



Good learners, faster learning

Overview on machine learning, stating problems and coordinate descent

Alencar Xavier

III International Meeting on Plant Breeding (2019)

Outline

1. Overview of Machine Learning

- Machine learning
- Context on breeding

2. Good Learners

- Metrics of success
- Case of genomics
- Case of phenomics
- Case of deep learning

3. Fast Learning

- Coordinate descent
- Test on REML for RR & GWAS
- Test on a non-Gaussian model

1. Overview of Machine Learning

- Machine learning
- Context on breeding

2. Good Learners

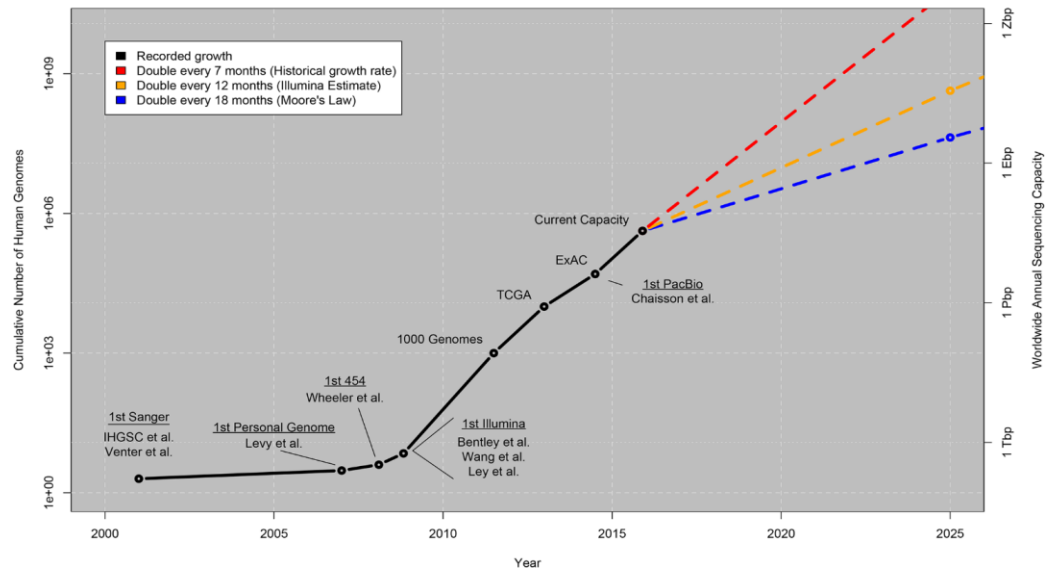
- Metrics of success
- Case of genomics
- Case of phenomics
- Case of deep learning

3. Fast Learning

- Coordinate descent
- Test on REML for RR & GWAS
- Test on a Laplacian model

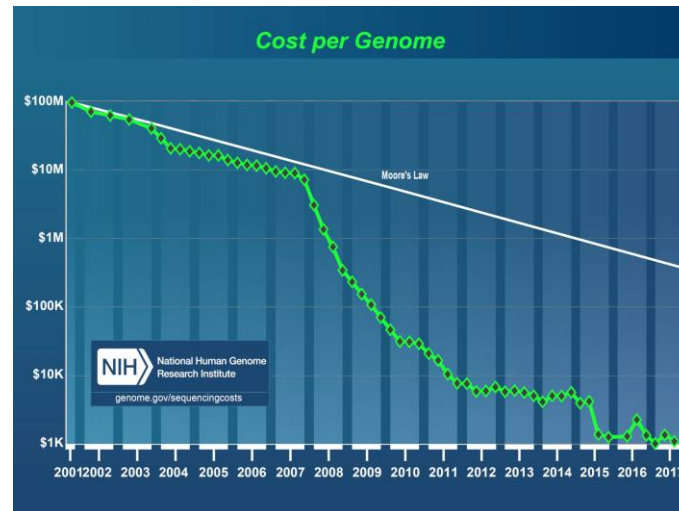
Part 1 – Overview on ML

Growth of DNA Sequencing



Stephens, Z. D. et al. (2015). Big data: astronomical or genomics? *PLoS biology*, 13(7), e1002195.

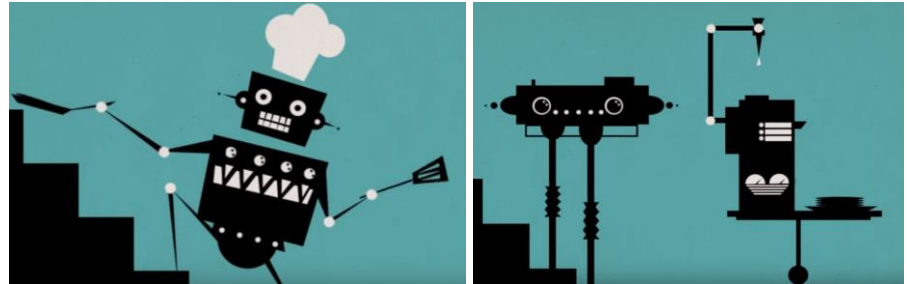
Cost per Genome



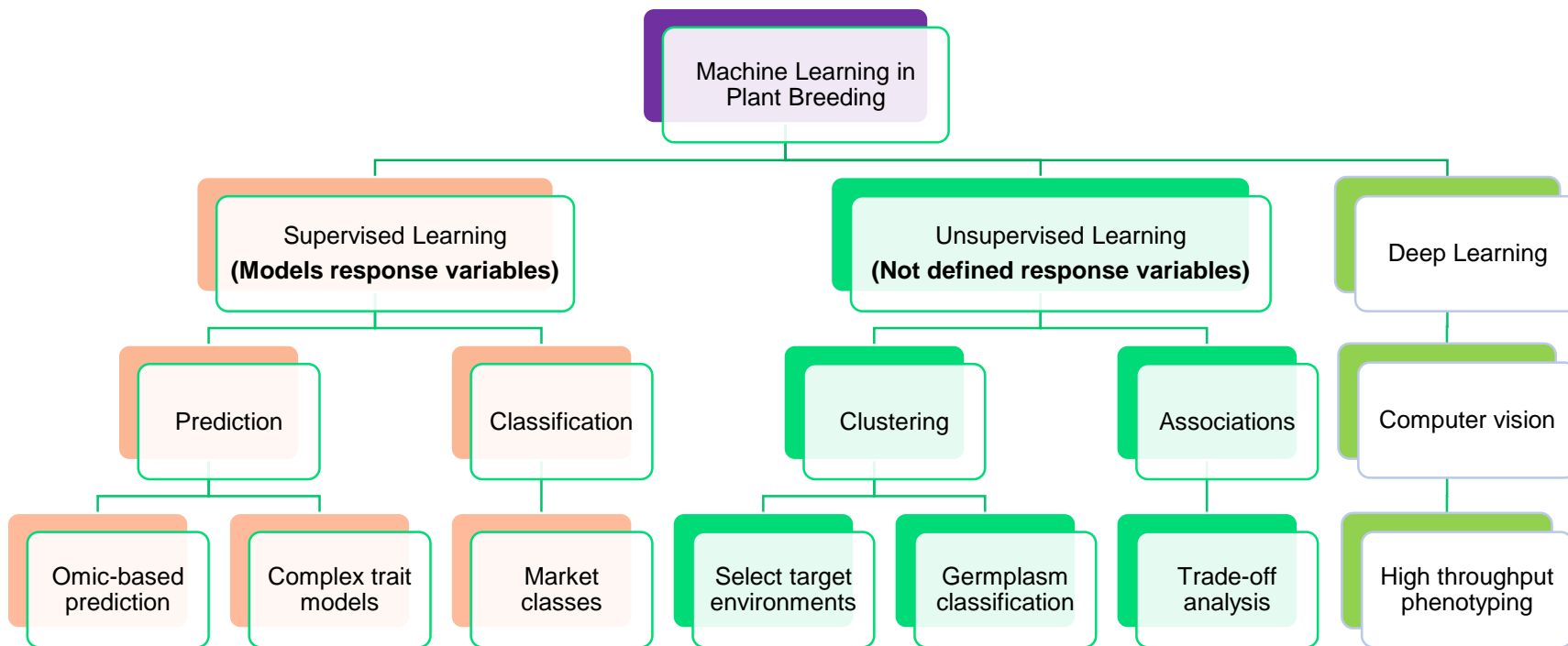
The Cost of Sequencing a Human Genome. NIH. <https://www.genome.gov/27565109/the-cost-of-sequencing-a-human-genome/>

