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 II by Laetitia

To be discussed in Part III by Anja
Significant pattern mining

Key aspects

<table>
<thead>
<tr>
<th>Pattern P is present</th>
<th>Pattern P is not present</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C = 1$</td>
<td>$a$</td>
</tr>
<tr>
<td>$C = 0$</td>
<td>$x - a$</td>
</tr>
<tr>
<td>$x$</td>
<td>$n - x$</td>
</tr>
</tbody>
</table>

Where

- $n$ : total number of transactions
- $n_1$ : number of transactions with class label $C = 1$
- $x$ : support of the pattern $P$, i.e. number of transactions where $P$ is present
- $a$ : support of the pattern $P$ 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

<table>
<thead>
<tr>
<th>Pattern $\mathcal{P}$ is present</th>
<th>Pattern $\mathcal{P}$ is not present</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C = 1$</td>
<td>$a$</td>
</tr>
<tr>
<td>$C = 0$</td>
<td>$x - a$</td>
</tr>
</tbody>
</table>

- Compute $p$-value based on $a$, $x$, $n_1$ and $n$
- Use Fisher’s Exact Test [R.A. Fisher, 1922]
  - $2 \times 2$ 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

<table>
<thead>
<tr>
<th>True null hypothesis</th>
<th>Number not rejected</th>
<th>Number rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$U$</td>
<td>$V$</td>
</tr>
<tr>
<td>Non-true null hypothesis</td>
<td>$T$</td>
<td>$S$</td>
</tr>
<tr>
<td></td>
<td>$m - R$</td>
<td>$R$</td>
</tr>
</tbody>
</table>

- $V$ 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

Adjustment 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 \( i = 1 \ldots m \) 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 \( k < m \)?
Multiple hypothesis testing

Adjustment 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 $i = 1 \ldots m$ 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

Adjustment 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 $i = 1 \ldots m$ 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

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>A</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controls</td>
<td>4</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>Cases</td>
<td>1</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Total</td>
<td>5</td>
<td>12</td>
<td>17</td>
</tr>
</tbody>
</table>

Example: Association test in GWAS

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

Enumeration of all matrices

\[
\begin{bmatrix} 5 & 5 \\ 0 & 7 \end{bmatrix} \quad \begin{bmatrix} 4 & 6 \\ 1 & 6 \end{bmatrix} \quad \begin{bmatrix} 3 & 7 \\ 2 & 5 \end{bmatrix} \quad \begin{bmatrix} 2 & 8 \\ 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 1 & 9 \\ 4 & 3 \end{bmatrix} \quad \begin{bmatrix} 0 & 10 \\ 5 & 2 \end{bmatrix}
\]

Where each \( p \) is obtained from the hyper-geometric distribution

\[
\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \quad \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \quad \begin{bmatrix} n \\ c_1 \end{bmatrix}
\]

\[
p = \frac{\binom{r_1}{x_{11}} \binom{r_2}{x_{21}}}{\binom{n}{c_1}}
\]

Fisher’s \( p \)-value

\[
p = 0.338235 = 0.040724 + 0.237557 + 0.056561 + 0.003394
\]

Using “biased matrices”
Testability

Deconstructing Fisher’s Exact Test

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>A</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controls</td>
<td>4</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>Cases</td>
<td>1</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Total</td>
<td>5</td>
<td>12</td>
<td>17</td>
</tr>
</tbody>
</table>

Example: Association test in GWAS

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

Enumeration of all matrices

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

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

\[
\begin{bmatrix}
x_{11} & x_{12} \\
x_{21} & x_{22} \\
\end{bmatrix}
\begin{bmatrix}
r_1 \\
r_2 \\
\end{bmatrix}
\rightarrow 
\begin{bmatrix}
x_{11} \\
x_{21} \\
\end{bmatrix}
\begin{bmatrix}
r_1 \\
r_2 \\
\end{bmatrix}
\rightarrow 
\begin{bmatrix}
\frac{r_1}{n} \times \frac{r_2}{c_1} \\
\frac{x_{11}}{n} \times \frac{x_{21}}{c_1} \\
\end{bmatrix}
\]

Fisher’s $p$-value

\[
p = 0.338235 = 0.040724 + 0.237557 + 0.056561 + 0.003394
\]

Using “biased matrices”
Testability

Deconstructing Fisher’s Exact Test

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>A</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controls</td>
<td>4</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>Cases</td>
<td>1</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Total</td>
<td>5</td>
<td>12</td>
<td>17</td>
</tr>
</tbody>
</table>

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

Enumeration of all matrices

\[
\begin{vmatrix}
5 & 5 \\
0 & 7 \\
\end{vmatrix}
\leftarrow
\begin{vmatrix}
4 & 6 \\
1 & 6 \\
\end{vmatrix}
\rightarrow
\begin{vmatrix}
3 & 7 \\
2 & 5 \\
\end{vmatrix}
\rightarrow
\begin{vmatrix}
2 & 8 \\
3 & 4 \\
\end{vmatrix}
\rightarrow
\begin{vmatrix}
1 & 9 \\
4 & 3 \\
\end{vmatrix}
\rightarrow
\begin{vmatrix}
0 & 10 \\
5 & 2 \\
\end{vmatrix}
\]

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

\[
\begin{pmatrix}
x_{11} & x_{12} \\
x_{21} & x_{22} \\
\end{pmatrix}
\begin{pmatrix}
r_1 \\
r_2 \\
\end{pmatrix}
\rightarrow
p = \frac{\binom{r_1}{x_{11}} \binom{r_2}{x_{21}} \binom{n}{c_1}}{\binom{n}{c_1}}
\]

Fisher’s $p$-value

\[p = 0.338235 = 0.040724 + 0.237557 + 0.056561 + 0.003394\]

Using “biased matrices”
Testability

Deconstructing Fisher’s Exact Test

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>A</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controls</td>
<td>4</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>Cases</td>
<td>1</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Total</td>
<td>5</td>
<td>12</td>
<td>17</td>
</tr>
</tbody>
</table>

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

**Enumeration of all matrices**

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

\( p = 0.040724 \quad p = 0.237557 \quad p = 0.407240 \quad p = 0.254525 \quad p = 0.056561 \quad p = 0.003394 \)

Where each \( p \) is obtained from the hyper-geometric distribution

\[
\begin{bmatrix}
X_{11} & X_{12} \\
X_{21} & X_{22} \\
\end{bmatrix}
\rightarrow \begin{bmatrix} r_1 \\
r_2 \\
\end{bmatrix} = \begin{bmatrix}
\binom{r_1}{x_{11}} \\
\binom{r_2}{x_{21}} \\
\end{bmatrix}
\rightarrow p = \frac{\binom{n}{r_1} \binom{n}{r_2}}{\binom{n}{c_1}}
\]

**Fisher’s \( p \)-value**

\[
p = 0.338235 = 0.040724 + 0.237557 + 0.056561 + 0.003394
\]

