

## Statistical Methods for Genome Enabled Prediction:

a mixed bag of tools for genome-assisted selection

IOWA STATE UNIVERSITY, APRIL 6-10, 2012

Daniel Gianola

Sewall Wright Professor of Animal Breeding and Genetics

UW-MADISON  
ANIMAL SCIENCES

University of Wisconsin



Dairy Science

Biostatistics & Medical Informatics

1

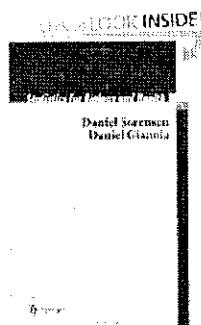
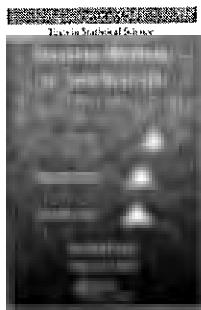
## TOPICS COVERED (order is approximate)

1. Evolution of statistical methods in quantitative genetics
2. Challenges from complexity and use of phenomic data
3. Brief review of Bayesian inference, Bayesian regression
4. Genome-enabled prediction: "Genomic Blup";  
the alphabet: Bayes A, Bayes B, Bayes C, Bayes L
5. Principles of cross-validation
6. The problem of dealing with interactions
7. Introduction to non-parametric regression: LOESS,  
kernel regression, RKHS, radial basis functions, neural networks (NN)
8. Results from animals and plants

PRE-REQUISITES: knowledge of quantitative genetics, mixed linear models,  
probability theory, some basics of R language 2

Statistic is tools to describe machine  
to make story about world

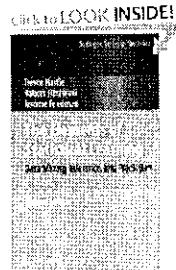
## SOME BIBLIOGRAPHY: Bayesian



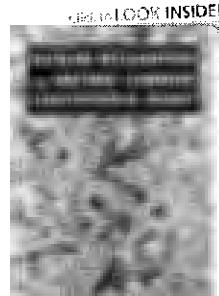
3

## Statistical learning: general

more S.I.!



My favorite  
→



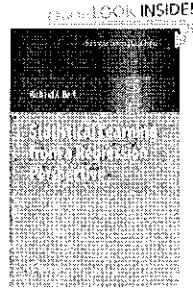
Free software: WEKA



4

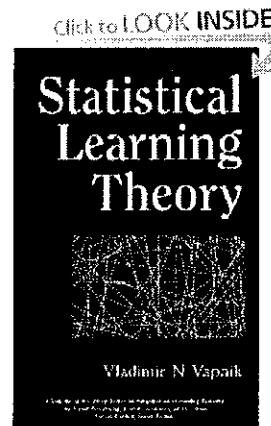
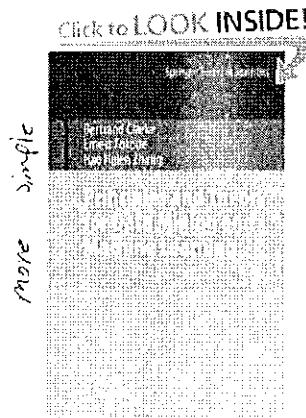
→ Pre-Prediction  
Impact on  
Results (first)

## Gentle introductions to non-parametric regression...

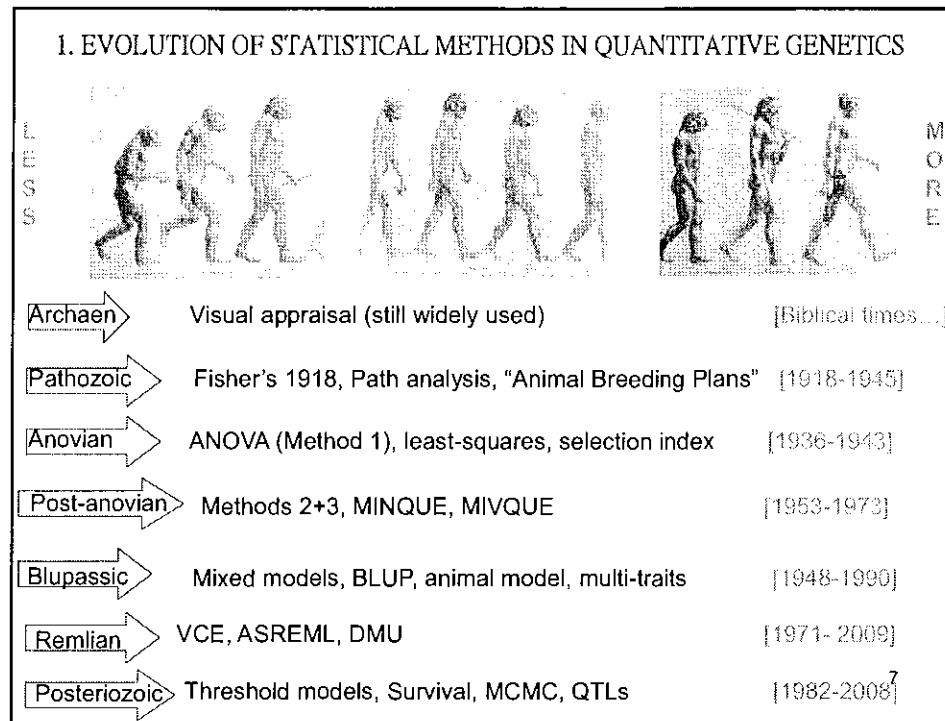


5

ONLY IF YOU REALLY WANT TO GO DEEPLY...



6



Balding et al. (2007) “Handbook of Statistical Genetics”. Wiley

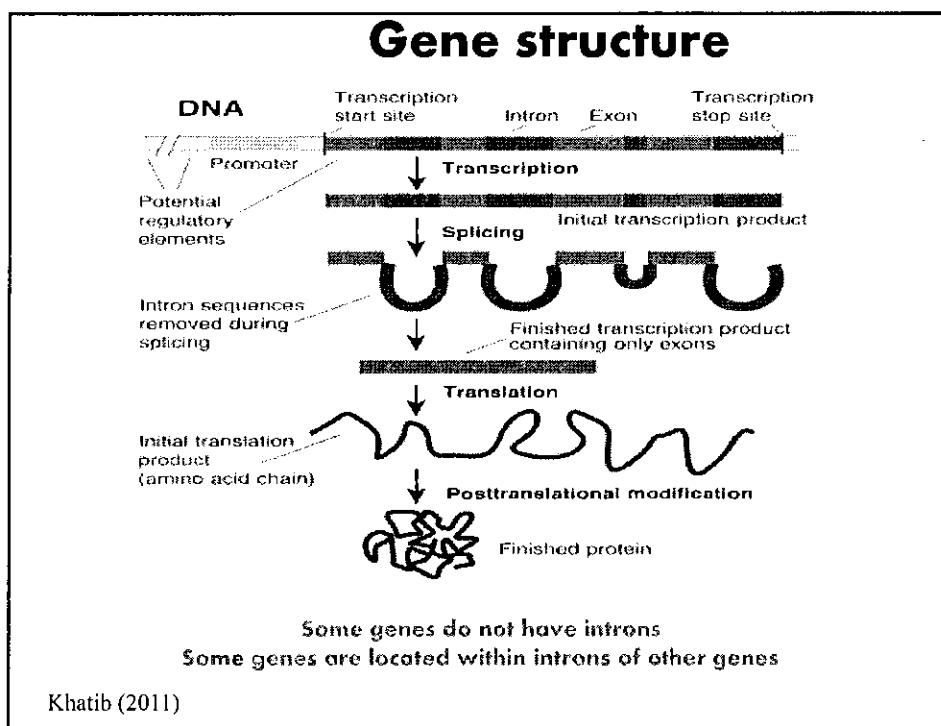
## Chapter 20

D. Gianola

*“Inferences from Mixed Models in Quantitative Genetics”*

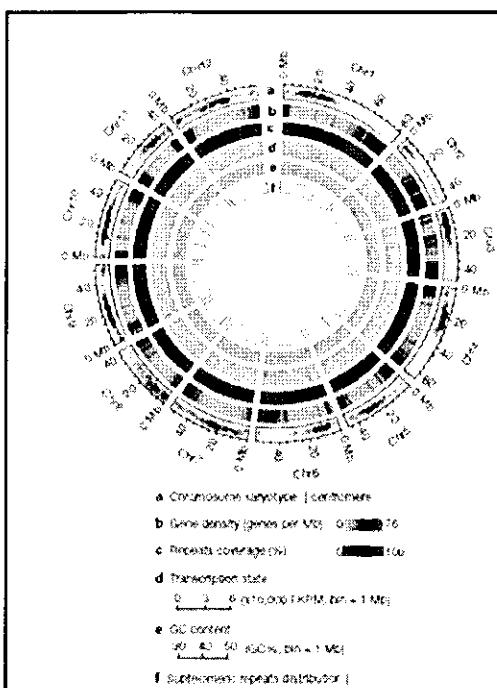
## 2. Challenges from complexity and use of phenomic data

9



## How many genes do we have?

<u>Organism</u>	<u>Genome size</u>	<u># of genes</u>	<u>DNA/gene</u>
• <i>Haemophilus influenzae</i>	1.8 Mb	~1,700	~ 1 Kb
• <i>Escherichia coli</i>	4.6 Mb	~4,300	~ 1 Kb
• Baker's Yeast  ( <i>Saccharomyces cerevisiae</i> )	12.1 Mb	~6,000	~ 2 Kb
• A worm  ( <i>Caenorhabditis elegans</i> )	97 Mb	~18,000	~5.4 Kb
• Fruit fly  ( <i>Drosophila melanogaster</i> )	185 Mb	~14,000	~13 Kb
• Human ( <i>Homo sapiens</i> )	3,000 Mb	~25,000	~ 86 Kb
• A flowering plant  ( <i>Arabidopsis thaliana</i> )	100 Mb	~25,000	~ 4 Kb



# POTATO GENOME (Nature 2011)

- Final assembly 727 Mb
  - Genome size 844 Mb
  - 1 SNP every 40 bp
  - 1 indel every 394 bp (average 12.8 bp)
  - 24,051 genes cluster with at least one of 11 genomes

# The Phenomic data (phenotypes+genomic)

- 1) Massive phenotypic data exist
- 2) Massive genomic data increasingly available

Example: SNPs (also gene expression)

→  $10^7$  SNPs dbSNP 124 (Nat. Center Biotechnology)

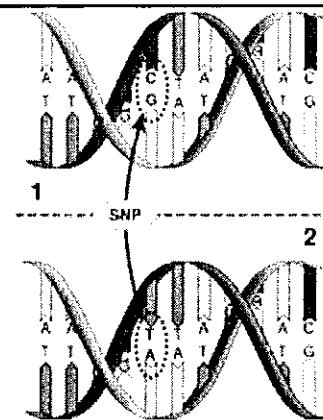
→ Perlegen: 1.58 million SNPs

→ Animals:

- Wong et al. (2004) -- chicken genetic variation map with **2.8 million SNPs**
- Hayes et al. (2004) -- **2500 SNPs** in salmon genome
- Poultry breeding companies -- Thousands of SNPs on sires/dams
- USA (2008) -- >**50,000 SNPs** in over **3000** Holstein sires
- Pigs -- **60,000 SNPs**
- All over developed world -- chips with **800,000 SNPs** in cattle

13

All you wanted to know about SNPs  
but were afraid to ask...