Machine Learning

- Good for solving **well-defined problem**
- Genomics
 - Prediction and selection
 - Germplasm analysis
- Phenomics
 - Automate existing traits
 - Identify new traits



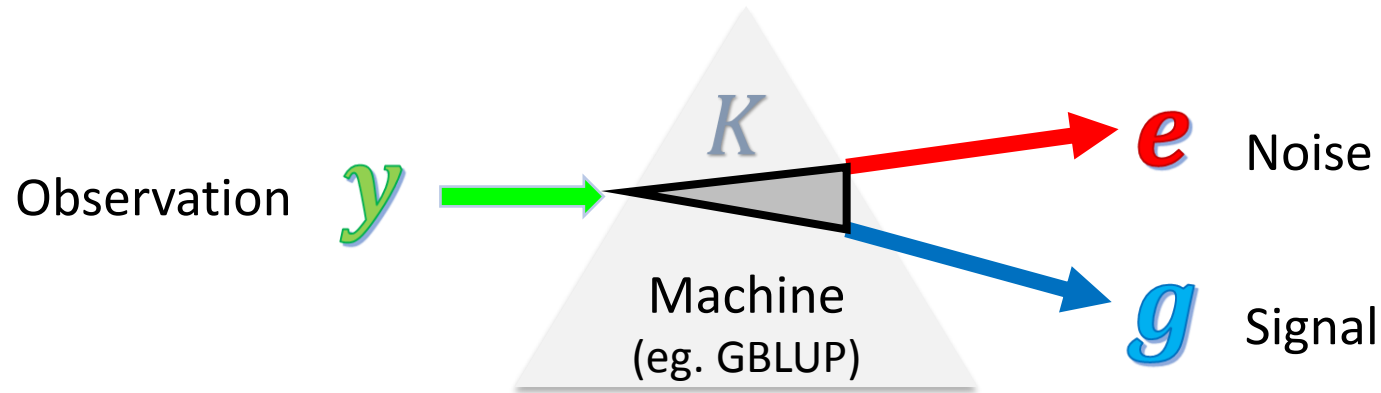
Source: <https://www.youtube.com/watch?v=MPR3o6Hnf2g>



* Not including spatial analysis and all the foundation research in genomics, genome assembly, expression, transcriptomics, proteomics and metabolomics

Supervised machine: use to distinguish signal from noise

$$y = \mu + g + e$$



1. Overview of Machine Learning

- Machine learning
- Context on breeding

2. Good Learners

- Metrics of success
- Case of genomics
- Case of phenomics
- Case of deep learning

3. Fast Learning

- Coordinate descent
- Test on REML for RR & GWAS
- Test on a Laplacian model

Part 2 – Good Learners

Defining the problem: Metrics for success

1. Scientist (why): to define the problem statistically (easy to get it wrong)
2. Metric (what): Correlations, MSPE, Jaccard index (top X %)
3. Testing (how): Simulation or CV real data? Predicting phenotype or GEBV?

Testing machines for different scenarios of genomic prediction

	Genotype	Environment	Prediction Difficulty	Usefulness
CV00	New	New	*****	*****
CV0	Observed	New	***	***
CV1	New	Observed	***	***
CV2	Observed	Observed	*	*

Adapted from Crossa et al. (2017) doi.org/10.1016/j.tplants.2017.08.011

Case 1: Test machines to recover additive signal

(on supervised learning)



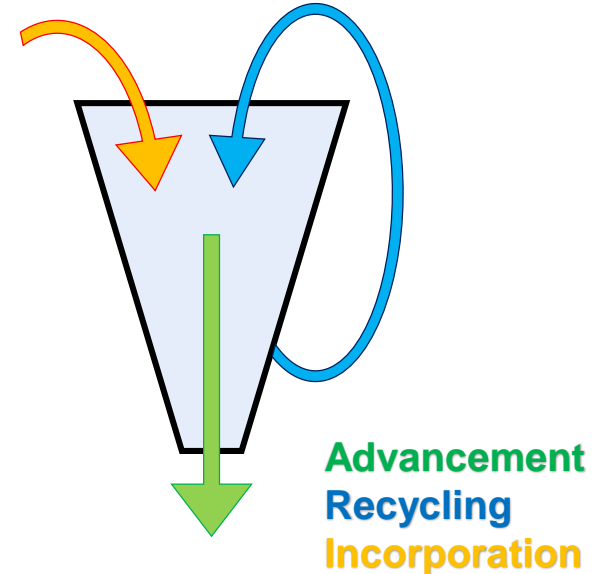
1. **Define prediction target**: Get GEBVs with the entire dataset
2. **How (Testing)**: Omit subset (e.g. family), train additive model
3. **What (Metric)**: Predict GEBV subset; correlate to target GEBV (not phenotype)

Predicted signal → Observed signal



Why is additive genetics ??

- Breeding value (**GEBV**)
 - *Pattern:* ADDITIVE GENETICS
 - *Method:* GBLUP, BayesABC, LASSO
 - *Suits:* **RECYCLING**, **ADVANCEMENT**
- Genomic value (**EGV**)
 - *Pattern:* ANY GENETICS
 - *Method:* RKHS, DNN, Random Forest
 - *Suits:* **ADVANCEMENT**



Case 2: HTP traits for yield improvement

(on unsupervised learning)

- **Task:** Search for HTP trait that is **additive-genetically correlated** (r_A) to yield



- $y_1 \rightarrow$ Trait of interest
- $y_2 \rightarrow$ Secondary trait

$$\text{Efficiency of indirect selection} = \frac{\text{Correlate Response}}{\text{Direct Response}} = \frac{h_{y_2}^2 \times r_A(y_1, y_2)}{h_{y_1}^2}$$


Where does r_A come from?

Where does additive genetic correlation (r_A) come from?

- For a given model used to fit two traits (y_1, y_2):

$$y = \mu + u + e, \quad u \sim N(0, G\sigma_a^2)$$

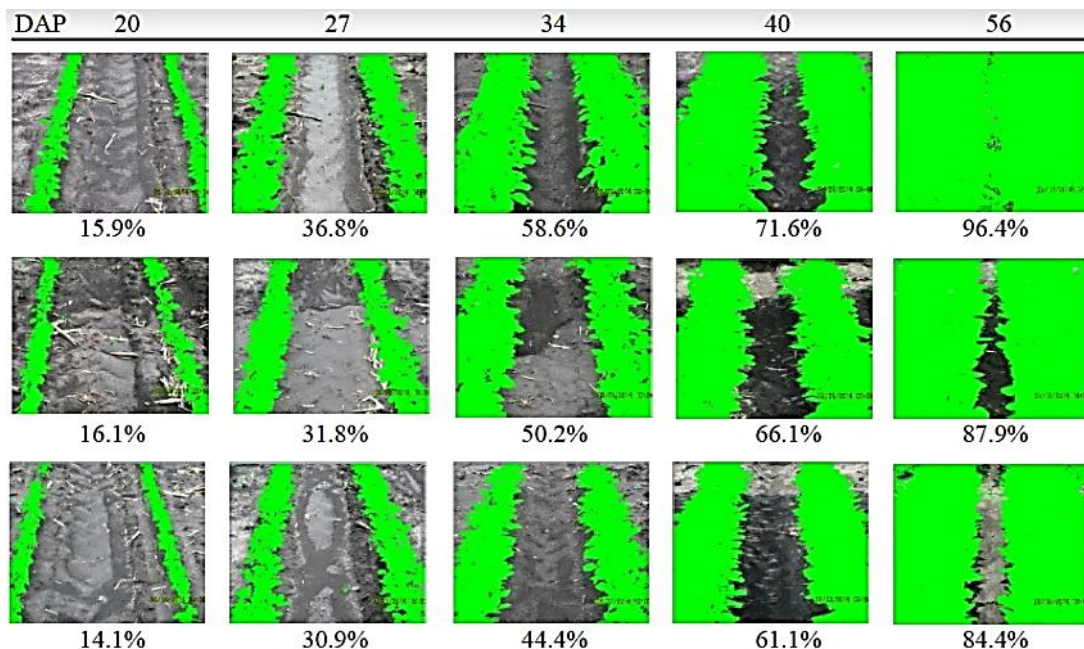
- r_A

$$r_A \cong \frac{\hat{u}_1' G^{-1} \hat{u}_2}{\sqrt{\hat{u}_1' G^{-1} \hat{u}_1 \times \hat{u}_2' G^{-1} \hat{u}_2}}$$