Using “biased matrices”
Testability

Minimum attainable $p$-value

\[
\begin{bmatrix}
5 & 5 \\
0 & 7
\end{bmatrix} \quad \begin{bmatrix}
4 & 6 \\
1 & 6
\end{bmatrix} \quad \begin{bmatrix}
3 & 7 \\
2 & 5
\end{bmatrix} \quad \begin{bmatrix}
2 & 8 \\
3 & 4
\end{bmatrix} \quad \begin{bmatrix}
1 & 9 \\
4 & 3
\end{bmatrix} \quad \begin{bmatrix}
0 & 10 \\
5 & 2
\end{bmatrix}
\]

\[p = 0.040724\quad p = 0.237557\quad p = 0.407240\quad p = 0.254525\quad p = 0.056561\quad p = 0.003394\]

Key elements

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

#### Minimum attainable $p$-value

\[
\begin{bmatrix}
5 & 5 \\
0 & 7 \\
\end{bmatrix} \quad \left\langle \begin{bmatrix}
4 & 6 \\
1 & 6 \\
\end{bmatrix} \right\rangle \quad \begin{bmatrix}
3 & 7 \\
2 & 5 \\
\end{bmatrix} \quad \begin{bmatrix}
2 & 8 \\
3 & 4 \\
\end{bmatrix} \quad \begin{bmatrix}
1 & 9 \\
4 & 3 \\
\end{bmatrix} \quad \begin{bmatrix}
0 & 10 \\
5 & 2 \\
\end{bmatrix}
\]

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

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

- If $r_1 \geq c_1$, then
  \[
  \begin{bmatrix}
  c_1 & x_{12} - x_{21} \\
  0 & x_{22} + x_{21} \\
  \end{bmatrix} \quad \frac{r_1}{r_2}
  \]
  \[
  \begin{bmatrix}
  0 & x_{12} + x_{11} \\
  c_1 & x_{22} - x_{11} \\
  \end{bmatrix} \quad \frac{n}{r_1}
  \]

- Otherwise
  \[
  \begin{bmatrix}
  c_1 & x_{12} - x_{21} \\
  0 & x_{22} + x_{21} \\
  \end{bmatrix} \quad \frac{r_1}{r_2}
  \]
  \[
  \begin{bmatrix}
  0 & x_{12} + x_{11} \\
  c_1 & x_{22} - x_{11} \\
  \end{bmatrix} \quad \frac{n}{r_1}
  \]

With $p_{\text{min}} = \left( \frac{r_1}{c_1} \right) / \left( \frac{n}{c_1} \right)$
Reducing the Bonferroni correction factor

An illustrative example

- Perform association tests on \( m = 5 \) SNPs
- Significance level \( \alpha = 0.05 \)
- With Bonferroni correction
  \[ \delta = \frac{\alpha}{m} = 0.01 \]

<table>
<thead>
<tr>
<th>Id</th>
<th>Observed</th>
<th>Fisher’s ( p )-value</th>
</tr>
</thead>
</table>
| SNP1 | \[
\begin{bmatrix}
2 \\
1 
\end{bmatrix}
\] | 0.2 |
| SNP2 | \[
\begin{bmatrix}
2 \\
2 
\end{bmatrix}
\] | 1.0 |
| SNP3 | \[
\begin{bmatrix}
2 \\
7 
\end{bmatrix}
\] | 0.015220 |
| SNP4 | \[
\begin{bmatrix}
3 \\
2 
\end{bmatrix}
\] | 1.0 |
| SNP5 | \[
\begin{bmatrix}
1 \\
3 
\end{bmatrix}
\] | 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?

Damian Roqueiro | Testability and correction for multiple hypothesis testing
COST Antwerp | 27 April 2016 | 13 / 23
Reducing the Bonferroni correction factor

An illustrative example

- Perform association tests on \( m = 5 \) SNPs
- Significance level \( \alpha = 0.05 \)
- With Bonferroni correction
  \[ \delta = \frac{\alpha}{m} = 0.01 \]

<table>
<thead>
<tr>
<th></th>
<th>( a )</th>
<th>( A )</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controls</td>
<td>( x_{11} )</td>
<td>( x_{12} )</td>
<td>( r_1 )</td>
</tr>
<tr>
<td>Cases</td>
<td>( x_{21} )</td>
<td>( x_{22} )</td>
<td>( r_2 )</td>
</tr>
<tr>
<td>Total</td>
<td>( c_1 )</td>
<td>( c_2 )</td>
<td>( n )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Id</th>
<th>Observed</th>
<th>Fisher’s ( p )-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNP_1</td>
<td>( [2 \ 6] )</td>
<td>0.2</td>
</tr>
<tr>
<td>SNP_2</td>
<td>( [2 \ 7] )</td>
<td>1.0</td>
</tr>
<tr>
<td>SNP_3</td>
<td>( [2 \ 1] )</td>
<td>0.015220</td>
</tr>
<tr>
<td>SNP_4</td>
<td>( [3 \ 11] )</td>
<td>1.0</td>
</tr>
<tr>
<td>SNP_5</td>
<td>( [1 \ 9] )</td>
<td>0.274510</td>
</tr>
</tbody>
</table>

- 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

<table>
<thead>
<tr>
<th>Id</th>
<th>Observed</th>
<th>Fisher’s $p$-value</th>
<th>Most biased</th>
<th>$p_{min}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNP1</td>
<td>[2 6] [1 6]</td>
<td>0.2</td>
<td>[0 8] [3 4]</td>
<td>0.076923</td>
</tr>
<tr>
<td>SNP2</td>
<td>[2 8] [2 7]</td>
<td>1.0</td>
<td>[0 10] [4 5]</td>
<td>0.032508</td>
</tr>
<tr>
<td>SNP3</td>
<td>[2 8] [7 1]</td>
<td>0.015220</td>
<td>[1 9] [8 0]</td>
<td>0.000206</td>
</tr>
<tr>
<td>SNP4</td>
<td>[3 11] [2 7]</td>
<td>1.0</td>
<td>[0 14] [5 4]</td>
<td>0.003745</td>
</tr>
<tr>
<td>SNP5</td>
<td>[1 9] [3 5]</td>
<td>0.274510</td>
<td>[0 10] [4 4]</td>
<td>0.022876</td>
</tr>
</tbody>
</table>