**SNP**= DNA sequence variation occurring when a single nucleotide - A, T, C, or G in the genome differs between members of a species (or between paired chromosomes)

**ABOVE:** two sequenced DNA fragments  
AAGCCTA to AAGCTTA, contain a difference in a single nucleotide.

we say that there are two alleles : C and T

14

## Copy number (CNV) of copy number polymorphisms (CNP): other source of information about genetic variation

- Individuals vary in number of copies of genomic regions
- Disease genes located in CNV regions

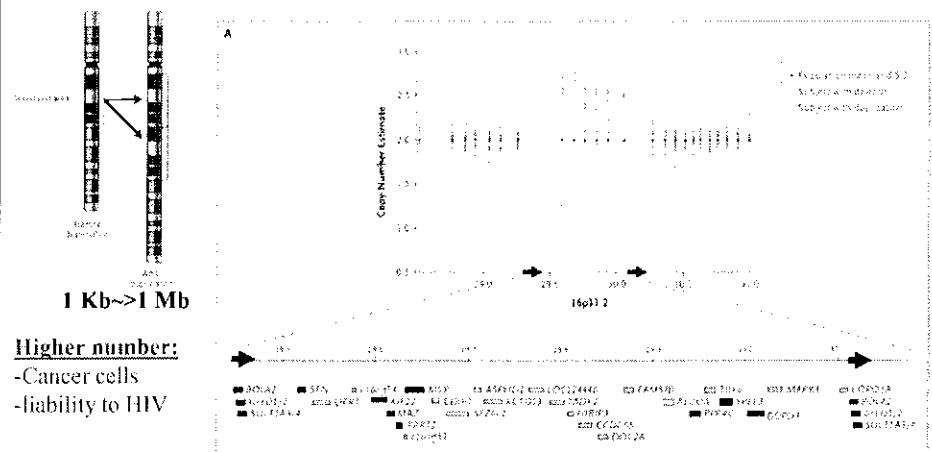


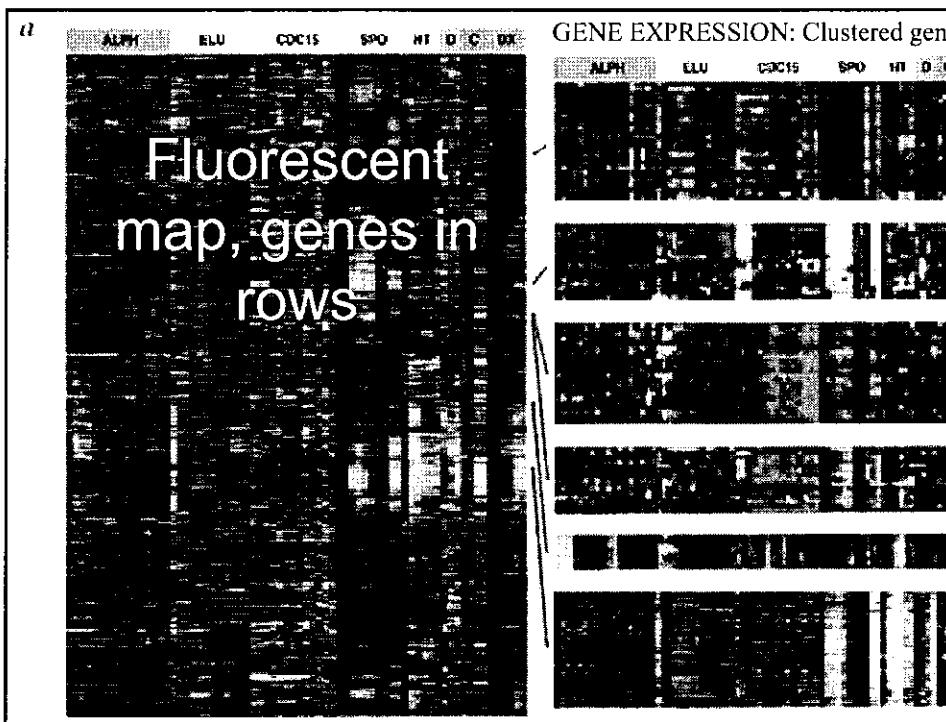
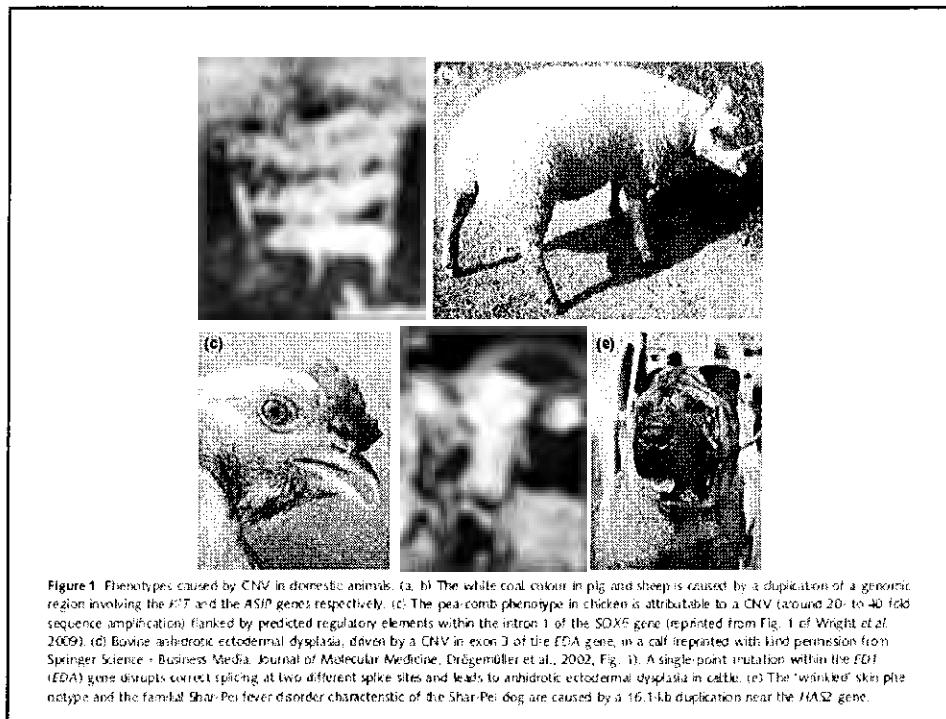
Table 1. Summary of CNV surveys performed in diverse mammalian livestock species.

Species	Number of samples	Number of CNVR	Mean size CNVR (kb)	Median size CNVR (kb)	Size range CNVR (kb)	Methods employed to detect CNVR	References
Cow	556	42	960.6	394.8	22.9-11050.6	BovineSNP50 BeadChip	Matukumali et al. (2009)
	265	368	171.5	128.3	50-200	BovineSNP50 BeadChip	Bae et al. (2010)
	20	304	723	16.7	1.7-2000	Bovine 2.1 M aCGH arrays	Fadista et al. (2010)
	90	177 <sup>1</sup>	159	89	18-1260	Bovine 385k aCGH arrays	Uu et al. (2010)
	539	682	204.9	131.1	32.5-5569	BovineSNP50 BeadChip	Hou et al. (2011)
Sheep	11	135	77.6	55.9	24.6-505	Bovine 385k aCGH arrays	Fontanesi et al. (2011a)
Goat	10	127	90.3	49.5	24.6-1070	Bovine 385k aCGH arrays	Fontanesi et al. (2010)
Pig	12	37	9.32	6.89	1.7-61.9	Porcine 385k aCGH arrays	Fadista et al. (2008) <sup>2</sup>
	55	49	754.6	170.9	44.7-10700	Porcine SNP60 Beadchip	Ramayo-Caldas et al. (2010)

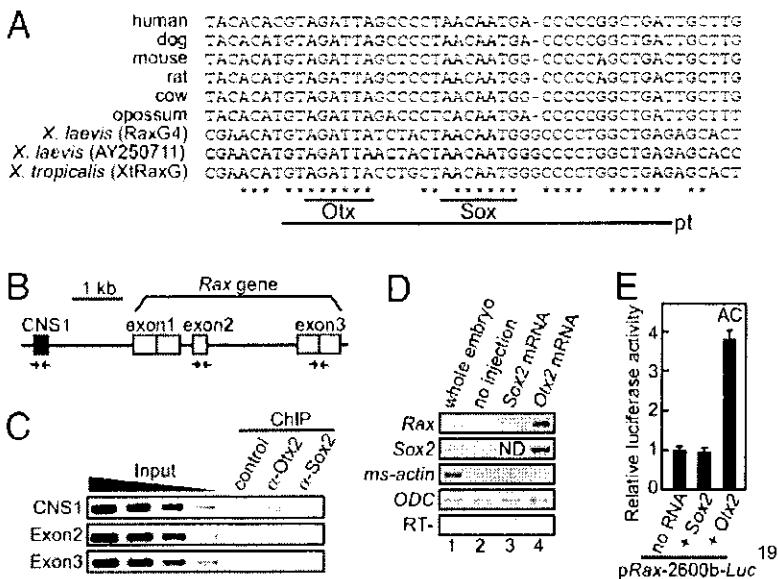
aCGH, array comparative genome hybridization; CNV, copy number variation; CNVR, copy number variation region.