- **Not** r_A

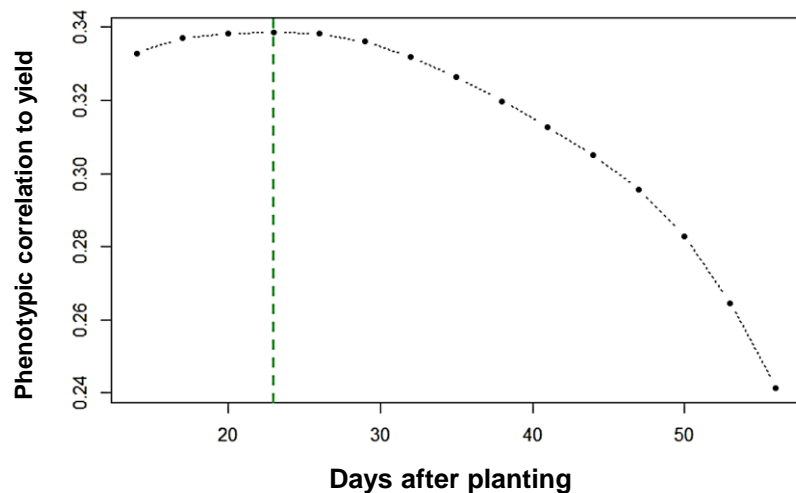
$$r = \frac{\hat{u}_1' \hat{u}_2}{\sqrt{\hat{u}_1' \hat{u}_1 \times \hat{u}_2' \hat{u}_2}}$$

The case of canopy coverage in soy



Drone HTP: Canopy coverage in soybeans

Raw phenotypes



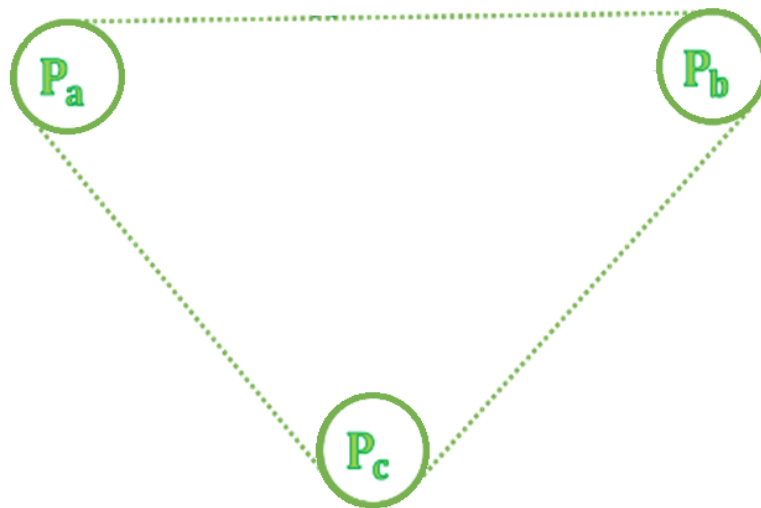
Multivariate mixed model

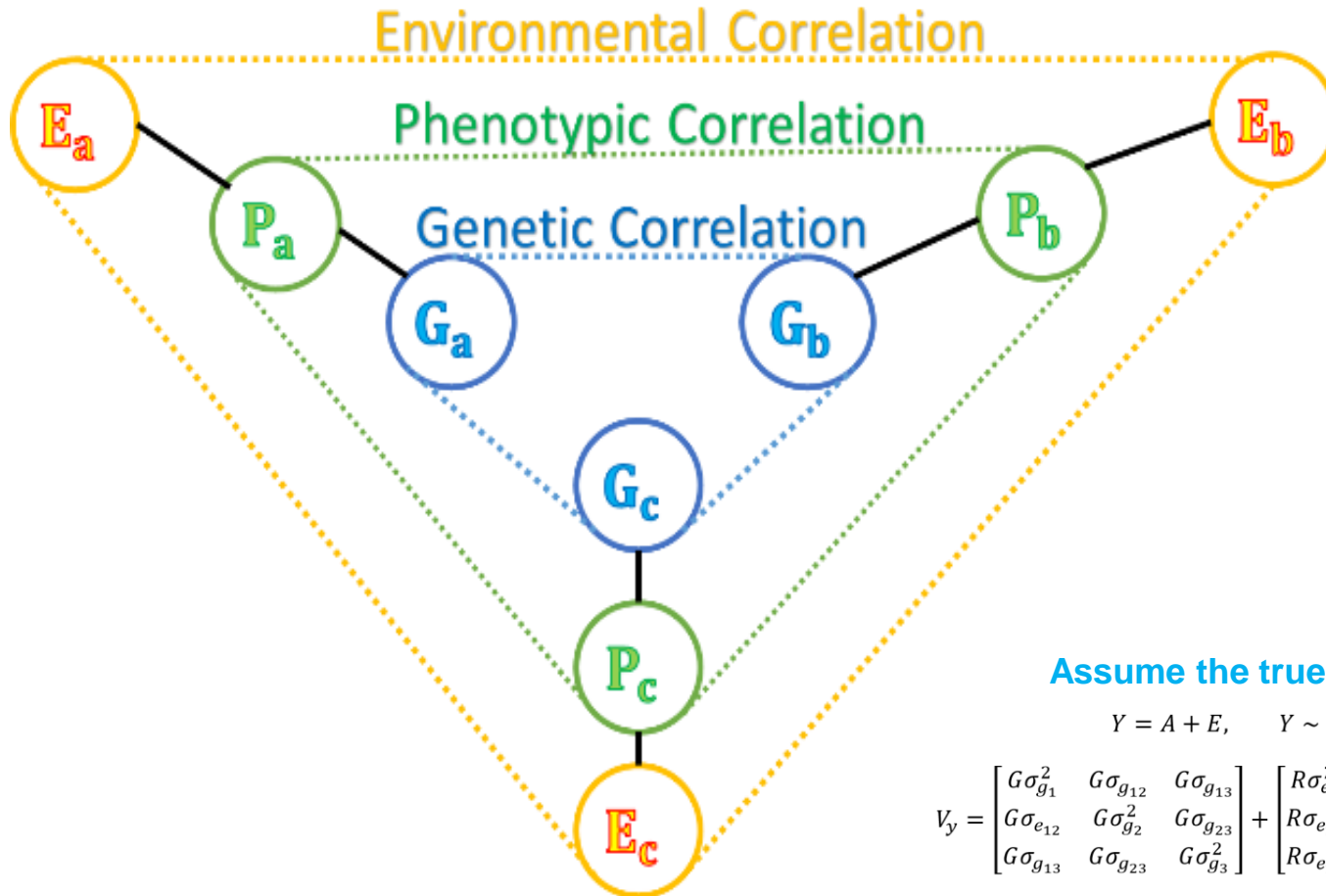
$$\rho_{cc,y} = \frac{CR}{R} = \frac{h_{cc}^2 \times r_{cc,y}}{h_y^2} = \frac{0.76 \times 0.88}{0.58} = 1.14$$

Xavier et al. (2017)

<https://doi.org/10.1534/genetics.116.198713>

Where does the additional information come from??





