ETH zürich

Machine learning to uncover biological interactions

Damian Roqueiro, Laetitia Papaxanthos and Anja Gumpinger
Karsten Borgwardt
Machine Learning & Computational Biology Lab
ETH Zürich

Antwerp, 27.04.2016

Part I
Testability and correction for multiple hypothesis testing

By Damian Roqueiro

Significant pattern mining

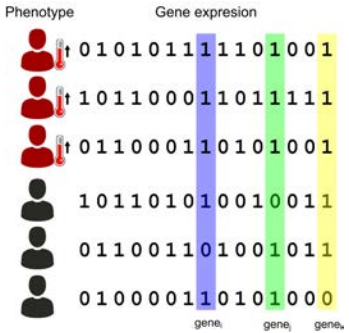
Definition F. Llinares-López et al. KDD 2015

- The goal of *significant pattern mining* is to identify sets of items that occur statistically significantly more often in one class than in the other.

Significant pattern mining

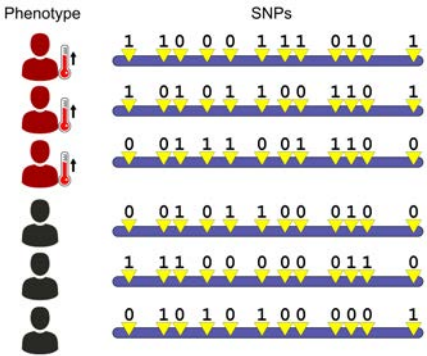
Definition F. Llinares-López et al. KDD 2015

- The goal of *significant pattern mining* is to identify sets of items that occur statistically significantly more often in one class than in the other.



Significant pattern mining

Two other motivating examples

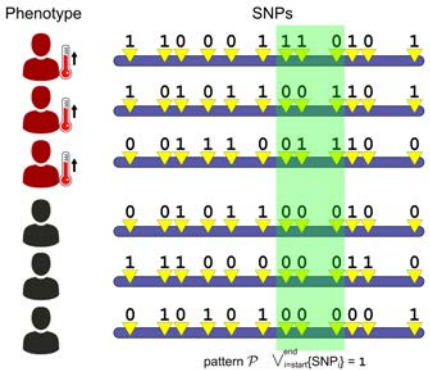


To be discussed in Part II by Laetitia

To be discussed in Part III by Anja

Significant pattern mining

Two other motivating examples

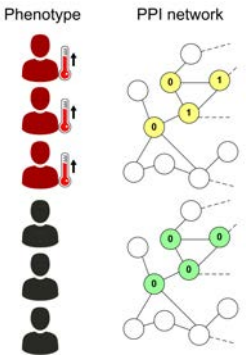


To be discussed in Part II by Laetitia

To be discussed in Part III by Anja

Significant pattern mining

Two other motivating examples



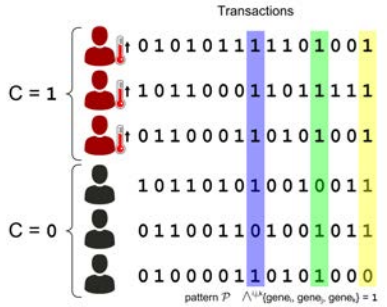
To be discussed in Part III by Anja

To be discussed in Part II by Laetitia

Significant pattern mining

Key aspects

	Pattern $\mathcal{P}$ is present	Pattern $\mathcal{P}$ is not present	
$C = 1$	a	$n_1 - a$	n_1
$C = 0$	$x - a$	$(n - n_1) - (x - a)$	$n - n_1$
	x	$n - x$	n



- Where

  n : total number of transactions

  n_1 : number of transactions with class label $C = 1$

  x : support of the pattern $\mathcal{P}$, i.e. number of transactions where $\mathcal{P}$ is present

  a : support of the pattern $\mathcal{P}$ in transactions of class $C = 1$

What is not significant pattern mining

Frequent itemset mining

Goal: Identify sets of products that are jointly bought by most customers

Significant pattern mining

Statistical association

	Pattern $\mathcal{P}$ is present	Pattern $\mathcal{P}$ is not present	
$C = 1$	a	$n_1 - a$	n_1
$C = 0$	$x - a$	$(n - n_1) - (x - a)$	$n - n_1$
	x	$n - x$	n

- Compute p-value based on a, x, n_1 and n
- Use Fisher's Exact Test R.A. Fisher, 1922
  - 2×2 contingency table
  - Marginals are assumed to be fixed (row and column totals)
- Must guarantee Family-wise Error Rate (FWER) $< \alpha$

Family-wise Error Rate (FWER)

Definition Y. Benjamini and Y. Hochberg, 1995

- Is the probability that at least one false discovery (type I error) occurs in multiple tests

	Number not rejected	Number rejected	
True null hypothesis	U	V	m_0
Non-true null hypothesis	T	S	m_1
	$m - R$	R	m

- V is the number of false positives
- FWER $= Pr(V \geq 1)$
- Increases at most linearly as the number of tests increases
  - Motivates the use of the Bonferroni correction

Multiple hypothesis testing

Adjustement of p-values

- Exponential growth in the number of patterns analyzed
  In our first example, all possible patterns of any size s in N genes, $\sum_{s=1}^{N} \binom{N}{s} = 2^N$
- Therefore, we must correct for multiple hypothesis testing

Bonferroni correction

- For each H_i, with $i = 1 \dots m$ we obtain a p-value p_i
- Corrected significance level $\delta = \frac{\alpha}{m}$
- Reject H_i if $p_i \leq \delta$
  - If m is large, we incur in loss of statistical power $\rightarrow$ nothing is significant
  - *Question: Can we correct using $k \ll m$?*

Multiple hypothesis testing

Adjustement of p-values

- Exponential growth in the number of patterns analyzed

  In our first example, all possible patterns of any size s in N genes, $\sum_{s=1}^{N} \binom{N}{s} = 2^N$
- Therefore, we must correct for multiple hypothesis testing

Bonferroni correction

- For each H_i, with $i = 1 \dots m$ we obtain a p-value p_i
- Corrected significance level $\delta = \frac{\alpha}{m}$
- Reject H_i if $p_i \leq \delta$
  - If m is large, we incur in loss of statistical power $\rightarrow$ nothing is significant
  - **Question**: Can we correct using $k \ll m$?

Multiple hypothesis testing

Adjustement of p-values

- Exponential growth in the number of patterns analyzed

  In our first example, all possible patterns of any size s in N genes, $\sum_{s=1}^{N} \binom{N}{s} = 2^N$
- Therefore, we must correct for multiple hypothesis testing

Bonferroni correction

- For each H_i, with $i = 1 \dots m$ we obtain a p-value p_i
- Corrected significance level $\delta = \frac{\alpha}{m}$
- Reject H_i if $p_i \leq \delta$
  - If m is large, we incur in loss of statistical power $\rightarrow$ nothing is significant
  - **Question**: Can we correct using $k \ll m$?

Testability

Deconstructing Fisher's Exact Test

	a	A	Total
Controls	**4**	**6**	10
Cases	**1**	**6**	7
Total	5	12	17

Example: Association test in GWAS

- p-value (two-sided) = 0.338235
- Null hypothesis: no association of alleles in cases/controls

- Enumeration of all matrices

$$\begin{bmatrix}5 & 5\\ 0 & 7\end{bmatrix} \quad \leftarrow \begin{bmatrix}\mathbf{4} & \mathbf{6}\\ \mathbf{1} & \mathbf{6}\end{bmatrix} \rightarrow \quad \begin{bmatrix}3 & 7\\ 2 & 5\end{bmatrix} \rightarrow \quad \begin{bmatrix}2 & 8\\ 3 & 4\end{bmatrix} \rightarrow \quad \begin{bmatrix}1 & 9\\ 4 & 3\end{bmatrix} \rightarrow \quad \begin{bmatrix}0 & 10\\ 5 & 2\end{bmatrix}$$

$p = 0.040724$ $\quad$ $p = 0.237557$ $\quad$ $p = 0.407240$ $\quad$ $p = 0.254525$ $\quad$ $p = 0.056561$ $\quad$ $p = 0.003394$

- Where each p is obtained from the hyper-geometric distribution

$$\begin{matrix}x_{11} & x_{12} & r_1\\ x_{21} & x_{22} & r_2\\ c_1 & c_2 & n\end{matrix} \rightarrow p = \frac{\binom{r_1}{x_{11}}\binom{r_2}{x_{21}}}{\binom{n}{c_1}}$$

- Fisher's p-value

$p = 0.338235 = 0.040724 + 0.237557 + 0.056561 + 0.003394$

Using "biased matrices"

Testability

Deconstructing Fisher's Exact Test

	a	A	Total
Controls	**4**	**6**	10
Cases	**1**	**6**	7
Total	5	12	17

Example: Association test in GWAS

- p-value (two-sided) = 0.338235
- Null hypothesis: no association of alleles in cases/controls

- Enumeration of all matrices

$$\begin{bmatrix}5 & 5\\ 0 & 7\end{bmatrix} \leftarrow \begin{bmatrix}\mathbf{4} & \mathbf{6}\\ \mathbf{1} & \mathbf{6}\end{bmatrix} \rightarrow \begin{bmatrix}3 & 7\\ 2 & 5\end{bmatrix} \rightarrow \begin{bmatrix}2 & 8\\ 3 & 4\end{bmatrix} \rightarrow \begin{bmatrix}1 & 9\\ 4 & 3\end{bmatrix} \rightarrow \begin{bmatrix}0 & 10\\ 5 & 2\end{bmatrix}$$

$p = 0.040724$ $p = 0.237557$ $p = 0.407240$ $p = 0.254525$ $p = 0.056561$ $p = 0.003394$

- Where each p is obtained from the hyper-geometric distribution

$$\begin{matrix}x_{11} & x_{12} & r_1\\ x_{21} & x_{22} & r_2\\ c_1 & c_2 & n\end{matrix} \rightarrow p = \frac{\binom{r_1}{x_{11}}\binom{r_2}{x_{21}}}{\binom{n}{c_1}}$$

- Fisher's p-value

$p = 0.338235 = 0.040724 + 0.237557 + 0.056561 + 0.003394$

Using "biased matrices"

Testability

Deconstructing Fisher's Exact Test

	a	A	Total
Controls	**4**	**6**	10
Cases	**1**	**6**	7
Total	5	12	17

Example: Association test in GWAS

- p-value (two-sided) = 0.338235
- Null hypothesis: no association of alleles in cases/controls

- Enumeration of all matrices

$$\begin{bmatrix}5 & 5\\ 0 & 7\end{bmatrix} \leftarrow \begin{bmatrix}\mathbf{4} & \mathbf{6}\\ \mathbf{1} & \mathbf{6}\end{bmatrix} \rightarrow \begin{bmatrix}3 & 7\\ 2 & 5\end{bmatrix} \rightarrow \begin{bmatrix}2 & 8\\ 3 & 4\end{bmatrix} \rightarrow \begin{bmatrix}1 & 9\\ 4 & 3\end{bmatrix} \rightarrow \begin{bmatrix}0 & 10\\ 5 & 2\end{bmatrix}$$

$p = 0.040724$ $p = 0.237557$ $p = 0.407240$ $p = 0.254525$ $p = 0.056561$ $p = 0.003394$

- Where each p is obtained from the hyper-geometric distribution

$$\begin{matrix}\begin{bmatrix}x_{11} & x_{12}\\ x_{21} & x_{22}\end{bmatrix} & \begin{matrix}r_1\\ r_2\end{matrix}\\ \begin{matrix}c_1 & c_2\end{matrix} & n\end{matrix} \rightarrow p = \frac{\binom{r_1}{x_{11}}\binom{r_2}{x_{21}}}{\binom{n}{c_1}}$$

- Fisher's p-value

$p = 0.338235 = 0.040724 + 0.237557 + 0.056561 + 0.003394$

Using "biased matrices"

Testability

Deconstructing Fisher's Exact Test

	a	A	Total
Controls	**4**	**6**	10
Cases	**1**	**6**	7
Total	5	12	17

Example: Association test in GWAS

- p-value (two-sided) = 0.338235
- Null hypothesis: no association of alleles in cases/controls

- Enumeration of all matrices

$$\begin{bmatrix}5 & 5\\ 0 & 7\end{bmatrix} \leftarrow \begin{bmatrix}\mathbf{4} & \mathbf{6}\\ \mathbf{1} & \mathbf{6}\end{bmatrix} \rightarrow \begin{bmatrix}3 & 7\\ 2 & 5\end{bmatrix} \rightarrow \begin{bmatrix}2 & 8\\ 3 & 4\end{bmatrix} \rightarrow \begin{bmatrix}1 & 9\\ 4 & 3\end{bmatrix} \rightarrow \begin{bmatrix}0 & 10\\ 5 & 2\end{bmatrix}$$

$p = 0.040724$ $p = 0.237557$ $p = 0.407240$ $p = 0.254525$ $p = 0.056561$ $p = 0.003394$

- Where each p is obtained from the hyper-geometric distribution

$$\begin{matrix}\begin{bmatrix}x_{11} & x_{12}\\ x_{21} & x_{22}\end{bmatrix} & \begin{matrix}r_1\\ r_2\end{matrix}\\ \begin{matrix}c_1 & c_2\end{matrix} & n\end{matrix} \rightarrow p = \frac{\binom{r_1}{x_{11}}\binom{r_2}{x_{21}}}{\binom{n}{c_1}}$$

- Fisher's p-value

$p = 0.338235 = 0.040724 + 0.237557 + 0.056561 + 0.003394$

Using "biased matrices"

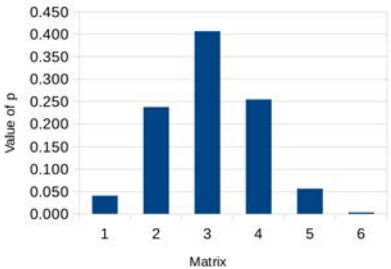
Testability

Minimum attainable p-value

1	2	3	4	5	6
$\begin{bmatrix} 5 & 5 \\ 0 & 7 \end{bmatrix}$	$\begin{bmatrix} 4 & 6 \\ 1 & 6 \end{bmatrix}$	$\begin{bmatrix} 3 & 7 \\ 2 & 5 \end{bmatrix}$	$\begin{bmatrix} 2 & 8 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 1 & 9 \\ 4 & 3 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 5 & 2 \end{bmatrix}$
$p = 0.040724$	$p = 0.237557$	$p = 0.407240$	$p = 0.254525$	$p = 0.056561$	$p = 0.003394$

Key elements

- Distribution of p is discrete
- p_{min} in most biased matrix
  - Statistical test on original matrix cannot give a p-value $< p_{min}$



Testability

Minimum attainable p-value

$$\begin{bmatrix}5 & 5\\ 0 & 7\end{bmatrix} \leftarrow \begin{bmatrix}\mathbf{4} & \mathbf{6}\\ \mathbf{1} & \mathbf{6}\end{bmatrix} \rightarrow \begin{bmatrix}3 & 7\\ 2 & 5\end{bmatrix} \rightarrow \begin{bmatrix}2 & 8\\ 3 & 4\end{bmatrix} \rightarrow \begin{bmatrix}1 & 9\\ 4 & 3\end{bmatrix} \rightarrow \begin{bmatrix}0 & 10\\ 5 & 2\end{bmatrix}$$

$p = 0.040724$ $\quad$ $p = 0.237557$ $\quad$ $p = 0.407240$ $\quad$ $p = 0.254525$ $\quad$ $p = 0.056561$ $\quad$ $p = 0.003394$

Most biased matrices (when $r_1 \leq r_2$)

if $r_1 \geq c_1$, then $\begin{bmatrix}c_1 & x_{12} - x_{21} & r_1\\ 0 & x_{22} + x_{21} & r_2\\ c_1 & c_2 & n\end{bmatrix}$

with $p_{min} = \binom{r_1}{c_1} \Big/ \binom{n}{c_1}$

otherwise $\begin{bmatrix}0 & x_{12} + x_{11} & r_1\\ c_1 & x_{22} - x_{11} & r_2\\ c_1 & c_2 & n\end{bmatrix}$

with $p_{min} = \binom{r_2}{c_1} \Big/ \binom{n}{c_1}$

Reducing the Bonferroni correction factor

An illustrative example

- Perform association tests on $m = 5$ SNPs
- Significance level $\alpha = 0.05$
- With Bonferroni correction

  $\rightarrow \delta = \frac{\alpha}{m} = 0.01$

	a	A	Total
Controls	x_{11}	x_{12}	r_1
Cases	x_{21}	x_{22}	r_2
Total	c_1	c_2	n

Id	Observed	Fisher's p-value
SNP_1	$\begin{bmatrix} 2 & 6 \\ 1 & 6 \end{bmatrix}$	0.2
SNP_2	$\begin{bmatrix} 2 & 8 \\ 2 & 7 \end{bmatrix}$	1.0
SNP_3	$\begin{bmatrix} 2 & 8 \\ 7 & 1 \end{bmatrix}$	0.015220
SNP_4	$\begin{bmatrix} 3 & 11 \\ 2 & 7 \end{bmatrix}$	1.0
SNP_5	$\begin{bmatrix} 1 & 9 \\ 3 & 5 \end{bmatrix}$	0.274510

- After correction for multiple hypothesis, there are no statistically significant associations
- How can we improve on these results using the p_{min} of each SNP?

Reducing the Bonferroni correction factor

An illustrative example

- Perform association tests on $m = 5$ SNPs
- Significance level $\alpha = 0.05$
- With Bonferroni correction

  $\rightarrow \delta = \frac{\alpha}{m} = 0.01$

	a	A	Total
Controls	x_{11}	x_{12}	r_1
Cases	x_{21}	x_{22}	r_2
Total	c_1	c_2	n

Id	Observed	Fisher's p-value
SNP_1	$\begin{bmatrix} 2 & 6 \\ 1 & 6 \end{bmatrix}$	0.2
SNP_2	$\begin{bmatrix} 2 & 8 \\ 2 & 7 \end{bmatrix}$	1.0
SNP_3	$\begin{bmatrix} 2 & 8 \\ 7 & 1 \end{bmatrix}$	0.015220
SNP_4	$\begin{bmatrix} 3 & 11 \\ 2 & 7 \end{bmatrix}$	1.0
SNP_5	$\begin{bmatrix} 1 & 9 \\ 3 & 5 \end{bmatrix}$	0.274510

- After correction for multiple hypothesis, there are no statistically significant associations
- How can we improve on these results using the p_{min} of each SNP?

