Genome-wide detection of intervals of genetic heterogeneity associated with complex traits.

by Llinares-López *et al*. *Bioinformatics*, 2015
STA426 Journal Club

Richard Affolter, Martin Emons, Philip Hartout

November 23, 2020

Introduction

Motivation: Genetic heterogenity

Genetic heterogenity

Several sequence variants give rise to the same pheontype

Task

Find regions in the genome that exhibit genetic heterogenity

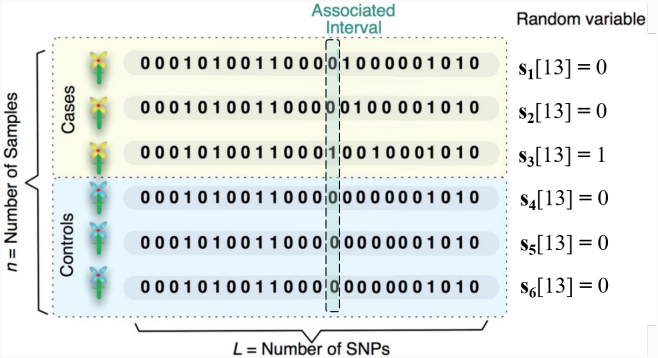
Setting

Given n individuals classified into two phenotypic groups, n_1 cases and n_2 controls

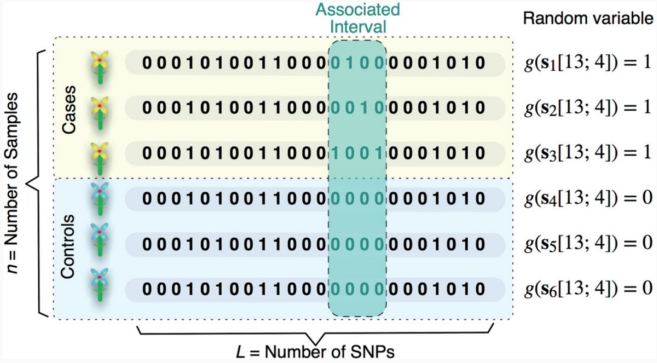
Each individual is represented by an ordered sequence of L binary genotypes

- binary SNPs in homozygous setting
- dominant/recessive encoding in heterozygous setting

Classical GWAS



Interval search



Problem of multiple hypthesis testing

how many possible intervals are there?

$$\begin{bmatrix} \tau_{1,1} & \tau_{1,2} & \tau_{1,3} & \cdots & \tau_{1,L} \\ & \tau_{2,2} & \tau_{2,3} & \cdots & \tau_{2,L} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & \tau_{L-1,L} \\ & & & & \tau_{L,L} \end{bmatrix}$$

$$D = \frac{L(L+1)}{2}$$

- D grows quadratic in L
- usual values for L are in the order of 10^5 or 10^6
  $\Rightarrow$ D in the order of 10^{10} - 10^{12}

Control of Family Wise Error Rate (FWER)

`Bonferroni correction`

$\delta_{\text{bon}} = \alpha / D$

- very simple to compute
- overly conservative,
  especially if D is a huge number

`Westfall-Young permutation testing`

Resample the dataset J times by random permutation of the class label
$\rightarrow$ destroys association between class labels and intervals
compute the minimal p-value across all D intervals
$\delta_{wy} = \alpha$-quantile of the set $\left\{ p_{\text{min}}^{(j)} \right\}_{j=1}^{J}$

- strong control of FWER
- computational effort unfeasable for reasonable values of J, often 10^3 or 10^4

Methods

Statistical testing

The contingency table contains *discrete* values:

	$g(\mathbf{s}[\tau; l]) = 1$	$g(\mathbf{s}[\tau; l]) = 0$	Row tot
$y = 1$	a	$n_1 - a$	n_1
$y = 0$	$x - a$	$n_0 - (x - a)$	n_2
Col tot	x	$n - x$	n

$$g(\mathbf{s}[\tau; l]) = \begin{cases} 1 \text{ if at least one interval contains 1} \\ 0 \text{ otherwise} \end{cases} \quad (1)$$

Fisher's exact test:

H_0: the proportion of $g(\mathbf{s}[\tau; l])$ does not influence the proportion of y.

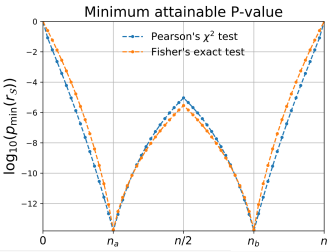
Minimal attainable p-value

The contingency table contains *discrete* values:

	$g(\mathbf{s}[\tau; l]) = 1$	$g(\mathbf{s}[\tau; l]) = 0$	Row tot
$y = 1$	15	0	15
$y = 0$	0	45	45
Col tot	15	45	60

Given one column total (x) along with n_1 and n_2

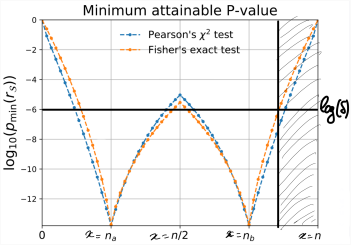
$\rightarrow$ one extreme case (table above) where $a = x$ with $p_{min} = 1.88 \cdot 10^{-14}$.