Assume the true model:

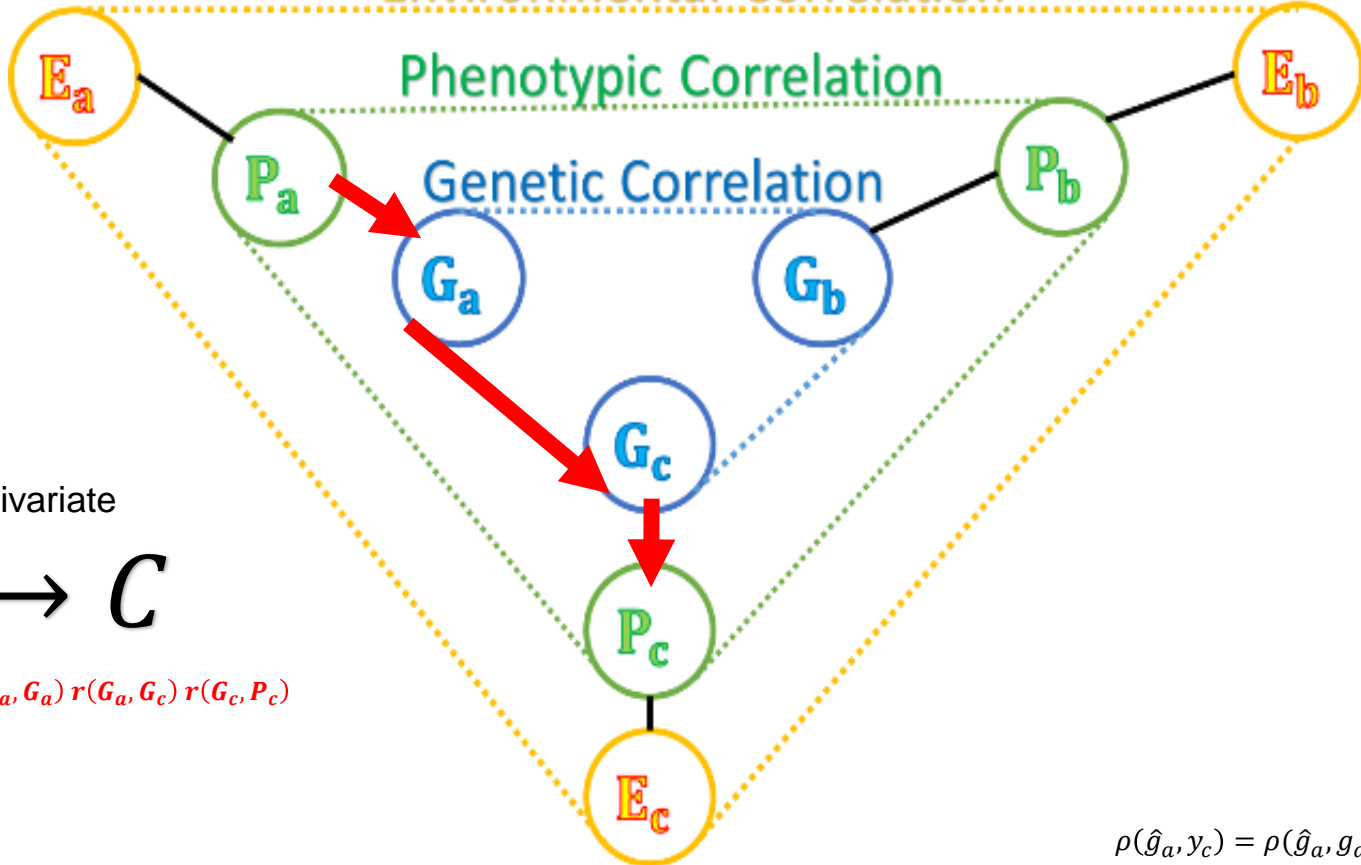
$$Y = A + E, \quad Y \sim N(0, V_y)$$

$$V_y = \begin{bmatrix} G\sigma_{g_1}^2 & G\sigma_{g_{12}} & G\sigma_{g_{13}} \\ G\sigma_{e_{12}} & G\sigma_{g_2}^2 & G\sigma_{g_{23}} \\ G\sigma_{g_{13}} & G\sigma_{g_{23}} & G\sigma_{g_3}^2 \end{bmatrix} + \begin{bmatrix} R\sigma_{e_1}^2 & R\sigma_{e_{12}} & R\sigma_{e_{13}} \\ R\sigma_{e_{12}} & R\sigma_{e_2}^2 & R\sigma_{e_{23}} \\ R\sigma_{e_{13}} & R\sigma_{e_{23}} & R\sigma_{e_3}^2 \end{bmatrix}$$

Environmental Correlation

Phenotypic Correlation

Genetic Correlation



univariate

$A \rightarrow C$

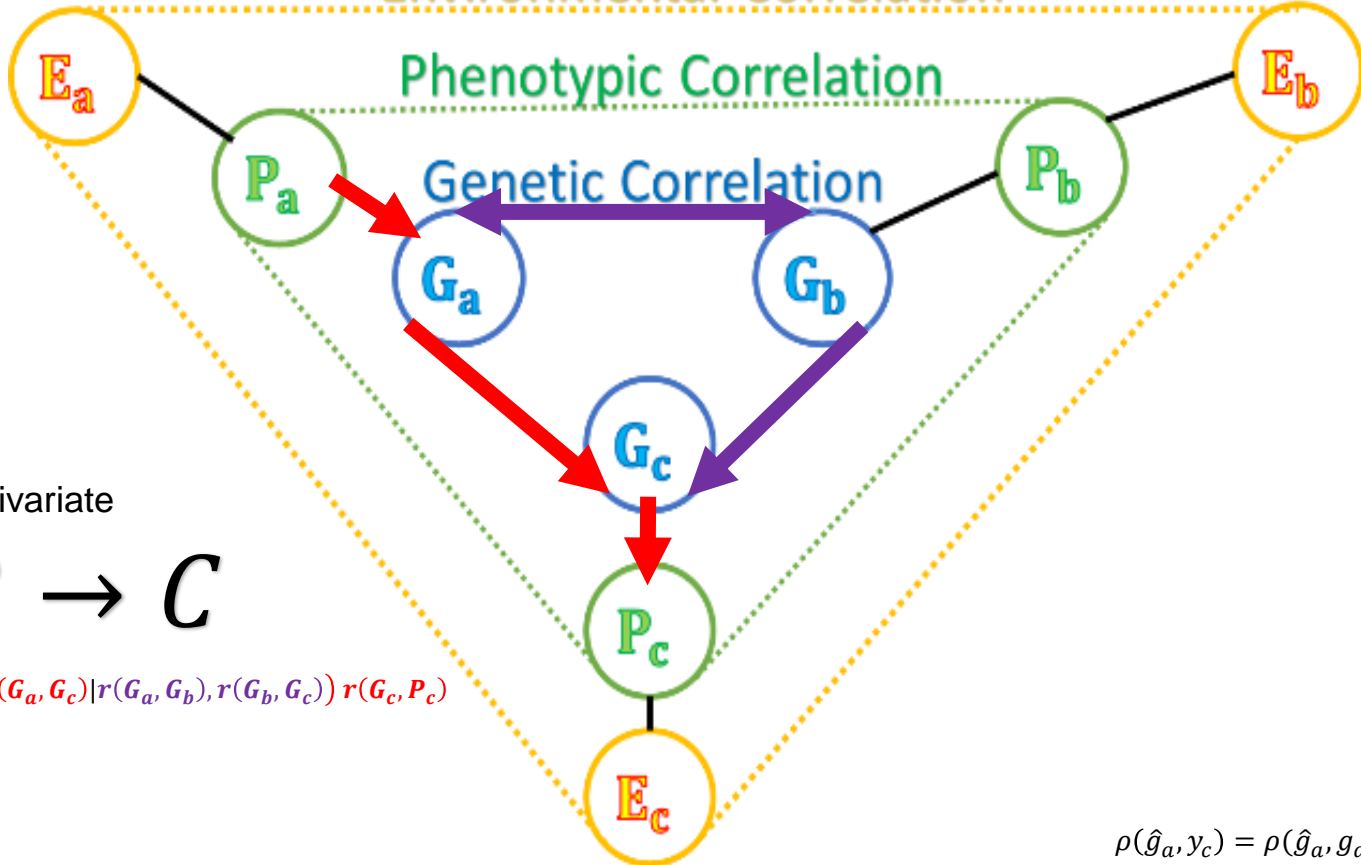
$$r(P_a, P_c) = r(P_a, G_a) r(G_a, G_c) r(G_c, P_c)$$

$$\rho(\hat{g}_a, y_c) = \rho(\hat{g}_a, g_a) \rho(g_a, g_c) \rho(g_c, y_c)$$

Environmental Correlation

Phenotypic Correlation

Genetic Correlation



bivariate

$$A|B \rightarrow C$$

$$r(P_a, P_b) = r(P_a, G_a) (r(G_a, G_c) | r(G_a, G_b), r(G_b, G_c)) r(G_c, P_c)$$

$$\rho(\hat{g}_a, y_c) = \rho(\hat{g}_a, g_a) \rho(g_a, g_c) \rho(g_c, y_c)$$

- **Linear model**

$$y = Zg + e, \quad y \sim N(0, V)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} Z_1 & 0 \\ 0 & Z_2 \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$$