Reducing the Bonferroni correction factor

Eliminate tests where $p_{min} < \alpha$ N. Manthel, 1980

Id	Observed	Fisher's p-value	Most biased	p_{min}
SNP_1	$\begin{bmatrix} 2 & 6 \\ 1 & 6 \end{bmatrix}$	0.2	$\begin{bmatrix} 0 & 8 \\ 3 & 4 \end{bmatrix}$	0.076923
SNP_2	$\begin{bmatrix} 2 & 8 \\ 2 & 7 \end{bmatrix}$	1.0	$\begin{bmatrix} 0 & 10 \\ 4 & 5 \end{bmatrix}$	0.032508
SNP_3	$\begin{bmatrix} 2 & 8 \\ 7 & 1 \end{bmatrix}$	0.015220	$\begin{bmatrix} 1 & 9 \\ 8 & 0 \end{bmatrix}$	0.000206
SNP_4	$\begin{bmatrix} 3 & 11 \\ 2 & 7 \end{bmatrix}$	1.0	$\begin{bmatrix} 0 & 14 \\ 5 & 4 \end{bmatrix}$	0.003745
SNP_5	$\begin{bmatrix} 1 & 9 \\ 3 & 5 \end{bmatrix}$	0.274510	$\begin{bmatrix} 0 & 10 \\ 4 & 4 \end{bmatrix}$	0.022876

- SNP_1 is eliminated from the analysis, its $p_{min} > \alpha$. It is untestable
- Then, $k = 4$ and $\delta = \frac{\alpha}{k} = 0.0125$. Yet, no statistically association after correction

Reducing the Bonferroni correction factor

Eliminate tests where $p_{min} < \alpha$ N. Manthel, 1980

Id	Observed	Fisher's p-value	Most biased	p_{min}
SNP_1	$\begin{bmatrix} 2 & 6 \\ 1 & 6 \end{bmatrix}$	0.2	$\begin{bmatrix} 0 & 8 \\ 3 & 4 \end{bmatrix}$	0.076923
SNP_2	$\begin{bmatrix} 2 & 8 \\ 2 & 7 \end{bmatrix}$	1.0	$\begin{bmatrix} 0 & 10 \\ 4 & 5 \end{bmatrix}$	0.032508
SNP_3	$\begin{bmatrix} 2 & 8 \\ 7 & 1 \end{bmatrix}$	0.015220	$\begin{bmatrix} 1 & 9 \\ 8 & 0 \end{bmatrix}$	0.000206
SNP_4	$\begin{bmatrix} 3 & 11 \\ 2 & 7 \end{bmatrix}$	1.0	$\begin{bmatrix} 0 & 14 \\ 5 & 4 \end{bmatrix}$	0.003745
SNP_5	$\begin{bmatrix} 1 & 9 \\ 3 & 5 \end{bmatrix}$	0.274510	$\begin{bmatrix} 0 & 10 \\ 4 & 4 \end{bmatrix}$	0.022876

- SNP_1 is eliminated from the analysis, its $p_{min} > \alpha$. It is untestable
- Then, $k = 4$ and $\delta = \frac{\alpha}{k} = 0.0125$. Yet, no statistically association after correction

Reducing the Bonferroni correction factor

Tarone's method R.E. Tarone, 1990

procedure main($\mathcal{H}$, α)
▷ $\mathcal{H}$: Set of all hypotheses
▷ α: Nominal significance level
    $k \leftarrow 0$
    repeat
        $k \leftarrow k + 1$
        $\mathcal{T} \leftarrow$ `get_testable_set`$(\mathcal{H}, \frac{\alpha}{k})$
    until $k \geq |\mathcal{T}|$
▷ Ready to perform Fisher's Exact Tests
    $\delta \leftarrow \frac{\alpha}{k}$
    `perform_fisher_exact_tests`$(\mathcal{H}_{\mathcal{T}}, \delta)$

function `get_testable_set`($\mathcal{H}$, δ)
▷ Determine all testable hypotheses
    $m \leftarrow |\mathcal{H}|$
    $\mathcal{T} \leftarrow \emptyset$
    for $i \leftarrow 1, m$ **do**
        if `is_testable`$(\mathcal{H}_i, \delta)$ **then**
            $\mathcal{T} \leftarrow \{\mathcal{T}\} \cup i$
    return $\mathcal{T}$

function `is_testable`(h, δ)
▷ Check if hypothesis h is testable
    $p_{min} \leftarrow$ `compute_min_pvalue`(h)
    if $p_{min} > \delta$ **then**
        return False
    return True

Reducing the Bonferroni correction factor

Tarone's method R.E. Tarone, 1990

- Intuition

  At the end of the loop we have $k \geq |\mathcal{T}|$
  This implies:

$$|\mathcal{T}| \leq k$$
$$\alpha|\mathcal{T}| \leq \alpha k$$
$$\frac{\alpha}{k}|\mathcal{T}| \leq \alpha$$

  Therefore FWER $\leq \delta|\mathcal{T}| \leq \alpha$

procedure main($\mathcal{H}$, α)
▷ $\mathcal{H}$: Set of all hypotheses
▷ α: Nominal significance level
  $k \leftarrow 0$
  repeat
    $k \leftarrow k + 1$
    $\mathcal{T} \leftarrow$ `get_testable_set`$(\mathcal{H}, \frac{\alpha}{k})$
  until $k \geq |\mathcal{T}|$

Reducing the Bonferroni correction factor

Tarone's method R.E. Tarone, 1990

repeat
$k \leftarrow k + 1$
$\mathcal{T} \leftarrow$ `get_testable_set`$(\mathcal{H}, \frac{\alpha}{k})$
until $k \geq |\mathcal{T}|$

Id	Observed	Most biased	Min. p-value
SNP_1	$\begin{bmatrix} 2 & 6 \\ 1 & 6 \end{bmatrix}$	$\begin{bmatrix} 0 & 8 \\ 3 & 4 \end{bmatrix}$	0.076923
SNP_2	$\begin{bmatrix} 2 & 8 \\ 2 & 7 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 4 & 5 \end{bmatrix}$	0.032508
SNP_3	$\begin{bmatrix} 2 & 8 \\ 7 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 9 \\ 8 & 0 \end{bmatrix}$	0.000206
SNP_4	$\begin{bmatrix} 3 & 11 \\ 2 & 7 \end{bmatrix}$	$\begin{bmatrix} 0 & 14 \\ 5 & 4 \end{bmatrix}$	0.003745
SNP_5	$\begin{bmatrix} 1 & 9 \\ 3 & 5 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 4 & 4 \end{bmatrix}$	0.022876

Reducing the Bonferroni correction factor

Tarone's method R.E. Tarone, 1990

repeat
$k \leftarrow k + 1$
$\mathcal{T} \leftarrow$ `get_testable_set`$(\mathcal{H}, \frac{\alpha}{k})$
until $k \geq |\mathcal{T}|$

- With $k = 1$, $\delta = 0.05$, $\mathcal{T} = \{2, 3, 4, 5\}$

  Condition $k \geq |\mathcal{T}|$ is False $\rightarrow$ next iteration

Id	Observed	Most biased	Min. p-value
SNP_1	$\begin{bmatrix} 2 & 6 \\ 1 & 6 \end{bmatrix}$	$\begin{bmatrix} 0 & 8 \\ 3 & 4 \end{bmatrix}$	0.076923
SNP_2	$\begin{bmatrix} 2 & 8 \\ 2 & 7 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 4 & 5 \end{bmatrix}$	0.032508
SNP_3	$\begin{bmatrix} 2 & 8 \\ 7 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 9 \\ 8 & 0 \end{bmatrix}$	0.000206
SNP_4	$\begin{bmatrix} 3 & 11 \\ 2 & 7 \end{bmatrix}$	$\begin{bmatrix} 0 & 14 \\ 5 & 4 \end{bmatrix}$	0.003745
SNP_5	$\begin{bmatrix} 1 & 9 \\ 3 & 5 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 4 & 4 \end{bmatrix}$	0.022876

Reducing the Bonferroni correction factor

Tarone's method R.E. Tarone, 1990

repeat
$k \leftarrow k + 1$
$\mathcal{T} \leftarrow \texttt{get_testable_set}(\mathcal{H}, \frac{\alpha}{k})$
until $k \geq |\mathcal{T}|$

- With $k = 1$, $\delta = 0.05$, $\mathcal{T} = \{2, 3, 4, 5\}$
  Condition $k \geq |\mathcal{T}|$ is False $\rightarrow$ next iteration
- With $k = 2$, $\delta = 0.025$, $\mathcal{T} = \{3, 4, 5\}$
  Condition $k \geq |\mathcal{T}|$ is False $\rightarrow$ next iteration

Id	Observed	Most biased	Min. p-value
SNP_1	$\begin{bmatrix} 2 & 6 \\ 1 & 6 \end{bmatrix}$	$\begin{bmatrix} 0 & 8 \\ 3 & 4 \end{bmatrix}$	0.076923
SNP_2	$\begin{bmatrix} 2 & 8 \\ 2 & 7 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 4 & 5 \end{bmatrix}$	0.032508
SNP_3	$\begin{bmatrix} 2 & 8 \\ 7 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 9 \\ 8 & 0 \end{bmatrix}$	0.000206
SNP_4	$\begin{bmatrix} 3 & 11 \\ 2 & 7 \end{bmatrix}$	$\begin{bmatrix} 0 & 14 \\ 5 & 4 \end{bmatrix}$	0.003745
SNP_5	$\begin{bmatrix} 1 & 9 \\ 3 & 5 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 4 & 4 \end{bmatrix}$	0.022876

Reducing the Bonferroni correction factor

Tarone's method R.E. Tarone, 1990

repeat
$k \leftarrow k + 1$
$\mathcal{T} \leftarrow \texttt{get_testable_set}(\mathcal{H}, \frac{\alpha}{k})$
until $k \geq |\mathcal{T}|$

- With $k = 1$, $\delta = 0.05$, $\mathcal{T} = \{2, 3, 4, 5\}$
  Condition $k \geq |\mathcal{T}|$ is False $\rightarrow$ next iteration
- With $k = 2$, $\delta = 0.025$, $\mathcal{T} = \{3, 4, 5\}$
  Condition $k \geq |\mathcal{T}|$ is False $\rightarrow$ next iteration
- With $k = 3$, $\delta = 0.0167$, $\mathcal{T} = \{3, 4\}$
  $k \geq |\mathcal{T}|$ evaluates to True $\rightarrow$ Stop

Id	Observed	Most biased	Min. p-value
SNP_1	$\begin{bmatrix} 2 & 6 \\ 1 & 6 \end{bmatrix}$	$\begin{bmatrix} 0 & 8 \\ 3 & 4 \end{bmatrix}$	0.076923
SNP_2	$\begin{bmatrix} 2 & 8 \\ 2 & 7 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 4 & 5 \end{bmatrix}$	0.032508
SNP_3	$\begin{bmatrix} 2 & 8 \\ 7 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 9 \\ 8 & 0 \end{bmatrix}$	0.000206
SNP_4	$\begin{bmatrix} 3 & 11 \\ 2 & 7 \end{bmatrix}$	$\begin{bmatrix} 0 & 14 \\ 5 & 4 \end{bmatrix}$	0.003745
SNP_5	$\begin{bmatrix} 1 & 9 \\ 3 & 5 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 4 & 4 \end{bmatrix}$	0.022876

Reducing the Bonferroni correction factor

Tarone's method R.E. Tarone, 1990

repeat
  $k \leftarrow k + 1$
  $\mathcal{T} \leftarrow \texttt{get_testable_set}(\mathcal{H}, \frac{\alpha}{k})$
until $k \geq |\mathcal{T}|$

▷ $k = 3$
▷ $\delta = \frac{\alpha}{k} = 0.0167$
▷ $\mathcal{T} = \{3, 4\}$

▷ Perform Fisher's Exact Test on SNP_3 and SNP_4

Id	Observed	Most biased	Min. p-value
SNP_1	$\begin{bmatrix} 2 & 6 \\ 1 & 6 \end{bmatrix}$	$\begin{bmatrix} 0 & 8 \\ 3 & 4 \end{bmatrix}$	0.076923
SNP_2	$\begin{bmatrix} 2 & 8 \\ 2 & 7 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 4 & 5 \end{bmatrix}$	0.032508
SNP_3	$\begin{bmatrix} 2 & 8 \\ 7 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 9 \\ 8 & 0 \end{bmatrix}$	0.000206
SNP_4	$\begin{bmatrix} 3 & 11 \\ 2 & 7 \end{bmatrix}$	$\begin{bmatrix} 0 & 14 \\ 5 & 4 \end{bmatrix}$	0.003745
SNP_5	$\begin{bmatrix} 1 & 9 \\ 3 & 5 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 4 & 4 \end{bmatrix}$	0.022876

Reducing the Bonferroni correction factor

Tarone's method R.E. Tarone, 1990

repeat
  $k \leftarrow k + 1$
  $\mathcal{T} \leftarrow$ `get_testable_set`$(\mathcal{H}, \frac{\alpha}{k})$
until $k \geq |\mathcal{T}|$

▷ $k = 3$
▷ $\delta = \frac{\alpha}{k} = 0.0167$
▷ $\mathcal{T} = \{3, 4\}$

Id	Observed	Most biased	Min. p-value
SNP_1	$\begin{bmatrix} 2 & 6 \\ 1 & 6 \end{bmatrix}$	$\begin{bmatrix} 0 & 8 \\ 3 & 4 \end{bmatrix}$	0.076923
SNP_2	$\begin{bmatrix} 2 & 8 \\ 2 & 7 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 4 & 5 \end{bmatrix}$	0.032508
SNP_3	$\begin{bmatrix} 2 & 8 \\ 7 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 9 \\ 8 & 0 \end{bmatrix}$	0.000206
SNP_4	$\begin{bmatrix} 3 & 11 \\ 2 & 7 \end{bmatrix}$	$\begin{bmatrix} 0 & 14 \\ 5 & 4 \end{bmatrix}$	0.003745
SNP_5	$\begin{bmatrix} 1 & 9 \\ 3 & 5 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 4 & 4 \end{bmatrix}$	0.022876

▷ Perform Fisher's Exact Test on SNP_3 and SNP_4

$SNP_3 \rightarrow p\text{-value} = 0.015220$

$SNP_4 \rightarrow p\text{-value} = 1.0$

- SNP_3 is statistically significant at level δ

Reducing the Bonferroni correction factor

Tarone's method R.E. Tarone, 1990

repeat
$k \leftarrow k + 1$
$\mathcal{T} \leftarrow$ `get_testable_set`$(\mathcal{H}, \frac{\alpha}{k})$
until $k \geq |\mathcal{T}|$
▷ $k = 3$
▷ $\delta = \frac{\alpha}{k} = 0.0167$
▷ $\mathcal{T} = \{3, 4\}$

Id	Observed	Most biased	Min. p-value
SNP_1	$\begin{bmatrix} 2 & 6 \\ 1 & 6 \end{bmatrix}$	$\begin{bmatrix} 0 & 8 \\ 3 & 4 \end{bmatrix}$	0.076923
SNP_2	$\begin{bmatrix} 2 & 8 \\ 2 & 7 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 4 & 5 \end{bmatrix}$	0.032508
SNP_3	$\begin{bmatrix} 2 & 8 \\ 7 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 9 \\ 8 & 0 \end{bmatrix}$	0.000206
SNP_4	$\begin{bmatrix} 3 & 11 \\ 2 & 7 \end{bmatrix}$	$\begin{bmatrix} 0 & 14 \\ 5 & 4 \end{bmatrix}$	0.003745
SNP_5	$\begin{bmatrix} 1 & 9 \\ 3 & 5 \end{bmatrix}$	$\begin{bmatrix} 0 & 10 \\ 4 & 4 \end{bmatrix}$	0.022876

▷ Perform Fisher's Exact Test on SNP_3 and SNP_4

$SNP_3 \rightarrow p\text{-value} = 0.015220$

$SNP_4 \rightarrow p\text{-value} = 1.0$

- **SNP_3 is statistically significant at level δ**

Reducing the Bonferroni correction factor

Tarone's method R.E. Tarone, 1990

- Contrast to Bonferroni correction with $m = 5$
  - $\delta = \frac{\alpha}{5} = 0.01$
  - No significant association would have been found

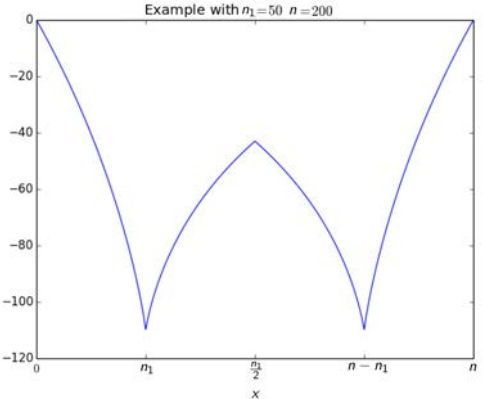
Id	Observed	Fisher's p-value
SNP_1	$\begin{bmatrix} 2 & 6 \\ 1 & 6 \end{bmatrix}$	0.2
SNP_2	$\begin{bmatrix} 2 & 8 \\ 2 & 7 \end{bmatrix}$	1.0
SNP_3	$\begin{bmatrix} 2 & 8 \\ 7 & 1 \end{bmatrix}$	0.015220
SNP_4	$\begin{bmatrix} 3 & 11 \\ 2 & 7 \end{bmatrix}$	1.0
SNP_5	$\begin{bmatrix} 1 & 9 \\ 3 & 5 \end{bmatrix}$	0.274510

Final thoughts

Pre-computing minimum attainable p-values

	Pattern $\mathcal{P}$ is present	Pattern $\mathcal{P}$ is not present	
$C = 1$	a	$n_1 - a$	n_1
$C = 0$	$x - a$	$(n - n_1) - (x - a)$	$n - n_1$
	x	$n - x$	n

- Margins are assumed to be equal for all H_i, e.g. imputed data in GWAS association test
- Therefore, p_{min} can be computed as a function of x



Conclusions of Part I

Key points

- Introduced key aspects of significant pattern mining
- Discussed the concept of minimum attainable p-value
- Applied the Tarone method to obtain a corrected significance level δ_k
- Found $k \ll m$ to correct for multiple hypothesis

Conclusions of Part I

Key points

- Introduced key aspects of significant pattern mining
- Discussed the concept of minimum attainable p-value
- Applied the Tarone method to obtain a corrected significance level δ_k
- Found $k \ll m$ to correct for multiple hypothesis

In Parts II and III

- How are the patterns defined?
- What test statistic is used?
- How is the search space pruned?
- Are the final results correlated in any way? Post-processing?

