GWAS multi-traits à partir de données groupées

Christophe Ambroise, Amin Madoui, Bargob Kakothy

Université d'Evry/CNRS/INRAe

Motivations

Tenebrio Molitor

The yellow mealworm (Tenebrio molitor) is a beetle that is particularly fond of cereal flours.

- It is found all over the world.
- It is capable of living in very dry stored foods.
- It is capable of eating certain forms of expanded polystyrene.
- It can live for up to 6 months.

Food

Mealworms are used

- In animal feed, especially to feed various insectivorous species such as birds, reptiles, and fish.
- In human food: they have a sweet, nutty flavor and can be used as substitutes in various dishes, including pastries and savory pies, or eaten fried.

Food

Mealworms are used

- In animal feed, especially to feed various insectivorous species such as birds, reptiles, and fish.
- In human food: they have a sweet, nutty flavor and can be used as substitutes in various dishes, including pastries and savory pies, or eaten fried.

Project YNFABRE i

- YNSECT Launches YNFABRE, the World's First Industrial Program Dedicated to Beetle Genetics
- Ynsect has identified a strain of mealworms through selection that allows for 25% faster growth than the original strain
- 30,000 tons of Molitor larva flour, rich in proteins,
- 60,000 tons of fertilizer, derived from the excretions of all these small animals
- **Improvement of the yield** of insect farms

Project YNFABRE ii

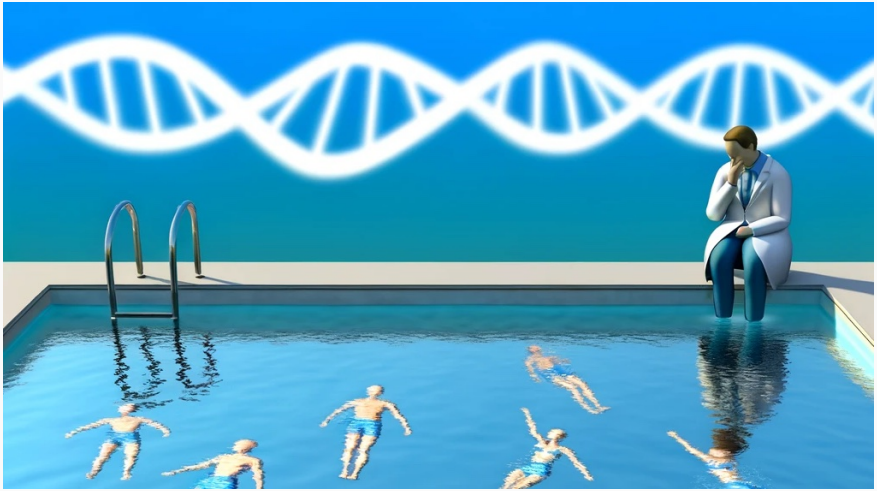
Politicians during the inauguration of the construction site for the vertical farm of Ÿnsect near Amiens (Libérarion 2021)

Project YNFABRE iii

Sequencing of 4,000 genotypes and haplotyping of Tenebrio molitor by high-throughput sequencing and Axiom biochips

- **GWAS**: Identification of phenotype-predictive loci by Genome Wide Association Studies
- **Genomic Estimation Breeding Value**: Construction and evaluation of a model based on pool-seq data and phenotypic data

Pool Data in Genomics



pool Data in Genomics

Classical data

SNP 1	SNP 2	...	SNP g
0	0	...	1
2	0	...	1
1	1	...	0
2	2	...	0
...	...	...	...
2	0	...	2

Table 1: Genotype

Phenotype 1	Phenotype 2	...	Phenotype d
100	5	...	10
20	10	...	10
10	10	...	50
200	5	...	40
...	...	...	...
200	2	...	60

Table 2: Phenotype

Pool data

Group	Freq. 1	Freq. 2	...	Freq. g
1	0.2	0.1	...	0.95
1	0.2	0.1	...	0.95
2	0.8	0.76	...	0.07
2	0.8	0.76	...	0.07
...	...	...	...	
K	0.3	0.1	...	0.21

Table 3: Group allele frequencies

Phenotype 1	Phenotype 2	...	Phenotype d
100	5	...	10
20	10	...	10
10	10	...	50
200	5	...	40
...	...	...	...
200	2	...	60

Table 4: Phenotype

Pool GWAS / pool Breeding value

Principle

- each individual phenotype (size, number of eggs...) is measured
- the genetic is processed by pool: only allele frequencies are available for each marker
- Goal: Adapt Classical analysis for pools of individuals:
  - GWAS
  - breeding value

Characteristics

- Cost effective
- Existing methodologies for pool-GWAS are limited and can only handle single phenotypes (to my knowledge)

Model

Missing Data

In contrast to classical GWAS approaches, pool-GWAS can be seen as **a missing data** challenge where

- Observed data: average genotypes of the pool of the individual,
- Missing data: complete individual genotypes.