<sup>1</sup>52 additional CNVRs were found to map to unknown chromosomal regions.

<sup>2</sup>Partial genome scan comprising chromosomes 4, 7, 14 and 17.



## SEQUENCES FOR THOUSANDS OF ANIMALS (WITHIN SPECIES) COMING SOON



Meuwissen, Hayes and Goddard (2001)

“Genomic selection”

Better terms:

“Genome-enabled selection”

“Genome-assisted selection”

SNP effects combined additively

$$y = \mu I_n + (\sum_i X_i g_i) + e,$$

Effect of chromosomal segment, allelic, haplotype

**ANIMAL BREEDING:  
USE ALL SNP MARKERS IN MODELS  
FOR GENOMIC-ASSISTED EVALUATION**

QUESTION: BYE-BYE QTLS, PEDIGREES, GENES?

20

## Essentials of genome-enabled prediction and selection

- Fit (train) some regression model (typically Bayesian) to a data set with markers and phenotypes
- Estimate marker effects
- Predict marked genetic value or phenotype in a new sample (testing or validation sample) for which only DNA information is available
- Once phenotype (or something related to phenotype) is observed, asses quality of prediction. For example, calculate predictive correlation or mean squared error of prediction (**choice of metric?**)
- Objective: gain reliability and if new sample is of juveniles, reduce generation interval. Dispense with progeny testing? Reduce frequency of phenotyping?

21

### TYPES OF QUESTIONS TO BE ANSWERED (*Drosophila*) OBER et al. (2012, Plos Genetics, forthcoming)

Implementing genomic prediction with full genome sequence data raises a number of questions. What is the most efficient way to incorporate the complete genomic information in prediction? How much predictive ability is gained by using whole genome sequence data compared to high density SNP panels? Is it possible to increase predictive ability by a pre-selection of SNPs or models with an internal feature selection? How comparable are the results of genomic prediction and genome wide association? Here, we address these questions empirically based on full genomic sequences of a population of *Drosophila melanogaster* inbred lines. The inbred lines have been sequenced, and constitute the Drosophila Genetics Reference Panel (DGRP), a new community resource for genetic studies of complex traits [27].

We report the results of a full sequence based genomic prediction for two quantitative traits, starvation stress resistance and locomotor startle response, both of which display considerable genetic variation in natural populations and respond rapidly to artificial selection [28–30]. We used whole-genome sequences determined on the Illumina platform for 157 (155) DGRP-lines for starvation resistance (startle response) [27]. Our reference method is a GBLUP approach in which ~ 2.5 million polymorphic SNPs are used to derive a genomic relationship matrix [8]. We evaluated predictive ability via cross-validation (CV), and compared prediction within vs. across sexes, various SNP densities, and training set sizes. We assessed whether BayesB is superior over GBLUP given full genome sequence data [26], and compared our genomic prediction results with those of GWAS conducted on the same DGRP lines [27].

## CROSS-VALIDATION

- Data available (genomic, phenotypic)
- Data generated according to unknown process
- Split into training (fitting)- testing (predictand) sets
- Fitting process essentially describes current data (model is typically wrong)
- Use training process to make statement about yet-to-be observed data (testing set)
- Prediction error (conditional and unconditional): point estimate
- Distribution of prediction errors (conditional or unconditional): interval estimate

23

### BREEDERS: FUNDAMENTAL THEOREM OF NATURAL SELECTION → additive effects

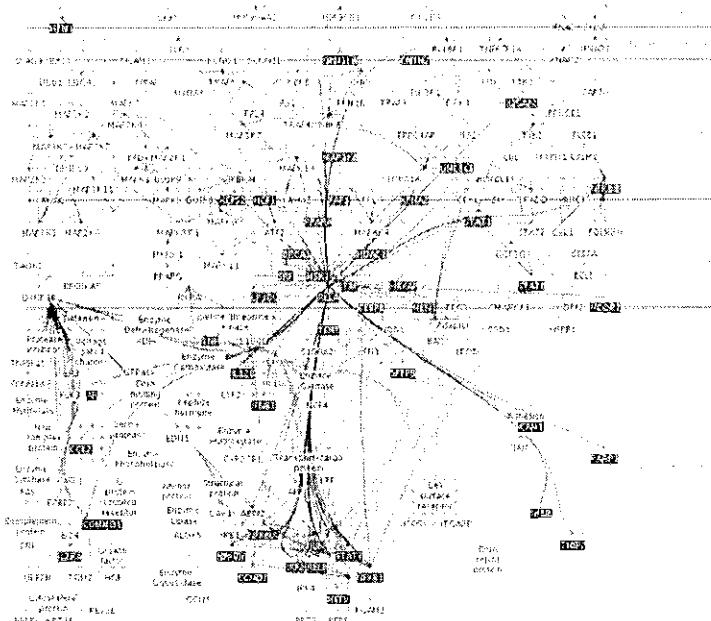
Schaeffer (2006):

A potential drawback of genome-wide selection may be the existence of interactions or epistatic effects between QTL. If epistatic effects are large, then the accuracy of GEBV may never reach 0.75. A statistical model could be written to account for interactions, but this would likely be very difficult to compute.

YES, IT WOULD BE DIFFICULT!  
SEE NEXT...

24

COULD WE WRITE A MODEL FOR SOMETHING LIKE THIS?  
A SYSTEMS BIOLOGY MAP OF THE BRAIN



Structuralism? Systems analysis?



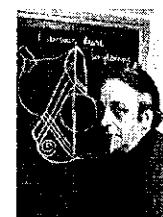
Levi-Strauss  
(1908-2009)



Lacan  
(1901-1981)



Foucault  
(1926-1984)



Althusser  
(1918-1990)



## Will “systems biology” help?

- von Bertalanffy (1968) wrote:

### Allgemeine Systemtheorie

“There exist models, principles, and laws that apply to generalized systems or their subclasses, irrespective of their particular kind, the nature of their component elements, and the relation or ‘forces’ between them.

It seems legitimate to ask for a theory, not of systems of a more or less special kind, but of universal principles applying to systems in general.

In this way we postulate a new discipline called *General System Theory*. Its subject matter is the formulation and derivation of those principles which are valid for ‘systems’ in general.

Concepts like those of organization, wholeness, directiveness, teleology, and differentiation are alien to conventional physics. However, they pop up everywhere in the biological, behavioural and social sciences, and are, in fact, indispensable for dealing with living organisms or social groups. Thus, a basic problem posed to modern science is a general theory of organization.

General system theory is, in principle, capable of giving exact definitions for such concepts and, in suitable cases, of putting them to quantitative analysis...

## Systems analysis is not new in the animal sciences...

### MODELING BEEF PRODUCTION SYSTEMS<sup>1</sup>

G. E. Joandet<sup>2</sup> and T. C. Cartwright

Texas A&M University, College Station

JOURNAL OF ANIMAL SCIENCE, Vol. 41, No. 4, 1975

### THE USE OF SYSTEMS ANALYSIS IN ANIMAL SCIENCE WITH EMPHASIS ON ANIMAL BREEDING<sup>1</sup>

T. C. Cartwright<sup>2</sup>

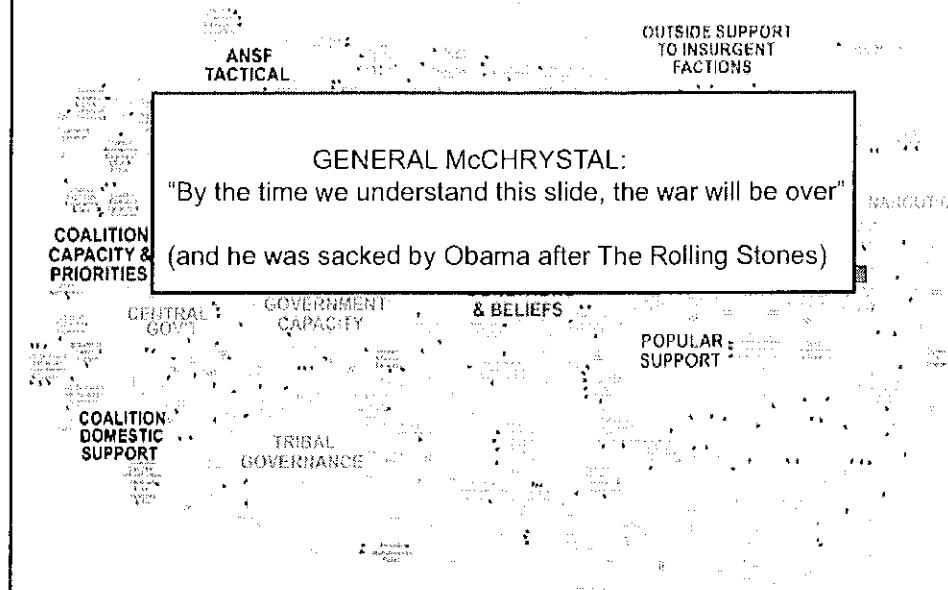
Texas A&M University, College Station 77843

JOURNAL OF ANIMAL SCIENCE, Vol. 49, No. 3 (1979)

## Where is the beef?

28

WHAT CAN WE EXPECT FROM SYSTEM ANALYSIS?  
SYSTEMS ANALYSIS IN ACTION: PENTAGON "SYSTEMS" VIEW OF  
THE WAR IN AFGHANISTAN



Dealing with epistatic interactions  
and non-linearities

gene x gene

gene x gene x gene

gene x gene x gene x gene

.....