- Covariance structure

$$V = Z(G \otimes \Sigma_a)Z' + I \otimes \Sigma_e = Z \left(G \otimes \begin{bmatrix} \sigma_{a_1}^2 & \sigma_{a_{12}} \\ \sigma_{a_{12}} & \sigma_{a_2}^2 \end{bmatrix} \right) Z' + I \otimes \begin{bmatrix} \sigma_{e_1}^2 & \sigma_{e_{12}} \\ \sigma_{e_{12}} & \sigma_{e_2}^2 \end{bmatrix}$$

- Mixed model equation

$$\begin{bmatrix} Z_1' \Sigma_e^{11} Z_1 + G^{-1} \Sigma_a^{11} & Z_1' \Sigma_e^{12} Z_2 + G^{-1} \Sigma_a^{12} \\ Z_2' \Sigma_e^{12} Z_1 + G^{-1} \Sigma_a^{12} & Z_2' \Sigma_e^{22} Z_2 + G^{-1} \Sigma_a^{22} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} Z_1' (\Sigma_e^{11} y_1 + \Sigma_e^{12} y_2) \\ Z_2' (\Sigma_e^{22} y_2 + \Sigma_e^{12} y_1) \end{bmatrix}$$

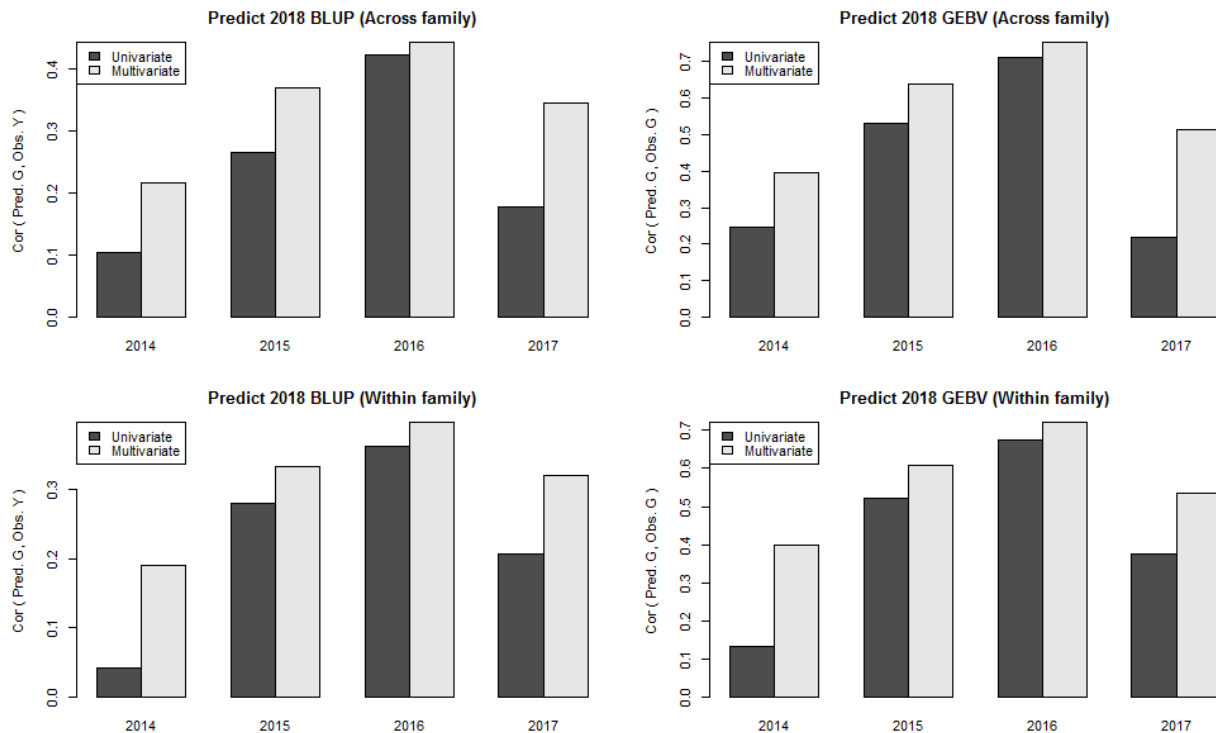
- **Univariate vs bivariate**

$$g_1 = (Z_1' \Sigma_e^{11} Z_1 + G^{-1} \Sigma_a^{11})^{-1} (Z_1' \Sigma_e^{11} y_1)$$

$$g_1 | g_2 = (Z_1' \Sigma_e^{11} Z_1 + G^{-1} \Sigma_a^{11})^{-1} (Z_1' (\Sigma_e^{11} y_1 + \underbrace{\Sigma_e^{12} y_2}_{\text{INFORMATION GAIN}}) - (Z_1' \Sigma_e^{12} Z_2 + G^{-1} \Sigma_a^{12}) g_2)$$



MV modeling increases the PA both within- and across-family

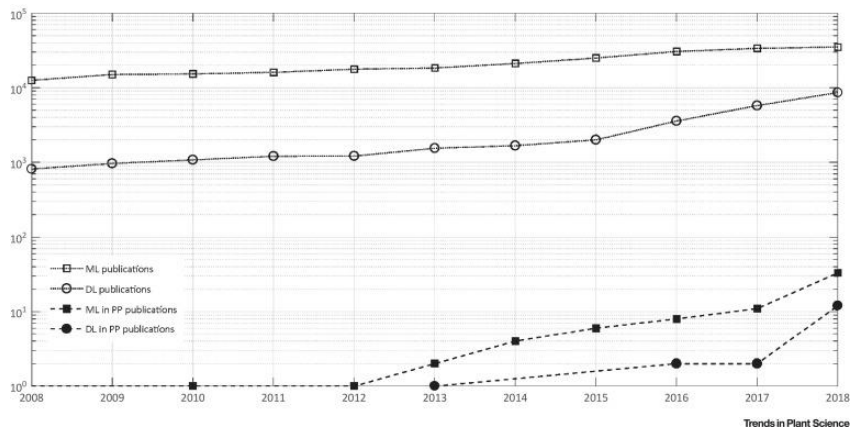


Source: Data not published

Case 3: No free lunch theorem

see Ho and Pepyne (2002), 10.1023/A:1021251113462

- A same prediction machine can be used for various applications
- Such an algorithm is unlikely to be equally good for all tasks, or consistently outperform simpler machines



Singh et al (2018) 10.1016/j.tplants.2018.07.004

Are DNN compared to **weak baselines**?? (arXiv:1907.06902v1)

Trait	DeepGS vs RR-BLUP
	MNV improvement (%)
GL	-0.30~9.12(0.39)
GW	0.01~5.46(0.46)
TW	0.97~23.63(2.63)
GP	2.32~54.50(3.39)
PHT	-5.12~199.48(-4.13)

MA (2017) 10.1101/241414



"compared with the widely used method RR-BLUP, DeepGS still yields a relative improvement ranging from 1.44% to 65.24%"

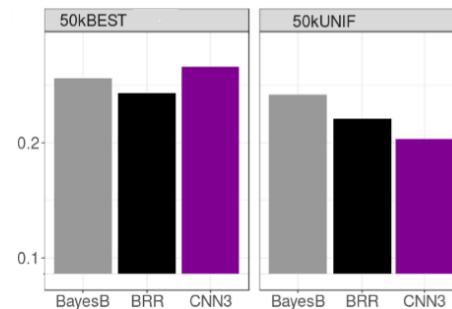
Case 3: No free lunch theorem

BL	RKHS	RBFNN
0.658	0.645	0.656

Gonzalez-Camacho et al (2012)
doi.org/10.1007/s00122-012-1868-9

Trait	BRR	Bayes B	RBFNN	BRNN
DTH	0.58 (0.12)	0.57 (0.12)	0.48 (0.14)	0.48 (0.14)
GY	0.57 (0.14)	0.70 (0.09)	0.56 (0.12)	0.65 (0.10)