SNP1 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

<table>
<thead>
<tr>
<th>Id</th>
<th>Observed</th>
<th>Fisher’s p-value</th>
<th>Most biased</th>
<th>( p_{min} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNP1</td>
<td>[2\ 6\ 1\ 6]</td>
<td>0.2</td>
<td>[0\ 8\ 3\ 4]</td>
<td>0.076923</td>
</tr>
<tr>
<td>SNP2</td>
<td>[2\ 8\ 2\ 7]</td>
<td>1.0</td>
<td>[0\ 10\ 4\ 5]</td>
<td>0.032508</td>
</tr>
<tr>
<td>SNP3</td>
<td>[2\ 8\ 7\ 1]</td>
<td>0.015220</td>
<td>[1\ 9\ 8\ 0]</td>
<td>0.000206</td>
</tr>
<tr>
<td>SNP4</td>
<td>[3\ 11\ 2\ 7]</td>
<td>1.0</td>
<td>[0\ 14\ 5\ 4]</td>
<td>0.003745</td>
</tr>
<tr>
<td>SNP5</td>
<td>[1\ 9\ 3\ 5]</td>
<td>0.274510</td>
<td>[0\ 10\ 4\ 4]</td>
<td>0.022876</td>
</tr>
</tbody>
</table>

- SNP1 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($\mathcal{H}, \alpha$)
▷ $\mathcal{H}$: Set of all hypotheses
▷ $\alpha$: Nominal significance level
  $k \leftarrow 0$
  repeat
    $k \leftarrow k + 1$
    $\mathcal{T} \leftarrow \text{get_testable_set}(\mathcal{H}, \frac{\alpha}{k})$
  until $k \geq |\mathcal{T}|$
▷ Ready to perform Fisher’s Exact Tests
  $\delta \leftarrow \frac{\alpha}{k}$
  perform_fisher_exact_tests($\mathcal{H}_T, \delta$)

function get_testable_set($\mathcal{H}, \delta$)
▷ 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($h, \delta$)
▷ Check if hypothesis $h$ is testable
  $p_{\text{min}} \leftarrow \text{compute_min_pvalue}(h)$
  if $p_{\text{min}} > \delta$ 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 $\text{FWER} \leq \delta |\mathcal{T}| \leq \alpha$

procedure main($\mathcal{H}$, $\alpha$)
  $\triangleright \mathcal{H}$: Set of all hypotheses
  $\triangleright \alpha$: Nominal significance level
  $k \leftarrow 0$
  repeat
    $k \leftarrow k + 1$
    $\mathcal{T} \leftarrow \text{get_testable_set}(\mathcal{H}, \frac{\alpha}{k})$
  until $k \geq |\mathcal{T}|$
Reducing the Bonferroni correction factor

**Tarone’s method** R.E. Tarone, 1990

repeat
  \[ k \leftarrow k + 1 \]
  \[ T \leftarrow \text{get\_testable\_set}(\mathcal{H}, \frac{\alpha}{k}) \]
until \( k \geq |T| \)

<table>
<thead>
<tr>
<th>Id</th>
<th>Observed</th>
<th>Most biased</th>
<th>Min. p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNP1</td>
<td>[2 6]</td>
<td>[0 8]</td>
<td>0.076923</td>
</tr>
<tr>
<td></td>
<td>[1 6]</td>
<td>[3 4]</td>
<td></td>
</tr>
<tr>
<td>SNP2</td>
<td>[2 8]</td>
<td>[0 10]</td>
<td>0.032508</td>
</tr>
<tr>
<td></td>
<td>[2 7]</td>
<td>[4 5]</td>
<td></td>
</tr>
<tr>
<td>SNP3</td>
<td>[2 8]</td>
<td>[1 9]</td>
<td>0.000206</td>
</tr>
<tr>
<td></td>
<td>[7 1]</td>
<td>[8 0]</td>
<td></td>
</tr>
<tr>
<td>SNP4</td>
<td>[3 11]</td>
<td>[0 14]</td>
<td>0.003745</td>
</tr>
<tr>
<td></td>
<td>[2 7]</td>
<td>[5 4]</td>
<td></td>
</tr>
<tr>
<td>SNP5</td>
<td>[1 9]</td>
<td>[0 10]</td>
<td>0.022876</td>
</tr>
<tr>
<td></td>
<td>[3 5]</td>
<td>[4 4]</td>
<td></td>
</tr>
</tbody>
</table>
Reducing the Bonferroni correction factor

**Tarone’s method**  R.E. Tarone, 1990

repeat
\[ k \leftarrow k + 1 \]
\[ T \leftarrow \text{get\_testable\_set}(\mathcal{H}, \frac{\alpha}{k}) \]
until \( k \geq |T| \)

- With \( k = 1, \delta = 0.05, T = \{2, 3, 4, 5\} \)
  
  Condition \( k \geq |T| \) is False → next iteration

<table>
<thead>
<tr>
<th>Id</th>
<th>Observed</th>
<th>Most biased</th>
<th>Min. p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNP1</td>
<td>2 6 1 6</td>
<td>0 8 3 4</td>
<td>0.076923</td>
</tr>
<tr>
<td>SNP2</td>
<td>2 8 2 7</td>
<td>0 10 4 5</td>
<td>0.032508</td>
</tr>
<tr>
<td>SNP3</td>
<td>2 8 7 1</td>
<td>1 9 8 0</td>
<td>0.000206</td>
</tr>
<tr>
<td>SNP4</td>
<td>3 11 2 7</td>
<td>0 14 5 4</td>
<td>0.003745</td>
</tr>
<tr>
<td>SNP5</td>
<td>1 9 3 5</td>
<td>0 10 4 4</td>
<td>0.022876</td>
</tr>
</tbody>
</table>
Reducing the Bonferroni correction factor

Tarone’s method  R.E. Tarone, 1990

repeat
  \[ k \leftarrow k + 1 \]
  \[ \mathcal{T} \leftarrow \text{get\_testable\_set}(\mathcal{H}, \frac{\alpha}{k}) \]
until \( k \geq |\mathcal{T}| \)

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

- With \( k = 2 \), \( \delta = 0.025 \), \( \mathcal{T} = \{3, 4, 5\} \)
  Condition \( k \geq |\mathcal{T}| \) is False → next iteration

<table>
<thead>
<tr>
<th>Id</th>
<th>Observed</th>
<th>Most biased</th>
<th>Min. p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNP1</td>
<td>[\begin{bmatrix} 2 &amp; 6 \ 1 &amp; 6 \end{bmatrix}]</td>
<td>[\begin{bmatrix} 0 &amp; 8 \ 3 &amp; 4 \end{bmatrix}]</td>
<td>0.076923</td>
</tr>
<tr>
<td>SNP2</td>
<td>[\begin{bmatrix} 2 &amp; 8 \ 2 &amp; 7 \end{bmatrix}]</td>
<td>[\begin{bmatrix} 0 &amp; 10 \ 4 &amp; 5 \end{bmatrix}]</td>
<td>0.032508</td>
</tr>
<tr>
<td>SNP3</td>
<td>[\begin{bmatrix} 2 &amp; 8 \ 7 &amp; 1 \end{bmatrix}]</td>
<td>[\begin{bmatrix} 1 &amp; 9 \ 8 &amp; 0 \end{bmatrix}]</td>
<td>0.000206</td>
</tr>
<tr>
<td>SNP4</td>
<td>[\begin{bmatrix} 3 &amp; 11 \ 2 &amp; 7 \end{bmatrix}]</td>
<td>[\begin{bmatrix} 0 &amp; 14 \ 5 &amp; 4 \end{bmatrix}]</td>
<td>0.003745</td>
</tr>
<tr>
<td>SNP5</td>
<td>[\begin{bmatrix} 1 &amp; 9 \ 3 &amp; 5 \end{bmatrix}]</td>
<td>[\begin{bmatrix} 0 &amp; 10 \ 4 &amp; 4 \end{bmatrix}]</td>
<td>0.022876</td>
</tr>
</tbody>
</table>
Reducing the Bonferroni correction factor