References

- R.A. Fisher. On the interpretation of χ^2 from contingency tables, and the calculation of P. Journal of the Royal Statistical Society 85 (1): 87–94. (1922)
- N. Mantel. Assessing laboratory evidence for neoplastic activity. Biometrics 36, 381-399. Correction, Biometrics 37, 875. (1980)
- R.E. Tarone. A modified Bonferroni method for discrete data. Biometrics 46, 515 (1990)
- F. Llinares-López, D.G. Grimm, D.A. Bodenham, U. Gieraths, M. Sugiyama, B. Rowan, K. Borgwardt. Genome-wide detection of intervals of genetic heterogeneity associated with complex traits. Bioinformatics. 31(12):i240-9 (2015)

Part II
Genome-wide genetic heterogeneity detection with categorical covariates

By Laetitia Papaxanthos

Outline

1. Genomic interactions problem statement
2. Statistical testing and correction for confounders
3. Methods: Fast Automatic Interval Search (`FAIS`) and `FastCMH` algorithms
4. Results on plant and human datasets
5. Summary and outlook

Table of Contents

1 Genomic interactions problem statement

2 Statistical testing and correction for confounders

3 Methods: Fast Automatic Interval Search (FAIS) and FastCMH algorithms

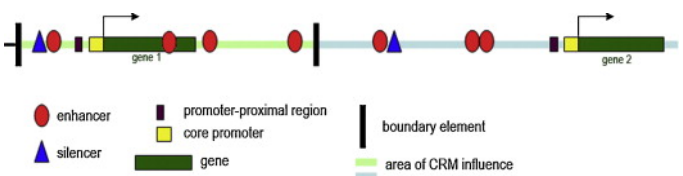
4 Results on plant and human datasets

5 Summary and outlook

Motivation

- Genetic heterogeneity: the phenomenon under which several variants have a common effect on a phenotype.
- High-order interactions discovery methods for complex traits, an attempt to explain the missing heritability.
- Detection of contiguous interactions between SNPs can reveal local Gene-Gene, cis-regulatory elements (CRE)-Gene or CRE-CRE interactions, ≈10bp to 100kb away.

Source: A systems biology approach to understanding cis-regulatory module function Cell and Developmental Biology, Jeziorska 2009



Propositions: Fast Automatic Interval Search (FAIS) and FastCMH

Baseline

10^5 SNPs lead to $\approx 10^9$ pairs of SNPs, $\approx 10^{14}$ triplets...

- Test high-order interactions: all genomic contiguous intervals, without prior discrimination of region function or length.
- Correct the multiple hypothesis testing problem by controlling FWER using Tarone.
- Scalable to $>$ 500000 SNPs and $>$ 5000 samples.

Propositions: Fast Automatic Interval Search (FAIS) and FastCMH

Categorical confounder correction with FastCMH

- Corrects for multiple categorical confounders such as phenotypical traits (age, height...) and population structure.
  - Enables to increase the number of samples by combining world-wide GWASs.

FAIS: Genome-wide detection of intervals of genetic heterogeneity associated with complex traits, Bioinformatics (2015), F. Llinares-Lopez, D. Grimm, D. Bodenham, U. Gieraths, M. Sugiyama, B. Rowan, K. Borgwardt
FastCMH: Genome-wide genetic-heterogeneity discovery with categorical covariates, submitted to Bioinformatics (2016), F. Llinares-Lopez*, L. Papaxanthos*, D. Bodenham, D. Roqueiro, COPDGene, K. Borgwardt

Table of Contents

1 Genomic interactions problem statement

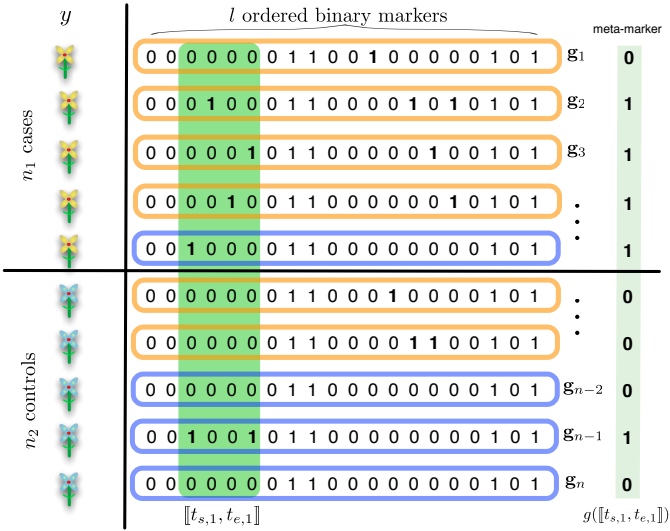
2 Statistical testing and correction for confounders

3 Methods: Fast Automatic Interval Search (FAIS) and FastCMH algorithms

4 Results on plant and human datasets

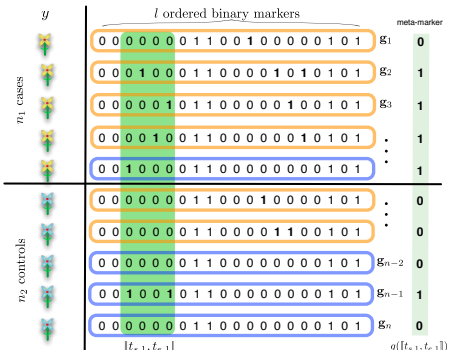
5 Summary and outlook

Genomic intervals coded as meta-markers in GWAS datasets



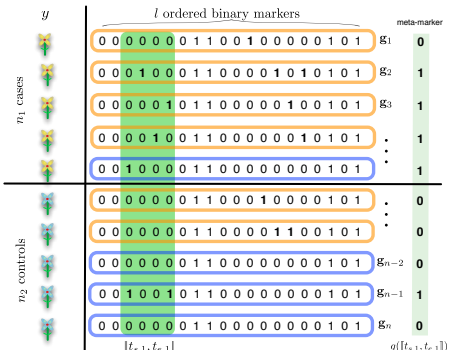
Association testing between meta-markers and phenotype

Variables	*Meta-marker* $= 1$	*Meta-marker* $= 0$	Row totals
$y = \text{case}$	a	$n_1 - a$	n_1 cases
$y = \text{control}$	$x - a$	$n_2 - (x - a)$	n_2 controls
Col totals	x	$n - x$	n



Association testing between meta-markers and phenotype

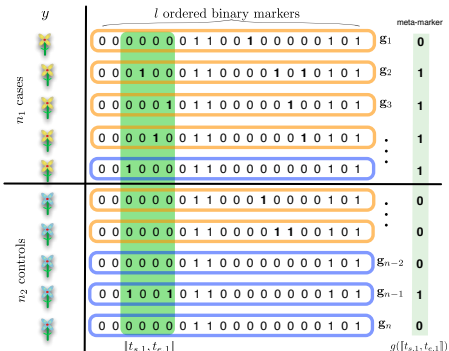
Variables	*Meta-marker* $= 1$	*Meta-marker* $= 0$	Row totals
$y = \text{case}$	a	$n_1 - a$	n_1 cases
$y = \text{control}$	$x - a$	$n_2 - (x - a)$	n_2 controls
Col totals	x	$n - x$	n



- Notation:
  - Genomic interval: $[\![t_e, t_s]\!]$
  - Binary meta-marker: $\mathbf{g}([\![t_e, t_s]\!]) = (g_1, ..., g_n)$

Association testing between meta-markers and phenotype

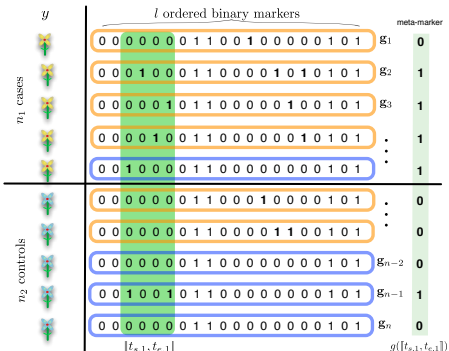
Variables	*Meta-marker* $= 1$	*Meta-marker* $= 0$	Row totals
$y = \text{case}$	a	$n_1 - a$	n_1 cases
$y = \text{control}$	$x - a$	$n_2 - (x - a)$	n_2 controls
Col totals	x	$n - x$	n



- Notation:
  - Genomic interval: $[\![t_e, t_s]\!]$
  - Binary meta-marker: $\mathbf{g}([\![t_e, t_s]\!]) = (g_1, ..., g_n)$
- Corresponding p-value based on entries a, x, n_1 and n_2
  - Fisher's Exact Test or Pearson's χ^2 Test

Association testing between meta-markers and phenotype

Variables	*Meta-marker* $= 1$	*Meta-marker* $= 0$	Row totals
$y =$ case	a	$n_1 - a$	n_1 cases
$y =$ control	$x - a$	$n_2 - (x - a)$	n_2 controls
Col totals	x	$n - x$	n



- Notation:
  - Genomic interval: $[\![t_e, t_s]\!]$
  - Binary meta-marker: $\mathbf{g}([\![t_e, t_s]\!]) = (g_1, ..., g_n)$
- Corresponding p-value based on entries a, x, n_1 and n_2
  - Fisher's Exact Test or Pearson's χ^2 Test
  - How to correct for confounders?

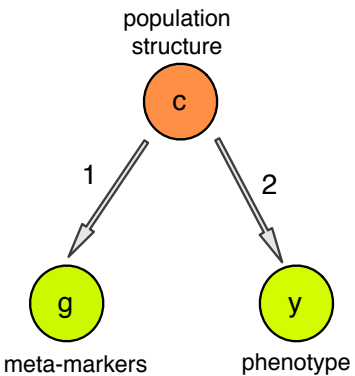
How to correct for confounders ?

Definition

- In statistical genetics, a confounder c is an extraneous variable that influences two conditionally independent variables, for example a phenotypic trait y and a marker g.

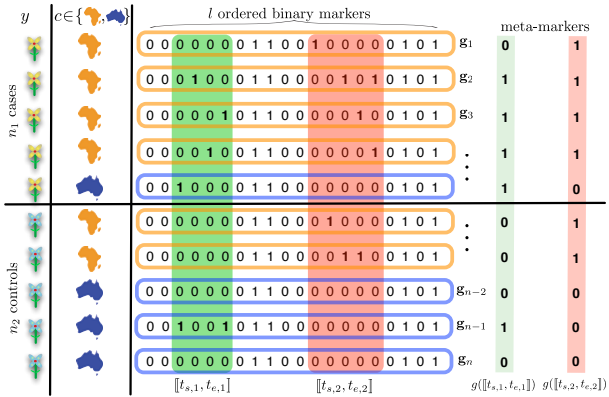
  $y \not\perp\!\!\!\perp g$ but $y \perp\!\!\!\perp g|c$

- It leads to **spurious associations** between the phenotypic trait y and the meta-marker g.



Illustration

Examples of non-confounded and confounded genomic intervals



Correcting for confounders with the Cochran-Mantel-Haenszel (CMH) Test

$\mathbf{g}([\![t_e, t_s]\!]) \in \mathcal{R}^n$ is a meta-marker and k the number of classes of the confounder.
For each class h we define:

- the contingency tables entries: $n_{1,h}$, $n_{2,h}$, x_h and a_h.

CMH Test

The CMH-test is based on the k-vectors $\mathbf{a}$, $\mathbf{x}$, $\mathbf{n_1}$ and $\mathbf{n_2}$.

$$T(\mathbf{a}, \mathbf{x}, \mathbf{n_1}, \mathbf{n_2}) = \frac{\left(\sum_{h=1}^{k} a_h - E(a_h)\right)^2}{\sum_{h=1}^{k} Var(a_h)}$$

$$= \frac{\left(\sum_{h=1}^{k} a_h - x_h \frac{n_{1,h}}{n_h}\right)^2}{\sum_{h=1}^{k} \frac{n_{1,h}}{n_h}\left(1 - \frac{n_{1,h}}{n_h}\right) x_h \left(1 - \frac{x_h}{n_h}\right)}$$

Correcting for confounders with the CMH Test

Corresponding p-value $\Psi(\mathbf{a}, \mathbf{x}, \mathbf{n_1}, \mathbf{n_2})$

$$\Psi(\mathbf{a}, \mathbf{x}, \mathbf{n_1}, \mathbf{n_2}) = 1 - F_{\chi^2}(T(\mathbf{a}, \mathbf{x}, \mathbf{n_1}, \mathbf{n_2}))$$

Table of Contents

1 Genomic interactions problem statement

2 Statistical testing and correction for confounders

3 Methods: Fast Automatic Interval Search (`FAIS`) and `FastCMH` algorithms

4 Results on plant and human datasets

5 Summary and outlook

FAIS and FastCMH architecture in brief

$\mathbf{g}(\llbracket t_e, t_s \rrbracket)$ represents a meta-marker n-vector.

Two steps:

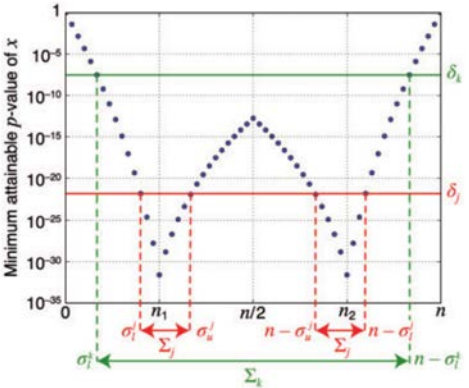
Input: Dataset of meta-markers $\mathcal{G} = \{\hat{\mathbf{g}}, \mathbf{y}, \mathbf{c}\}$, desired FWER α.
Output: Set of non-overlapping (conditionally) associated genomic regions $\mathcal{R}_{sig,filt} = \{\llbracket t_s, t_e \rrbracket \mid p(\llbracket t_s, t_e \rrbracket) \leq \delta_{tar}\}$ and Tarone significance threshold δ_{tar}.

1. $(\delta_{tar}, \mathcal{R}_T(\delta_{tar})) \leftarrow$ `get_significant_regions`$(\mathcal{G}, \alpha)$
2. $\mathcal{R}_{sig,filt} \leftarrow$ `filter_overlapping_regions`$(\mathcal{R}_T(\delta_{tar}))$

Return: $\mathcal{R}_{sig,filt}$

1. Routine `get_significant_regions`: initialization

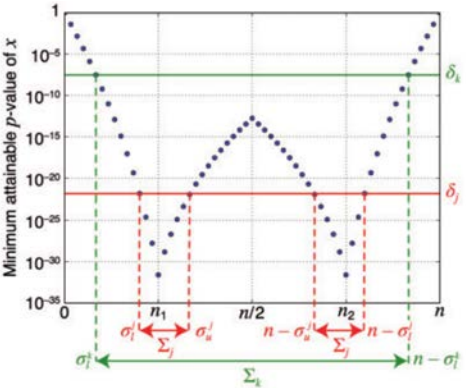
- $\delta \leftarrow 1$, $\mathcal{I}_T(\delta) \leftarrow \{\}$



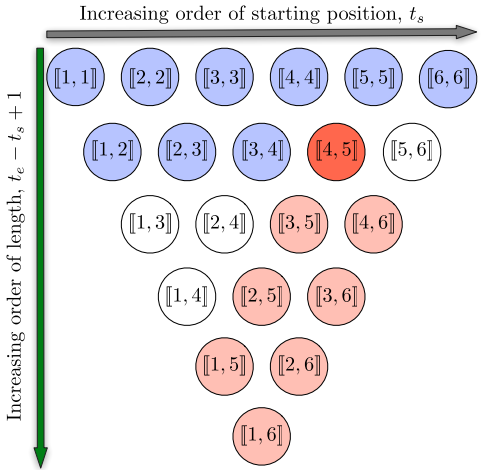
1. Routine `get_significant_regions`: initialization

- $\delta \leftarrow 1$, $\mathcal{I}_T(\delta) \leftarrow \{\}$

For all $[\![t_s, t_e]\!] \in \mathcal{R}_{cand}$, in increasing order of starting position t_s, and then length $t_e - t_s$:



1. Routine `get_sigificant_regions`: interval enumeration

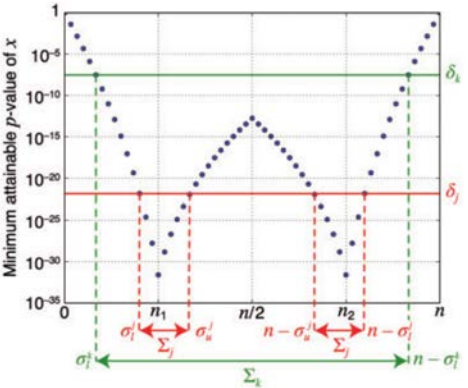


1. Routine `get_significant_regions`: interval processing

- $\delta \leftarrow 1$, $\mathcal{I}_T(\delta) \leftarrow \{\}$

For all $[\![t_s, t_e]\!] \in \mathcal{R}_{cand}$, in increasing order of starting position t_s and then length $t_e - t_s$:

- Compute $x_{[\![t_s,t_e]\!]}$

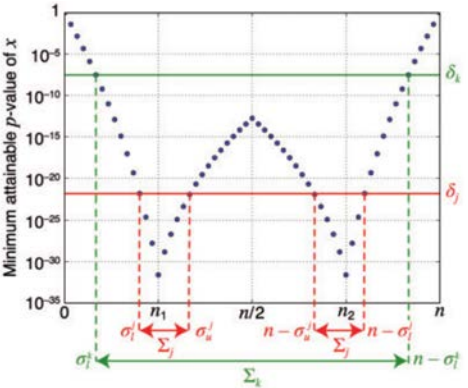


1. Routine `get_significant_regions`: interval processing

- $\delta \leftarrow 1$, $\mathcal{I}_T(\delta) \leftarrow \{\}$

For all $[\![t_s, t_e]\!] \in \mathcal{R}_{cand}$, in increasing order of starting position t_s and then length $t_e - t_s$:

- Compute $x_{[\![t_s,t_e]\!]}$
- If $\Phi(x_{[\![t_s,t_e]\!]}) \leq \delta$: → Tarone's testability criterion



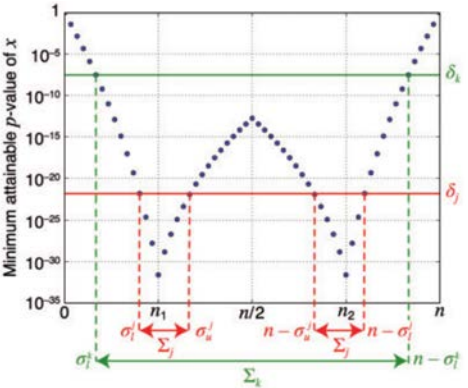
1. Routine `get_significant_regions`: interval processing

- $\delta \leftarrow 1$, $\mathcal{I}_T(\delta) \leftarrow \{\}$

For all $[\![t_s, t_e]\!] \in \mathcal{R}_{cand}$, in increasing order of starting position t_s and then length $t_e - t_s$:

- Compute $x_{[\![t_s,t_e]\!]}$
- If $\Phi(x_{[\![t_s,t_e]\!]}) \leq \delta$: → Tarone's testability criterion

As a reminder:
$\Phi(\mathbf{x}_{[\![t_s,t_e]\!]}) = \min_{\mathbf{a} \in [\![0, \mathbf{x}_{[\![t_s,t_e]\!]}]\!]} \Psi(a, \mathbf{x}_{t_s,t_e})$ is the minimum attainable p-value.