Perez-Rodriguez et al (2011)
doi.org/10.1534/g3.112.003665



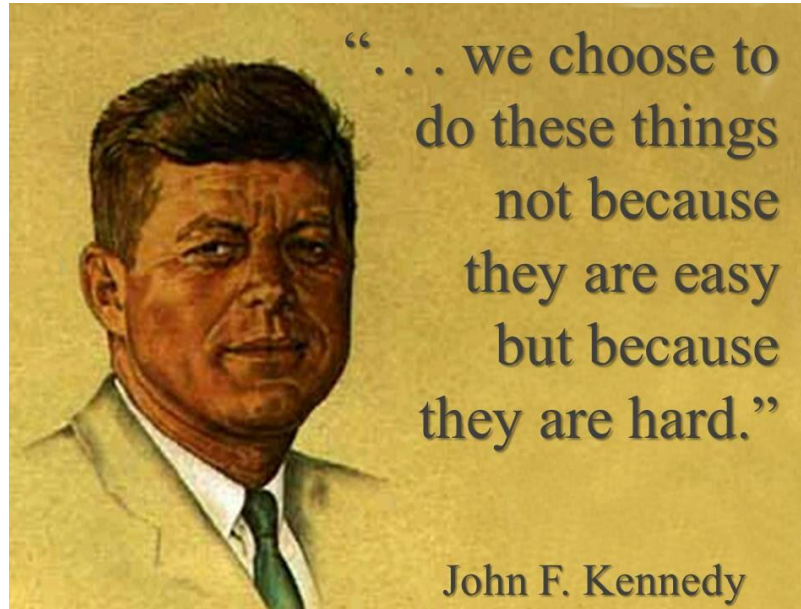
Bellot et al. (2018)
doi.org/10.1534/genetics.118.301298

Similar GS outcome observed by

- A Montesinos-López et al (2018) doi.org/10.1534/g3.118.200740
- AO Montesinos-López et al (2018) doi.org/10.1534/g3.118.200728
- JM González-Camacho et al (2018) 10.3835/plantgenome2017.11.0104

To get some hands-on experience with DNN see
Pérez-Enciso and Zingaretti (2019)
doi: 10.3390/genes10070553

Breeding values from DNN vs GBLUP



1. Overview of Machine Learning

- Machine learning
- Context on breeding

2. Good Learners

- Metrics of success
- Case of genomics
- Case of phenomics
- Case of deep learning

3. Fast Learning

- Coordinate descent
- Conditioning and approximation for RR & GWAS
- Test on more complicated machines

Part 3 – Fast Learning

OR *“The art of enabling large analysis with restricted computational resources”*

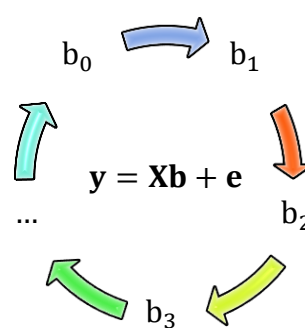
Coordinate Descent (CD)

(see Friedman et al. 2010, [dx.doi.org/10.18637/jss.v033.i01](https://doi.org/10.18637/jss.v033.i01))

- **Key concept:** Reduce the dimensionality of the problem to univariate through “**conditioning**”

- *How does CD work?*

1. Update a coefficient
2. Update residuals



- *Other popular names for CD*

- **Gradient descent** – another popular nomenclature in ML literature
- **Gauss-Seidel residual update** (see Legarra & Misztal (2008) 10.3168/jds.2007-0403)
- **Full-conditional expectation** – The backbone of *Bayesian Gibbs sampling* methods

CD works on Variance Components

$$\sigma_i^2 = \frac{\mathbf{u}_i' \mathbf{K}_i^{-1} \mathbf{u}_i + \text{tr}(\mathbf{K}_i^{-1} \mathbf{C}^{ii}) \sigma_e^2}{q_i} = \text{HARD TO GET (for large and dense datasets)}$$

$$\sigma_e^2 = \frac{\mathbf{y}' \mathbf{e}}{n - r_X} = \text{EASY TO GET}$$

- Consider the model

$$\mathbf{y} = \mu + \mathbf{M}\mathbf{a} + \mathbf{e}$$

- If you condition the response variable to a single marker



$$\tilde{\mathbf{y}} = \mathbf{y} - (\mu + \mathbf{M}_{-j} \mathbf{a}_{-j}) = \mathbf{m}_j \mathbf{a}_j + \mathbf{e}$$

- The genetic variance becomes

$$\sigma_{\text{SNP}}^2 = \frac{\mathbf{a}' \mathbf{K}^{-1} \mathbf{a} + \text{tr}(\mathbf{K}^{-1} \mathbf{C}^{ii}) \sigma_e^2}{q} = \mathbf{a}^2 + \frac{\sigma_e^2}{\mathbf{m}' \mathbf{m} + \lambda}$$

Example 1A: conditioning and approximation of RR-BLUP

$$\mathbf{y} = \mu + \mathbf{M}\mathbf{a} + \mathbf{e}, \quad \mathbf{a} \sim N(0, \sigma_{\text{SNP}}^2)$$

- A simple speed up using

1. Conditioning: Use **CD** to marker effects (**a**)



2. Approximation: To get genetic variance (σ_{SNP}^2)



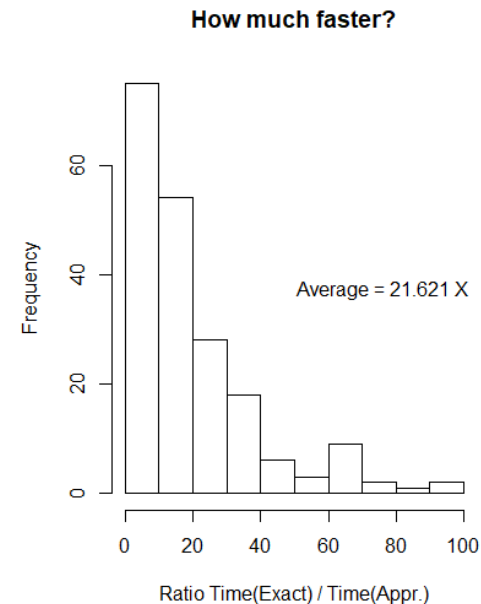
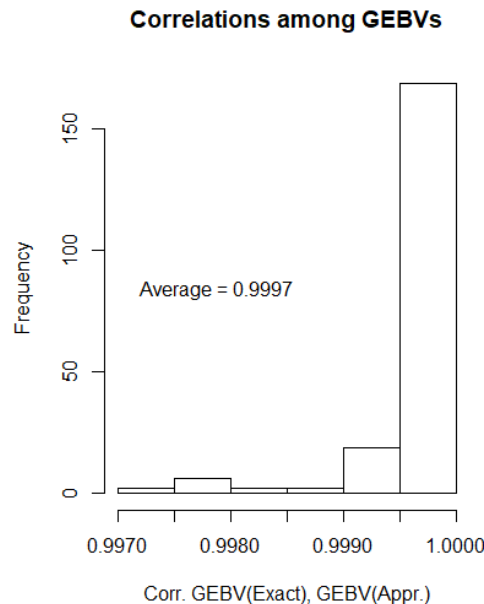
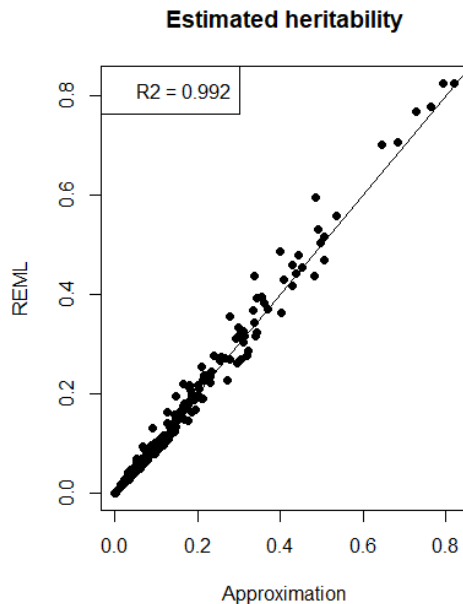
$$\sigma_{\text{SNP}}^2 \cong \frac{(\mathbf{M}\mathbf{a})'(\mathbf{y} - \mu)}{\text{tr}(\mathbf{M}'\mathbf{S}\mathbf{M})} = \frac{\sigma_y^2 - \sigma_e^2}{\sum \sigma_{\mathbf{M}_j}^2}$$

