Set Covering Maching for SNPs discovery

Vivien Goepp

March, 23rd 2020
CBIO Meeting

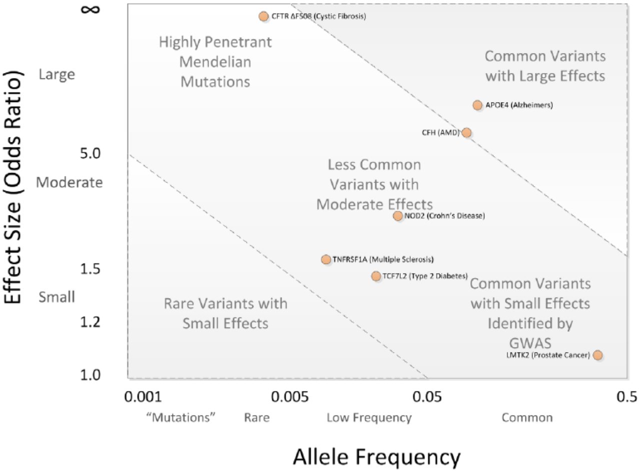
1. Short reminder on GWAS

Goal of Genome-wise association studies (GWAS)

- **Goal**: Discover gene mutations *linked* to a disease.
- GWAS will not provide: causality relations or biological understanding of a disease.
- Applications to common diseases:
  - Inflammatory Bowel Diseases
  - Auto-immune diseases
  - Metabolic diseases (T2 diabetes, obesity, BMI)
  - Multiple sclerosis
  - Cancer

The CD/CV hypothesis

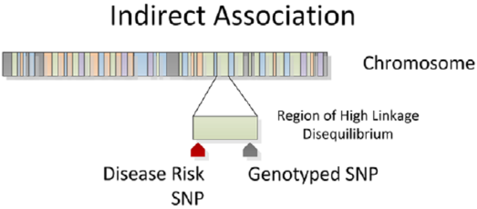
- *Common diseases* are partly caused by *common variants*
- Consequence: each mutation can only have a *small effect*



Source: Bush et al (2012)

Linkage disequilibrium

Linkage disequilibrium (LD): correlation between close-by alleles on the genome



Source: Bush et al (2012)

Linkage disequilibrium

Idea of using single nucleotide polymorphisms (SNPs):

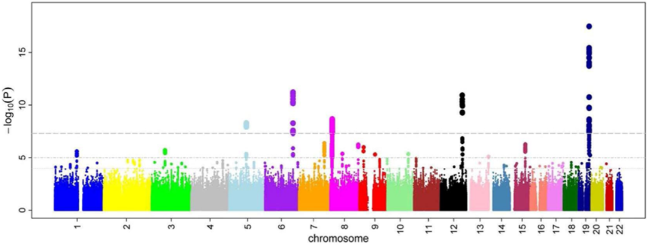
- SNP: Single nucleotide polymorphism
- There are many high-LD blocks on the genome
- We can use SNPs as markers of an LD block

GWAS

- Gather $n \sim 10^3$ individuals
- Observe the phenotype:
  - quantitative (BMI, cholesterol, height)
  - or qualitative (case-control for common disease).
- Observe the genotype of $p \sim 10^6$ SNPs

GWAS

- Gather $n \sim 10^3$ individuals
- Observe the phenotype:
  - quantitative (BMI, cholesterol, height)
  - or qualitative (case-control for common disease).
- Observe the genotype of $p \sim 10^6$ SNPs



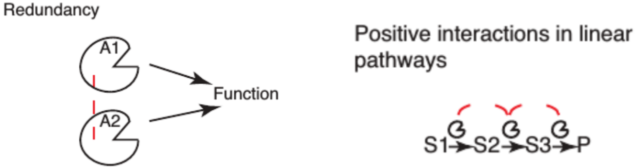
Source: Ikram et al (2010)

Epistasis and interactions between SNPs

- Epistasis: “The masking of the effects of one variant by another” (Bateson 1909).