K-Pools

Consider K pools constructed according to the procedure described above:

- $p_{k,j}$ is the minor allele frequency (MAF) of SNP j in pool k.
- it estimates of the true proportion of minor alleles in pool k.

Notations

Matrix of the p MAFs in the K pools

$F = \{p_{k,j}\}_{k,j} \in [0,1]^{K \times p}$

Cluster matrix

$C = \{C_{ik}\}_{ik} \in \{0,1\}^{n \times K}$ the pool membership binary matrix where $C_{ik} = \mathbb{I}_{\text{(ind. i comes from pool k)}}$. We will also denote by $c(i)$ the pool of individual i.

Design matrix

$$X = CF$$

Matrix of phenotypes

$$Y \in \mathbb{R}^{n \times d}$$

Unobserved genotype of individual i at locus j

$$G_{ij} \sim Bin(2, p_{c(i),j})$$

which we approximate by a Gaussian distribution !

$$G_{ij} \approx X_{ij} + \underbrace{\sqrt{2p_{c(i),j}(1-p_{c(i),j})}\epsilon_{ij}}_{Z_{ij}}$$

where

$$\epsilon_{ij} \sim \mathcal{N}(0,1)$$

and X_{ij} is the average allele frequency of the pool $c(i)$.

A mixed linear model with missing data

$$y_i = \mu + Bg_i + e_i,$$

where

- μ is a fixed part,
- B is a d by p matrix of weight parameters
- the genetic contribution $g_i = (G_{ij})_{j=1\cdots p}^T$,
- e_i is random vector modeling the environmental contribution to the phenotype.

A mixed linear model with missing data

$$y_i = \mu + B\underbrace{(x_i + z_i)}_{g_i} + e_i.$$

where

- x_i is fixed (minor allele frequencies)
- $z_i \sim \mathcal{N}_p(0, V_g)$
- $e_i \sim \mathcal{N}_d(0, V_e)$

Relation to Factor analysis

If we disregard the average frequency, the model mentioned above closely resembles classical factor analysis (Murphy 2012):

$$y_i = \mu + Bz_i + e_i$$

For $y = \mu + B\underbrace{(x+z)}_{g} + e$

- Marginal distribution : $y \sim \mathcal{N}_d(\mu + Bx, \Sigma_{yy} = BV_gB^T + V_e)$
- Posterior distribution: $z|y \sim \mathcal{N}_p(\mu_{z|y}, \Sigma_{z|y})$

where

- $\Sigma_{z|y} = \Sigma_{zz} - \Sigma_{zy}\Sigma_{yy}^{-1}\Sigma_{yz} = V_g - B^T(BV_gB^T + V_e)^{-1}B = S$
- $\mu_{z|y} = \Sigma_{z|y}\Sigma_{yy}^{-1}(y - \mu - Bx)$

EM algorithm for pool GWAS

- μ, B and V_e are estimated using a simple Expectation Maximisation algorithm (Dempster, Laird, and Rubin 1977)
- EM algorithm slow to converge in such a high dimensional problem
- Tackling the problem from a machine learning perspective using asymptotic arguments

A faster alternative approach: Noise injection

The expression of genotypes as the sum of observed frequencies and a missing part can also be related to noise injection (Grandvalet, Canu, and Boucheron 1997), (Grandvalet 2000):

- acts as a form of regularization,
- prevents overfitting
- improves generalization and robustness

Noise injection can be seen as a kind of data aumentation where the individual genotypes from their MAF are regenerated.

Theoritical risk

We consider the classical risk assuming $e_i \perp\!\!\!\perp z_i$

$$
\begin{aligned}
J &= -\mathbb{E}_{e_i,z_i}\left[\log f(e_i, z_i)\right] \\
&= -\mathbb{E}_{e_i,z_i}\left[\log f(e_i|z_i)\right] - \underbrace{\mathbb{E}_{e_i,z_i}\left[\log f(z_i)\right]}_{-\frac{1}{2}(\log\det(V_g)+\log 2\pi)} \\
&= -E_{e_i,z_i}\left[-\frac{\|y_i - Bx_i + Bz_i\|^2_{V_e^{-1}}}{2}\right] + cst, \\
&= E_{e_i,z_i}\left[-\frac{\|y_i - Bx_i\|^2_{V_e^{-1}}}{2}\right] + \frac{\mathbb{E}\left[z_i^T B^T V_e^{-1} B z_i\right]}{2} + cst, \\
&= E_{e_i,z_i}\left[-\frac{\|y_i - Bx_i\|^2_{V_e^{-1}}}{2}\right] + \frac{trace\left[V_g B^T V_e^{-1} B\right]}{2} + cst,
\end{aligned}
$$