Schaeffer (1986)

10.3168/jds.S0022-0302(86)80743-3

- Compare to EMMA REML

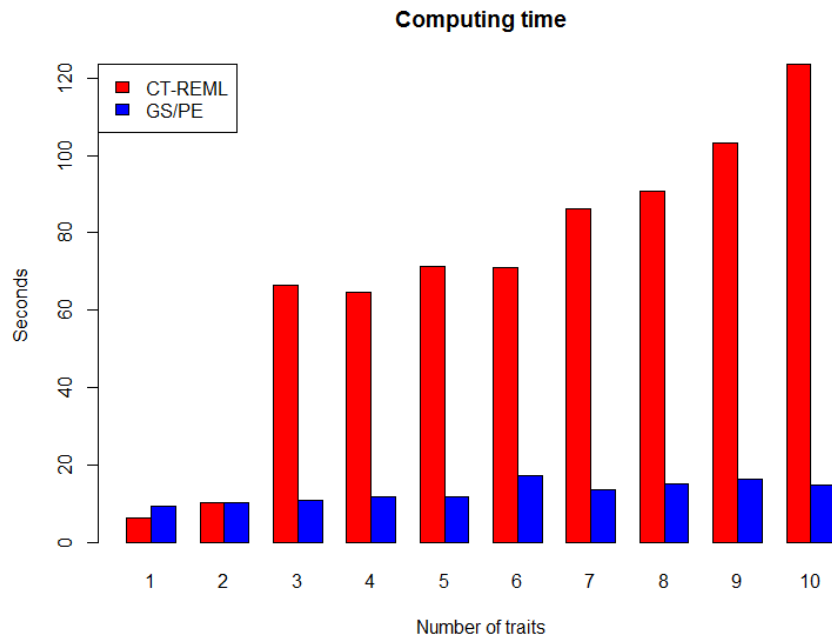
1A) Variance components and breeding values



200 simulated datasets, varying N, P and h^2

Some contribution from Ramasubramanian (Beavis Lab, Iowa State)

Even greater improvements in performance can be observed in multivariate modeling



Wheat data from BGLR
(env 5-10 were augmentations)

1B) conditioning and approximation of GWAS

- Test SNPs via likelihood ratio costs only 1 extra round of CD
- Reduced model (*does not contain the marker of interest*)

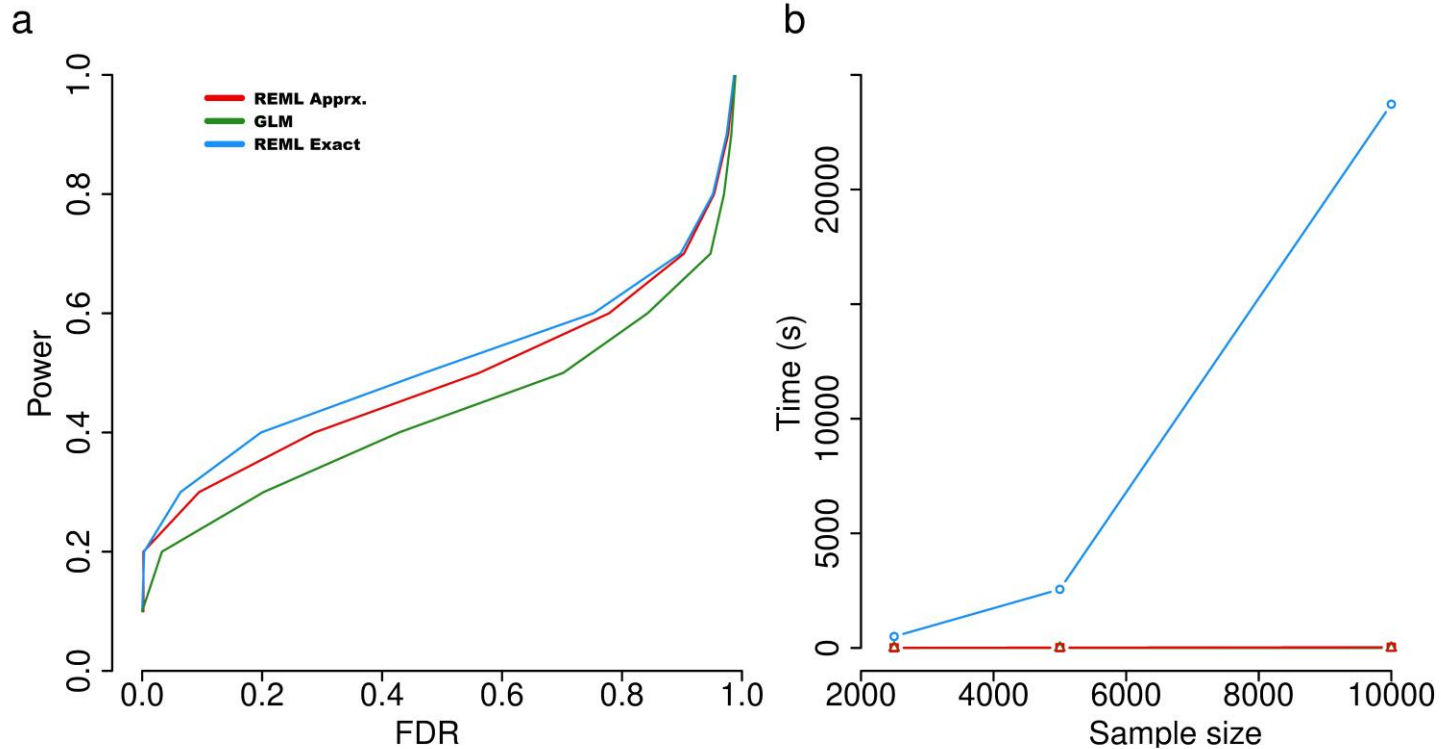
$$\mathbf{y} = \mu + \mathbf{M}_{-j}\mathbf{a}_{-j} + \mathbf{e}$$

- Full model (*contains the marker of interest*)

$$\mathbf{y} = \mu + \mathbf{m}_j\mathbf{a}_j + \mathbf{M}_{-j}\mathbf{a}_{-j} + \mathbf{e}$$

Legend
Fixed effects
Random effects

Power, false positives, computing time



Analysis provided by Meng Huang (Rainey Lab, Purdue)

Example 2: conditioning a non-Gaussian model

A. First, we test as Whole-Genome Regression (WGR)

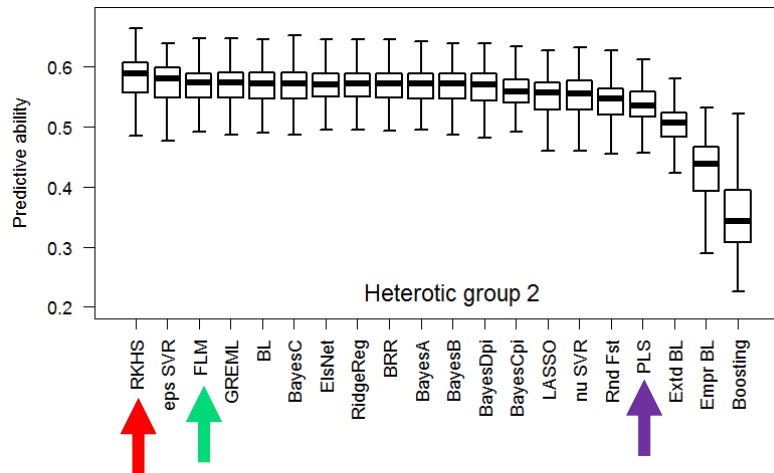
$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{M}\mathbf{a} + \mathbf{e}, \quad \mathbf{a} \sim N(0, T^2 \sigma_e^2)$$