Epistasis and interactions between SNPs

- Epistasis: “The masking of the effects of one variant by another” (Bateson 1909).
- Examples

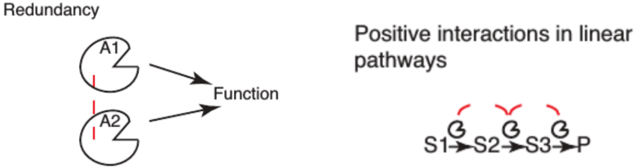


Examples of biological causes for epistasis (Source: Lehner 2011)

Epistasis and interactions between SNPs

- Epistasis: “The masking of the effects of one variant by another” (Bateson 1909).

- Examples



Examples of biological causes for epistasis (Source: Lehner 2011)

- The genetic mutations are in interaction

- Need to consider SNPs *jointly*

The statistical problem

We do feature selection with:

- $\frac{p}{n} \sim 1000$

The statistical problem

We do feature selection with:

- $\frac{p}{n} \sim 1000$
- Correlation between features (SNPs)

The statistical problem

We do feature selection with:

- $\frac{p}{n} \sim 1000$
- Correlation between features (SNPs)
- Interaction between features

The statistical problem

We do feature selection with:

- $\frac{p}{n} \sim 1000$
- Correlation between features (SNPs)
- Interaction between features

Very small statistical power

2. Set covering machines for SNPs discovery

Encoding of SNP alleles

Each SNP has value $\in \{\mathrm{aa}, \mathrm{aA}, \mathrm{AA}\}$.

Genotype	Dominant	Recessive	Allelic dosage	One-hot
aa	0	0	0	100
aA	1	0	1	010
AA	1	1	2	001

Encoding of SNP alleles

Each SNP has value $\in \{\mathrm{aa}, \mathrm{aA}, \mathrm{AA}\}$.

Genotype	Dominant	Recessive	Allelic dosage	One-hot
aa	0	0	0	100
aA	1	0	1	010
AA	1	1	2	001

We will use one-hot encoding → binary features

Setting of the set covering machine (SCM)

- $y_i \in \{0, 1\}$ (case/control GWAS)
- $x_{i,j} \in \{0, 1\}$ (one-hot encoding)
- $p \gg n$ with true model assumed to be *very sparse*

Setting of the set covering machine (SCM)

- $y_i \in \{0, 1\}$ (case/control GWAS)
- $x_{i,j} \in \{0, 1\}$ (one-hot encoding)
- $p \gg n$ with true model assumed to be *very sparse*

- SCM learns a boolean function of the features:

$$f(\mathbf{x}) = \bigwedge_{j \in \mathcal{R}} h_j(\mathbf{x}),$$

  where $\mathcal{R} \in$ is the set of rules to learn.

- Here a rules h_j is the one-hot encoding of a SNP.

Setting of the set covering machine (SCM)

- $y_i \in \{0, 1\}$ (case/control GWAS)
- $x_{i,j} \in \{0, 1\}$ (one-hot encoding)
- $p \gg n$ with true model assumed to be *very sparse*

- SCM learns a boolean function of the features:

$$f(\mathbf{x}) = \bigwedge_{j \in \mathcal{R}} h_j(\mathbf{x}),$$

where $\mathcal{R} \in$ is the set of rules to learn.

- Here a rules h_j is the one-hot encoding of a SNP.
- SCM only learns a conjunction of SNPs to explain the phenotype.

The set covering problem (1/2)

Haussler algorithm:

- Assume there is a combination of features that perfectly classifies the dataset: $\mathbf{y} = \bigwedge_{j \in \mathcal{R}} h_j$
- How to find the sparsest possible combination of features?
- Only consider rules that correctly classify *all* positive examples ($y_i = 1$).

The set covering problem (2/2)

- Example: what conjunction of h_js equals $\mathbf{y}$?

	$\mathbf{y}$	h_1	h_2	h_3	h_4
$\mathcal{N}$ (negative examples)	0	0	0	1	1
	0	1	0	0	1
	0	1	1	0	0
	0	1	1	1	0
$\mathcal{P}$ (positive examples)	1	1	1	1	1
	1	1	1	1	1

The set covering problem (2/2)

- Example: what conjunction of h_js equals $\mathbf{y}$?

	$\mathbf{y}$	h_1	h_2	h_3	h_4	$h_2 \wedge h_3$
$\mathcal{N}$	0	0	0	1	1	0
	0	1	0	0	1	0
	0	1	1	0	0	0
	0	1	1	1	0	0
$\mathcal{P}$	1	1	1	1	1	1
	1	1	1	1	1	1

The set covering problem (2/2)

- Example: what conjunction of h_js equals $\mathbf{y}$?