(Alice in Wonderland)

## Fixed effects models (unravelling “physiological epistasis” a la Cheverud?)

- Lots of “main effects”
- Splendid non-orthogonality
- Lots of 2-factor interactions
- Lots of 3-factor interactions
- Lots of non-estimability
- Lots of uninterpretable high-order interactions
- Run out of “degrees of freedom”



Epistatic networks will probably involve a few genes of large effect

## Example of epistatic network

Old fashioned, Ford-T car

Modern Swedish car

Say one knows genes A, B, C, D. Do ANOVA:

A

B

C

D

AB → Significant at 0.05

AC → Significant at 0.01

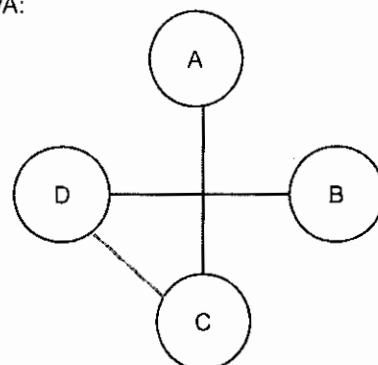
AD

BC

BD → Significant at 0.01

CD → Significant at 0.001

Yawn. nobody will publish...



Publish in Nature and claim  
new paradigm for epistasis 32



## RANDOM EFFECTS MODELS FOR ASSESSING EPISTASIS REST ON: Cockerham (1954) and Kempthorne (1954)

--Orthogonal partition of genetic variance into additive, dominance, additive x additive, etc. **ONLY** if

- No selection
- No inbreeding
- No assortative mating
- No mutation
- No migration
- Linkage equilibrium

Just consider  
Linkage disequilibrium

ALL  
ASSUMPTIONS  
VIOLATED!

33

## A VIEW OF LINEAR MODELS (as employed in q. genetics)

Mathematically, can be viewed as a "local" approximation of a complex process

$$f(x) = f(a) + f'(a)(x - a) + \underbrace{\frac{f''(a)}{2!}(x - a)^2 + \frac{f'''(a)}{3!}(x - a)^3 + \dots}_{\text{Linear approximation}} + \frac{f^{(n)}(a)}{n!}(x - a)^n + \dots$$

Linear approximation

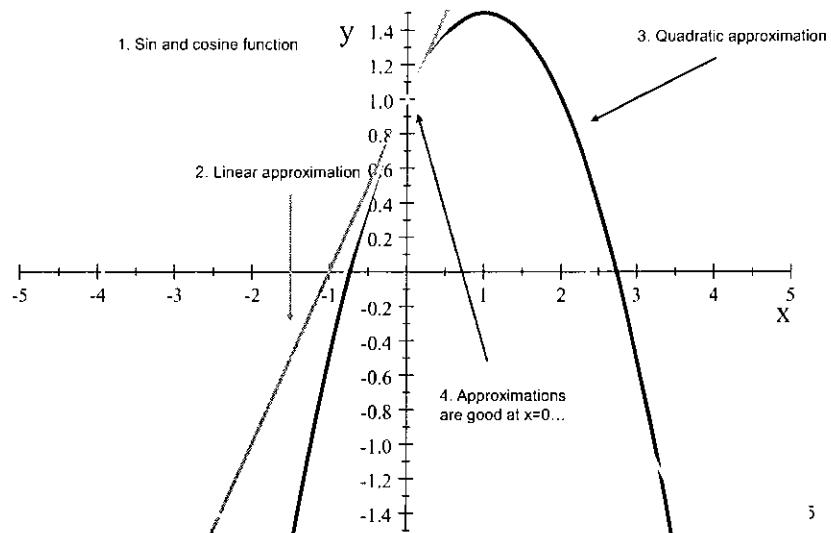
Quadratic approximation

$n^{\text{th}}$  order approximation

FELDMAN and LEWONTIN (1975)  
CHEVALET (1994)

How good are linear and quadratic approximations? A Taylor series provides a local approximation only...

$$y = g(x) + e \quad g(x) = \sin(x) + \cos(x)$$



### CLOSE ENCOUNTERS OF THE PREHISTORIC KIND

Homo sapiens

Neanderthal

GENOMICS AND  
COMPLEX BIOLOGY

NO! THE ADDITIVE  
GENETIC MODEL

36

A prevailing view, and for good reasons  
(Hill et al., 2008; Crow, 2010; Hill, 2010)

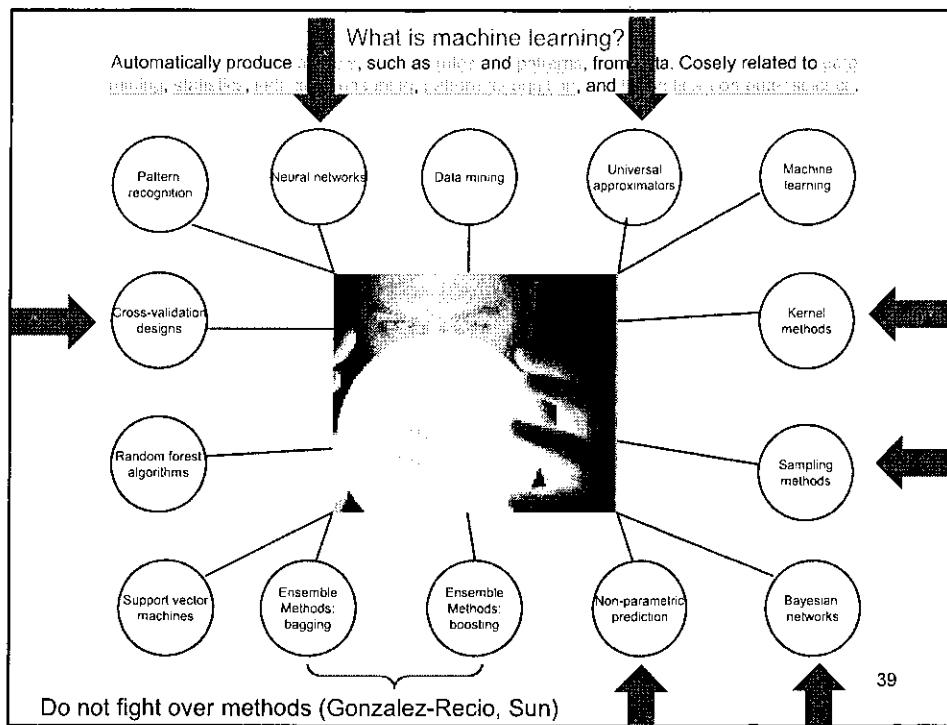
- Fisher's theorem of natural selection
- Interactions are second-order effects; likely tiny and hard to detect
- Epistasis probably arises with genes of large effects, unlikely to be observed in outbred populations
- Epistatic systems generate additive variance and “release” it, so why worry?

37

A much less popular view  
(Gianola and a few others)

- If everything behaves as additive, can additive models allow us to learn about “genetic architecture”?
- In areas where phenotypic prediction is crucial (medicine, precision mating) can the exploitation of interaction have added value?
- Is so, should we consider enriching our battery of statistical tricks?

38



## Distinctive aspects of non-parametric fitting

- Investigate patterns free of strictures imposed by parametric models
- Regression coefficients appear but (typically) do not have an obvious interpretation
- Often: very good predictive performance in cross-validation
- Tuning methods and algorithms (maximization, MCMC) similar to those of parametric methods
- Often produce surprising results

40

## PENALIZED and BAYESIAN METHODS for functional inference play a role

- The idea of “penalty is ad-hoc
- It does not arise “naturally” in classical inference
- It appears very naturally in Bayesian inference
  - $L_2$  penalty: equivalent to Gaussian prior
  - $L_1$  penalty: equivalent to double exponential prior
  - Penalties on covariance matrices equivalent to priors (e.g., inverse Wishart)



Bayesian methods arise naturally in predictive inference

41

The concept of penalized likelihood  
(example: ridge regression viewed from this perspective)

$$y = X\beta + e; e \sim N(0, I\sigma_e^2)$$

$$SSR = (y - X\beta)'(y - X\beta)$$

$$L(\beta|y) \sim \exp\left[-\frac{(y - X\beta)'(y - X\beta)}{2\sigma_e^2}\right]$$

$$\text{Penalty} \sim \exp\left[-\frac{\beta'\beta}{2\sigma_\beta^2}\right]$$

$$\text{Penalized likelihood} \sim \exp\left[-\frac{(y - X\beta)'(y - X\beta)}{2\sigma_e^2}\right] \exp\left[-\frac{\beta'\beta}{2\sigma_\beta^2}\right]$$

Penalized sum of squares =  $-2 \log[\text{Penalized likelihood}]$

$$= \frac{(y - X\beta)'(y - X\beta)}{2\sigma_e^2} + \frac{\beta'\beta}{2\sigma_\beta^2}$$

42

Ridge regression estimator obtained by minimizing penalized SS over  $\beta$

$$\frac{\partial(\text{Penalized sum of squares})}{\partial \beta} = -X' \frac{(y - X\beta)}{\sigma_e^2} + \frac{\beta}{\sigma_\beta^2}$$

$\Rightarrow$  Set to 0

$$\left( X'X + I \frac{\sigma_e^2}{\sigma_\beta^2} \right) \hat{\beta} = X'y$$

$$\hat{\beta} = (X'X + I\lambda)^{-1}X'y; \quad \lambda = \frac{\sigma_e^2}{\sigma_\beta^2}$$

Verify minimum:

$$\frac{\partial^2(\text{Penalized sum of squares})}{\partial \beta \partial \beta'} = \left( \frac{X'X}{\sigma_e^2} + \frac{I}{\sigma_\beta^2} \right) = \left( X'X + I \frac{\sigma_e^2}{\sigma_\beta^2} \right) \sigma_e^2$$

Positive-definite  $\Rightarrow$  minimum

43

### The concept of penalized likelihood (example in the mixed linear model)

$y = X\beta + Zu + e$ $y \beta, u, R \sim N(X\beta + Zu, R)$ $u \sim N(\mathbf{0}, G)$
--

$$p(y|\beta, u, R) = \frac{1}{(2\pi)^{\frac{N}{2}} |R|^{\frac{1}{2}}} \exp \left[ -\frac{1}{2} (y - X\beta - Zu)' R^{-1} (y - X\beta - Zu) \right]$$

$$p(u|G) = \frac{1}{(2\pi)^{\frac{q}{2}} |G|^{\frac{1}{2}}} \exp \left[ -\frac{1}{2} u' G^{-1} u \right]$$

