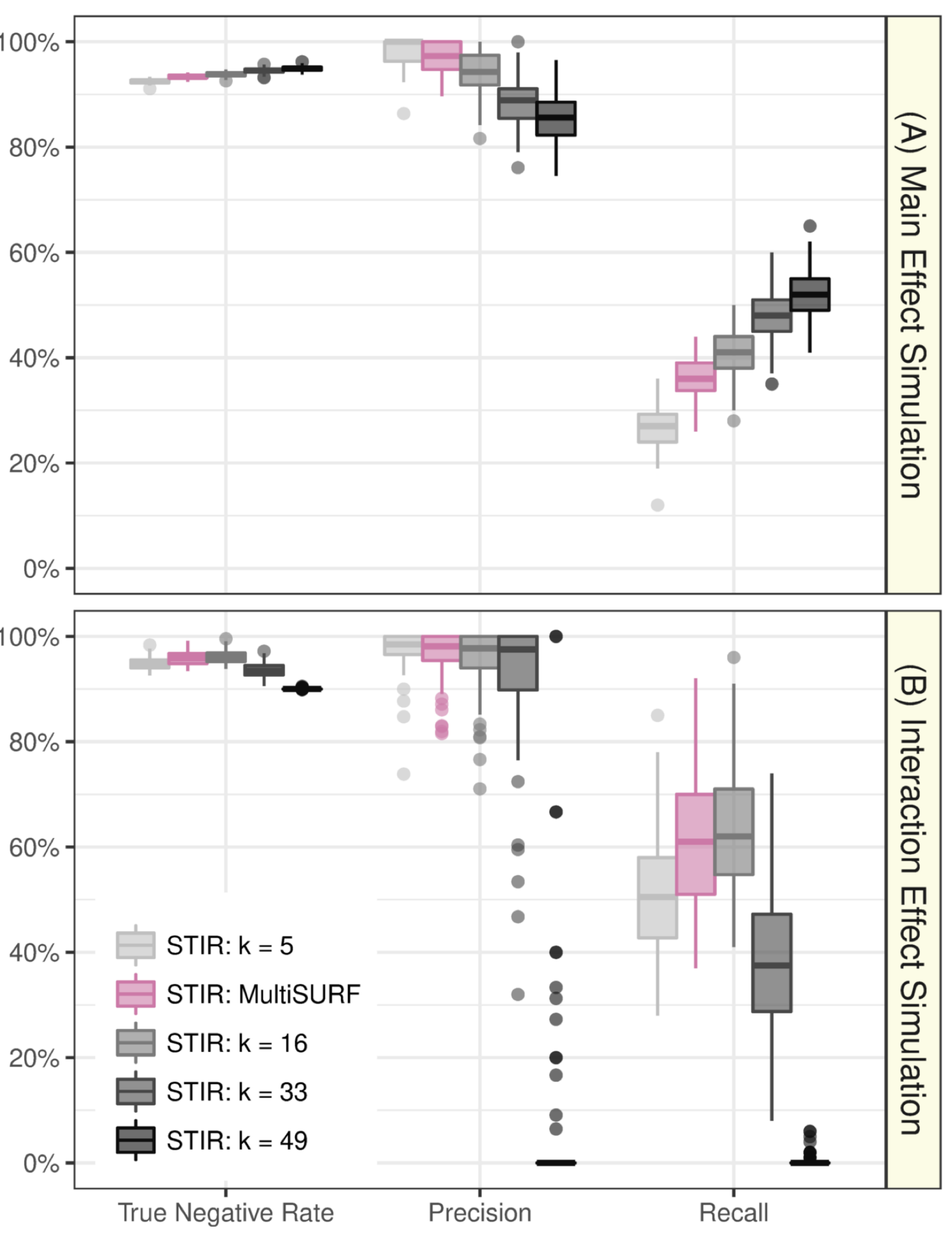
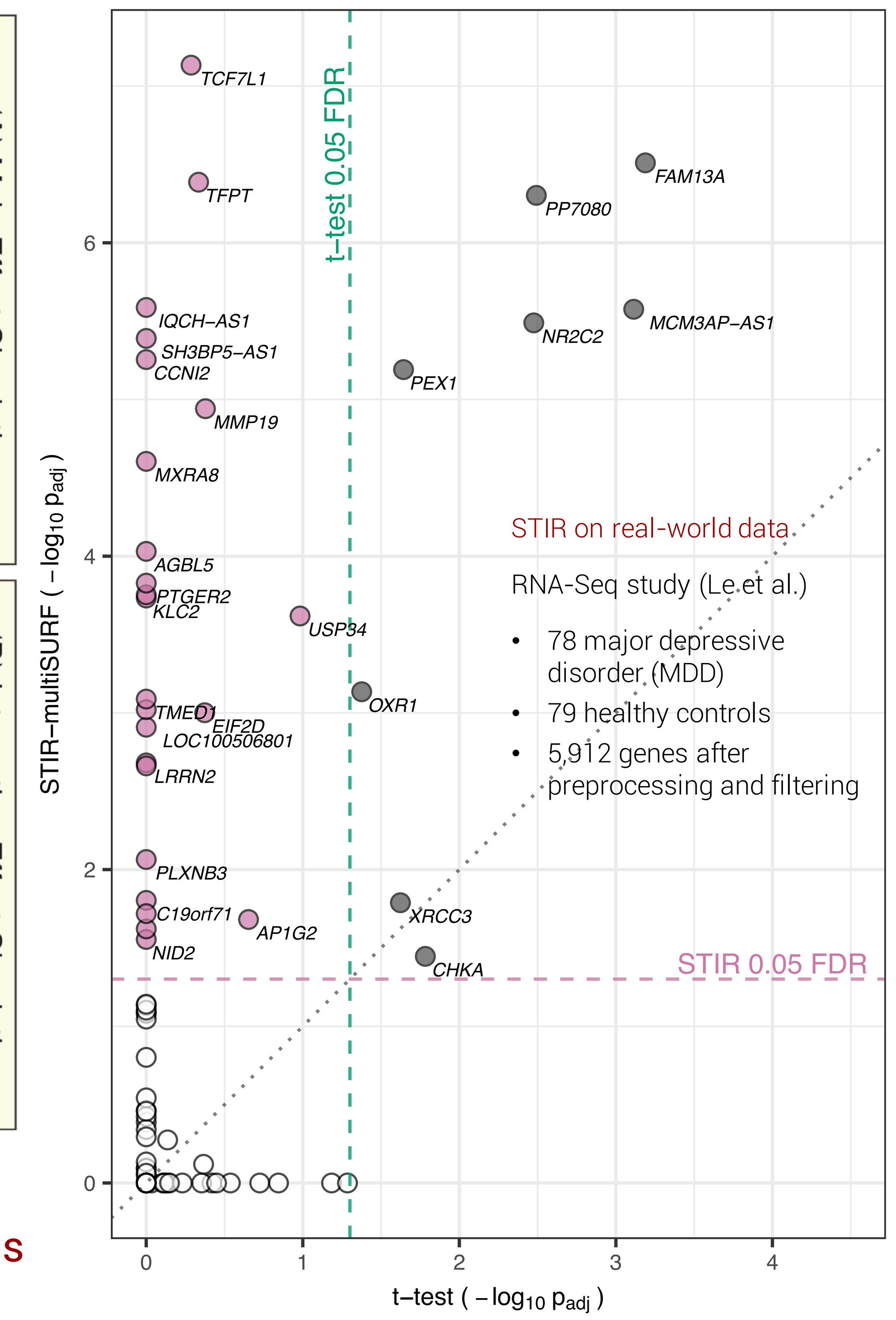