Empirical risk

$$\begin{aligned}
\hat{J}(B) &= \frac{1}{n}\sum_i \|y_i - Bx_i\|^2_{V_e^{-1}} + trace(V_e^{-1}BV_gB^T), \\
&= \frac{1}{n}trace\left(\sum_i (y_i - Bx_i)^T V_e^{-1}(y_i - Bx_i)\right) + trace(V_e^{-1}BV_gB^T) \\
&= \frac{1}{n}trace\Big(V_e^{-1}(Y^TY + BX^TXB^T - 2Y^TXB^T\Big) + trace(V_e^{-1}BV_g
\end{aligned}$$

where Y is n by d matrix of phenotype such that $Y = (y_{ij})_{ij}$.

Minimization

The gradient writes

$$\nabla_B \hat{J}(B) = \frac{2}{n} V_e^{-1} B X^T X - \frac{2}{n} V_e^{-1} Y^T X + 2 V_e^{-1} B V_g = 0,$$

By canceling the gradient we get:

$$\begin{aligned} V_e^{-1} B \left(X^T X + n V_g \right) &= V_e^{-1} Y^T X, \\ B &= Y^T X \left(X^T X + n V_g \right)^{-1}. \end{aligned}$$

- d parallele adaptive ridge regressions with known penalties
- d BLUPs

Algorithm

Practical consideration

Estimation of B

Tested implementations

1. Gradient descent for Adaptive Ridge
2. Singular Value Decomposition of X
   - X of rank K (of the order of 10 to 30)
   - Faster than brute force Gradient descent

p-values computation and correction

1. Simple (dirty) normal assumption of test stat (from GWALPHA)
2. Knockoff (very slow)

Application

Simulation

`PhenotypeSimulator` (Meyer and Birney 2018) was used to simulate multivariate complex phenotypes with multiple loci involved, other genetic and non-genetic factors, and observational noise structure.

Parameters

- h^2: Heritability
- K: Pool size
- n: Population size
- p: Number of Markers

Compared methods

- EMMA (StatGenGWAS)
- PooGawM
- Chi square
- GWalpha (Fournier-Level, Robin, and Balding 2017)

Baseline method using classical GWAS (without pool)

Single trait GWAS in the statgenGWAS package follows the approach of (Kang et al. 2010):

1. population structure and kinship (relatedness) among individuals estimated using linear mixed model.
2. using an F-test, for each SNP in turn using previous step to estimate the covariance structure among individuals

GWalpha (Fournier-Level, Robin, and Balding 2017)

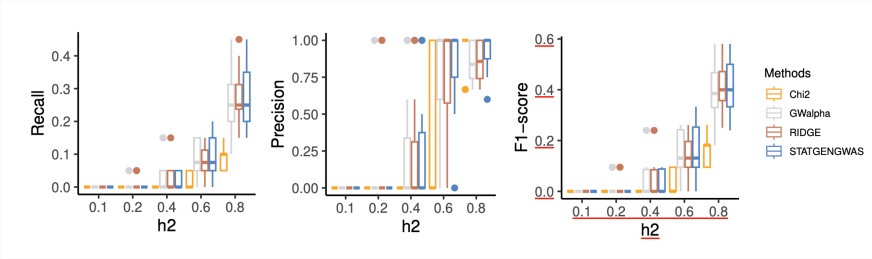
1. ranked Y binned into K pools based on their trait values.
2. Inverse-quantile transformed into $[0, 1]$
3. For each pool $1 \cdots K$, $q_1, \cdots, q_K$ dist. of a specific allele accross the pools
4. GWAlpha estimates the parameters of the beta distribution of the q_k, both for a specified allele and for all alternative
5. Test stat: $\hat{\alpha} = W(\frac{\hat{\mu}_{Allele} - \hat{\mu}_{Alternative}}{\sigma_y})$

Scores

- **Precision**: ratio of true positive predictions to the total positive predictions
- **Recall** : ratio of true positive predictions to all positive
- **F1 score** : harmonic mean of precision and recall