Assuming known variance components, the log of the joint density of the data and random effects is termed "penalized likelihood"

$$\rightarrow l(\beta, u | y, R, G) = K - \frac{1}{2}(y - X\beta - Zu)'R^{-1}(y - X\beta - Zu) - \frac{1}{2}u'G^{-1}u$$

$$-2l(\beta, u | y, R, G) = K + (y - X\beta - Zu)'(y - X\beta - Zu) + u'G^{-1}u \quad \text{Penalized SS}$$

$$\rightarrow \frac{\partial l(\beta, u | y, R, G)}{\partial \beta} = X'R^{-1}(y - X\beta - Zu)$$

$$\frac{\partial l(\beta, u | y, R, G)}{\partial u} = Z'R^{-1}(y - X\beta - Zu) - G^{-1}u$$

Setting the derivatives to 0 yields

$$\begin{bmatrix} X'R^{-1}X & X'R^{-1}Z \\ Z'R^{-1}X & Z'R^{-1}Z + G^{-1} \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} X'R^{-1}y \\ Z'R^{-1}y \end{bmatrix}$$

- The solution to these equations produces the "maximum penalized likelihood" estimates of  $\beta$  and  $u$
- These solutions are also the BLUE( $\beta$ ) and BLUP( $u$ )

# OVERVIEW OF BAYESIAN INFERENCE

Rev. Thomas Bayes

1702 London, England

1761 Tunbridge Wells, Kent, England

1763. "An essay towards solving a problem in the doctrine of chances".  
*Philosophical Transactions of the Royal Society of London* 53, 370-418.



Pierre-Simon Laplace

1749 Beaumont-en-Auge, France  
1827 Paris, France

1774. "Mémoire sur la probabilité des causes par les événements".<sup>1</sup>  
*Savants étrangers* 6, 621-656. *Oeuvres* 8, 27-65

## HISTORICAL NOTES

- Karl Pearson (without knowing) used Bayes
- Fisher (likelihood, fiducial inference)
- Lack of admissibility of classical procedures (James-Stein)
- Revival: Neo-Bayesianism (Lindley, Box, Zellner)
- MCMC procedures (Metropolis, Geman and Geman)
- Bayesian methods in genetics: Haldane (1948), Dempfle (1977), Gianola and Fernando (1986)
- Explosion of Bayesianism in statistics: Gelfand and Smith (1990)
- Explosion in genetics as well

2

## Bayesian methods in Genetics: today

- Classification of genotypes
- Molecular evolution
- Linkage mapping
- QTL cartography
- Genetic risk analysis
  - Gaussian linear and non-linear models
    - (cross-sectional)
    - (longitudinal univariate)
    - (multivariate)
  - Generalized linear models
  - Survival analysis
  - Thick-tailed processes
  - Mixtures
  - Semi-parametrics
  - Transcriptional analysis
  - Structural equation modeling
  - Bayesian proteomics with wavelets
  - Methods for genomic selection
    - (the Bayesian Alphabet -- A, B, C-pi, L... and more)
  - Bayesian non-parametrics (Dirichlet process priors)

RED: animal breeders

3

## THE BAYESIAN APPROACH IN A NUTSHELL

- All unknowns in statistical system treated as random
- Randomness reflects (typically) **subjective** uncertainty
- Can include as unknowns:
  - The model (distribution, functional form)
  - Its parameters (heritability, inbreeding coefficient)
  - Genetic effects, number of QTL loci, marker effects
  - Combine with what is known a priori with information from data: Bayesian learning
- Bayesian approach can also be used for developing predictors of future observations without taking inference too seriously

4

## HOW DOES ONE DO THIS?

- Introduce a prior distribution for all unknowns (PRIOR)
- Define a distribution for the data under a certain model (LIKELIHOOD)
- Arrive at conditional distribution of all unknowns given data (POSTERIOR)
- Derive marginal or conditional posterior distributions of interest by standard probability theory
- Display summaries or entire distribution
- Interpret results probabilistically
- Example: the posterior probability of  $H_0$  is 8%

5

## BAYES THEOREM: DISCRETE

- $M$  disjoint hypotheses about some mechanism. Assign probabilities to the events "the hypothesis is true":

$$0 \leq p(H_i) \leq 1, \quad i = 1, 2, \dots, M,$$

$$p(H_i \cap H_j) = 0, \quad i \neq j,$$

$$\sum_{i=1}^M p(H_i) = 1.$$

Mutually exclusive

Exhaustive

6

- N observable effects. Given that a hypothesis holds, one observes events with probabilities

$$0 \leq p(E_i|H) \leq 1, \quad i = 1, 2, \dots, M, \quad j = 1, 2, \dots, N,$$

$$p(E_j \cap (E_i, H)) = 0, \quad j \neq i,$$

$$\sum_{i=1}^M p(E_i|H) = 1.$$

Probability distribution of events under hypothesis  
("likelihood")

7

- Assume that events E and the hypotheses H have joint distribution:

$$p(H = H_i, E = E_j) = p(E_j|H_i)p(H_i)$$

- The conditional probability that a hypothesis holds, given the observed effects is:

**Bayes theorem**

$$p(H_i|E_j) = \frac{p(H_i, E_j)}{p(E_j)} = \frac{p(E_j|H_i)p(H_i)}{p(E_j)}$$

Posterior probability

Prior probability

Likelihood

Marginal distribution of data

$$p(E_j) = \sum_{i=1}^M p(E_j|H_i)p(H_i) = E_H[p(E_j|H)]$$

## THE "PROPORTIONAL TO" REPRESENTATION:

$$p(H_i|E_j) = \frac{p(E_j|H_i)p(H_i)}{E_H[p(E_j|H_i)]}$$

$\times p(E_j|H_i)p(H_i).$

The "Pac-Man" operator: eats anything that does not depend on  $H_i$ .

48

## BAYESIAN LEARNING: DISCRETE

- Let 2 bits of evidence accumulate. Then:

$$p(H_i|E'_j, E_j) = \frac{p(E'_j, E_j|H_i)p(H_i)}{E_H[p(E'_j, E_j|H_i)]}$$

$\times p(E'_j, E_j|H_i)p(H_i)$

$\times p(E'_j|E_j, H_i)p(E_j|H_i)p(H_i)$

$\times p(E'_j|E_j, H_i)p(H_i|E_j).$

- Prior before bit 2 is posterior after bit 1

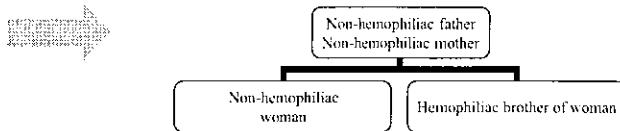
- If, given the hypothesis, the 2 bits of evidence are independent:

$$p(E'_j|E_j, H_i)p(E_j|H_i) = \underbrace{p(E'_j|H_i)p(E_j|H_i)}_{\text{Conditional independence}} \quad 49$$

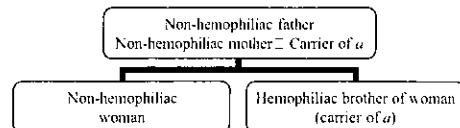
### EXAMPLE OF DISCRETE PROBLEM: HEMOPHILIA IN HUMANS

(1995). Hemophilia is a genetic disease in humans. The locus responsible for its expression is located in the sex chromosomes (these are denoted as XX in women, and XY in men). The condition is observed in women only in double recessive individuals (aa), and in men that are carriers of the a allele in the X-chromosome.

DATA



IMPLICATION

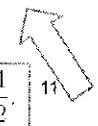


**Problem:** Probability(woman is carrier) = Probability( $\theta=1$ ) ?

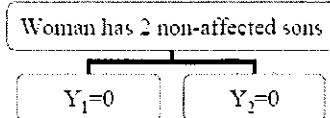
PRIOR IMPLIED BY THIS INFORMATION

(mother must be  $Aa$ )

$$\Pr(\theta = 1) = \Pr(\theta = 0) = \frac{1}{2}$$



### MORE DATA BECOME AVAILABLE



Given that  $\theta = 1$ , the probability of the observed data is

$$\begin{aligned} & \Pr(Y_1 = 0, Y_2 = 0 | \theta = 1) \\ &= \Pr(Y_1 = 0 | \theta = 1) \Pr(Y_2 = 0 | \theta = 1) = \left(\frac{1}{2}\right)^2 \times \left(\frac{1}{2}\right) \end{aligned}$$

On the other hand, if she is not a carrier ( $\theta = 0$ ),

$$\begin{aligned} & \Pr(Y_1 = 0, Y_2 = 0 | \theta = 0) \\ &= \Pr(Y_1 = 0 | \theta = 0) \Pr(Y_2 = 0 | \theta = 0) = 1 \times 1 = 1. \end{aligned}$$

THE DATA CONFER 4 TIMES MORE LIKELIHOOD TO NON-CARRIER HYPOTHESIS

## POSTERIOR PROBABILITIES

$$\begin{aligned}
 \Pr(\theta = 1|Y_1 = 0, Y_2 = 0) &= \frac{\Pr(\theta = 1)\Pr(Y_1 = 0, Y_2 = 0|\theta = 1)}{\Pr(Y_1 = 0, Y_2 = 0)} \\
 &= \frac{\Pr(\theta = 1)\Pr(Y_1 = 0, Y_2 = 0|\theta = 1)}{\sum_{i=0,1} \Pr(\theta = i)\Pr(Y_1 = 0, Y_2 = 0|\theta = i)} \\
 &= \frac{\frac{1}{2}\frac{1}{2}}{\frac{1}{2}\frac{1}{2} + \frac{1}{2}\frac{1}{2}} = \frac{1}{2}
 \end{aligned}$$

and

$$\Pr(\theta = 0|Y_1 = 0, Y_2 = 0) = 1 - \frac{1}{2} = \frac{1}{2}.$$

- WE MOVED FROM 0.5; 0.5 TO 0.2; 0.8
- SHARPER STATE OF KNOWLEDGE BUT CANNOT RULE OUT HYPOTHESIS WOMAN IS A CARRIER
- STILL UNCERTAINTY...MORE DATA NEEDED

13

EVEN MORE DATA BECOME AVAILABLE...