Testability and pruning

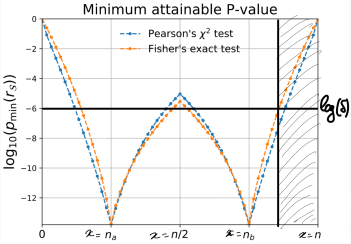
Testability and pruning

Testability:
Concept introduced by Tarones [1].
If $p_{min} > \delta$, we can discard all values of x which yield p_{min} or higher.
Graphically:



Testability and pruning

Testability:
Concept introduced by Tarones [1].
If $p_{min} > \delta$, we can discard all values of x which yield p_{min} or higher.
Graphically:



Pruning:
If (τ, I) is non-testable, and $(\tau, I) \subset (\tau', I')$, then there is no need to evaluate (τ', I').

This leads to speed-ups in the computation, as a large quantity of intervals don't need to be evaluated.

Fast automatic interval search (FAIS)

Objective: obtain corrected significance threshold δ^*.

Fast automatic interval search (FAIS)

Objective: obtain corrected significance threshold δ^*.

1. Initialize δ such that all intervals are *testable*.

Fast automatic interval search (FAIS)

Objective: obtain corrected significance threshold δ^*.

1. Initialize δ such that all intervals are *testable*.
2. Sequentially enumerate intervals in increasing order of length
   - If testable: processed, and δ is adjusted

Fast automatic interval search (FAIS)

Objective: obtain corrected significance threshold δ^*.

1. Initialize δ such that all intervals are *testable*.
2. Sequentially enumerate intervals in increasing order of length
   - If testable: processed, and δ is adjusted
   - If not: *prune* intervals containing current interval

FAIS - two variants

FAIS - two variants

FAIS – δ^*_{Tarone}

$\rightarrow$ Standard procedure highlighted above

FAIS - two variants

FAIS – δ^*_{Tarone}

→ Standard procedure highlighted above

FAIS-WY – $\delta^*_{\text{Westfall-Young}}$

→ add Westfall-Young component. Consists of adding the following steps:

- Permutation-based procedure to produce an initial set of p_{min} across different permutations.
- FWER can be estimated from average of p_{min}, and the threshold is adjusted as needed during the interval enumeration.

Differences?

FAIS-WY is slightly slower, but tends to have higher power.

Experiments

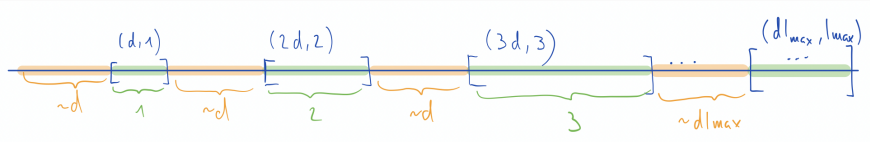
Set-up

Objective:

1. Test the two presented algorithms `FAIS` and `FAIS-WY` on simulated data and on real data from *Arabidopsis thaliana*
2. Benchmarking against known methods
   - 2.1 `BRUTE` - Bonferroni Correction
   - 2.2 `BRUTE-WY` - Westfall-Young Version of `BRUTE`
   - 2.3 `UFE` - Univariate Fisher's Exact Test - (standard GWAS)

Creation of Simulated Data I

- n binary sequences of length L
- n_1 sequences for cases, n_2 sequences for controls
- sampled every entry of a sequence $s_i[j]$ - Background noise
  - $s_i[j] \sim B(1, \rho_0), \qquad i = 1, 2, ..n, \; j = 1, ..., L$
  - $s_i[j] = 1$ with probability ρ_0
- inserted l_{max} significant intervals
  - $(d, 1), (2d, 2), ..., (dl_{max}, l_{max})$ with $d > l_{max}$
  - d is the space between the significant intervals