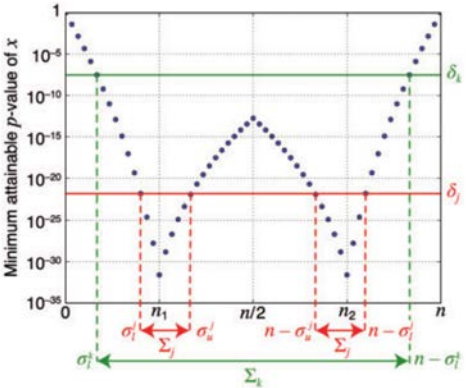


1. Routine `get_significant_regions`: interval processing

- $\delta \leftarrow 1$, $\mathcal{I}_T(\delta) \leftarrow \{\}$

For all $[\![t_s, t_e]\!] \in \mathcal{R}_{cand}$, in increasing order of starting position t_s and then length $t_e - t_s$:

- Compute $x_{[\![t_s,t_e]\!]}$
- If $\Phi(x_{[\![t_s,t_e]\!]}) \leq \delta$: → Tarone's testability criterion
  - $\mathcal{I}_T(\delta) \leftarrow \mathcal{I}_T(\delta) \cup \{[\![t_s, t_e]\!]\}$

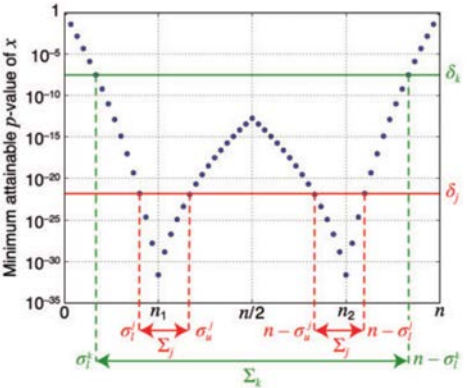


1. Routine `get_significant_regions`: interval processing

- $\delta \leftarrow 1$, $\mathcal{I}_T(\delta) \leftarrow \{\}$

For all $[\![t_s, t_e]\!] \in \mathcal{R}_{cand}$, in increasing order of starting position t_s and then length $t_e - t_s$:

- Compute $x_{[\![t_s,t_e]\!]}$
- If $\Phi(x_{[\![t_s,t_e]\!]}) \leq \delta$: → Tarone's testability criterion
  - $\mathcal{I}_T(\delta) \leftarrow \mathcal{I}_T(\delta) \cup \{[\![t_s, t_e]\!]\}$
  - While $\delta\,|\mathcal{I}_T(\delta)| > \alpha$: → check $\widehat{FWER}$

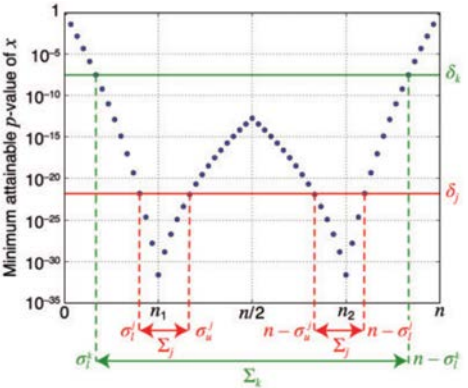


1. Routine `get_significant_regions`: interval processing

- $\delta \leftarrow 1$, $\mathcal{I}_T(\delta) \leftarrow \{\}$

For all $[\![t_s, t_e]\!] \in \mathcal{R}_{cand}$, in increasing order of starting position t_s and then length $t_e - t_s$:

- Compute $x_{[\![t_s,t_e]\!]}$
- If $\Phi(x_{[\![t_s,t_e]\!]}) \leq \delta$: → Tarone's testability criterion
  - $\mathcal{I}_T(\delta) \leftarrow \mathcal{I}_T(\delta) \cup \{[\![t_s, t_e]\!]\}$
  - While $\delta\,|\mathcal{I}_T(\delta)| > \alpha$: → check $\widehat{FWER}$
    - Decrease δ

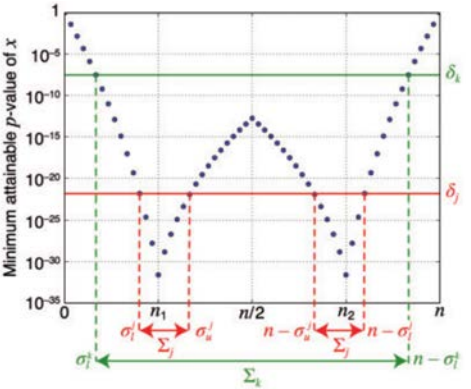


1. Routine `get_significant_regions`: interval processing

- $\delta \leftarrow 1$, $\mathcal{I}_T(\delta) \leftarrow \{\}$

For all $[\![t_s, t_e]\!] \in \mathcal{R}_{cand}$, in increasing order of starting position t_s and then length $t_e - t_s$:

- Compute $x_{[\![t_s,t_e]\!]}$
- If $\Phi(x_{[\![t_s,t_e]\!]}) \leq \delta$: → Tarone's testability criterion
  - $\mathcal{I}_T(\delta) \leftarrow \mathcal{I}_T(\delta) \cup \{[\![t_s, t_e]\!]\}$
  - While $\delta\,|\mathcal{I}_T(\delta)| > \alpha$: → check $\widehat{FWER}$
    - Decrease δ
    - Remove newly untestable intervals from $\mathcal{I}_T(\delta)$

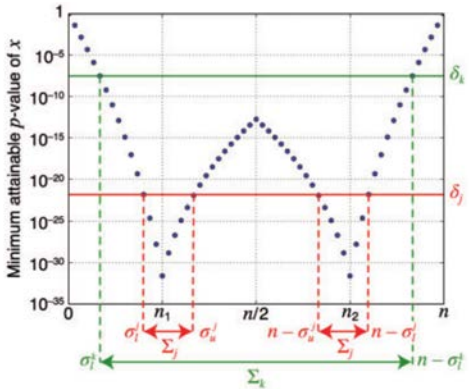


1. Routine `get_significant_regions`: interval processing

- $\delta \leftarrow 1$, $\mathcal{I}_T(\delta) \leftarrow \{\}$

For all $[\![t_s, t_e]\!] \in \mathcal{R}_{cand}$, in increasing order of starting position t_s and then length $t_e - t_s$:

- Compute $x_{[\![t_s,t_e]\!]}$
- If $\Phi(x_{[\![t_s,t_e]\!]}) \leq \delta$: $\rightarrow$ Tarone's testability criterion
  - $\mathcal{I}_T(\delta) \leftarrow \mathcal{I}_T(\delta) \cup \{[\![t_s, t_e]\!]\}$
  - While $\delta\,|\mathcal{I}_T(\delta)| > \alpha$: $\rightarrow$ check $\widehat{FWER}$
    - Decrease δ
    - Remove newly untestable intervals from $\mathcal{I}_T(\delta)$
- If **pruning_condition**$(x_{[\![t_s,t_e]\!]})$ then: $\Rightarrow$ depends on the test statistic

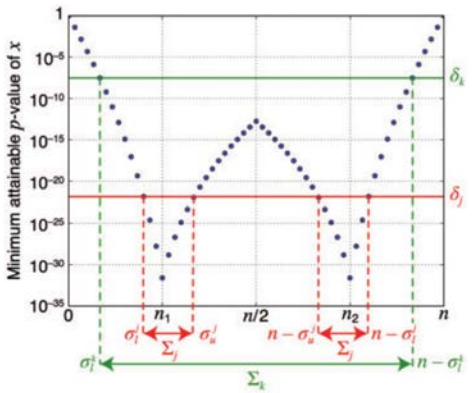


1. Routine `get_significant_regions`: interval processing

- $\delta \leftarrow 1$, $\mathcal{I}_T(\delta) \leftarrow \{\}$

For all $[\![t_s, t_e]\!] \in \mathcal{R}_{cand}$, in increasing order of starting position t_s and then length $t_e - t_s$:

- Compute $x_{[\![t_s,t_e]\!]}$
- If $\Phi(x_{[\![t_s,t_e]\!]}) \leq \delta$: $\rightarrow$ Tarone's testability criterion
  - $\mathcal{I}_T(\delta) \leftarrow \mathcal{I}_T(\delta) \cup \{[\![t_s, t_e]\!]\}$
  - While $\delta\,|\mathcal{I}_T(\delta)| > \alpha$: $\rightarrow$ check $\widehat{FWER}$
    - Decrease δ
    - Remove newly untestable intervals from $\mathcal{I}_T(\delta)$
- If **pruning_condition**$(x_{[\![t_s,t_e]\!]})$ then: $\Rightarrow$ depends on the test statistic
  - Prune all intervals $[\![t'_s, t'_e]\!] \supset [\![t_s, t_e]\!]$ from $\mathcal{R}_{cand}$



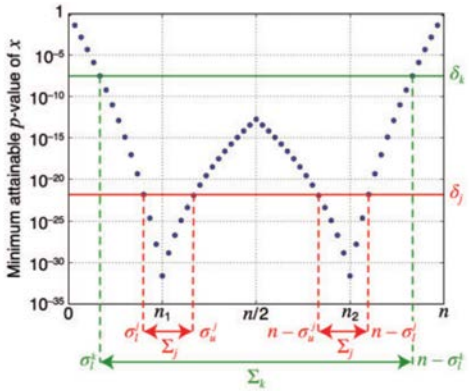
1. Routine `get_significant_regions`: interval processing

- $\delta \leftarrow 1$, $\mathcal{I}_T(\delta) \leftarrow \{\}$

For all $[\![t_s, t_e]\!] \in \mathcal{R}_{cand}$, in increasing order of starting position t_s and then length $t_e - t_s$:

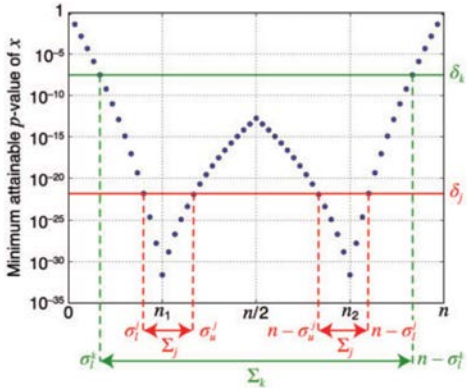
- Compute $x_{[\![t_s,t_e]\!]}$
- If $\Phi(x_{[\![t_s,t_e]\!]}) \leq \delta$: $\rightarrow$ Tarone's testability criterion
  - $\mathcal{I}_T(\delta) \leftarrow \mathcal{I}_T(\delta) \cup \{[\![t_s, t_e]\!]\}$
  - While $\delta\,|\mathcal{I}_T(\delta)| > \alpha$: $\rightarrow$ check $\widehat{FWER}$
    - Decrease δ
    - Remove newly untestable intervals from $\mathcal{I}_T(\delta)$
- If **pruning_condition**$(x_{[\![t_s,t_e]\!]})$ then: $\Rightarrow$ depends on the test statistic
  - Prune all intervals $[\![t'_s, t'_e]\!] \supset [\![t_s, t_e]\!]$ from $\mathcal{R}_{cand}$

Return: δ_{tar} and $\mathcal{R}_T(\delta_{tar})$

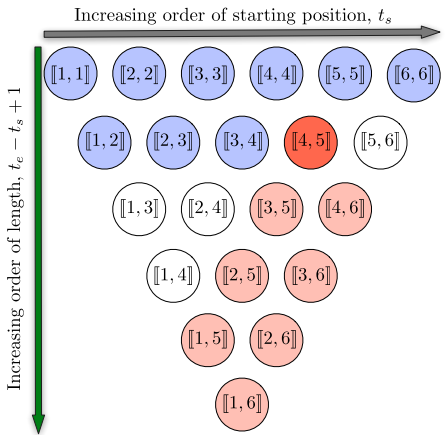


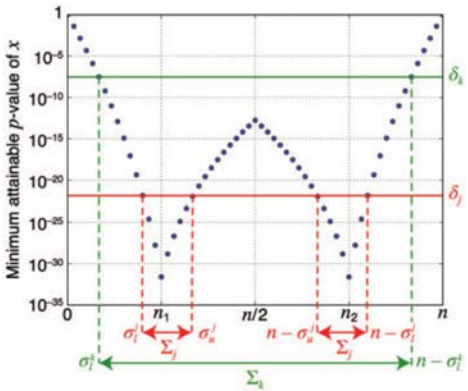
Pruning conditions for FAIS

- FAIS: χ^2, Fisher exact test.
- The minimum attainable p-value is monotonically increasing as x increases in $\mathcal{R}_{cor} = [\max(n_1, n_2), n]$.
- The pruning condition is straight forward:
  $x_{[\![t_s,t_e]\!]} \geq \max(n_1, n_2)$ and
  $\Phi(x_{[\![t_s,t_e]\!]}) > \delta$



1. Routine `get_significant_regions`: interval pruning

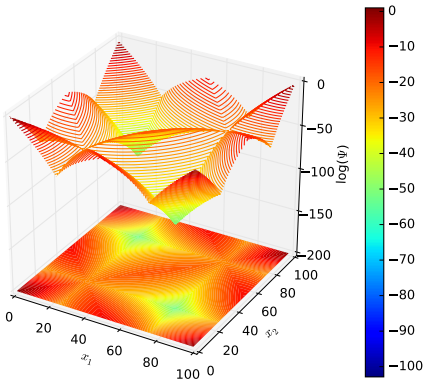




Pruning conditions for `FastCMH`

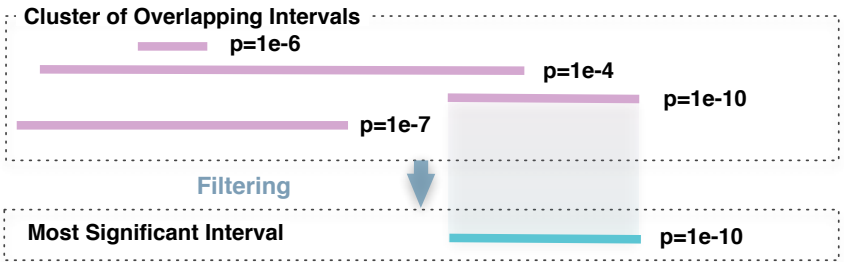
- `FastCMH`: CMH-test.
- The minimum attainable p-value $\Phi(x_{[\![t_s,t_e]\!]})$ is not monotonic for $\mathbf{x}_{[\![t_s,t_e]\!]} \in \mathcal{R}_{cor} = [\![\max(n_{1,h}, n_{2,h}), n]\!]_{h=1}^{k}$.

  - We compute a monotonic lower bound to the p-value surface in the prunable search space $\mathcal{R}_{cor}$.
  - Runtime scales as $O(k \log(k))$



2. Routine `filter_overlapping_regions`

- Selection of the interval with the smallest *p*-value



- Advantage: Corrects for redundancy, LD partly;
- Limitation: Dependent statistical tests:
  - Solution: Permutation testing, implemented with FAIS-WY but not with FastCMH.