Woman has 3 non-affected sons



Using posterior  
as prior for new data.  
assuming conditional  
independence

$$\begin{aligned}
 \Pr(\theta = 1|Y_1 = 0, Y_2 = 0, Y_3 = 0) &= \frac{\frac{1}{2}\Pr(Y_3 = 0|\theta = 1)}{\frac{1}{2}\Pr(Y_3 = 0|\theta = 1) + \frac{1}{2}\Pr(Y_3 = 0|\theta = 0)} \\
 &= \frac{\frac{1}{2}\frac{1}{2}}{\frac{1}{2}\frac{1}{2} + \frac{1}{2}\frac{1}{2}} = \frac{1}{9}
 \end{aligned}$$

Using prior before  
Any progeny data.  
And combining all  
information

$$\Pr(\theta = 1|Y_1 = 0, Y_2 = 0, Y_3 = 0) = \frac{\frac{1}{2} \cdot (\frac{1}{2})^3}{\frac{1}{2} \cdot (\frac{1}{2})^3 + \frac{1}{2} \cdot (1)^3} = \frac{1}{9}$$

WOMAN COULD STILL BE A CARRIER!

$$\begin{aligned}
 \Pr(\theta = 1 | Y_1 = 0, Y_2 = 0, \dots, Y_N = 0) &= \frac{\Pr(\theta = 1) \left(\frac{1}{2}\right)^n}{\Pr(\theta = 1) \left(\frac{1}{2}\right)^n + \Pr(\theta = 0) 1^n} \\
 &= \frac{\left(\frac{1}{2}\right)^n}{\left(\frac{1}{2}\right)^n + \frac{\Pr(\theta=0)}{\Pr(\theta=1)}} \\
 &= \frac{1}{1 + \frac{\Pr(\theta=0)}{\Pr(\theta=1)} 2^n}
 \end{aligned}$$

TENDS TO 0 AS  $n$  GOES TO INFINITY. HOWEVER.

$$\Pr(\theta = 1 | Y_1 = 0, Y_2 = 0, \dots, Y_N = 0, Y_{N+1} = 1) = 1$$

IF WOMAN HAS AT LEAST ONE HEMOPHILIAC SON.

54

## BAYES THEOREM: CONTINUOUS

- Evidence is now given by a vector of observations  $y$
- Hypothesis is a vector of unknowns  $\theta$
- A probability model  $M$  poses joint distribution  $[\theta, y | M]$  with density

$$h(\theta, y) = g(\theta)f(y|\theta) = m(y)p(\theta|y)$$

- Assume that both the unknowns and the parameter are continuous-valued

55

## BAYES THEOREM IN A NUTSHELL

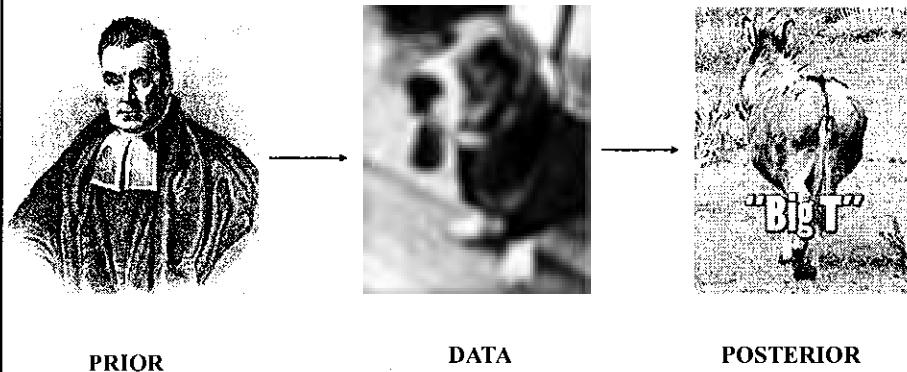
$$p(\theta|y) = \frac{g(\theta)f(y|\theta)}{m(y)} \propto g(\theta)f(y|\theta)$$

Prior density  
Likelihood function  
Marginal data density

Posterior density

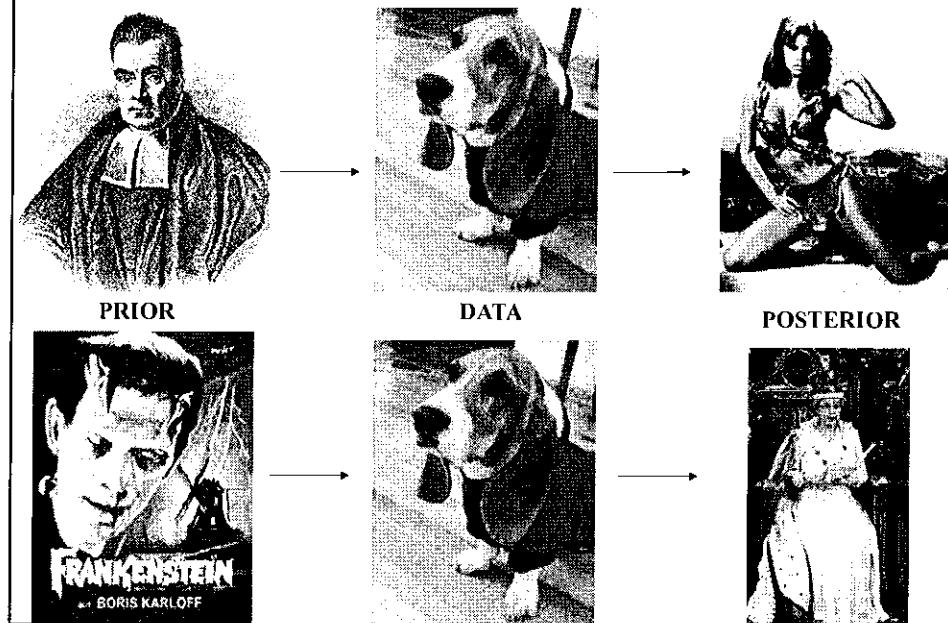
17

## BAYESIAN INFERENCE and MCMC (can fit any model)



*Most of the times the prior comes "out of the blue"* 18

THUS: THE ANTI-BAYESIAN ARGUMENT...



## LARGE SAMPLES

- “Asymptotic domination” of the prior by the data (likelihood)
- For parameters on which there is a lot of information from the data, the prior matters little
- Prior may be influential in small samples; worthwhile to investigate sensitivity
- What is a small sample?
- Even if prior matters little, Bayesian approach allows to use probability theory to measure uncertainty

What about asymptotics in situations where  $n \ll p$ , and where there are strong non-linearities? Can one learn about marker effects?<sup>20</sup>

Singly manne  
method is a  
biased method  
because manne  
are not orthogonal  
overfitting) and  
not of multiple  
test correctly  
(FDL, ...), work

CAN ONE ESTIMATE MARKER EFFECTS FROM DATA WHEN  $n \ll p$  WITHOUT BRINGING EXTERNAL INFORMATION?



$$y = X\beta + e; e \sim N(0, I\sigma_e^2)$$

$$L(\beta|y) \sim \exp \left[ -\frac{(y - X\beta)'(y - X\beta)}{2\sigma_e^2} \right]$$

$$\log[L(\beta|y)] = -\frac{(y - X\beta)'(y - X\beta)}{2\sigma_e^2}$$



If  $p \gg n$

$$\text{Information}(\beta)_{p \times p} = -\frac{\partial^2}{\partial \beta \partial \beta'} \log[L(\beta|y)] = -\frac{\partial}{\partial \beta'} \left[ -X' \frac{(y - X\beta)}{\sigma_e^2} \right] = \frac{X'X}{\sigma_e^2}$$

$$\text{Expected Information}(\beta)_{p \times p} = \frac{X'X}{\sigma_e^2} \Rightarrow \text{Does not have full-rank}$$

$\Rightarrow$  MARKER EFFECTS ARE NOT INDIVIDUALLY ESTIMABLE

**VERY IMPORTANT TO KEEP IN MIND**

21

### ORDINARY LEAST-SQUARES (MAXIMUM LIKELIHOOD UNDER NORMALITY)

"Full model"



$$y = X\beta + e$$

$$= X_1\beta_1 + X_2\beta_2 + e$$

Rank( $X_1, X_2$ ) =  $p \leq n$

"OLS" estimator



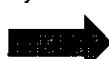
$$\begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix} = \begin{bmatrix} X'_1 X_1 & X'_1 X_2 \\ X'_2 X_1 & X'_2 X_2 \end{bmatrix}^{-1} \begin{bmatrix} X'_1 y \\ X'_2 y \end{bmatrix}$$

$$= [X'X]^{-1} X'y$$

$$E(\hat{\beta}|X) = [X'X]^{-1} X'E(y)$$

$$= [X'X]^{-1} X'X\beta = \beta$$

"OLS" is biased If full model holds and one fits "smaller" model (e.g., single marker Regressions, or just additive effects)



$$y = X_1\beta_1 + e$$

$$E(\hat{\beta}_1|X_1) = (X'_1 X_1)^{-1} E(y)$$

$$= (X'_1 X_1)^{-1} [X_1\beta_1 + X_2\beta_2]$$

$$= \beta_1 + (X'_1 X_1)^{-1} X'_1 X_2 \beta_2$$

Typical of GWAS

Example:  $n=4, p=5$

$$\rightarrow X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \end{bmatrix}$$

$$\rightarrow X^T X = \begin{bmatrix} 4 & 6 & 14 & 36 & 98 \\ 6 & 14 & 36 & 98 & 276 \\ 14 & 36 & 98 & 276 & 794 \\ 36 & 98 & 276 & 794 & 2316 \\ 98 & 276 & 794 & 2316 & 6818 \end{bmatrix}$$

$\rightarrow |X^T X| = 0$  LEAST-SQUARES DOES NOT ALLOW  
ESTIMATION OF INDIVIDUAL REGRESSIONS  
IN THE  $n << p$  SITUATION