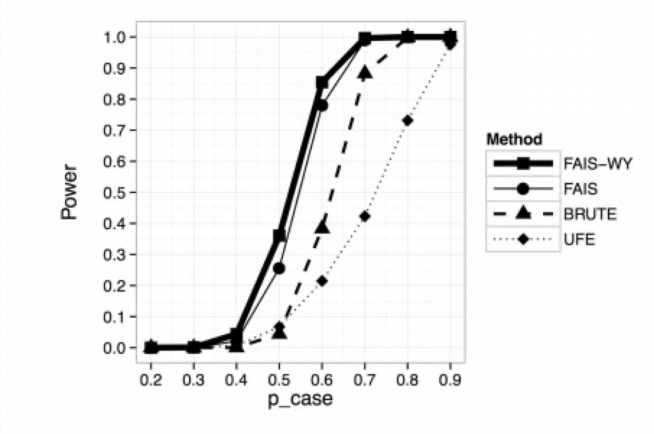


Creation of Simulated Data II

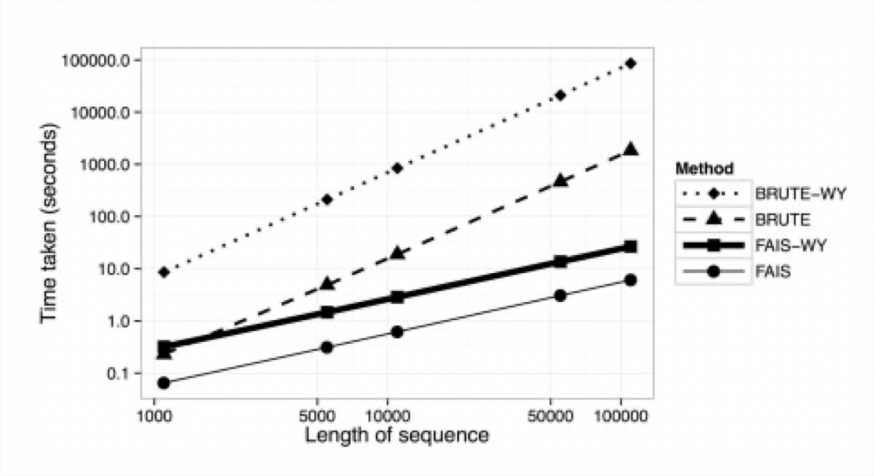
- Cases: replaced elements in significant intervals $s_i[dl; l]$ with new sequences
  - Probability of at least one 1 occurring in $s_i[dl; l] = \rho_{case}$
  - Sampling from Bernoulli distribution again
- Controls: Same procedure for controls with ρ_{con}

Experiments on Simulated Data - Power and FWER

- parameter setting:
  $n_1 = 100, n_2 = 100, d = 1000, l_{max} = 10, \alpha = 0.05, \rho_0 = 0.1, \rho_{con} = 0.2$



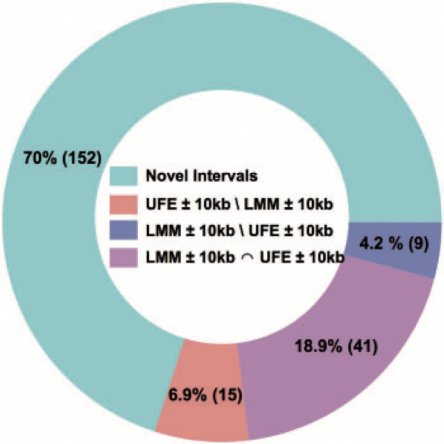
Experiments on Simulated Data - Running Time Comparisons



Experiments on Real Data - Heterogeneity Detection in *A. thaliana* I

- *Arabidopsis thaliana* GWAS dataset
- 21 defense and developmental related phenotypes
- assessed population structure via logistic regression $\rightarrow$ genomic control inflation factor λ
- used UFE and LMM to assess confounding due to population structure
  - LMM $\rightarrow$ corrects population structure
  - UFE does not
  - how many sites are found by `FAIS` and `FAIS-WY` that are found by UFE but not LMM $\rightarrow$ due to population structure

Experiments on Real Data - Heterogeneity Detection in *A. thaliana* II



Discussion & Conclusion

Strengths

- Very good performance
- High accuracy
- Can detect intervals with higher sensitivity

Weaknesses

- Does not account for population structure
- Only consider contiguous intervals of SNPs $\rightarrow$ they could be dispersed
- Encoding sensitive $\rightarrow$ changing binary encoding of a SNP will affect the result

Questions?

https://github.com/mjemons/Latex-Documents

References i


Robert E Tarone.
A modified bonferroni method for discrete data.
Biometrics, pages 515–522, 1990.


Felipe Llinares-López, Dominik G Grimm, Dean A Bodenham, Udo Gieraths, Mahito Sugiyama, Beth Rowan, and Karsten Borgwardt.
Genome-wide detection of intervals of genetic heterogeneity associated with complex traits.
Bioinformatics, 31(12):i240–i249, 2015.

Experiments on Real Data - Heterogeneity Detection in *A. thaliana* II

Phenotype name	λ-GC	UFE hits	LMM hits	FAIS hits	FAIS-WY hits
Chlorosis 16	1.01	0	0	0	0
Chlorosis 10	1.02	0	1	0	0
Leaf roll 22	1.17	0	0	0	0
Emco5	1.18	0	4	0	1
Emoy	1.18	1	2	0	0
Hiks1	1.2	0	1	0	0
Noco2	1.25	1	0	0	1
Anthocyanin 16	1.33	0	0	0	1
Anthocyanin 10	1.44	0	1	0	1
Anthocyanin 22	1.47	0	0	0	1
Emwa1	1.5	0	0	0	1
avrRpt2	1.52	5	8	2	5
avrB	1.63	16	14	13	15
Leaf roll 16	1.65	0	1	0	1
avrRpm1	1.68	15	14	13	14
Chlorosis 22	1.71	2	0	0	3
Leaf roll 10	1.79	1	1	1	3
avrPphB	1.92	14	9	7	16
LES	2.22	8	9	1	11
LY	2.54	36	2	9	40
YEL	3.41	21	76	11	103