Tarone’s method  R.E. Tarone, 1990

repeat
\[ k \leftarrow k + 1 \]
\[ T \leftarrow \text{get\_testable\_set}(\mathcal{H}, \frac{\alpha}{k}) \]
until \( k \geq |T| \)

- With \( k = 1, \delta = 0.05, T = \{2, 3, 4, 5\} \)
  
  Condition \( k \geq |T| \) is False → next iteration

- With \( k = 2, \delta = 0.025, T = \{3, 4, 5\} \)
  
  Condition \( k \geq |T| \) is False → next iteration

- With \( k = 3, \delta = 0.0167, T = \{3, 4\} \)
  
  \( k \geq |T| \) evaluates to True → Stop

<table>
<thead>
<tr>
<th>Id</th>
<th>Observed</th>
<th>Most biased</th>
<th>Min. p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNP1</td>
<td>[2 6]</td>
<td>[0 8]</td>
<td>0.076923</td>
</tr>
<tr>
<td>SNP2</td>
<td>[2 8]</td>
<td>[0 10]</td>
<td>0.032508</td>
</tr>
<tr>
<td>SNP3</td>
<td>[2 8]</td>
<td>[1 9]</td>
<td>0.000206</td>
</tr>
<tr>
<td>SNP4</td>
<td>[3 11]</td>
<td>[0 14]</td>
<td>0.003745</td>
</tr>
<tr>
<td>SNP5</td>
<td>[1 9]</td>
<td>[0 10]</td>
<td>0.022876</td>
</tr>
</tbody>
</table>
Reducing the Bonferroni correction factor

Tarone’s method  R.E. Tarone, 1990

\[
\begin{align*}
\text{repeat} & \\
& k \leftarrow k + 1 \\
& T \leftarrow \text{get\_testable\_set}(\mathcal{H}, \frac{\alpha}{k}) \\
\text{until } & k \geq |T| \\
& \triangleright k = 3 \\
& \triangleright \delta = \frac{\alpha}{k} = 0.0167 \\
& \triangleright T = \{3, 4\}
\end{align*}
\]

\[
\text{\triangleright Perform Fisher’s Exact Test on SNP}_3 \text{ and SNP}_4
\]

<table>
<thead>
<tr>
<th>Id</th>
<th>Observed</th>
<th>Most biased</th>
<th>Min. p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNP₁</td>
<td>[\begin{bmatrix} 2 &amp; 6 \ 1 &amp; 6 \end{bmatrix}]</td>
<td>[\begin{bmatrix} 0 &amp; 8 \ 3 &amp; 4 \end{bmatrix}]</td>
<td>0.076923</td>
</tr>
<tr>
<td>SNP₂</td>
<td>[\begin{bmatrix} 2 &amp; 8 \ 2 &amp; 7 \end{bmatrix}]</td>
<td>[\begin{bmatrix} 0 &amp; 10 \ 4 &amp; 5 \end{bmatrix}]</td>
<td>0.032508</td>
</tr>
<tr>
<td>SNP₃</td>
<td>[\begin{bmatrix} 2 &amp; 8 \ 7 &amp; 1 \end{bmatrix}]</td>
<td>[\begin{bmatrix} 1 &amp; 9 \ 8 &amp; 0 \end{bmatrix}]</td>
<td>0.000206</td>
</tr>
<tr>
<td>SNP₄</td>
<td>[\begin{bmatrix} 3 &amp; 11 \ 2 &amp; 7 \end{bmatrix}]</td>
<td>[\begin{bmatrix} 0 &amp; 14 \ 5 &amp; 4 \end{bmatrix}]</td>
<td>0.003745</td>
</tr>
<tr>
<td>SNP₅</td>
<td>[\begin{bmatrix} 1 &amp; 9 \ 3 &amp; 5 \end{bmatrix}]</td>
<td>[\begin{bmatrix} 0 &amp; 10 \ 4 &amp; 4 \end{bmatrix}]</td>
<td>0.022876</td>
</tr>
</tbody>
</table>
Reducing the Bonferroni correction factor

Tarone’s method  R.E. Tarone, 1990

repeat
  
  $k \leftarrow k + 1$

  $\mathcal{T} \leftarrow \text{get_testable_set}(\mathcal{H}, \frac{\alpha}{k})$

until $k \geq |\mathcal{T}|$

$\triangleright k = 3$

$\triangleright \delta = \frac{\alpha}{k} = 0.0167$

$\triangleright \mathcal{T} = \{3, 4\}$

$\triangleright$ Perform Fisher’s Exact Test on SNP$_3$ and SNP$_4$

SNP$_3$ $\rightarrow$ $p$-value = 0.015220

SNP$_4$ $\rightarrow$ $p$-value = 1.0

SNP$_3$ is statistically significant at level $\delta$
Reducing the Bonferroni correction factor

Tarone’s method  R.E. Tarone, 1990

repeat

    \( k \leftarrow k + 1 \)
    \( \mathcal{T} \leftarrow \text{get\_testable\_set}(\mathcal{H}, \frac{\alpha}{k}) \)

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\(_4\)

SNP\(_3\) \( \rightarrow p\text{-value} = 0.015220 \)
SNP\(_4\) \( \rightarrow p\text{-value} = 1.0 \)

- SNP\(_3\) is statistically significant at level \( \delta \)

<table>
<thead>
<tr>
<th>Id</th>
<th>Observed</th>
<th>Most biased</th>
<th>Min. p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNP(_1)</td>
<td>2 1 6</td>
<td>0 3 4</td>
<td>0.076923</td>
</tr>
<tr>
<td>SNP(_2)</td>
<td>2 8 7</td>
<td>0 4 5</td>
<td>0.032508</td>
</tr>
<tr>
<td>SNP(_3)</td>
<td>2 8 1 9 8 0</td>
<td>0.000206</td>
<td></td>
</tr>
<tr>
<td>SNP(_4)</td>
<td>3 11 7 4 14</td>
<td>0.003745</td>
<td></td>
</tr>
<tr>
<td>SNP(_5)</td>
<td>1 9 3 4 10 4</td>
<td>0.022876</td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Id</th>
<th>Observed</th>
<th>Fisher's p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNP₁</td>
<td>[2\ 6]</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>[1 \ 6]</td>
<td></td>
</tr>
<tr>
<td>SNP₂</td>
<td>[2\ 8]</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>[2 \ 7]</td>
<td></td>
</tr>
<tr>
<td>SNP₃</td>
<td>[2\ 8]</td>
<td>0.015220</td>
</tr>
<tr>
<td></td>
<td>[7 \ 1]</td>
<td></td>
</tr>
<tr>
<td>SNP₄</td>
<td>[3\ 11]</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>[2 \ 7]</td>
<td></td>
</tr>
<tr>
<td>SNP₅</td>
<td>[1\ 9]</td>
<td>0.274510</td>
</tr>
<tr>
<td></td>
<td>[3 \ 5]</td>
<td></td>
</tr>
</tbody>
</table>
Final thoughts