Table of Contents

1 Genomic interactions problem statement

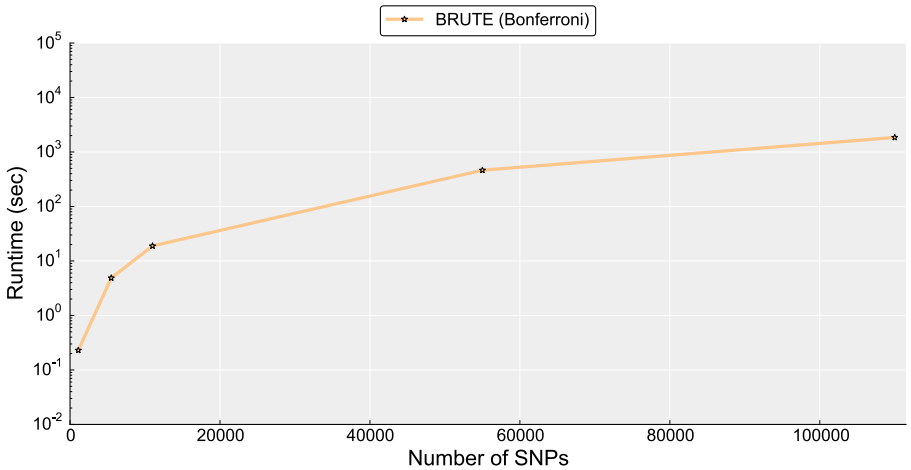
2 Statistical testing and correction for confounders

3 Methods: Fast Automatic Interval Search (FAIS) and FastCMH algorithms

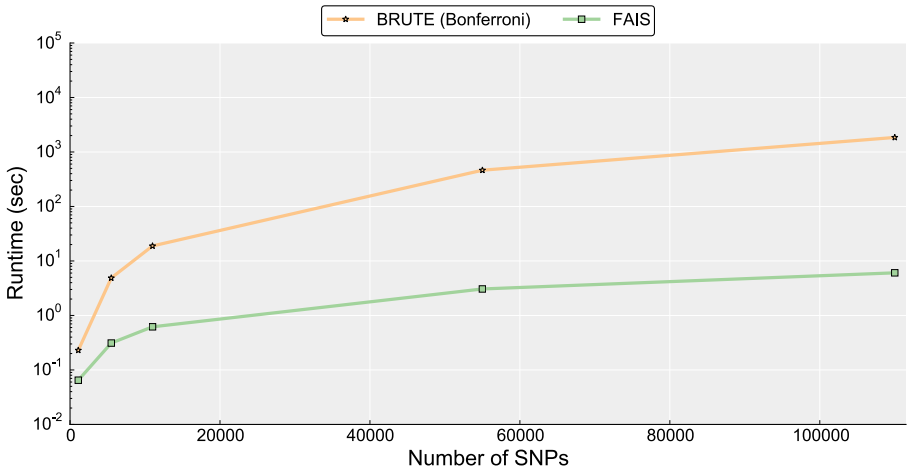
4 Results on plant and human datasets

5 Summary and outlook

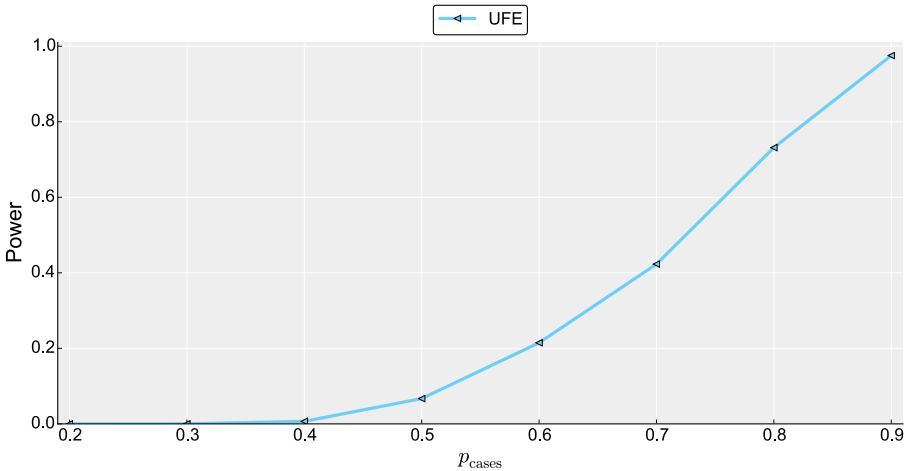
FAIS: runtime simulation



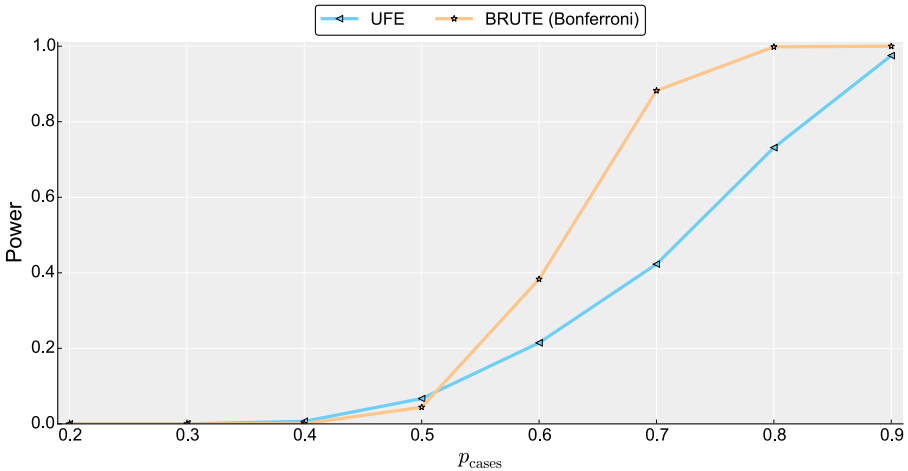
FAIS: runtime simulation



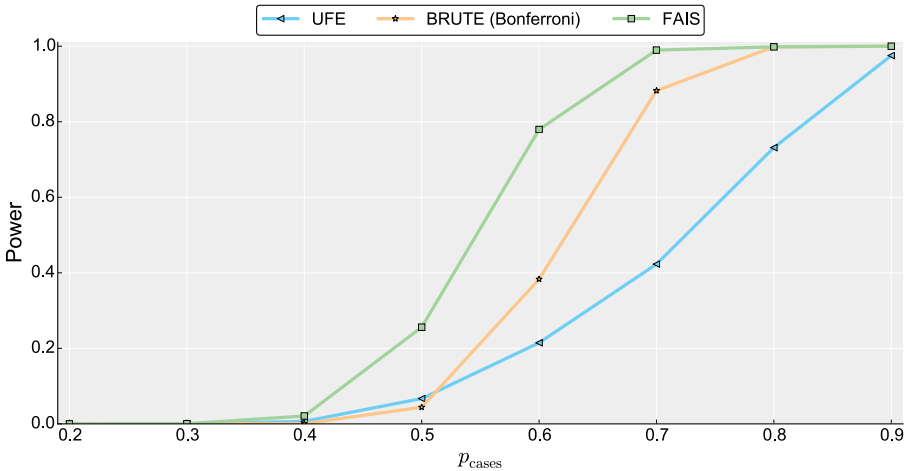
FAIS: power simulation



FAIS: power simulation



FAIS: power simulation



FAIS: genetic heterogeneity detection in *Arabidopsis thaliana*

Dataset (Atwell 2010)

- 21 defense and development binary phenotypes

FAIS: genetic heterogeneity detection in *Arabidopsis thaliana*

Dataset (Atwell 2010)

- 21 defense and development binary phenotypes
- Sample sizes between 76 and 177

FAIS: genetic heterogeneity detection in *Arabidopsis thaliana*

Dataset (Atwell 2010)

- 21 defense and development binary phenotypes
- Sample sizes between 76 and 177
- 214,051 homozygous SNPs (inbred)

FAIS: genetic heterogeneity detection in *Arabidopsis thaliana*

Dataset (Atwell 2010)

- 21 defense and development binary phenotypes
- Sample sizes between 76 and 177
- 214,051 homozygous SNPs (inbred)
- Compare findings of FAIS-WY with univariate methods: Fisher's Exact Test (UFE), Linear Mixed Model (LMM).

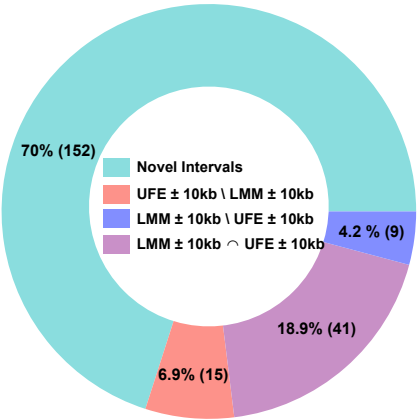
FAIS: genetic heterogeneity detection in *Arabidopsis thaliana*

Dataset (Atwell 2010)

- 21 defense and development binary phenotypes
- Sample sizes between 76 and 177
- 214,051 homozygous SNPs (inbred)
- Compare findings of FAIS-WY with univariate methods: Fisher's Exact Test (UFE), Linear Mixed Model (LMM).

Sources for intervals found

- True genetic heterogeneity



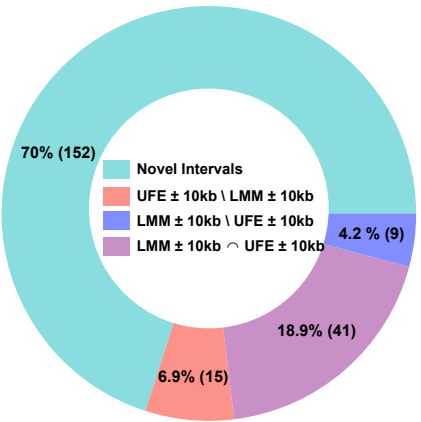
FAIS: genetic heterogeneity detection in *Arabidopsis thaliana*

Dataset (Atwell 2010)

- 21 defense and development binary phenotypes
- Sample sizes between 76 and 177
- 214,051 homozygous SNPs (inbred)
- Compare findings of FAIS-WY with univariate methods: Fisher's Exact Test (UFE), Linear Mixed Model (LMM).

Sources for intervals found

- True genetic heterogeneity
- Linkage to causal SNPs



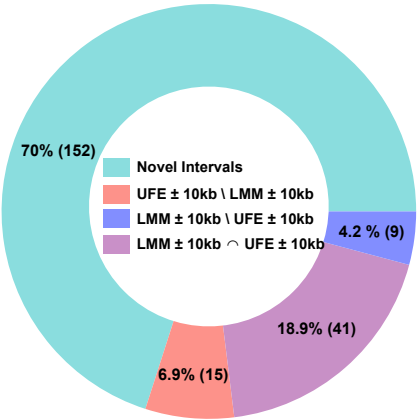
FAIS: genetic heterogeneity detection in *Arabidopsis thaliana*

Dataset (Atwell 2010)

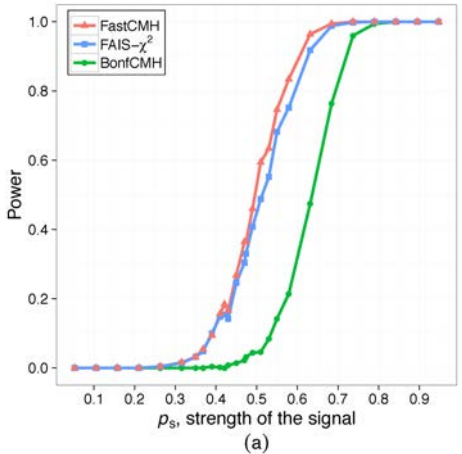
- 21 defense and development binary phenotypes
- Sample sizes between 76 and 177
- 214,051 homozygous SNPs (inbred)
- Compare findings of FAIS-WY with univariate methods: Fisher's Exact Test (UFE), Linear Mixed Model (LMM).

Sources for intervals found

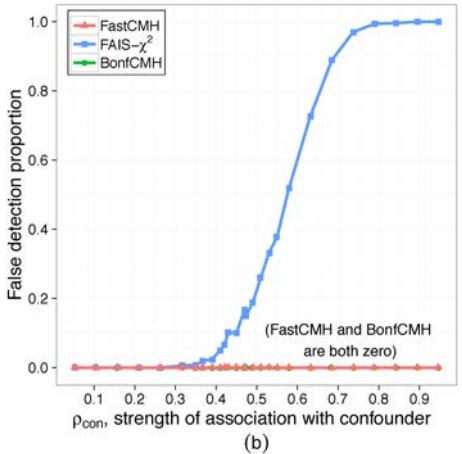
- True genetic heterogeneity
- Linkage to causal SNPs
- Structural variation in the region



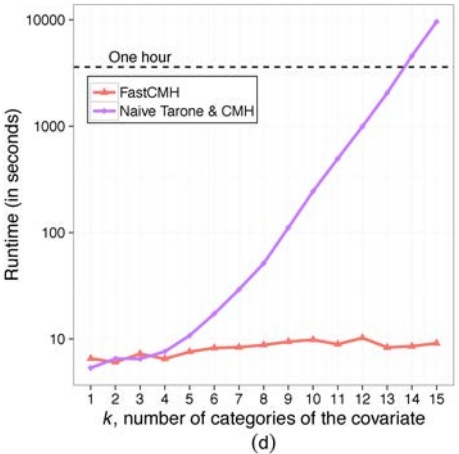
FastCMH: simulations show a high power, low false detection proportion and high-speed detection



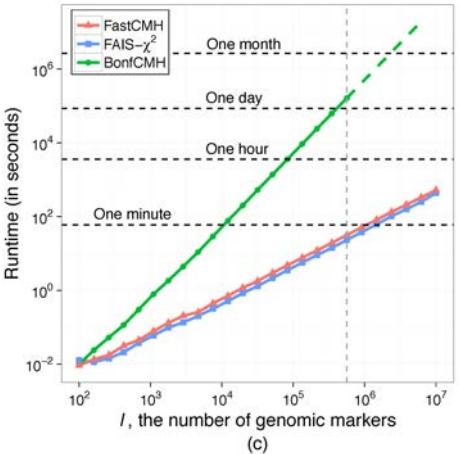
FastCMH: simulations show a high power, low false detection proportion and high-speed detection



FastCMH: simulations show a high power, low false detection proportion and high-speed detection



FastCMH: simulations show a high power, low false detection proportion and high-speed detection



Datasets

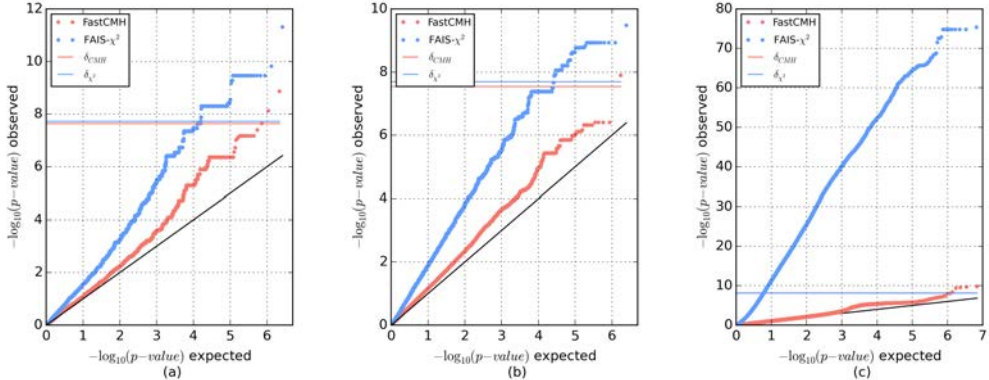
COPD case/control study

- Binary phenotype: COPD cases vs. controls.
- 8,011 samples, 3,633 are cases and 4,378 are controls.
- Approximately 615,906 SNPs, binarized using a dominant encoding, to study the risk factor of any minor-allele
- 2,665 African-American and 5,346 non-Hispanic whites.

Arabidopsis thaliana dataset

- 5 binary phentoypes
- 2-5 geographical origins (Eigenstrat, Price 2006).

FastCMH: correcting for confounders in COPD and Arabidopsis thaliana case/control studies



QQplots for: (a) LES phenotype, (b) LY phenotype, (c) COPD study

FastCMH reports novel genomic regions

COPD case/control study

- Each of the 3 reported regions overlaps with a gene in: CHRNA5-CHRNA3-CHRNB4, a nicotine receptor (nAChR).
- None of the SNPs alone shows an association with COPD.
- Separated studies (AA and NHW alone) do not find those three significant hits.

A. thaliana studies

- `FastCMH` reports 33 genomic regions and `FAIS-`χ^2 reports 81
- Decrease of the genomic inflation factor.
- 45% of the total number of reported SNPs are not into genes.

Burden tests: a genome-scale approach to study high-order interactions

Burden tests collapse SNPs into genes and test for the association of the entire region with the phenotypic trait (Lee 2014).
We used:

- a logistic regression model
- two encodings: (1) OR combination of SNPs inside the genes and (2) minor-allele counts.
- three covariate corrections: (1) principal components of the kinship matrix (only for *Arabidopsis th.*), (2) $k-1$ dummy variables for k classes and (3) CMH-test.

Limitations:Test a small subset of all possible regions in a genome by discriminating them on their function.

FastCMH finds genomic regions that can not be found by burden tests

COPD case/control study

- None of the three genes in CHRNA5-CHRNA3-CHRNB4 are reported by the burden tests.
- `FastCMH`'s advantage: significant regions do not span the entire genes.

Arabidopsis thaliana studies

- High variability among the hits
- Low to medium confounder correction.

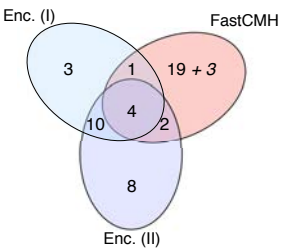


Table of Contents

1 Genomic interactions problem statement

2 Statistical testing and correction for confounders

3 Methods: Fast Automatic Interval Search (FAIS) and FastCMH algorithms

4 Results on plant and human datasets

5 Summary and outlook

Summary

- `FastCMH` enables to discover *all* candidate genomic regions of genetic heterogeneity, efficiently, with high power and while correcting for confounders.
- Principled approach for meta-analysis.
- Code available:
  https://www.bsse.ethz.ch/mlcb/research/bioinformatics-and-computational-biology.html

Outlook

- Implementing the permutation testing version to correct for dependency between the tests.
- Extending `FastCMH` to heterozygous genotypes and continuous phenotypes.
- Including long-range interactions by enabling all combinations of SNPs (submitted work).
- Adding biological prior:
  - Differentiating between SNPs that prevent or cause a disease.
  - Detecting significant gene clusters in pathways (Part III).

References I

- Agrawal, R., Imieliński, T., and Swami, A. (1993).
  Mining association rules between sets of items in large databases.
  SIGMOD Rec., 22(2):207–216.
- Arora, A., Sachan, M., and Bhattacharya, A. (2014).
  Mining statistically significant connected subgraphs in vertex labeled graphs.
  In *SIGMOD*, pages 1003–1014.
- Atwell, S., Huang, Y. S., Vilhjálmsson, B. J., Willems, G., Horton, M., Li, Y., Meng, D., Platt, A., Tarone, A. M., Hu, T. T., et al. (2010).
  Genome-wide association study of 107 phenotypes in *Arabidopsis Thaliana* inbred lines.
  Nature, 465(7298):627–631.
- Bay, S. D. and Pazzani, M. (2001).
  Detecting group differences: Mining contrast sets.
  Data Mining and Knowledge Discovery, 5(3):213–246.
- Benjamini, Y. and Hochberg, Y. (1995).
  Controlling the false discovery rate: A practical and powerful approach to multiple testing.
  Journal of the Royal Statistical Society. Series B, 57(1):289–300.
- Bonferroni, C. E. (1936).
  Teoria statistica delle classi e calcolo delle probabilità.
  Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze, 8:3–62.
- Dudoit, S. and van der Laan, M. J. (2007).
  Multiple Testing Procedures with Applications to Genomics (Springer Series in Statistics).
  Springer.