Example again:  $n=4, p=5$  BUT ADD 5 TO DIAGONALS OF  $X^T X$

$$\rightarrow X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \end{bmatrix}$$

$$\rightarrow X^T X = \begin{bmatrix} 4 & 6 & 14 & 36 & 98 \\ 6 & 14 & 36 & 98 & 276 \\ 14 & 36 & 98 & 276 & 794 \\ 36 & 98 & 276 & 794 & 2316 \\ 98 & 276 & 794 & 2316 & 6818 \end{bmatrix} X^T X + I \times 5 = \begin{bmatrix} 9 & 6 & 14 & 36 & 98 \\ 6 & 19 & 36 & 98 & 276 \\ 14 & 36 & 103 & 276 & 794 \\ 36 & 98 & 276 & 799 & 2316 \\ 98 & 276 & 794 & 2316 & 6823 \end{bmatrix}$$



Determinant= 0

Determinant= 25782105

## RIDGE REGRESSION

Classical view: "estimator" of a fixed vector of regression coefficients

$$\begin{aligned}
 \hat{\beta}_{Ridge} &= [X'X + I\lambda]^{-1} X'y \\
 &= [I + (X'X)^{-1}\lambda]^{-1} (X'X)^{-1} X'y \\
 &= [I + (X'X)^{-1}\lambda]^{-1} \hat{\beta}_{OLS} \\
 E(\hat{\beta}_{Ridge}|X) &= [I + (X'X)^{-1}\lambda]^{-1} E(\hat{\beta}_{OLS}) \\
 &= [I + (X'X)^{-1}\lambda]^{-1} \beta
 \end{aligned}$$

Can assess by cross-validation  
Shrinkage towards 0  
Biased estimator but more precise

Classical view: "predictor" of a random vector of regression coefficients  $\beta \sim (0, \sigma_\beta^2)$ .

$$\hat{\beta} = \text{BLUP}, E(\hat{\beta}) = E(\beta), \text{ taking } \lambda = \frac{\sigma_\epsilon^2}{\sigma_\beta^2} \text{ as known}$$

Bayesian view: Mean of posterior distribution of regressions under prior  $\beta \sim (0, \sigma_\beta^2)$ , normal likelihood and known variance parameters

## EXAMPLE OF CONTINUOUS CASE

Inferring the Poisson parameter (ML)

N independent samples

$$p(y_1, y_2, \dots, y_N | \lambda) = \frac{\lambda^{\sum y_i} e^{-N\lambda}}{\prod y_i!} \quad \xrightarrow{\text{likelihood}} \quad l(\lambda|y) \propto \lambda^{\sum y_i} e^{-N\lambda}$$

$$\xrightarrow{\text{Log-likelihood}} \quad L(\lambda|y) = K + \sum y_i \log(\lambda) - N\lambda$$

$$\frac{dL(\lambda|y)}{d\lambda} = \frac{\sum y_i}{\lambda} - N$$

$$\boxed{MLE(\lambda) = \frac{\sum y_i}{N}}$$

$$-E \frac{d^2 L(\lambda|y)}{(d\lambda)^2} = E\left(\frac{\sum y_i}{\lambda^2}\right) = \frac{N}{\lambda}$$

$$\boxed{\widehat{AsyVar}(\hat{\lambda}) = \frac{\hat{\lambda}}{N}}$$

## Inferring the Poisson parameter (Bayes)

Gamma prior  $\rightarrow p(\lambda|\alpha, \beta) = \frac{1}{\beta^\alpha \Gamma(\alpha)} \lambda^{\alpha-1} \exp\left(-\frac{\lambda}{\beta}\right)$

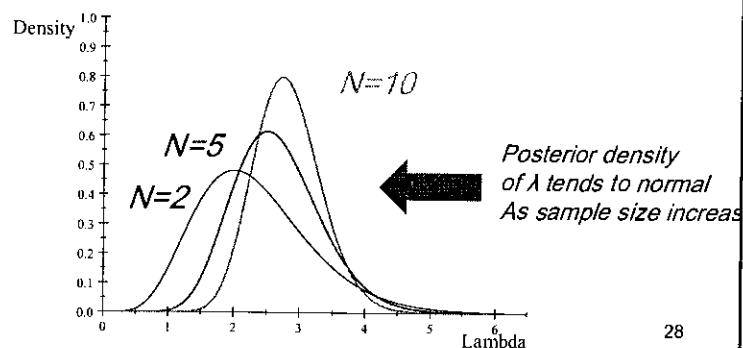
$$\begin{aligned}
 p(\lambda|y, \alpha, \beta) &\propto l(\lambda|y)p(\lambda|\alpha, \beta) \\
 &\propto \lambda^{\sum r_i} e^{-N\bar{y}} \lambda^{\alpha-1} \exp\left(-\frac{\lambda}{\beta}\right) \\
 &\propto \lambda^{\sum r_i + \alpha - 1} \exp\left[-\left(N + \frac{1}{\beta}\right)\lambda\right] \\
 &\propto \lambda^{N\bar{y} + \alpha - 1} \exp\left[-\frac{\lambda}{\left(\frac{\beta}{N\bar{y} + 1}\right)}\right]
 \end{aligned}$$

Posterior is Gamma as well (Conjugacy)

27

$$\begin{aligned}
 \lambda|y, \alpha, \beta &\sim \text{Gamma}\left(N\bar{y} + \alpha, \frac{N\beta + 1}{\beta}\right)^{-1} \\
 p(\lambda|y, \alpha, \beta) &= \frac{\left(\frac{N\beta + 1}{\beta}\right)^{N\bar{y} + \alpha}}{\Gamma(N\bar{y} + \alpha)} \lambda^{N\bar{y} + \alpha - 1} e^{-\left(\frac{N\beta + 1}{\beta}\right)\lambda}
 \end{aligned}$$

Suppose the mean of the observations is 3 and that  $N=2, 5, 10$ .  $\alpha=\beta=1$ . The posterior densities look like



$$\rightarrow E(\lambda|\alpha, \beta) = \alpha\beta; \quad Var(\lambda|\alpha, \beta) = \alpha\beta^2$$

$$E(\lambda|y, \alpha, \beta) = (N\bar{y} + \alpha)\left(\frac{\beta}{N\beta + 1}\right)$$

$$= \left(\frac{N\beta}{N\beta + 1}\right)\bar{y} + \left(\frac{1}{N\beta + 1}\right)\alpha\beta$$

$$= \left(\frac{N}{N + \frac{1}{\beta}}\right)\bar{y} + \left(\frac{\frac{1}{\beta}}{N + \frac{1}{\beta}}\right)\alpha\beta$$

$$\rightarrow Var(\lambda|\alpha, \beta) = (N\bar{y} + \alpha)\left(\frac{\beta}{N\beta + 1}\right)^2$$

$$= (N\bar{y} + \alpha)\left(\frac{1}{N + \frac{1}{\beta}}\right)^2$$

$$= N\left(\frac{1}{N + \frac{1}{\beta}}\right)^2 MLE(\lambda) + \left(\frac{1}{N + \frac{1}{\beta}}\right)^2 \alpha$$

- 1) Weighted ave.  
of MLE and prior mean  
2) When  $N$  goes to infinity, expectation tends to MLE.

$$\lim_{N \rightarrow \infty} Var(\lambda|\alpha, \beta) = \frac{MLE(\lambda)}{N}$$

Tends to AsyVar of MLE estimator

## Joint, Conditional and Marginal Posterior Distributions

- Put  $\theta = [\theta_1' \ \theta_2']'$  representing distinct features of models, (e.g., means and variances)

- Then, elicit a joint prior density

$$g(\theta_1, \theta_2) = g(\theta_1|\theta_2)g(\theta_2) = g(\theta_2|\theta_1)g(\theta_1)$$

where  $g(\theta_1)$  is the marginal prior and  $g(\theta_2|\theta_1)$  is a conditional prior

- Joint posterior density is

$$p(\theta_1, \theta_2|y) = \frac{L(\theta_1, \theta_2|y)g(\theta_1, \theta_2)}{\int \int L(\theta_1, \theta_2|y)g(\theta_1, \theta_2)d\theta_1 d\theta_2}$$

$$\propto L(\theta_1, \theta_2|y)g(\theta_1, \theta_2),$$

- Must decide which is the object of inference

- Joint, conditional or marginal posterior probability statements?

b<  
Conjoint  
and joint  
inference prior

## Marginal posterior densities

- Obtained directly from probability calculus as:

$$p(\theta_1|y) = \int p(\theta_1, \theta_2|y) d\theta_2$$

$$p(\theta_2|y) = \int p(\theta_1, \theta_2|y) d\theta_1$$

- Additional marginalizing may be needed if  $\theta_i = [\theta_{i1}, \theta_{i2}]^T$

$$\begin{aligned} p(\theta_{1,i}|y) &= \int \int p(\theta_1, \theta_2|y) d\theta_{1B} d\theta_2 \\ &= \int p(\theta_1|y) d\theta_{1B}. \end{aligned}$$

31

## Conditional posterior distributions

- By definition of conditional density:

$$p(\theta_1|\theta_2, y) = \frac{p(\theta_1, \theta_2|y)}{p(\theta_2|y)}$$

- Here, one is interested in variation about  $\theta_1$  only

$$\begin{aligned} p(\theta_1|\theta_2, y) &\propto p(\theta_1, \theta_2|y) \\ &\propto L(\theta_1, \theta_2|y) p(\theta_1, \theta_2) \\ &\propto L(\theta_1, \theta_2|y) p(\theta_1|\theta_2) \\ &\propto L(\theta_1|\theta_2, y) p(\theta_1|\theta_2). \end{aligned}$$

- Identifying conditional posterior distributions: important for implementing MCMC methods (sampling from posteriors)

32

## BAYESIAN LINEAR REGRESSION MODEL (normal distribution of residuals)

- MAKE DISTINCTION BETWEEN 2 SETS OF LOCATION PARAMETERS

$$\begin{aligned}
 & \text{Dummy variates} \quad \text{Treatment effects} \quad \text{regressions} \\
 & \downarrow \qquad \qquad \qquad \downarrow \\
 & y = X_1\beta_1 + X_2\beta_2 + e \\
 & = \begin{bmatrix} X_1 & X_2 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} + e = X\beta + e,
 \end{aligned}$$

regressors

Maximum likelihood (also least-squares) estimator:

$$\begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix} = C^{-1} \begin{bmatrix} X'_1 y \\ X'_2 y \end{bmatrix}$$