Pre-computing minimum attainable $p$-values

<table>
<thead>
<tr>
<th>Pattern $\mathcal{P}$ is present</th>
<th>Pattern $\mathcal{P}$ is not present</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C = 1$</td>
<td>$a$</td>
</tr>
<tr>
<td>$C = 0$</td>
<td>$x - a$</td>
</tr>
<tr>
<td></td>
<td>$x$</td>
</tr>
</tbody>
</table>

- Margins are assumed to be equal for all $H_i$, e.g. imputed data in GWAS association test
- Therefore, $p_{\text{min}}$ 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 $\delta_k$
- 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 $\delta_k$
- 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


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
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, \(\approx 10 \text{bp} \text{ to } 100 \text{kb 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.


# 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

<table>
<thead>
<tr>
<th></th>
<th>( y )</th>
<th>( l ) ordered binary markers</th>
<th>meta-marker</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_1 ) cases</td>
<td>[image of intervals]</td>
<td>[image of binary markers]</td>
<td>( g_1 ) 0</td>
</tr>
<tr>
<td>( n_2 ) controls</td>
<td>[image of intervals]</td>
<td>[image of binary markers]</td>
<td>( g_2 ) 1</td>
</tr>
<tr>
<td></td>
<td>( [t_{s,1}, t_{c,1}] )</td>
<td>( g_{n-2} ) 0</td>
<td></td>
</tr>
</tbody>
</table>
### Association testing between meta-markers and phenotype

<table>
<thead>
<tr>
<th>Variables</th>
<th>Meta-marker = 1</th>
<th>Meta-marker = 0</th>
<th>Row totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y = \text{case}$</td>
<td>$a$</td>
<td>$n_1 - a$</td>
<td>$n_1$ cases</td>
</tr>
<tr>
<td>$y = \text{control}$</td>
<td>$x - a$</td>
<td>$n_2 - (x - a)$</td>
<td>$n_2$ controls</td>
</tr>
<tr>
<td>Col totals</td>
<td>$x$</td>
<td>$n - x$</td>
<td>$n$</td>
</tr>
</tbody>
</table>

#### Notation:
- Genomic interval: $J_t, t_{K}$
- Binary meta-marker: $g(J_t, t_{K}) = (g_1, \ldots, g_n)$

Corresponding $p$-value based on entries $a, x, n_1$ and $n_2$

Fisher’s Exact Test or Pearson’s $\chi^2$ Test

How to correct for confounders?

Laetitia Papaxanthos  
Genetic heterogeneity detection with categorical covariates  
COST Antwerp | 27 April 2016 | 10 / 39
Association testing between meta-markers and phenotype

<table>
<thead>
<tr>
<th>Variables</th>
<th>Meta-marker = 1</th>
<th>Meta-marker = 0</th>
<th>Row totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y =$ case</td>
<td>$a$</td>
<td>$n_1 - a$</td>
<td>$n_1$ cases</td>
</tr>
<tr>
<td>$y =$ control</td>
<td>$x - a$</td>
<td>$n_2 - (x - a)$</td>
<td>$n_2$ controls</td>
</tr>
<tr>
<td>Col totals</td>
<td>$x$</td>
<td>$n - x$</td>
<td>$n$</td>
</tr>
</tbody>
</table>

Notation:

- Genomic interval: $[t_e, t_s]$
- Binary meta-marker: $g([t_e, t_s]) = (g_1, \ldots, g_n)$
### Association testing between meta-markers and phenotype

<table>
<thead>
<tr>
<th>Variables</th>
<th>Meta-marker = 1</th>
<th>Meta-marker = 0</th>
<th>Row totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>y = case</td>
<td></td>
<td></td>
<td>n₁</td>
</tr>
<tr>
<td>y = control</td>
<td></td>
<td></td>
<td>n₁ - a</td>
</tr>
<tr>
<td>Col totals</td>
<td></td>
<td></td>
<td>n₂</td>
</tr>
</tbody>
</table>

- \( n₁ \) cases
- \( n₂ \) controls

#### Notation:
- Genomic interval: \( [t_e, t_s] \)
- Binary meta-marker: \( g([t_e, t_s]) = (g₁, \ldots, gₙ) \)
- Corresponding \( p \)-value based on entries \( a, x, n₁ \) and \( n₂ \)
- Fisher’s Exact Test or Pearson’s \( \chi^2 \) Test

#### Metamarker

<table>
<thead>
<tr>
<th>( g )</th>
<th>( [t_e, t_s] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g₁ )</td>
<td>( [t_e, t_s] )</td>
</tr>
<tr>
<td>( g₂ )</td>
<td>0</td>
</tr>
<tr>
<td>( gₙ₋₁ )</td>
<td>1</td>
</tr>
<tr>
<td>( gₙ )</td>
<td>0</td>
</tr>
</tbody>
</table>

**Note:**
- \( l \) ordered binary markers
- \( a, x \) entries
- \( n₁ \) cases
- \( n₂ \) controls

---

Laetitia Papaxanthos | Genetic heterogeneity detection with categorical covariates
---

COST Antwerp | 27 April 2016 | 10 / 39
Association testing between meta-markers and phenotype

<table>
<thead>
<tr>
<th>Variables</th>
<th>Meta-marker = 1</th>
<th>Meta-marker = 0</th>
<th>Row totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y = \text{case} )</td>
<td>( a )</td>
<td>( n_1 - a )</td>
<td>( n_1 ) cases</td>
</tr>
<tr>
<td>( y = \text{control} )</td>
<td>( x - a )</td>
<td>( n_2 - (x - a) )</td>
<td>( n_2 ) controls</td>
</tr>
<tr>
<td>Col totals</td>
<td>( x )</td>
<td>( n - x )</td>
<td>( n )</td>
</tr>
</tbody>
</table>

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

\[ y \perp\!\!\!\!\!\!\perp g \text{ but } y \perp g \mid c \]