References II

- Epstein, M. P., Duncan, R., Jiang, Y., Conneely, K. N., Allen, A. S., and Satten, G. A. (2012).
  A permutation procedure to correct for confounders in case-control studies, including tests of rare variation.
  Am J Hum Genet, 91:215–223.
- Fan, W., Zhang, K., Cheng, H., Gao, J., Yan, X., Han, J., Yu, P. S., and Verscheure, O. (2008).
  Direct mining of discriminative and essential frequent patterns via model-based search tree.
  In *SIGKDD*, pages 230–238.
- Fischer, J., Heun, V., and Kramer, S. (2005).
  Fast frequent string mining using suffix arrays.
  In *IEEE ICDM*, pages 609–612.
- Fisher, R. A. (1922).
  On the Interpretation of χ^2 from Contingency Tables, and the Calculation of P.
  Journal of the Royal Statistical Society, 85(1):87–94.
- Lee, B.-K., Bhinge, A. A., Battenhouse, A., McDaniell, R. M., Liu, Z., Song, L., Ni, Y., Birney, E., Lieb, J. D., Furey, T. S., Crawford, G. E., and Iyer, V. R. (2012).
  Cell-type specific and combinatorial usage of diverse transcription factors revealed by genome-wide binding studies in multiple human cells.
  Genome Research, 22(1):9–24.
- Lee, S., Abecasis, G. R., Boehnke, M., and Lin, X. (2014).
  Rare-variant association analysis: Study designs and statistical tests.
  The American Journal of Human Genetics, 95(1):5 – 23.
- Lee, S. and De Raedt, L. (2005).
  An efficient algorithm for mining string databases under constraints.
  In *Knowledge Discovery in Inductive Databases*, volume 3377 of *LNCS*, pages 108–129.

References III

- Li, G., Semerci, M., Yener, B., and Zaki, M. J. (2012).
  Effective graph classification based on topological and label attributes.
  Statistical Analysis and Data Mining, 5(4):265–283.
- Llinares-López, F., Grimm, D., Bodenham, D. A., Gieraths, U., Sugiyama, M., Rowan, B., and Borgwardt, K. M. (2015).
  Genome-wide detection of intervals of genetic heterogeneity associated with complex traits.
  Bioinformatics, 31(12):240–249.
- Minato, S., Uno, T., Tsuda, K., Terada, A., and Sese, J. (2014).
  A fast method of statistical assessment for combinatorial hypotheses based on frequent itemset enumeration.
  In *ECMLPKDD*, volume 8725 of *LNCS*, pages 422–436.
- Novak, P. K., Lavrač, N., and Webb, G. I. (2009).
  Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining.
  JMLR, 10:377–403.
- Orenstein, Y. and Shamir, R. (2013).
  Design of shortest double-stranded DNA sequences covering all *k*-mers with applications to protein-binding microarrays and synthetic enhancers.
  Bioinformatics, 29(13):i71–i79.
- Pearson, K. (1900).
  X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling.
  Philosophical Magazine Series 6, 50:157–175.
- Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. (2011).
  Weisfeiler-Lehman graph kernels.
  JMLR, 12:2359–2561.

References IV

- Sugiyama, M., Llinares López, F., Kasenburg, N., and Borgwardt, K. M. (2015).
  Mining significant subgraphs with multiple testing correction.
  In *SIAM SDM*.
  (accepted, preliminary version is available at `http://arxiv.org/abs/1407.0316`).
- Tarone, R. E. (1990).
  A modified bonferroni method for discrete data.
  Biometrics, 46(2):515–522.
- Terada, A., Okada-Hatakeyama, M., Tsuda, K., and Sese, J. (2013a).
  Statistical significance of combinatorial regulations.
  Proceedings of the National Academy of Sciences, 110(32):12996–13001.
- Terada, A., Tsuda, K., and Sese, J. (2013b).
  Fast westfall-young permutation procedure for combinatorial regulation discovery.
  In *IEEE International Conference on Bioinformatics and Biomedicine*, pages 153–158.
- Webb, G. I. (2007).
  Discovering significant patterns.
  Machine Learning, 68(1):1–33.
- Westfall, P. H. and Young, S. S. (1993).
  Resampling-based multiple testing.
  Statistics in Medicine, 13(10):1084–1086.
- Zimmermann, A., Bringmann, B., and Rückert, U. (2010).
  Fast, effective molecular feature mining by local optimization.
  In *ECMLPKDD*, volume 6323 of *LNCS*, pages 563–578.

Part III
Significant Subgraph Search in Protein-Protein Interaction Networks

By Anja Gumpinger

Outline

1 Searching for significant subgraphs: motivation and problem statement

2 State of the art: dmGWAS

3 The Tarone method in significant subgraph search

4 Application to gene expression data

5 Summary and future work

Table of Contents

1 Searching for significant subgraphs: motivation and problem statement

2 State of the art: dmGWAS

3 The Tarone method in significant subgraph search

4 Application to gene expression data

5 Summary and future work

Motivation

Paradigm

- Univariate analysis of SNPs only account for small amount of total phenotypic variation [Manolio et al., 2009]
- Several variants, each with weak association to phenotype, orchestrate to manifest phenotype

Motivation

Paradigm

- Univariate analysis of SNPs only account for small amount of total phenotypic variation [Manolio et al., 2009]
- Several variants, each with weak association to phenotype, orchestrate to manifest phenotype

Idea

- Genes do not interact randomly with each other, but are organized in pathways
- Include biological prior knowledge into interaction search
- Use protein-protein interaction (PPI) networks
  - KEGG pathways [Kanehisa and Goto, 2000]
  - PINA [Cowley et al., 2011]

Problem statement

Initial setup:

- Dataset of n individuals that can be classified into two phenotypic groups:
  - n_1 cases
  - n_2 controls
- Protein-protein interaction network that will serve as biological prior knowledge

Problem statement

Initial setup:

- Dataset of n individuals that can be classified into two phenotypic groups:
  - n_1 cases
  - n_2 controls
- Protein-protein interaction network that will serve as biological prior knowledge

Problem statement: significant subgraph search

- Find subgraphs of genes within the PPI, such that the genotypes of the genes in the subgraphs are significantly associated with the phenotype
- Rigorous correction for multiply hypothesis testing by controlling the family wise error rate

Table of Contents

1 Searching for significant subgraphs: motivation and problem statement

2 State of the art: dmGWAS

3 The Tarone method in significant subgraph search

4 Application to gene expression data

5 Summary and future work

State of the art: dmGWAS

BIOINFORMATICS ORIGINAL PAPER
Vol. 27 no. 1 2011, pages 95–102
doi:10.1093/bioinformatics/btq615

Genetics and population analysis
Advance Access publication November 2, 2010

dmGWAS: dense module searching for genome-wide association studies in protein–protein interaction networks

Peilin Jia[1,2,†], Siyuan Zheng[1,3,†], Jirong Long[4], Wei Zheng[4,5] and Zhongming Zhao[1,2,3,5,6,*]

[1]Department of Biomedical Informatics, [2]Department of Psychiatry, [3]Functional Genomics Shared Resource, [4]Vanderbilt Epidemiology Center, [5]Vanderbilt-Ingram Cancer Center and [6]Bioinformatics Resource Center, Vanderbilt University, Nashville, TN 37232, USA

Associate Editor: Jeffrey Barrett

- Method [Jia et al., 2011] to identify subgraphs or genes for complex diseases
- Achieved by integrating the association signal from GWAS datasets into human protein-protein interaction networks

dmGWAS - Implementation: Input/Output

R implementation of dmGWAS available.

Input

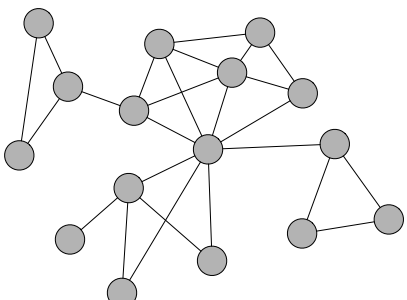
- Protein-protein interaction network
- P-values p_i for each gene in network
- User-specified parameters

Output

- List of subgraphs within the protein-protein interaction network, enriched with low p-value genes
- Subgraphs ranked by subgraph score

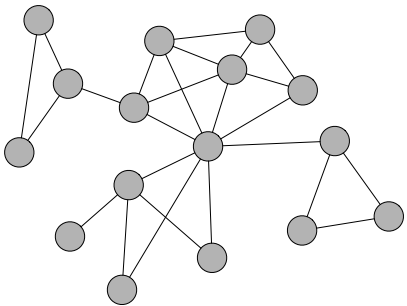
dmGWAS - Greedy search for subgraphs

1. Transformation of p-values, $z_i = \Phi^{-1}(1 - p_i)$



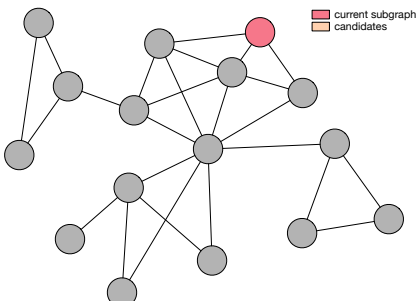
dmGWAS - Greedy search for subgraphs

1. Transformation of p-values, $z_i = \Phi^{-1}(1 - p_i)$
2. At each gene in the PPI network: start greedy search for subgraphs with high scores



dmGWAS - Greedy search for subgraphs

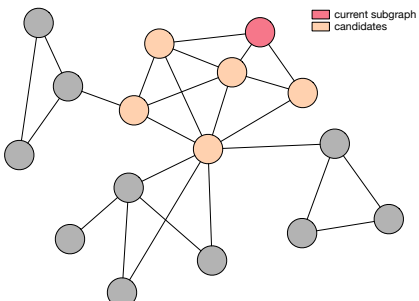
1. Transformation of p-values, $z_i = \Phi^{-1}(1 - p_i)$
2. At each gene in the PPI network: start greedy search for subgraphs with high scores



(i) Compute subgraph score $Z_{current} = \frac{\sum z_i}{\sqrt{k}}$

dmGWAS - Greedy search for subgraphs

1. Transformation of p-values, $z_i = \Phi^{-1}(1 - p_i)$
2. At each gene in the PPI network: start greedy search for subgraphs with high scores



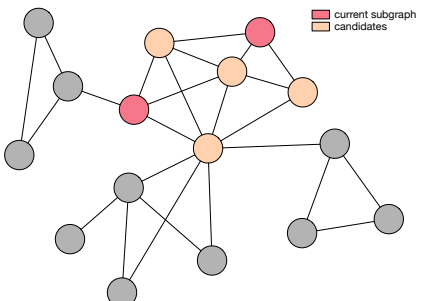
(i) Compute subgraph score $Z_{current} = \frac{\sum z_i}{\sqrt{k}}$

(ii) Find neighbors with distance smaller or equal to d (here $d = 2$)

(iii) For each neighbor: compute a tentative new subgraph score Z_{new}

dmGWAS - Greedy search for subgraphs

1. Transformation of p-values, $z_i = \Phi^{-1}(1 - p_i)$
2. At each gene in the PPI network: start greedy search for subgraphs with high scores



(i) Compute subgraph score $Z_{current} = \frac{\sum z_i}{\sqrt{k}}$

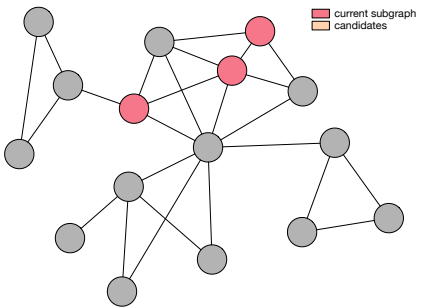
(ii) Find neighbors with distance smaller or equal to d (here $d = 2$)

(iii) For each neighbor: compute a tentative new subgraph score Z_{new}

(iv) Pick neighbor with maximal Z_{new}: if $Z_{new} \geq Z_{current}(1 + r)$: add node (plus nodes in shortest path) to subgraph

dmGWAS - Greedy search for subgraphs

1. Transformation of p-values, $z_i = \Phi^{-1}(1 - p_i)$
2. At each gene in the PPI network: start greedy search for subgraphs with high scores



(i) Compute subgraph score $Z_{current} = \frac{\sum z_i}{\sqrt{k}}$

(ii) Find neighbors with distance smaller or equal to d (here $d = 2$)

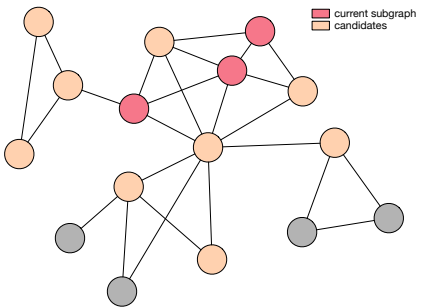
(iii) For each neighbor: compute a tentative new subgraph score Z_{new}

(iv) Pick neighbor with maximal Z_{new}: if $Z_{new} \geq Z_{current}(1 + r)$: add node (plus nodes in shortest path) to subgraph

(v) Repeat (i) - (iv), until $Z_{new} \ngeq Z_{current}(1 + r)$

dmGWAS - Greedy search for subgraphs

1. Transformation of p-values, $z_i = \Phi^{-1}(1 - p_i)$
2. At each gene in the PPI network: start greedy search for subgraphs with high scores



(i) Compute subgraph score $Z_{current} = \frac{\sum z_i}{\sqrt{k}}$

(ii) Find neighbors with distance smaller or equal to d (here $d = 2$)

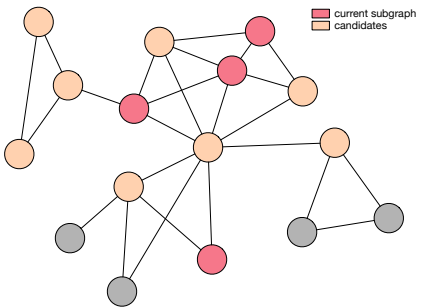
(iii) For each neighbor: compute a tentative new subgraph score Z_{new}

(iv) Pick neighbor with maximal Z_{new}: if $Z_{new} \geq Z_{current}(1 + r)$: add node (plus nodes in shortest path) to subgraph

(v) Repeat (i) - (iv), until $Z_{new} \ngeq Z_{current}(1 + r)$

dmGWAS - Greedy search for subgraphs

1. Transformation of p-values, $z_i = \Phi^{-1}(1 - p_i)$
2. At each gene in the PPI network: start greedy search for subgraphs with high scores



(i) Compute subgraph score $Z_{current} = \frac{\sum z_i}{\sqrt{k}}$

(ii) Find neighbors with distance smaller or equal to d (here $d = 2$)

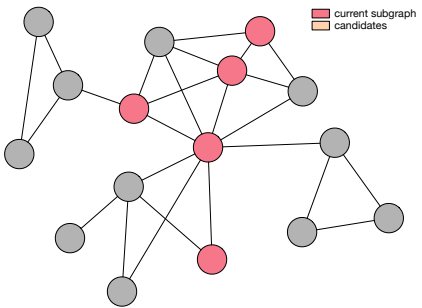
(iii) For each neighbor: compute a tentative new subgraph score Z_{new}

(iv) Pick neighbor with maximal Z_{new}: if $Z_{new} \geq Z_{current}(1 + r)$: add node (plus nodes in shortest path) to subgraph

(v) Repeat (i) - (iv), until $Z_{new} \ngeq Z_{current}(1 + r)$

dmGWAS - Greedy search for subgraphs

1. Transformation of p-values, $z_i = \Phi^{-1}(1 - p_i)$
2. At each gene in the PPI network: start greedy search for subgraphs with high scores



(i) Compute subgraph score $Z_{current} = \frac{\sum z_i}{\sqrt{k}}$

(ii) Find neighbors with distance smaller or equal to d (here $d = 2$)

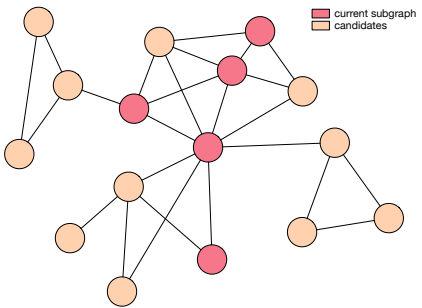
(iii) For each neighbor: compute a tentative new subgraph score Z_{new}

(iv) Pick neighbor with maximal Z_{new}: if $Z_{new} \geq Z_{current}(1 + r)$: add node (plus nodes in shortest path) to subgraph

(v) Repeat (i) - (iv), until $Z_{new} \ngeq Z_{current}(1 + r)$

dmGWAS - Greedy search for subgraphs

1. Transformation of p-values, $z_i = \Phi^{-1}(1 - p_i)$
2. At each gene in the PPI network: start greedy search for subgraphs with high scores



(i) Compute subgraph score $Z_{current} = \frac{\sum z_i}{\sqrt{k}}$

(ii) Find neighbors with distance smaller or equal to d (here $d = 2$)

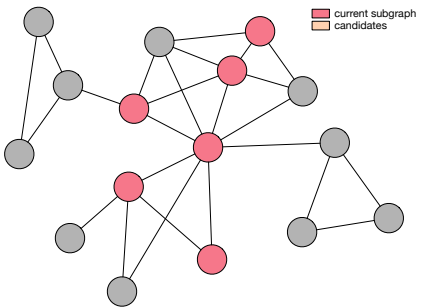
(iii) For each neighbor: compute a tentative new subgraph score Z_{new}

(iv) Pick neighbor with maximal Z_{new}: if $Z_{new} \geq Z_{current}(1 + r)$: add node (plus nodes in shortest path) to subgraph

(v) Repeat (i) - (iv), until $Z_{new} \ngeq Z_{current}(1 + r)$

dmGWAS - Greedy search for subgraphs

1. Transformation of p-values, $z_i = \Phi^{-1}(1 - p_i)$
2. At each gene in the PPI network: start greedy search for subgraphs with high scores



(i) Compute subgraph score $Z_{current} = \frac{\sum z_i}{\sqrt{k}}$

(ii) Find neighbors with distance smaller or equal to d (here $d = 2$)

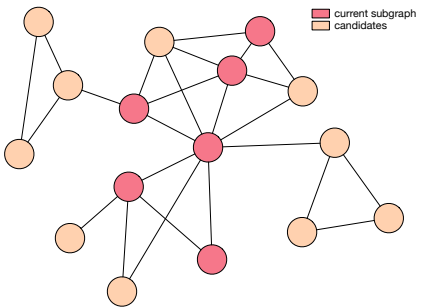
(iii) For each neighbor: compute a tentative new subgraph score Z_{new}