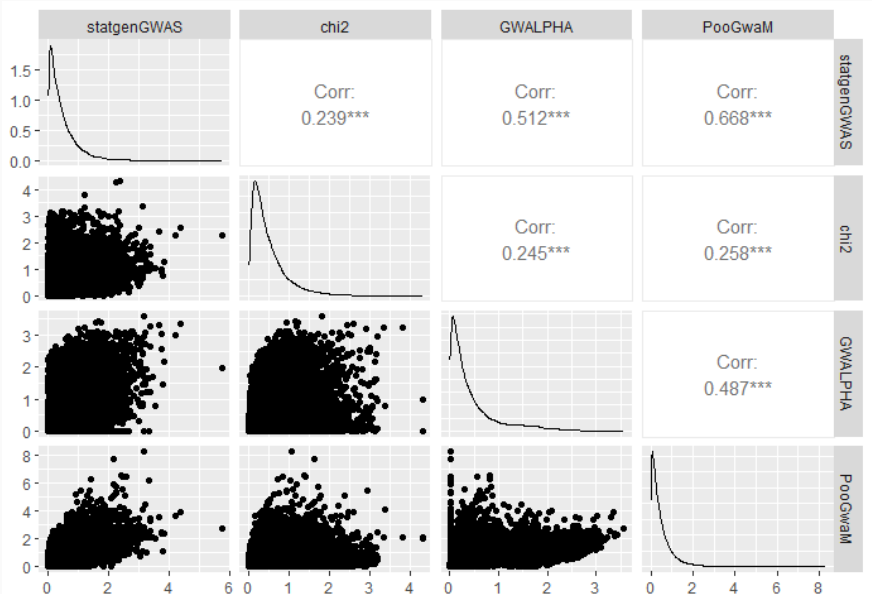
$$F1.score = 2\frac{Precision + Recall}{Precision \times Recall}$$

Heritability: h^2

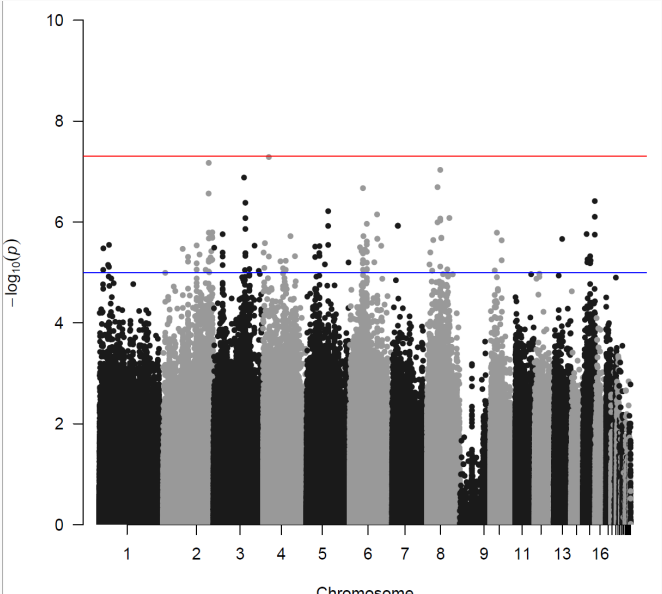


- Recall comparable to EMMA and GWalpha
- Precision slightly better than GWalpha

Comparison with EMMA on real INRAe Maize data



Worm Weight at 56 days



Summary and perspectives

In summary

Simple model for

- considering multi-traits
- computation of pseudo breeding value computation

Perspectives

- Take into account dependence structure among phenotypes
- Improve the multiple testing strategy

References i

Dempster, Arthur P, Nan M Laird, and Donald B Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society: Series B (Methodological)* 39 (1): 1–22.

Fournier-Level, Alexandre, Charles Robin, and David J. Balding. 2017. "GWAlpha: Genome-Wide Estimation of Additive Effects (Alpha) Based on Trait Quantile Distribution from Pool-Sequencing Experiments." *Bioinformatics* 33 (April): 1246–47. https://doi.org/10.1093/BIOINFORMATICS/BTW805.

Grandvalet, Yves. 2000. "Anisotropic Noise Injection for Input Variables Relevance Determination." *IEEE Transactions on Neural Networks* 11 (6): 1201–12.

References ii

Grandvalet, Yves, Stéphane Canu, and Stéphane Boucheron. 1997. "Noise Injection: Theoretical Prospects." *Neural Computation* 9 (5): 1093–1108.

Kang, Hyun Min, Jae Hoon Sul, Susan K. Service, Noah A. Zaitlen, Sit Yee Kong, Nelson B. Freimer, Chiara Sabatti, and Eleazar Eskin. 2010. "Variance Component Model to Account for Sample Structure in Genome-Wide Association Studies." *Nature Genetics* 42 (April): 348–54. https://doi.org/10.1038/NG.548.

References iii

Meyer, Hannah Verena, and Ewan Birney. 2018. "PhenotypeSimulator: A Comprehensive Framework for Simulating Multi-Trait, Multi-Locus Genotype to Phenotype Relationships." *Bioinformatics (Oxford, England)* 34 (September): 2951–56. https://doi.org/10.1093/BIOINFORMATICS/BTY197.

Murphy, Kevin P. 2012. *Machine Learning: A Probabilistic Perspective*. MIT press.