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

Examples of non-confounded and confounded genomic intervals

<table>
<thead>
<tr>
<th>y</th>
<th>c ∈ {alive, dead}</th>
<th>l ordered binary markers</th>
<th>meta-markers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 0 1 0 1</td>
<td></td>
<td>0 1</td>
</tr>
<tr>
<td></td>
<td>0 0 0 1 0 0 0 1 1 0 0 0 1 0 1 0 1 0 1</td>
<td></td>
<td>1 1</td>
</tr>
<tr>
<td></td>
<td>0 0 0 0 0 1 0 1 1 0 0 0 0 1 0 0 0 1 0 1</td>
<td></td>
<td>1 1</td>
</tr>
<tr>
<td></td>
<td>0 0 0 0 1 0 0 1 1 0 0 0 0 0 1 0 1 0 1</td>
<td></td>
<td>1 1</td>
</tr>
<tr>
<td></td>
<td>0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1</td>
<td></td>
<td>1 0</td>
</tr>
<tr>
<td></td>
<td>0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 0 1 0</td>
<td></td>
<td>0 1</td>
</tr>
<tr>
<td></td>
<td>0 0 0 0 0 0 1 1 0 0 0 1 1 0 0 1 0 1</td>
<td></td>
<td>0 1</td>
</tr>
<tr>
<td></td>
<td>0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 1</td>
<td></td>
<td>0 0</td>
</tr>
<tr>
<td></td>
<td>0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 1 0 1</td>
<td></td>
<td>1 0</td>
</tr>
<tr>
<td></td>
<td>0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 1</td>
<td></td>
<td>0 0</td>
</tr>
</tbody>
</table>

$g([t_{e,1}, t_{e,2}])$ $g([t_{e,2}, t_{e,2}])$
Correcting for confounders with the Cochran-Mantel-Haenszel (CMH) Test

\( g([t_e, t_s]) \in \mathcal{R}^n \) 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 \( a, x, n_1 \) and \( n_2 \).

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

\[
= \frac{\left( \sum_{h=1}^{k} a_h - x_h \frac{n_{1,h}}{n_h} \right)^2}{\sum_{h=1}^{k} \frac{n_{1,h}}{n_h} \left( 1 - \frac{n_{1,h}}{n_h} \right) x_h \left( 1 - \frac{x_h}{n_h} \right)}
\]
Correcting for confounders with the CMH Test

Corresponding \( p \)-value \( \Psi(a, x, n_1, n_2) \)

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

$g([t_e, t_s])$ represents a meta-marker n-vector.

**Two steps:**

**Input:** Dataset of meta-markers $\mathcal{G} = \{\hat{g}, y, c\}$, desired FWER $\alpha$.

**Output:** Set of non-overlapping (conditionally) associated genomic regions $\mathcal{R}_{\text{sig, filt}} = \{[t_s, t_e] \mid p([t_s, t_e]) \leq \delta_{\text{tar}}\}$ and Tarone significance threshold $\delta_{\text{tar}}$.

1. $(\delta_{\text{tar}}, \mathcal{R}_T(\delta_{\text{tar}})) \leftarrow \text{get\_significant\_regions}(\mathcal{G}, \alpha)$

2. $\mathcal{R}_{\text{sig, filt}} \leftarrow \text{filter\_overlapping\_regions}(\mathcal{R}_T(\delta_{\text{tar}}))$

**Return:** $\mathcal{R}_{\text{sig, filt}}$
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 \([t_s, t_e] \in \mathcal{R}_{\text{cand}}, \) in increasing order of starting position \(t_s,\) and then length \(t_e - t_s:\)
1. Routine get_significant_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 \( \Phi(x_{[t_s,t_e]}) \leq \delta: \quad \to \text{Tarone’s testability criterion} \)
1. Routine get\_significant\_regions: interval processing

\[ \delta \leftarrow 1, I_T(\delta) \leftarrow \{\} \]

For all \([t_s, t_e] \in R_{cand}\), 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:

\[ \Phi(x_{[t_s, t_e]}) = \min_{a \in [0, x_{[t_s, t_e]}]} \Psi(a, x_{t_s, t_e}) \] is the minimum attainable \(p\)-value.
1. Routine get\_significant\_regions: interval processing

- $\delta \leftarrow 1$, $\mathcal{I}_T(\delta) \leftarrow \{\}$
- For all $[t_s, t_e] \in \mathcal{R}_{\text{cand}}$, 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
  - $\mathcal{I}_T(\delta) \leftarrow \mathcal{I}_T(\delta) \cup \{[t_s, t_e]\}$
1. Routine \texttt{get\_significant\_regions}: interval processing

\begin{itemize}
  \item $\delta \leftarrow 1$, $\mathcal{I}_T(\delta) \leftarrow \{\}$
  \item For all $[t_s, t_e] \in \mathcal{R}_{\text{cand}}$, in increasing order of starting position $t_s$ and then length $t_e - t_s$:
    \begin{itemize}
      \item Compute $x_{[t_s, t_e]}$
      \item If $\Phi(x_{[t_s, t_e]}) \leq \delta$: $\rightarrow$ Tarone’s testability criterion
        \begin{itemize}
          \item $\mathcal{I}_T(\delta) \leftarrow \mathcal{I}_T(\delta) \cup \{[t_s, t_e]\}$
          \item While $\delta |\mathcal{I}_T(\delta)| > \alpha$: $\rightarrow$ check FWER
        \end{itemize}
    \end{itemize}
\end{itemize}
1. Routine \texttt{get\_significant\_regions: interval processing}

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

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

- Compute \( x_{[t_s, t_e]} \)

- If \( \Phi(x_{[t_s, t_e]}) \leq \delta : \rightarrow \text{Tarone’s testability criterion} \)

  - \( \mathcal{I}_T(\delta) \leftarrow \mathcal{I}_T(\delta) \cup \{[t_s, t_e]\} \)

  - While \( \delta |\mathcal{I}_T(\delta)| > \alpha: \rightarrow \text{check FWER} \)
    - Decrease \( \delta \)
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 \(\Phi(x_{[[t_s, t_e]]}) \leq \delta\) → 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 FWER
    - Decrease \(\delta\)
    - 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 $[t_s, t_e] \in \mathcal{R}_{\text{cand}}$, 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 FWER
    - Decrease $\delta$
    - 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

- \( \delta \leftarrow 1, I_T(\delta) \leftarrow \{ \} \)

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
  - \( I_T(\delta) \leftarrow I_T(\delta) \cup \{[t_s, t_e]\} \)
  - While \( \delta |I_T(\delta)| > \alpha \): \( \rightarrow \) check FWER
    - Decrease \( \delta \)
    - Remove newly untestable intervals from \( I_T(\delta) \)

- 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 \texttt{get\_significant\_regions: interval processing}

- \( \delta \leftarrow 1, \ I_T(\delta) \leftarrow \{\} \)