(iv) Pick neighbor with maximal Z_{new}: if $Z_{new} \geq Z_{current}(1 + r)$: add node (plus nodes in shortest path) to subgraph

(v) Repeat (i) - (iv), until $Z_{new} \ngeq Z_{current}(1 + r)$

dmGWAS - Greedy search for subgraphs

1. Transformation of p-values, $z_i = \Phi^{-1}(1 - p_i)$
2. At each gene in the PPI network: start greedy search for subgraphs with high scores



(i) Compute subgraph score $Z_{current} = \frac{\sum z_i}{\sqrt{k}}$

(ii) Find neighbors with distance smaller or equal to d (here $d = 2$)

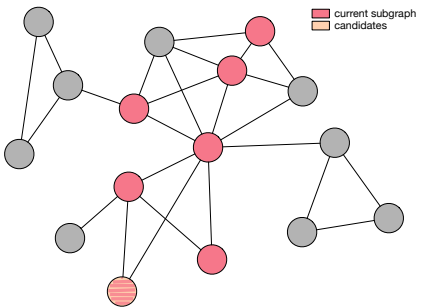
(iii) For each neighbor: compute a tentative new subgraph score Z_{new}

(iv) Pick neighbor with maximal Z_{new}: if $Z_{new} \geq Z_{current}(1 + r)$: add node (plus nodes in shortest path) to subgraph

(v) Repeat (i) - (iv), until $Z_{new} \ngeq Z_{current}(1 + r)$

dmGWAS - Greedy search for subgraphs

1. Transformation of p-values, $z_i = \Phi^{-1}(1 - p_i)$
2. At each gene in the PPI network: start greedy search for subgraphs with high scores



(i) Compute subgraph score $Z_{current} = \frac{\sum z_i}{\sqrt{k}}$

(ii) Find neighbors with distance smaller or equal to d (here $d = 2$)

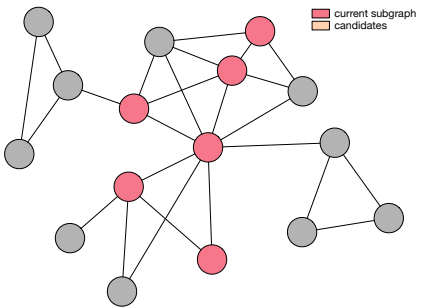
(iii) For each neighbor: compute a tentative new subgraph score Z_{new}

(iv) Pick neighbor with maximal Z_{new}: if $Z_{new} \geq Z_{current}(1 + r)$: add node (plus nodes in shortest path) to subgraph

(v) Repeat (i) - (iv), until $Z_{new} \ngeq Z_{current}(1 + r)$

dmGWAS - Greedy search for subgraphs

1. Transformation of p-values, $z_i = \Phi^{-1}(1 - p_i)$
2. At each gene in the PPI network: start greedy search for subgraphs with high scores



(i) Compute subgraph score $Z_{current} = \frac{\sum z_i}{\sqrt{k}}$

(ii) Find neighbors with distance smaller or equal to d (here $d = 2$)

(iii) For each neighbor: compute a tentative new subgraph score Z_{new}

(iv) Pick neighbor with maximal Z_{new}: if $Z_{new} \geq Z_{current}(1 + r)$: add node (plus nodes in shortest path) to subgraph

(v) Repeat (i) - (iv), until $Z_{new} \ngeq Z_{current}(1 + r)$

dmGWAS - Characteristics

- Greedy approach, based on computation of gene-wise p-values
- No p-value, but ranking of subgraphs with high scores
- Outcome (number of subgraphs, sizes of subgraphs) highly dependent on setting of parameters d and r
  - Suggestions by authors:
    - $d = 2$: median distance between any two genes in PPI < 5 [Chuang et al., 2007]
    - r: test various values and take reasonable one
- Postprocessing of output:
  - Upper bound on number of reported subgraphs: number of genes in PPI
  - Suggestion of authors: use top 10% ranked subgraphs
  - Analysis of induced sugraph of top-ranked subgraphs (consensus graph)

State of the art: other methods

DAPPLE: Disease Association Protein-Protein Link Evaluator [Rossin et al., 2011]

- Network of genes associated with phenotype are more densely connected than expected by pure chance
- To show this: random permutation of underlying network

State of the art: other methods

DAPPLE: Disease Association Protein-Protein Link Evaluator [Rossin et al., 2011]

- Network of genes associated with phenotype are more densely connected than expected by pure chance
- To show this: random permutation of underlying network

SConES: Selecting CONnected Explanatory SNPs [Azencott et al., 2013]

- Finding subgraphs in network with maximized association, connectivity and sparsity
- Can be written as optimization problem
- Code available at:
  https://www.bsse.ethz.ch/mlcb/research/bioinformatics-and-computational-biology/scones.html

Table of Contents

1 Searching for significant subgraphs: motivation and problem statement

2 State of the art: dmGWAS

3 The Tarone method in significant subgraph search

4 Application to gene expression data

5 Summary and future work

Short Revision: Tarone Trick

- Conducting high number of statistical significance test $\rightarrow$ multiple hypothesis testing problem

Short Revision: Tarone Trick

- Conducting high number of statistical significance test $\rightarrow$ multiple hypothesis testing problem
- Control family-wise error rate (FWER)

$$\text{FWER} = \Pr\left(\text{FP} \geq 1\right) \leq \alpha \tag{1}$$

- Need to find the maximum significance threshold δ such that Eq. 1 holds

Short Revision: Tarone Trick

- Conducting high number of statistical significance test $\rightarrow$ multiple hypothesis testing problem
- Control family-wise error rate (FWER)

$$\text{FWER} = \Pr\left(\text{FP} \geq 1\right) \leq \alpha \tag{1}$$

- Need to find the maximum significance threshold δ such that Eq. 1 holds
- Bonferroni correction: $\delta = \frac{\alpha}{\text{number of tests}}$

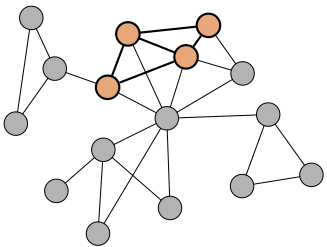
Short Revision: Tarone Trick

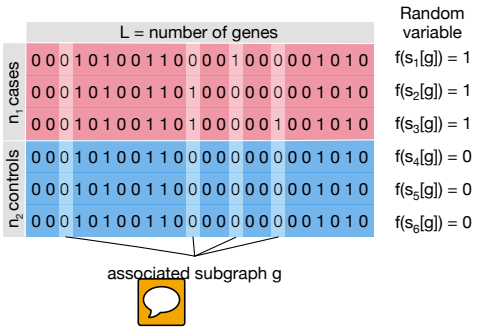
- Conducting high number of statistical significance test $\rightarrow$ multiple hypothesis testing problem
- Control family-wise error rate (FWER)

$$\text{FWER} = \Pr\left(\text{FP} \geq 1\right) \leq \alpha \tag{1}$$

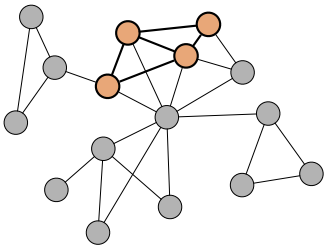
- Need to find the maximum significance threshold δ such that Eq. 1 holds
- Bonferroni correction: $\delta = \frac{\alpha}{\text{number of tests}}$
- Minimum attainable p-value: subgraphs that are not testable at a significance threshold δ cannot become false positives, thus no correction is required for those
- Tarone correction: $\delta = \frac{\alpha}{\text{number of testable subgraphs}}$

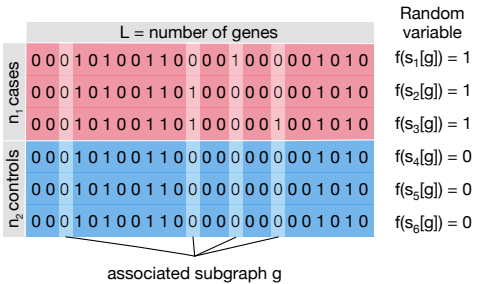
Tarone method for graphs: contingency tables





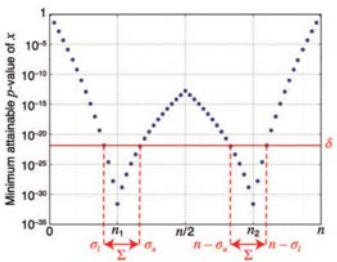
Tarone method for graphs: contingency tables





Variables	f(s[g]) = 1	f(s[g]) = 0	Row totals
y = case	α_g	$n_1 - \alpha_g$	n_1
y = control	$x_g - \alpha_g$	$n_2 - (x_g - \alpha_g)$	n_2
Col. totals	x_g	$N - x$	n

Tarone method: intervals vs. subgraphs



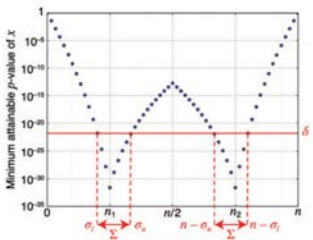
Interval search:

- Exploration of search space: subsequently combining intervals
- Pruning of search space: intervals containing non-testable intervals are non-testable

Subgraph search:

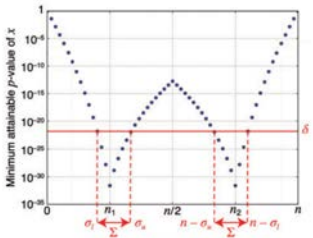
- Exploration of search space: growing subgraphs by subsequently adding nodes
- Pruning of search space: supergraphs of non-testable subgraph is non-testable

Network Tarone: growing and pruning graphs



- Subgraph g with x_g

Network Tarone: growing and pruning graphs

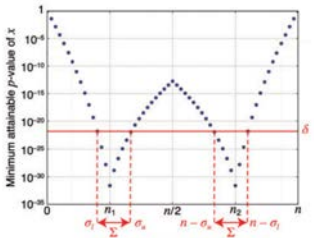


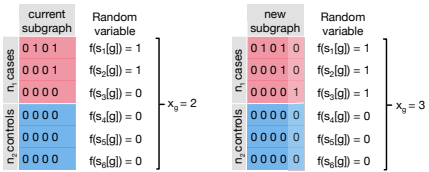
- Subgraph g with x_g
- **Monotonicity**: adding a new gene to a subgraph can only increase x_g

	current subgraph	Random variable	
n_1 cases	0 1 0 1	$f(s_1[g]) = 1$	$x_g = 2$
	0 0 0 1	$f(s_2[g]) = 1$	
	0 0 0 0	$f(s_3[g]) = 0$	
n_2 controls	0 0 0 0	$f(s_4[g]) = 0$	
	0 0 0 0	$f(s_5[g]) = 0$	
	0 0 0 0	$f(s_6[g]) = 0$	

	new subgraph	Random variable	
n_1 cases	0 1 0 1 0	$f(s_1[g]) = 1$	$x_g = 3$
	0 0 0 1 0	$f(s_2[g]) = 1$	
	0 0 0 0 1	$f(s_3[g]) = 1$	
n_2 controls	0 0 0 0 0	$f(s_4[g]) = 0$	
	0 0 0 0 0	$f(s_5[g]) = 0$	
	0 0 0 0 0	$f(s_6[g]) = 0$	

Network Tarone: growing and pruning graphs

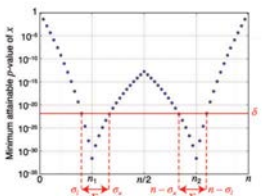




- Subgraph g with x_g
- **Monotonicity**: adding a new gene to a subgraph can only increase x_g
- Pruning: only subgraphs with $n - \sigma_l < x_g$ can be pruned from search space
  - If subgraph is non-testable: adding genes will always result in non-testable supergraph
  - Once subgraph is non-testable with $n - \sigma_l < x_g$: can stop growing graph

Network Tarone: adjusting the significance threshold

1. Compute minimum attainable p-value $\Psi(x_g)$ of current subgraph g with x_g

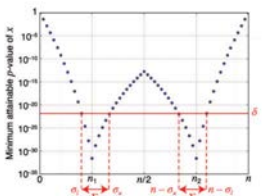


Network Tarone: adjusting the significance threshold

1. Compute minimum attainable p-value $\Psi(x_g)$ of current subgraph g with x_g
2. Subgraph is testable (i.e. $\Psi(x_g) \leq \delta$):
   1. Number of subgraphs that have to be corrected for increased
   2. Lower significance threshold δ s.t. FWER criterion is fulfilled

$$\delta * |\text{testable subgraphs}| \leq \alpha$$

   3. Add next gene to subgraph and return to step 1

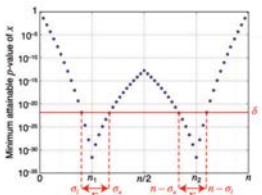


Network Tarone: adjusting the significance threshold

1. Compute minimum attainable p-value $\Psi(x_g)$ of current subgraph g with x_g
2. Subgraph is testable (i.e. $\Psi(x_g) \leq \delta$):
   1. Number of subgraphs that have to be corrected for increased
   2. Lower significance threshold δ s.t. FWER criterion is fulfilled

$$\delta * |\text{testable subgraphs}| \leq \alpha$$

   3. Add next gene to subgraph and return to step 1
3. Subgraph is non-testable (i.e. $\Psi(x_g) > \delta$):
   1. $x_g < n - \sigma_l$: Add next gene to subgraph an return to step 1
   2. $x_g > n - \sigma_l$: Stop growing subgraph



Steps in Network Tarone

1. Binarization of input data
   - GWAS data
   - Gene expression data

Steps in Network Tarone

1. Binarization of input data
   - GWAS data
   - Gene expression data
2. Application of Network Tarone: finding significant subgraphs in PPI network
   - Efficiently enumerating subgraphs in network
   - Accounting for multiple hypothesis testing

Steps in Network Tarone

1. Binarization of input data
   - GWAS data
   - Gene expression data
2. Application of Network Tarone: finding significant subgraphs in PPI network
   - Efficiently enumerating subgraphs in network
   - Accounting for multiple hypothesis testing
3. Evaluation of output
   - Reducing high number of often very similar significant subgraphs (clustering)
   - Reporting of results and biological interpretation

Binarization of data

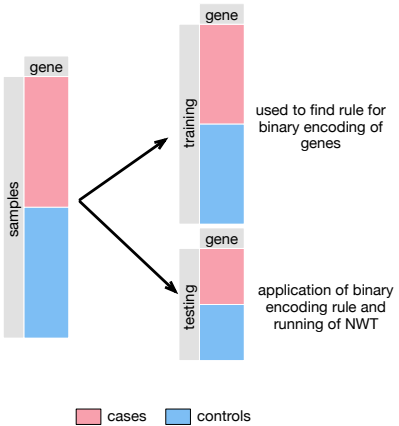
- Binarization depends on type of data used
  - Gene expression data: differential expression
  - GWAS data:
    - Approach based on allele frequencies
    - Machine learning approaches

Binarization of data

- Binarization depends on type of data used
  - Gene expression data: differential expression
  - GWAS data:
    - Approach based on allele frequencies
    - Machine learning approaches

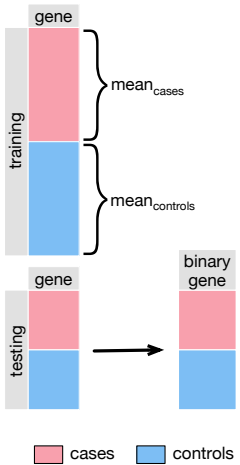
Idea: Risk gene encoding

- For one sample, binary status of a gene reflects whether sample can rather be assigned as case or control, based on only that gene
- Approaches require splitting of data into training and test set



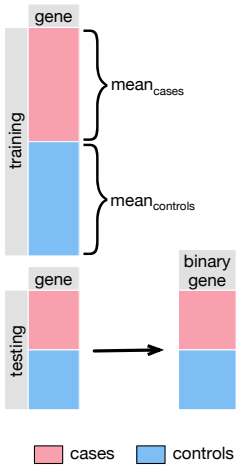
Binarization of gene-expression data

1. For each gene, compute the mean of cases mean_{cases} and controls $\text{mean}_{controls}$ in training set



Binarization of gene-expression data

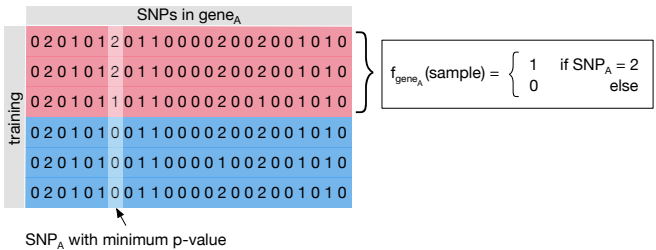
1. For each gene, compute the mean of cases mean_{cases} and controls $\text{mean}_{controls}$ in training set
2. Use data in test set to run Network Tarone:
   1. Binarize the data in test set by assigning the gene the label of the group with the smaller distance to the mean
   2. Use binary data as input for NWT



Binarization of GWAS data using allele counts

Building a classification rule

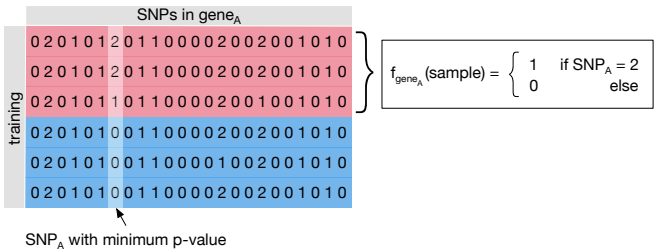
1. Represent gene by all SNPs in or near gene



Binarization of GWAS data using allele counts

Building a classification rule

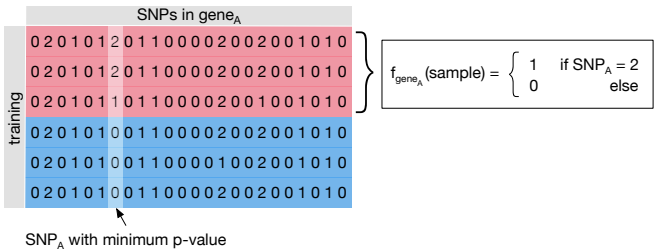
1. Represent gene by all SNPs in or near gene
2. Compute univariate p-values for each SNP in gene (using PLINK, FaSTLMM, ...)
3. Represent gene by SNP with lowest p-value