B. Test a single-step: Plugging WGR into a mixed model via conditioning

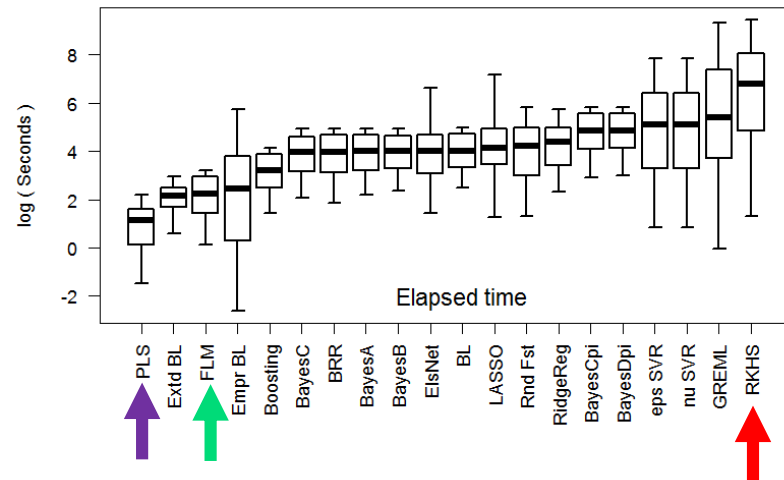
$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{Z}\mathbf{u} + \mathbf{e}, \quad \text{where } \mathbf{u} = \mathbf{M}\mathbf{a} + \boldsymbol{\varepsilon}$$

2A) Fast Laplace machine (FLM) on genomic prediction

Accuracy



Speed

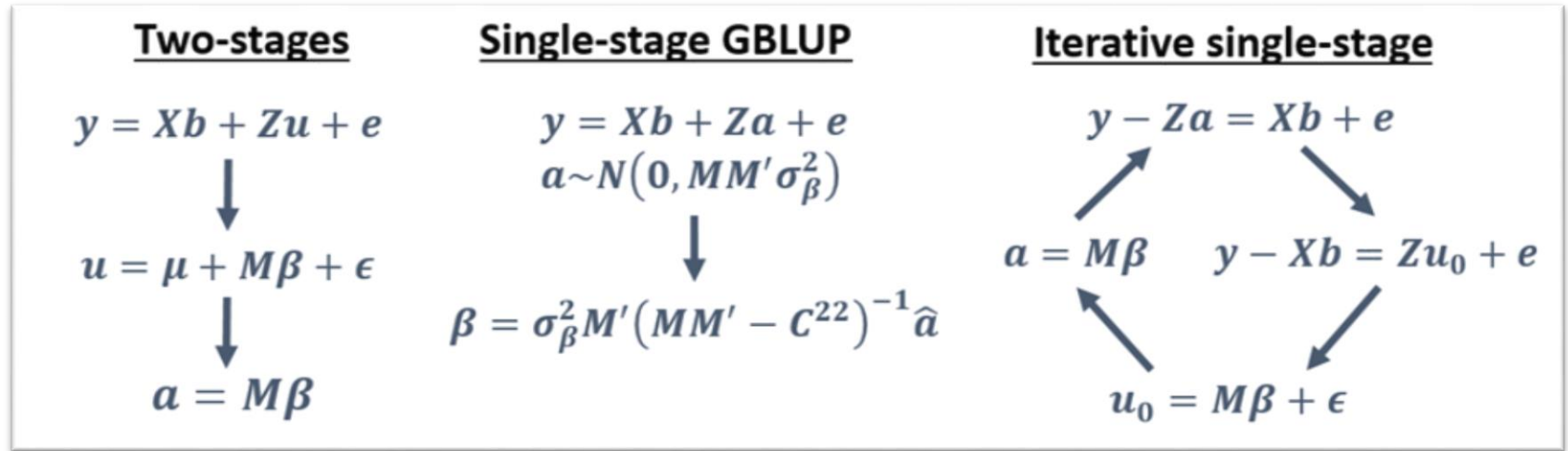


Maize datasets (from a heterotic groups in 10 geographies, 5-fold cross-validations 20x): Box plot of prediction accuracy (left) and computing time (right) across methods. Models ordered based on the average performance.

* No free-lunch: A model cannot be fastest and most accurate

Source: Xavier (2019) G3, 10.1534/g3.119.400728

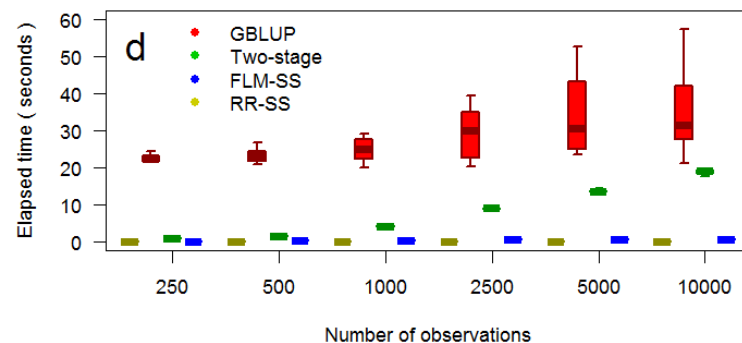
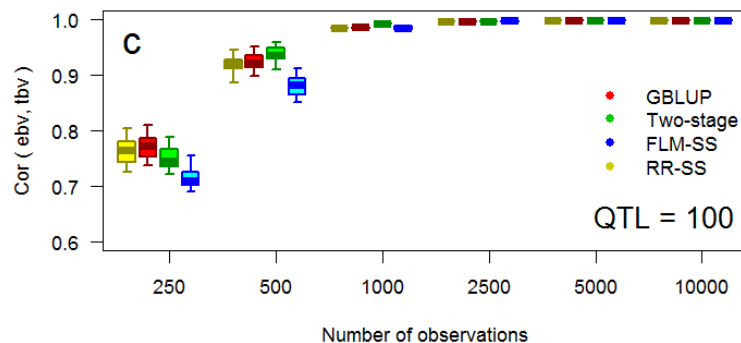
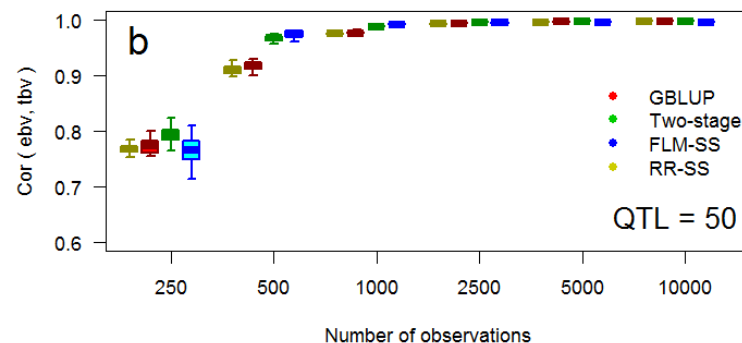
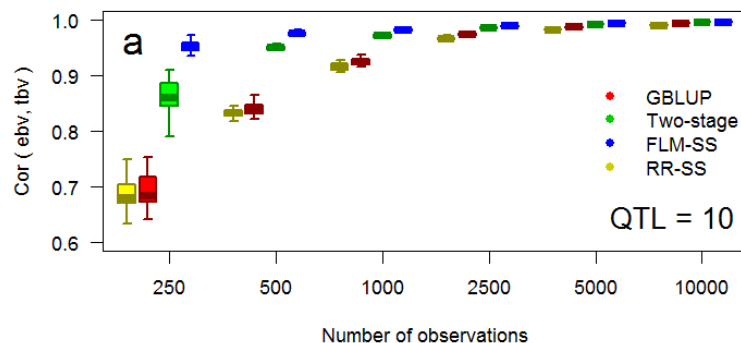
2B) Conditioning the WGR into a mixed model to speed up computation



BLUEs + BLUPs

Classical GBLUP

Conditioning



This comparison was performed across a variable number of individuals ($n = 250, 500, 1000, 2500, 5000$ and 10000) and architecture ($QTL = 10, 50, 100$), each scenario was repeated 20x with different seeds to sample the individuals.

Source: Xavier (2019) G3, 10.1534/g3.119.400728

Concluding Remarks (🔑's)

1. Machines respond to well defined problems
2. Metrics of success are critical (e.g. r_A and $gebv$'s)
3. Conditioning and approximations can make good machines

Concluding Remarks (🔑's)

1. Machines respond to well defined problems
2. Metrics of success are critical (e.g. r_A and prediction targets)
3. Conditioning and approximations can make good machines

Concluding Remarks

1. Machines respond to well defined problems
2. Metrics of success are critical
3. Conditioning and approximation make accurate/scalable machines

Thank you for your attention!

Questions??

Alencar Xavier

Alencar.Xavier@Corteva.com

<http://alenxav.wix.com/home>