For all \([t_s, t_e] \in \mathcal{R}_{\text{cand}}\), 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
  - \(I_T(\delta) \leftarrow I_T(\delta) \cup \{[t_s, t_e]\}\)
  - While \(\delta |I_T(\delta)| > \alpha\): \(\rightarrow\) check \(FWER\)
    - Decrease \(\delta\)
    - Remove newly untestable intervals from \(I_T(\delta)\)
- If \texttt{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 \(\mathcal{R}_{\text{cand}}\)

Return: \(\delta_{\text{tar}}\) and \(\mathcal{R}_T(\delta_{\text{tar}})\)
Pruning conditions for FAIS

- FAIS: $\chi^2$, Fisher exact test.
- The minimum attainable p-value is monotonically increasing as $x$ increases in $R_{cor} = [\max(n_1, n_2), n]$.
- The pruning condition is straightforward:
  \[ x_{[t_s,t_e]} \geq \max(n_1, n_2) \text{ and } \Phi(x_{[t_s,t_e]}) > \delta \]
1. Routine `get_significant_regions`: interval pruning

Increasing order of starting position, $t_s$

Increasing order of length, $t_e - t_s + 1$

Minimum attainable p-value of $x$

$\delta_k$

$\delta_j$
Pruning conditions for FastCMH

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

![Diagram showing selection of intervals with different $p$-values](image)

- **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

![Graph showing runtime simulation results for BRUTE (Bonferroni)]
FAIS: runtime simulation

![Graph showing runtime simulation for BRUTE (Bonferroni) and FAIS methods.](image)
FAIS: power simulation

![Graph showing the relationship between power and \( p_{cases} \).]
FAIS: power simulation

![Power Simulation Graph](image)
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)
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 phenotypes
- 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-$\chi^2$ 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>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Genomic interactions problem statement</td>
</tr>
<tr>
<td>2</td>
<td>Statistical testing and correction for confounders</td>
</tr>
<tr>
<td>3</td>
<td>Methods: Fast Automatic Interval Search (FAIS) and FastCMH algorithms</td>
</tr>
<tr>
<td>4</td>
<td>Results on plant and human datasets</td>
</tr>
<tr>
<td>5</td>
<td>Summary and outlook</td>
</tr>
</tbody>
</table>
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.
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


References II


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.

References IV


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
State of the art: dmGWAS

- 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

\[ z_i = \Phi^{-1}(1 - p_i) \]

\[ Z_{\text{current}} \]

At each gene in the PPI network:
1. Compute subgraph score
2. Find neighbors with distance smaller or equal to \( d \) (here \( d = 2 \))
3. For each neighbor: compute a tentative new subgraph score
4. Pick neighbor with maximal subgraph score:
   - If \( Z_{\text{new}} > Z_{\text{current}}(1 + r) \):
     - Add node (plus nodes in shortest path) to subgraph
5. Repeat (i) - (iv), until \( Z_{\text{new}} \leq Z_{\text{current}}(1 + r) \)
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 $Z_{\text{current}} = \frac{\sum z_i}{\sqrt{k}}$
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 \( Z_{new} \neq Z_{current}(1 + r) \)
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 \( Z_{\text{current}} = \frac{\sum z_i}{\sqrt{k}} \)
(ii) Find neighbors with distance smaller or equal to \( d \) (here \( d = 2 \))
(iii) For each neighbor: compute a tentative new subgraph score \( Z_{\text{new}} \)
(iv) Pick neighbor with maximal \( Z_{\text{new}} \): if \( Z_{\text{new}} \geq Z_{\text{current}}(1 + r) \): add node (plus nodes in shortest path) to subgraph
(v) Repeat (i) - (iv), until \( Z_{\text{new}} \not\geq Z_{\text{current}}(1 + r) \)
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 \( Z_{new} \neq Z_{current}(1 + r) \)
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 $Z_{\text{current}} = \frac{\sum z_i}{\sqrt{k}}$
- (ii) Find neighbors with distance smaller or equal to $d$ (here $d = 2$)
- (iii) For each neighbor: compute a tentative new subgraph score $Z_{\text{new}}$
- (iv) Pick neighbor with maximal $Z_{\text{new}}$: if $Z_{\text{new}} \geq Z_{\text{current}}(1 + r)$: add node (plus nodes in shortest path) to subgraph
- (v) Repeat (i) - (iv), until $Z_{\text{new}} \not\geq Z_{\text{current}}(1 + r)$
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 \( Z_{\text{current}} = \frac{\sum z_i}{\sqrt{k}} \)

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

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

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

(v) Repeat (i) - (iv), until \( Z_{\text{new}} \neq Z_{\text{current}}(1 + r) \)
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 \( Z_{\text{current}} = \frac{\sum z_i}{\sqrt{k}} \)

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

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

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

   (v) Repeat (i) - (iv), until \( Z_{\text{new}} < Z_{\text{current}}(1 + r) \)
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 $Z_{new} < Z_{current}(1 + r)$
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 $Z_{\text{current}} = \frac{\sum z_i}{\sqrt{k}}$
- (ii) Find neighbors with distance smaller or equal to $d$ (here $d = 2$)
- (iii) For each neighbor: compute a tentative new subgraph score $Z_{\text{new}}$
- (iv) Pick neighbor with maximal $Z_{\text{new}}$: if $Z_{\text{new}} \geq Z_{\text{current}}(1 + r)$: add node (plus nodes in shortest path) to subgraph
- (v) Repeat (i) - (iv), until $Z_{\text{new}} \not\geq Z_{\text{current}}(1 + r)$
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 \( Z_{\text{current}} = \frac{\sum z_i}{\sqrt{k}} \)
   
   (ii) Find neighbors with distance smaller or equal to \( d \) (here \( d = 2 \))
   
   (iii) For each neighbor: compute a tentative new subgraph score \( Z_{\text{new}} \)
   
   (iv) Pick neighbor with maximal \( Z_{\text{new}} \): if \( Z_{\text{new}} \geq Z_{\text{current}}(1 + r) \): add node (plus nodes in shortest path) to subgraph
   
   (v) Repeat (i) - (iv), until \( Z_{\text{new}} \neq Z_{\text{current}}(1 + r) \)
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 subgraph 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
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 → multiple hypothesis testing problem

Control family-wise error rate (FWER)

$$\text{FWER} = \Pr (\text{FP})$$

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

Bonferroni correction:

$$= \frac{\alpha}{\text{number of tests}}$$

Minimum attainable p-value: subgraphs that are not testable at a significance threshold cannot become false positives, thus no correction is required for those

Tarone correction:

$$= \frac{\alpha}{\text{number of testable subgraphs}}$$
Short Revision: Tarone Trick

- Conducting high number of statistical significance test → multiple hypothesis testing problem
- Control family-wise error rate (FWER)
  \[ \text{FWER} = \Pr (FP \geq 1) \leq \alpha \]  
- Need to find the maximum significance threshold \( \delta \) such that Eq. 1 holds
Conducting high number of statistical significance test → multiple hypothesis testing problem