Binarization of GWAS data using allele counts

Building a classification rule

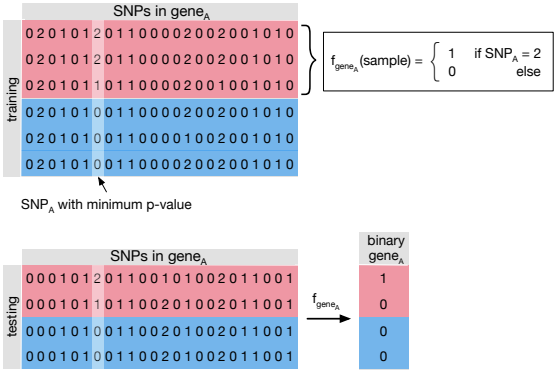
1. Represent gene by all SNPs in or near gene
2. Compute univariate p-values for each SNP in gene (using PLINK, FaSTLMM, ...)
3. Represent gene by SNP with lowest p-value
4. Determine most frequent genotype of selected SNP in cases and use this as classification rule



Binarization of GWAS data using allele counts

Classification of samples in test set

1. Represent gene by SNP with lowest p-value in training set
2. Apply classification rule found on training set to get binary representation of gene for each sample in test set



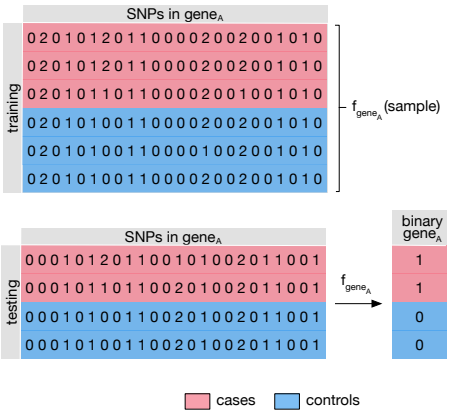
Binarization of GWAS data using machine learning (work in progress)

Building classification rule

1. Represent gene by all SNPs in or near gene
2. Determine a classification rule for each gene using all SNPs to predict risk encoding

Classification of samples in testing set

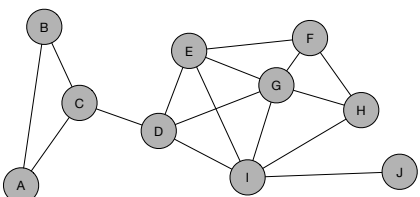
1. Represent gene by all SNPs in or near gene
2. Apply classification rule found on training set to get binary representation of gene for each sample in test set



Exploring the network

Growing the subgraphs

Need computationally efficient way to enumerate subgraphs in order to avoid visiting same subgraphs multiple times. Approach based on [Wernicke, 2006].



1. Indexing of nodes

Exploring the network

Growing the subgraphs

Need computationally efficient way to enumerate subgraphs in order to avoid visiting same subgraphs multiple times. Approach based on [Wernicke, 2006].

1. Indexing of nodes
2. Add one node at a time as seed gene

Exploring the network

Growing the subgraphs

Need computationally efficient way to enumerate subgraphs in order to avoid visiting same subgraphs multiple times. Approach based on [Wernicke, 2006].

1. Indexing of nodes
2. Add one node at a time as seed gene
3. Grow all possible subgraphs including seed gene

Exploring the network

Growing the subgraphs

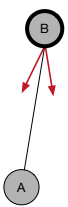
Need computationally efficient way to enumerate subgraphs in order to avoid visiting same subgraphs multiple times. Approach based on [Wernicke, 2006].

1. Indexing of nodes
2. Add one node at a time as seed gene
3. Grow all possible subgraphs including seed gene
4. For each newly grown subgraph check testability and process accordingly

Exploring the network

Growing the subgraphs

Need computationally efficient way to enumerate subgraphs in order to avoid visiting same subgraphs multiple times. Approach based on [Wernicke, 2006].

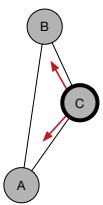


1. Indexing of nodes
2. Add one node at a time as seed gene
3. Grow all possible subgraphs including seed gene
4. For each newly grown subgraph check testability and process accordingly

Exploring the network

Growing the subgraphs

Need computationally efficient way to enumerate subgraphs in order to avoid visiting same subgraphs multiple times. Approach based on [Wernicke, 2006].

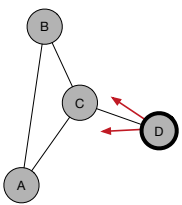


1. Indexing of nodes
2. Add one node at a time as seed gene
3. Grow all possible subgraphs including seed gene
4. For each newly grown subgraph check testability and process accordingly

Exploring the network

Growing the subgraphs

Need computationally efficient way to enumerate subgraphs in order to avoid visiting same subgraphs multiple times. Approach based on [Wernicke, 2006].

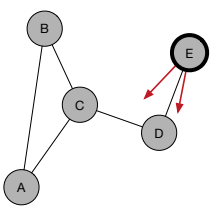


1. Indexing of nodes
2. Add one node at a time as seed gene
3. Grow all possible subgraphs including seed gene
4. For each newly grown subgraph check testability and process accordingly

Exploring the network

Growing the subgraphs

Need computationally efficient way to enumerate subgraphs in order to avoid visiting same subgraphs multiple times. Approach based on [Wernicke, 2006].



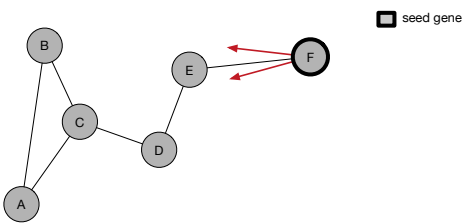
seed gene

1. Indexing of nodes
2. Add one node at a time as seed gene
3. Grow all possible subgraphs including seed gene
4. For each newly grown subgraph check testability and process accordingly

Exploring the network

Growing the subgraphs

Need computationally efficient way to enumerate subgraphs in order to avoid visiting same subgraphs multiple times. Approach based on [Wernicke, 2006].

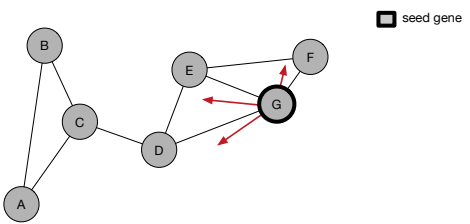


1. Indexing of nodes
2. Add one node at a time as seed gene
3. Grow all possible subgraphs including seed gene
4. For each newly grown subgraph check testability and process accordingly

Exploring the network

Growing the subgraphs

Need computationally efficient way to enumerate subgraphs in order to avoid visiting same subgraphs multiple times. Approach based on [Wernicke, 2006].

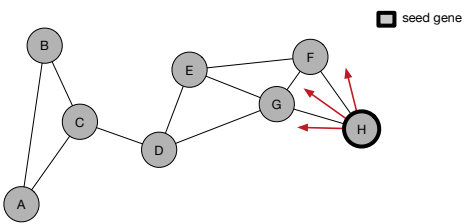


1. Indexing of nodes
2. Add one node at a time as seed gene
3. Grow all possible subgraphs including seed gene
4. For each newly grown subgraph check testability and process accordingly

Exploring the network

Growing the subgraphs

Need computationally efficient way to enumerate subgraphs in order to avoid visiting same subgraphs multiple times. Approach based on [Wernicke, 2006].



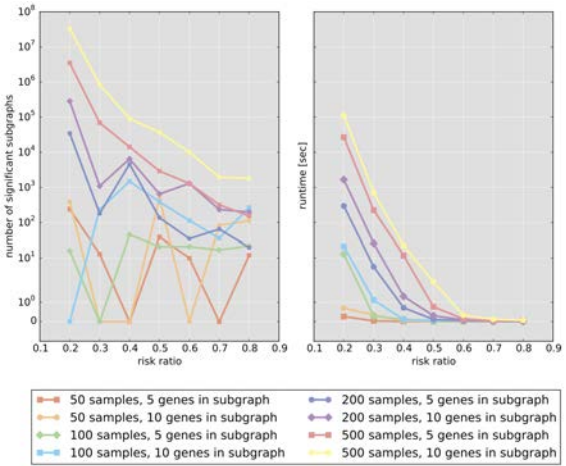
1. Indexing of nodes
2. Add one node at a time as seed gene
3. Grow all possible subgraphs including seed gene
4. For each newly grown subgraph check testability and process accordingly

Some results of NWT on artificial data

Artificial generation of binary data

- Generation of binary data with known ground truth (true significant subgraph)
- using R package 'bindata'
- Parameters to set:
  - Size of data set
  - Sizes of associated subgraphs
  - Risk ratio (ratio of 1/0 in binarized data)
  - Strength of association between subgraph and phenotype
- Size of underlying network: 68 nodes, 84 edges

Some results of NWT on artificial data



Postprocessing: clustering of significant subgraphs

Idea

Cluster all significant subgraphs, use subgraph with lowest p-value from each cluster as final output

Postprocessing: clustering of significant subgraphs

Idea

Cluster all significant subgraphs, use subgraph with lowest p-value from each cluster as final output

Structural clustering

- Subgraphs that overlap belong to the same cluster

Postprocessing: clustering of significant subgraphs

Idea

Cluster all significant subgraphs, use subgraph with lowest p-value from each cluster as final output

Structural clustering

- Subgraphs that overlap belong to the same cluster

Functional clustering (work in progress)

- Cluster significant subgraphs by their encoding
- Subgraphs with similar effects belong to the same cluster

Postprocessing: clustering of significant subgraphs

Idea

Cluster all significant subgraphs, use subgraph with lowest p-value from each cluster as final output

Structural clustering

- Subgraphs that overlap belong to the same cluster

Functional clustering (work in progress)

- Cluster significant subgraphs by their encoding
- Subgraphs with similar effects belong to the same cluster

DBSCAN clustering (work in progress)

- Create graph of subgraphs, where each subgraph corresponds to node, edge weighted by Jaccard-index, correlation, ...

Table of Contents

1 Searching for significant subgraphs: motivation and problem statement

2 State of the art: dmGWAS

3 The Tarone method in significant subgraph search

4 Application to gene expression data

5 Summary and future work

Application to breast cancer mRNA profiles

Dataset:
mRNA expression profiling obtained from a study of breast cancer patients [Buffa et al., 2011]

- Number of samples: 207
- Number of mRNAs measured: 24.385

Application to breast cancer mRNA profiles

Dataset:
mRNA expression profiling obtained from a study of breast cancer patients [Buffa et al., 2011]

- Number of samples: 207
- Number of mRNAs measured: 24.385
- Patients in study are divided into two groups
  - Estrogen receptor positive (ER+)
  - Estrogen receptor negative (ER-)

Application to breast cancer mRNA profiles

Dataset:
mRNA expression profiling obtained from a study of breast cancer patients [Buffa et al., 2011]

- Number of samples: 207
- Number of mRNAs measured: 24.385
- Patients in study are divided into two groups
  - Estrogen receptor positive (ER+)
  - Estrogen receptor negative (ER-)
- Tumors from two groups show different molecular patterns in terms of cell differentiation, proliferation, survival, invasion, angiogenesis
- In general: better prognosis and treatment of ER+ patients compared to ER- patients

Application to breast cancer mRNA profiles

1. Binarization of data
   - 107 samples in test set, 100 samples in training set
   - risk ratio: 0.27

Application to breast cancer mRNA profiles

1. Binarization of data
   - 107 samples in test set, 100 samples in training set
   - risk ratio: 0.27
2. Application of NWT approach to 11 KEGG pathways
   - 7 signaling pathways
   - 2 pathways linked to cell adhesion
   - 2 pathways linked to cell cycle and apoptosis
3. **Results**: Found significant subgraphs in 9 KEGG pathways

Application to breast cancer mRNA profiles

KEGG pathway	Pathway description	genes in pathway	significant subgraphs	average size	runtime (in sec)
04115	p53 signaling pathway	65	2049	6.83	10.35
04150	mTOR signaling pathway	48	0		0.49
04330	Notch signaling pathway	46	93	5.42	1.73
04064	NF-kappa B signaling pathway	70	1	5	3.50
04012	ErbB signaling pathway	83	12	4.75	4.72
04010	MAPK signaling pathway	240	29063	7.79	149.34
04310	Wnt signaling pathway	127	91	5.58	176.74
04510	Focal adhesion	195	670	8.10	2824.77
04520	Adherens junction	68	0		2.19
04110	Cell cycle	114	45	6.69	33.25
04210	Apoptosis	75	21	6.33	0.96

Table of Contents

1 Searching for significant subgraphs: motivation and problem statement

2 State of the art: dmGWAS

3 The Tarone method in significant subgraph search

4 Application to gene expression data

5 Summary and future work

Summary of Network Tarone approach

Network Tarone approach

- Search for significant subgraphs in networks
- Rigorous correction for multiple hypothesis testing by controlling the FWER

Summary of Network Tarone approach

Network Tarone approach

- Search for significant subgraphs in networks
- Rigorous correction for multiple hypothesis testing by controlling the FWER
- Exploit testability of subgraphs: only subgraphs that are testable have to be corrected for
- Restriction of search space: non-testable subgraphs and their supergraphs can be pruned
- Efficient network exploration allows for growing subgraphs without visiting same subgraph multiple times

Summary of Network Tarone approach

Network Tarone approach

- Search for significant subgraphs in networks
- Rigorous correction for multiple hypothesis testing by controlling the FWER
- Exploit testability of subgraphs: only subgraphs that are testable have to be corrected for
- Restriction of search space: non-testable subgraphs and their supergraphs can be pruned
- Efficient network exploration allows for growing subgraphs without visiting same subgraph multiple times
- Less conservative significance threshold than classical approaches, such as Bonferroni correction

Future work

- Application to more sophisticated PPI networks
  - Networks with directed edges (pathways are directed)
    - → reduces number of networks to test

Future work

- Application to more sophisticated PPI networks
  - Networks with directed edges (pathways are directed)
    - → reduces number of networks to test
- Machine learning to binarize GWAS data
  - → Use information of all SNPs overlapping with gene

Future work

- Application to more sophisticated PPI networks
  - Networks with directed edges (pathways are directed)
    - → reduces number of networks to test
- Machine learning to binarize GWAS data
  - → Use information of all SNPs overlapping with gene
- Improve runtime for datasets with large sample sizes
  - → GPU implementation

Future work

- Application to more sophisticated PPI networks
  - Networks with directed edges (pathways are directed)
    - → reduces number of networks to test
- Machine learning to binarize GWAS data
  - → Use information of all SNPs overlapping with gene
- Improve runtime for datasets with large sample sizes
  - → GPU implementation
- Include correction for covariates
  - → Analogously to CMH

References I

- Azencott, C.-A., Grimm, D., Sugiyama, M., Kawahara, Y., and Borgwardt, K. M. (2013).
  Efficient network-guided multi-locus association mapping with graph cuts.
  Bioinformatics, 29(13):i171–i179.
- Buffa, F. M., Camps, C., Winchester, L., Snell, C. E., Gee, H. E., Sheldon, H., Taylor, M., Harris, A. L., and Ragoussis, J. (2011).
  microrna-associated progression pathways and potential therapeutic targets identified by integrated mrna and microrna expression profiling in breast cancer.
  Cancer research, 71(17):5635–5645.
- Chuang, H.-Y., Lee, E., Liu, Y.-T., Lee, D., and Ideker, T. (2007).
  Network-based classification of breast cancer metastasis.
  Molecular systems biology, 3(1).
- Cowley, M. J., Pinese, M., Kassahn, K. S., Waddell, N., Pearson, J. V., Grimmond, S. M., Biankin, A. V., Hautaniemi, S., and Wu, J. (2011).
  Pina v2. 0: mining interactome modules.
  Nucleic acids research, page gkr967.
- Jia, P., Zheng, S., Long, J., Zheng, W., and Zhao, Z. (2011).
  dmgwas: dense module searching for genome-wide association studies in protein–protein interaction networks.
  Bioinformatics, 27(1):95–102.
- Kanehisa, M. and Goto, S. (2000).
  Kegg: kyoto encyclopedia of genes and genomes.
  Nucleic acids research, 28(1):27–30.
- Lippert, C., Listgarten, J., Liu, Y., Kadie, C. M., Davidson, R. I., and Heckerman, D. (2011).
  Fast linear mixed models for genome-wide association studies.
  Nature methods, 8(10):833–835.

References II

- Llinares-López, F., Grimm, D. G., Bodenham, D. A., Gieraths, U., Sugiyama, M., Rowan, B., and Borgwardt, K. (2015).
  Genome-wide detection of intervals of genetic heterogeneity associated with complex traits.
  Bioinformatics, 31(12):i240–i249.
- Manolio, T. A., Collins, F. S., Cox, N. J., Goldstein, D. B., Hindorff, L. A., Hunter, D. J., McCarthy, M. I., Ramos, E. M., Cardon, L. R., Chakravarti, A., et al. (2009).
  Finding the missing heritability of complex diseases.
  Nature, 461(7265):747–753.
- Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A., Bender, D., Maller, J., Sklar, P., De Bakker, P. I., Daly, M. J., et al. (2007).
  Plink: a tool set for whole-genome association and population-based linkage analyses.
  The American Journal of Human Genetics, 81(3):559–575.
- Rossin, E. J., Lage, K., Raychaudhuri, S., Xavier, R. J., Tatar, D., Benita, Y., Cotsapas, C., Daly, M. J., Constortium, I. I. B. D. G., et al. (2011).
  Proteins encoded in genomic regions associated with immune-mediated disease physically interact and suggest underlying biology.
  PLoS Genet, 7(1):e1001273.
- Wernicke, S. (2006).
  Efficient detection of network motifs.
  IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), 3(4):347–359.

Thank you

- Felipe Llinares López
- Dean Bodenham
- Udo Gieraths
- Dominik Grimm
- Elisabetta Ghisu
- Anja Gumpinger
- Xiao He
- Laetitia Papaxanthos
- Damian Roqueiro
- Menno Witteveen
- Birgit Knapp

Machine Learning & Computational Biology Lab

Department of Biosystems Science and Engineering, Basel

Karsten Borgwardt
karsten.borgwardt@bsse.ethz.ch

MLCB

ETH zürich

Thank you for your patience and attention