Figure 4. MDD gene scatter plot of -log10 adjusted significance for STIR-multiSURF and standard *t*-test for RNA-Seq differential expression.



STIR on simulated data

Main and interaction effect (**Fig. 2**):

- STIR (mauve) ~ permutation-Relief (blue).

Interaction effect (**Fig. 2B**):

- no *t*-tests are true positive: no main effects and the *t*-test (green) has zero Precision and Recall
- STIR still has high Precision and Recall (Relief sensitive to interactions)

Effect of *k* in detecting functional attributes

Main effect (**Fig. 3A**), as *k* increases

- STIR gains more power to detect the functional attributes (Recall ↑) and with an expected increase in false positive attributes (Precision ↓).
- ReliefF becomes more myopic

Interaction effect (**Fig. 3B**):

- No longer monotonic
- Recall reaches max at ~ *k=m/6*
- Near 0 at k_{max}
- multiSURF neighborhood constitutes a compromise between main and interaction effect performance.

STIR on real-world data (**Fig. 4**)

- 32 significant STIR genes include all 8 significant genes from standard t-test
- STIR genes outside of the intersection with *t*-test (mauve) may be good candidates for interaction effects.

Conclusion

STIR is the first method to use a theoretical distribution to calculate the statistical significance of Relief attribute scores without the computational expense of permutation.

- STIR p-values ~ permutation p-values → one can use STIR pseudo t-test instead of costly permutation testing.
- STIR formalism **generalizes to all Relief-based** neighbor finding algorithms, including MultiSURF.
- ***k=m/6* offers a better default** than the pervasive use of *k=10* (arbitrary choice in the early literature).
- Extensions of STIR: multi-class data; quantitative trait data (regression); correction for covariates; missing data; application to GWAS data.