	$\mathbf{y}$	h_1	h_2	h_3	h_4	$h_2 \wedge h_3$
$\mathcal{N}$	0	0	0	1	1	0
	0	1	0	0	1	0
	0	1	1	0	0	0
	0	1	1	1	0	0
$\mathcal{P}$	1	1	1	1	1	1
	1	1	1	1	1	1

- Smallest number of sets $\{1\}, \{1,2\}, \{2,3\}, \{3,4\}$ whose union is $\{1,2,3,4\}$.

The set covering problem (2/2)

- Example: what conjunction of h_js equals $\mathbf{y}$?

	$\mathbf{y}$	h_1	h_2	h_3	h_4	$h_2 \wedge h_3$
$\mathcal{N}$	0	0	0	1	1	0
	0	1	0	0	1	0
	0	1	1	0	0	0
	0	1	1	1	0	0
$\mathcal{P}$	1	1	1	1	1	1
	1	1	1	1	1	1

- Smallest number of sets $\{1\}, \{1,2\}, \{2,3\}, \{3,4\}$ whose union is $\{1,2,3,4\}$.
- This is the set cover problem (NP hard)

The set covering problem (2/2)

- Example: what conjunction of h_js equals $\mathbf{y}$?

	$\mathbf{y}$	h_1	h_2	h_3	h_4	$h_2 \wedge h_3$
$\mathcal{N}$	0	0	0	1	1	0
	0	1	0	0	1	0
	0	1	1	0	0	0
	0	1	1	1	0	0
$\mathcal{P}$	1	1	1	1	1	1
	1	1	1	1	1	1

- Smallest number of sets $\{1\}, \{1,2\}, \{2,3\}, \{3,4\}$ whose union is $\{1,2,3,4\}$.
- This is the set cover problem (NP hard)
- We use a greedy approach

Set covering machine

We choose the rule with maximum usefulness:

$$U_h = |\mathcal{A}_h| - q|\mathcal{B}_h|,$$

$\mathcal{A}_h$: negative examples correctly classified

$\mathcal{B}_h$: positive examples uncorrectly classified

Set covering machine

We choose the rule with maximum usefulness:

$$U_h = |\mathcal{A}_h| - q|\mathcal{B}_h|,$$

$\mathcal{A}_h$: negative examples correctly classified

$\mathcal{B}_h$: positive examples uncorrectly classified

- We allow errors on positive examples
- q controls this error

Set covering machine: example

Example (with $q = 1$):

	$\mathbf{y}$	h_1	h_2	h_3	h_4
$\mathcal{N}$	0	0	0	1	1
	0	1	0	0	1
	0	1	1	0	0
	0	1	1	1	0
$\mathcal{P}$	1	1	0	1	1
	1	1	0	0	1

Set covering machine: example

Example (with $q = 1$):

	$\mathbf{y}$	h_1	h_2	h_3	h_4
$\mathcal{N}$	0	0	0	1	1
	0	1	0	0	1
	0	1	1	0	0
	0	1	1	1	0
$\mathcal{P}$	1	1	0	1	1
	1	1	0	0	1
	$\lvert\mathcal{A}_h\rvert$	1	2	2	2

Set covering machine: example

Example (with $q = 1$):

	$\mathbf{y}$	h_1	h_2	h_3	h_4
$\mathcal{N}$	0	0	0	1	1
	0	1	0	0	1
	0	1	1	0	0
	0	1	1	1	0
$\mathcal{P}$	1	1	0	1	1
	1	1	0	0	1
	$\lvert\mathcal{A}_h\rvert$	1	2	2	2
	$\lvert\mathcal{B}_h\rvert$	0	2	1	0

Set covering machine: example

Example (with $q = 1$):

	$\mathbf{y}$	h_1	h_2	h_3	h_4
$\mathcal{N}$	0	0	0	1	1
	0	1	0	0	1
	0	1	1	0	0
	0	1	1	1	0
$\mathcal{P}$	1	1	0	1	1
	1	1	0	0	1
	$\lvert\mathcal{A}_h\rvert$	1	2	2	2
	$\lvert\mathcal{B}_h\rvert$	0	2	1	0
	$\mathcal{U}_h$	1	0	1	2