Control family-wise error rate (FWER)

\[
\text{FWER} = \Pr (FP \geq 1) \leq \alpha
\] (1)

Need to find the maximum significance threshold \( \delta \) 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)

\[
\text{FWER} = \Pr (FP \geq 1) \leq \alpha
\]  

(1)

- Need to find the maximum significance threshold $\delta$ 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 $\delta$ 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

$L = \text{number of genes}$

Associated subgraph $g$

Random variable

$f(s_1[g]) = 1$

$f(s_2[g]) = 1$

$f(s_3[g]) = 1$

$f(s_4[g]) = 0$

$f(s_5[g]) = 0$

$f(s_6[g]) = 0$
### Tarone method for graphs: contingency tables

#### Variables

<table>
<thead>
<tr>
<th>Variables</th>
<th>$f(s[g]) = 1$</th>
<th>$f(s[g]) = 0$</th>
<th>Row totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y = \text{case}$</td>
<td>$\alpha_g$</td>
<td>$n_1 - \alpha_g$</td>
<td>$n_1$</td>
</tr>
<tr>
<td>$y = \text{control}$</td>
<td>$x_g - \alpha_g$</td>
<td>$n_2 - (x_g - \alpha_g)$</td>
<td>$n_2$</td>
</tr>
<tr>
<td>Col. totals</td>
<td>$x_g$</td>
<td>$N - x$</td>
<td>$n$</td>
</tr>
</tbody>
</table>

#### Random variable

- $f(s_1[g]) = 1$
- $f(s_2[g]) = 1$
- $f(s_3[g]) = 1$
- $f(s_4[g]) = 0$
- $f(s_5[g]) = 0$
- $f(s_6[g]) = 0$

$L = \text{number of genes}$

Associated subgraph $g$
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 $x_g$

<table>
<thead>
<tr>
<th>current subgraph</th>
<th>Random variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_1$ cases</td>
<td>$n_2$ controls</td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>$f(s_1[g]) = 1$</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>$f(s_2[g]) = 1$</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>$f(s_3[g]) = 0$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>new subgraph</th>
<th>Random variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_1$ cases</td>
<td>$n_2$ controls</td>
</tr>
<tr>
<td>0 1 0 1 0</td>
<td>$f(s_1[g]) = 1$</td>
</tr>
<tr>
<td>0 0 0 1 0</td>
<td>$f(s_2[g]) = 1$</td>
</tr>
<tr>
<td>0 0 0 0 0</td>
<td>$f(s_3[g]) = 0$</td>
</tr>
</tbody>
</table>

![Graph showing subgraph growth and pruning with random variable $x_g$](image-url)
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

- Current subgraph
  - $f(s_6[g]) = 0$
  - $f(s_5[g]) = 0$
  - $f(s_4[g]) = 0$
  - $f(s_3[g]) = 0$
  - $f(s_2[g]) = 1$
  - $f(s_1[g]) = 1$

- Random variable $x_g = 2$

- New subgraph
  - $f(s_6[g]) = 0$
  - $f(s_5[g]) = 0$
  - $f(s_4[g]) = 0$
  - $f(s_3[g]) = 1$
  - $f(s_2[g]) = 1$
  - $f(s_1[g]) = 1$

- Random variable $x_g = 3$
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 $\delta$ s.t. FWER criterion is fulfilled
      $$\delta \times |\text{testable subgraphs}| \leq \alpha$$

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 $\delta$ s.t. FWER criterion is fulfilled
      $$\delta \times |\text{testable subgraphs}| \leq \alpha$$
3. Add next gene to subgraph and return to step 1
3. Subgraph is non-testable (i.e. $\Psi(x_g) > \delta$):
   1. $x_g < n - \sigma_I$: Add next gene to subgraph and return to step 1
   2. $x_g > n - \sigma_I$: Stop growing subgraph
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
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 $\text{mean}_{\text{cases}}$ and controls $\text{mean}_{\text{controls}}$ in training set.

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.

Anja Gumpinger | Significant subgraph search in protein-protein interaction networks
COST Antwerp | 27 April 2016 | 21 / 38
1. For each gene, compute the mean of cases \( \text{mean}_{\text{cases}} \) and controls \( \text{mean}_{\text{controls}} \) 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

\[
f_{\text{gene}_A}(\text{sample}) = \begin{cases} 1 & \text{if } \text{SNP}_A = 2 \\ 0 & \text{else} \end{cases}
\]
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

\[
f_{\text{gene}_A}(\text{sample}) = \begin{cases} 
1 & \text{if } \text{SNP}_A = 2 \\
0 & \text{else} 
\end{cases}
\]
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

\[
f_{\text{gene}_A} (\text{sample}) = \begin{cases} 1 & \text{if SNP}_A = 2 \\ 0 & \text{else} \end{cases}
\]

SNPs in gene_A

\[
\begin{array}{cccccccccccc}
0 & 2 & 0 & 1 & 0 & 1 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 2 & 0 & 1 & 0 & 1 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 2 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 2 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 2 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 2 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
\end{array}
\]
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

\[
f_{\text{gene}_A}^{(\text{sample})} = \begin{cases} 
1 & \text{if } \text{SNP}_A = 2 \\
0 & \text{else} 
\end{cases}
\]
**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

#### Table: SNPs in geneA

<table>
<thead>
<tr>
<th></th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>f_{geneA}</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>cases</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>controls</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Figure: Binary Representation of Gene A

```
controls: 0 2 0 1 0 1 2 0 1 1 0 0 0 0 2 0 0 1 0 0 1 0 1 0
0 2 0 1 0 1 2 0 1 1 0 0 0 0 2 0 0 1 0 0 1 0 1 0
0 2 0 1 0 1 1 0 1 0 0 0 0 2 0 0 1 0 0 1 0 1 0 1 0
0 2 0 1 0 1 0 0 1 1 0 0 0 2 0 0 1 0 0 1 0 1 0 1 0
0 2 0 1 0 1 0 0 1 1 0 0 0 1 0 0 2 0 0 1 0 0 1 0 1 0
0 2 0 1 0 1 0 0 1 1 0 0 0 2 0 0 2 0 0 1 0 0 1 0 1 0

binary geneA: 1 1 0 0
```
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].

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].

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

![Graph showing the number of significant subgraphs and runtime for different sample sizes and subgraph sizes.](image)

Legend:
- 50 samples, 5 genes in subgraph
- 50 samples, 10 genes in subgraph
- 100 samples, 5 genes in subgraph
- 100 samples, 10 genes in subgraph
- 200 samples, 5 genes in subgraph
- 200 samples, 10 genes in subgraph
- 500 samples, 5 genes in subgraph
- 500 samples, 10 genes in subgraph

Risk ratio on the x-axis, and number of significant subgraphs and runtime on the y-axis.
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

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

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

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

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

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


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

Thank you for your patience and attention