Set covering machine: example

Example (with $q = 1$):

	$\mathbf{y}$	h_1	h_2	h_3	h_4
$\mathcal{N}$	0	0	0	1	1
	0	1	0	0	1
	0	1	1	0	0
	0	1	1	1	0
$\mathcal{P}$	1	1	0	1	1
	1	1	0	0	1
	$\lvert\mathcal{A}_h\rvert$	1	2	2	2
	$\lvert\mathcal{B}_h\rvert$	0	2	1	0
	$\mathcal{U}_h$	1	0	1	2

- $\mathcal{R} \leftarrow \{h_4\}$

Set covering machine: example

Example (with $q = 1$):

	$\mathbf{y}$	h_1	h_2	h_3	h_4
$\mathcal{N}$	0	0	0	1	1
	0	1	0	0	1
	0	1	1	0	0
	0	1	1	1	0
$\mathcal{P}$	1	1	0	1	1
	1	1	0	0	1
	$\lvert\mathcal{A}_h\rvert$	1	2	2	2
	$\lvert\mathcal{B}_h\rvert$	0	2	1	0
	$\mathcal{U}_h$	1	0	1	2

- $\mathcal{R} \leftarrow \{h_4\}$
- $\mathcal{N} \leftarrow \mathcal{N} \setminus \mathcal{A}_{h_4}$

Set covering machine: example

Example (with $q = 1$):

	$\mathbf{y}$	h_1	h_2	h_3	h_4
$\mathcal{N}$	0	0	0	1	1
	0	1	0	0	1
	0	1	1	0	0
	0	1	1	1	0
$\mathcal{P}$	1	1	0	1	1
	1	1	0	0	1
	$\lvert\mathcal{A}_h\rvert$	1	2	2	2
	$\lvert\mathcal{B}_h\rvert$	0	2	1	0
	$\mathcal{U}_h$	1	0	1	2

- $\mathcal{R} \leftarrow \{h_4\}$
- $\mathcal{N} \leftarrow \mathcal{N} \setminus \mathcal{A}_{h_4}$
- $\mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{B}_{h_4}$

Set covering machine: example (2)

	$\mathbf{y}$	h_1	h_2	h_3
$\mathcal{N}$	0	0	0	1
	0	1	0	0
$\mathcal{P}$	1	1	0	1
	1	1	0	0

Set covering machine: example (2)

	$\mathbf{y}$	h_1	h_2	h_3
$\mathcal{N}$	0	0	0	1
	0	1	0	0
$\mathcal{P}$	1	1	0	1
	1	1	0	0
	$\|\mathcal{A}_h\|$	1	2	1

Set covering machine: example (2)

	$\mathbf{y}$	h_1	h_2	h_3
$\mathcal{N}$	0	0	0	1
	0	1	0	0
$\mathcal{P}$	1	1	0	1
	1	1	0	0
	$\lvert\mathcal{A}_h\rvert$	1	2	1
	$\lvert\mathcal{B}_h\rvert$	0	2	1

Set covering machine: example (2)

	$\mathbf{y}$	h_1	h_2	h_3
$\mathcal{N}$	0	0	0	1
	0	1	0	0
$\mathcal{P}$	1	1	0	1
	1	1	0	0
	$\lvert\mathcal{A}_h\rvert$	1	2	1
	$\lvert\mathcal{B}_h\rvert$	0	2	1
	$\mathcal{U}_h$	1	0	1

Stopping criteria

- We have finished the job: $\mathcal{N} = \emptyset$
- Early stopping: $|\mathcal{R}| \geq s$ with parameter $s \geq 1$
- There remains only useless rules: $|\mathcal{A}_h| = |\mathcal{B}_h| = 0$

Computation time

- Each greedy step is fast to compute:
  - Let $\mathcal{I}_{\mathcal{N}}$ be the (current) indices of the negative examples.
  - $|\mathcal{A}_h| = |\mathcal{I}_{\mathcal{N}}| - \sum_{i \in \mathcal{I}_{\mathcal{N}}} x_{i,j}$ if h is the presence rule of feature j
  - Similar for $|\mathcal{B}_h|$.

Computation time

- Each greedy step is fast to compute:
  - Let $\mathcal{I}_\mathcal{N}$ be the (current) indices of the negative examples.
  - $|\mathcal{A}_h| = |\mathcal{I}_\mathcal{N}| - \sum_{i \in \mathcal{I}_\mathcal{N}} x_{i,j}$ if h is the presence rule of feature j
  - Similar for $|\mathcal{B}_h|$.
- Overall complexity $\mathcal{O}\left(|\mathcal{R}|ns\right)$

Computation time

- Each greedy step is fast to compute:
    - Let $\mathcal{I}_{\mathcal{N}}$ be the (current) indices of the negative examples.
    - $|\mathcal{A}_h| = |\mathcal{I}_{\mathcal{N}}| - \sum_{i \in \mathcal{I}_{\mathcal{N}}} x_{i,j}$ if h is the presence rule of feature j
    - Similar for $|\mathcal{B}_h|$.
- Overall complexity $\mathcal{O}\left(|\mathcal{R}|ns\right)$
- Limited memory usage

Upper bound on the risk (1/2)

SCM is a *sample compression algorithm*

- Given a model f learnt by an SCM, there exist
  - a set of individuals $\mathcal{Z} \in \{1, \cdots, n\}$
  - a message string σ containing additional information,

  such that h can be reconstructed from $\mathcal{Z}$.

Upper bound on the risk (1/2)

SCM is a *sample compression algorithm*

- Given a model f learnt by an SCM, there exist
  - a set of individuals $\mathcal{Z} \in \{1, \cdots, n\}$
  - a message string σ containing additional information,

  such that h can be reconstructed from $\mathcal{Z}$.

- Then there exists a bound on the risk

$$R(h) = \mathbb{E}_{(\mathbf{x},y)\sim D}\left[\mathbb{1}_{f(\mathbf{x})\neq y}\right].$$

Upper bound on the risk (1/2)

SCM is a *sample compression algorithm*

- Given a model f learnt by an SCM, there exist
  - a set of individuals $\mathcal{Z} \in \{1, \cdots, n\}$
  - a message string σ containing additional information,

  such that h can be reconstructed from $\mathcal{Z}$.

- Then there exists a bound on the risk

$$R(h) = \mathbb{E}_{(\mathbf{x},y)\sim D}\left[\mathbb{1}_{f(\mathbf{x})\neq y}\right].$$

- Marchand and Sokolova (2006) established that:

$$\mathbb{P}\left(\forall S \sim D, \forall h, R(h) \leq \varepsilon(h, S, \delta)\right) \geq 1 - \delta$$

  - ε depends on $\mathcal{Z}$ and the of classif. errors made on $S \setminus \mathcal{Z}$.
  - ε does not depend on p.

Upper bound on the risk (2/2)

Consequences:

- The bound does not depend on p: theoretical performance guarantee
- It can be used for hyperparameter selection (Marchand et Shawe-Taylor, 2002).

Conclusion

- Set covering machine
  - learns a boolean conjunction of SNPs
  - runs fast
  - does not suffer from $p \gg n$
- However there are other issues:
  - Many SNPs can have same $\mathcal{U}_h$: which one to choose?
  - Only conjunctions of SNPs

Thank you

References

- SCM: Marchand, M. and Shawe-Taylor, J.,*The set covering machine*, JMLR, 2002
- Risk bound for SCM: Marchand, M and Sokolova, M., *Learning with Decision Lists of Data-Dependent Features*, JMLR, 2005
- GWAS: Bush, W, Moore, J., Lewitter, F., and Kann, M., *Chapter 11: Genome-Wide Association Studies*, Plos Comp. Biol., 2012
- Epistasis: Lehner, B., *Molecular mechanisms of epistasis within and between genes*, Trends in Gen., 2011

Appendix: Sample compression bound

$$\varepsilon(h, S, \delta) = 1 - \exp\left(\frac{-1}{n - |\mathcal{Z}| - r}\left[\log\binom{m}{|\mathcal{Z}|} + \log\binom{m - |\mathcal{Z}|}{r} + |h|\log(2\mathcal{N}(\mathcal{Z})) + \log\Omega\right]\right)$$

- with $\Omega = \frac{\pi^6(|h|+1)^2(r+1)^2(|\mathcal{Z}|+1)^2}{216\delta}$,
- where r is the number of classif. errors on $S \setminus \mathcal{Z}$.