Vector of right-hand sides

Coefficient matrix →

$$C = \begin{bmatrix} X'_1 X_1 & X'_1 X_2 \\ X'_2 X_1 & X'_2 X_2 \end{bmatrix}$$

$$\begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix} \sim \left( \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}, \begin{bmatrix} X'_1 X_1 & X'_1 X_2 \\ X'_2 X_1 & X'_2 X_2 \end{bmatrix}^{-1} \sigma_e^2 \right)$$

If you left only  $X_1$  then  $(X'X)^{-1}$  would be smaller than

## LIKELIHOOD FUNCTION

• Under standard conditional independence

$$L(\beta_1, \beta_2, \sigma^2 | \mathbf{y}) \propto (\sigma^2)^{-\frac{n}{2}} \exp \left[ -\frac{(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)}{2\sigma^2} \right]$$

• Decompose

$$(\mathbf{y} - \mathbf{X}_1\beta_1 - \mathbf{X}_2\beta_2)'(\mathbf{y} - \mathbf{X}_1\beta_1 - \mathbf{X}_2\beta_2) = S_e + S_\beta$$

$$S_e = (\mathbf{y} - \mathbf{X}_1\hat{\beta}_1 - \mathbf{X}_2\hat{\beta}_2)'(\mathbf{y} - \mathbf{X}_1\hat{\beta}_1 - \mathbf{X}_2\hat{\beta}_2)$$

$$S_\beta = \begin{bmatrix} (\beta_1 - \hat{\beta}_1)' & (\beta_2 - \hat{\beta}_2)' \end{bmatrix} \mathbf{C} \begin{bmatrix} \beta_1 - \hat{\beta}_1 \\ \beta_2 - \hat{\beta}_2 \end{bmatrix}$$

### Inference using improper (flat) priors

$$p(\beta_1, \beta_2, \sigma^2 | \mathbf{y}) \propto (\sigma^2)^{-\frac{n}{2}} \exp \left[ -\frac{S_e + S_\beta}{2\sigma^2} \right]$$

Joint posterior is proportional to likelihood

#### a) Conditional posterior of coefficients, given variance

$$p(\beta_1, \beta_2 | \sigma^2, \mathbf{y}) \propto \exp \left[ -\frac{S_\beta}{2\sigma^2} \right]$$

$$p(\beta_1, \beta_2 | \sigma^2, \mathbf{y}) \propto \exp \left[ -\frac{\begin{bmatrix} (\beta_1 - \hat{\beta}_1)' & (\beta_2 - \hat{\beta}_2)' \end{bmatrix} \mathbf{C} \begin{bmatrix} \beta_1 - \hat{\beta}_1 \\ \beta_2 - \hat{\beta}_2 \end{bmatrix}}{2\sigma^2} \right]$$



“Similar” (but not same) as distribution of maximum likelihood (OLS) estimators under normality

$$\begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} \mid \sigma^2, \mathbf{y} \propto N\left(\begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{X}'_1 \mathbf{X}_1 & \mathbf{X}'_1 \mathbf{X}_2 \\ \mathbf{X}'_2 \mathbf{X}_1 & \mathbf{X}'_2 \mathbf{X}_2 \end{bmatrix}^{-1} \sigma^2\right)$$

b) Conditional posterior distribution of coefficients, given variance and other coefficients

$$\beta_1 | \beta_2, \sigma^2, \mathbf{y} \propto N(\tilde{\beta}_1, (\mathbf{X}'_1 \mathbf{X}_1)^{-1} \sigma^2)$$

$$\beta_2 | \beta_1, \sigma^2, \mathbf{y} \propto N(\tilde{\beta}_2, (\mathbf{X}'_2 \mathbf{X}_2)^{-1} \sigma^2)$$

$$\tilde{\beta}_i = (\mathbf{X}'_i \mathbf{X}_i)^{-1} \mathbf{X}'_i (\mathbf{y} - \underbrace{(\mathbf{X}'_j \beta_j)}_{\text{“offset”}}), \quad i = 1, 2, i \neq j.$$

c) Conditional posterior of individual coefficient, given the variance and all other coefficients

Normal, with mean and variance:

$$\tilde{\beta}_k = \frac{\mathbf{x}'_k (\mathbf{y} - \mathbf{X}_{-k} \beta_{-k})}{\mathbf{x}'_k \mathbf{x}_k}$$

↑ without parameter  $k$   
Column  $k$  of  $\mathbf{X}$

$$Var(\beta_k | \beta_{-k}, \sigma^2, \mathbf{y}) = \frac{\sigma^2}{\mathbf{x}'_k \mathbf{x}_k}$$

d) Conditional posterior of variance, given all coefficients

$$p(\sigma^2 | \beta_1, \beta_2, \mathbf{y}) \propto (\sigma^2)^{-\frac{n}{2}} \exp\left[-\frac{S_e + S_\beta}{2\sigma^2}\right]$$

$$p(\sigma^2 | \beta_1, \beta_2, \mathbf{y}) \propto (\sigma^2)^{-\left(\frac{n-2}{2}+1\right)} \exp\left[-\frac{S_e + S_\beta}{2\sigma^2}\right].$$

$$\sigma^2 | \beta_1, \beta_2, \mathbf{y} \sim (n-2) \left( \frac{S_e + S_\beta}{n-2} \right) \chi_{n-2}^{-2}$$

Curious loss  
of 2 d.f. (due to prior)

Wishart { scaled } dist.  
+ gamma

$$\rightarrow \sigma^2 |\beta_1, \beta_2, y \sim (n-2) \left( \frac{S_e + S_\beta}{n-2} \right) \chi_{n-2}^{-2}$$

$$E(\chi_v^{-2}) = \frac{1}{v-2}; Var(\chi_v^{-2}) = \frac{2v^2}{(v-2)^2(v-4)}$$

$$E(\sigma^2 |\beta_1, \beta_2, y) = (n-2) \left( \frac{S_e + S_\beta}{n-2} \right) E(\chi_{n-2}^{-2})$$

$$\rightarrow = \frac{S_e + S_\beta}{n-4}$$

$$Var(\sigma^2 |\beta_1, \beta_2, y) = \left[ (n-2) \left( \frac{S_e + S_\beta}{n-2} \right) \right]^2 \frac{2(n-2)^2}{(n-4)^2(n-6)}$$

$$\rightarrow = \frac{2(S_e + S_\beta)^2 (n-2)^2}{(n-4)^2(n-6)}$$

## MULTIVARIATE-t DISTRIBUTION

Let:

$$y|\mu, \Sigma, w \sim N(y|\mu, \frac{\Sigma}{w})$$

and

$$w \sim Ga(\frac{v}{2}, \frac{v}{2}); v > 0$$

Joint density:

$$\begin{aligned} p(y, w | \mu, \Sigma, v) &= p(y|\mu, \Sigma, w)p(w|v) \\ &= \left[ 2\pi \left( \frac{\Sigma}{w} \right) \right]^{-\frac{1}{2}} \exp \left[ -\frac{1}{2} (y - \mu)' \left( \frac{\Sigma}{w} \right)^{-1} (y - \mu) \right] \\ &\quad \times \frac{(v/2)^{\frac{v}{2}}}{\Gamma(v/2)} w^{\frac{v}{2}-1} \exp \left[ -\frac{vw}{2} \right]. \end{aligned}$$

Marginal density of  $\mathbf{y}$ :

$$p(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = |2\pi\boldsymbol{\Sigma}|^{-\frac{1}{2}} \frac{(\nu/2)^{\frac{\nu}{2}}}{\Gamma(\nu/2)} \\ \times \int_0^{\infty} w^{\frac{n+\nu}{2}-1} \exp\left[-w \frac{(\mathbf{y}-\boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{y}-\boldsymbol{\mu}) + \nu}{2}\right] dw.$$

Integrand is kernel of

$$Ga\left(w \mid \frac{n+\nu}{2}, \frac{(\mathbf{y}-\boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{y}-\boldsymbol{\mu}) + \nu}{2}\right) \\ \int_0^{\infty} w^{\frac{n+\nu}{2}-1} \exp\left[-w \frac{(\mathbf{y}-\boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{y}-\boldsymbol{\mu}) + \nu}{2}\right] dw \\ = \frac{\Gamma(\frac{n+\nu}{2})}{\left[\frac{(\mathbf{y}-\boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{y}-\boldsymbol{\mu}) + \nu}{2}\right]^{\frac{n+\nu}{2}}}.$$

Multivariate-t density:

$$p(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \frac{(\nu)^{\frac{\nu}{2}} \Gamma(\frac{n+\nu}{2})}{\Gamma(\frac{\nu}{2}) |\pi \boldsymbol{\Sigma}|^{\frac{1}{2}}} \left[ (\mathbf{y}-\boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{y}-\boldsymbol{\mu}) + \nu \right]^{-\frac{n+\nu}{2}} \\ = \boxed{\frac{\Gamma(\frac{n+\nu}{2})}{\Gamma(\frac{\nu}{2}) |\nu \pi \boldsymbol{\Sigma}|^{\frac{1}{2}}} \left[ 1 + \frac{(\mathbf{y}-\boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{y}-\boldsymbol{\mu})}{\nu} \right]^{-\frac{n+\nu}{2}}}.$$

Degrees of freedom

dimension

$$E(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \boldsymbol{\mu}$$

$$Var(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \frac{\nu}{\nu-2} \boldsymbol{\Sigma}$$

“Scale matrix”

Two variables ( $y_1, y_2$ ) could be uncorrelated but have bivariate t-distr.

All marginal and conditional distributions are multivariate or univariate t

Starting from

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} | \mu, \Sigma, w \sim N \left( \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \frac{1}{w} \right)$$

all marginal and conditional distributions are normal. Integration over

$$Ga \left( w \frac{n+v}{2}, \frac{(y-\mu)' \Sigma^{-1} (y-\mu) + v}{2} \right)$$

yields t-distributions. For example, the  $n_1$  dimensional distribution  $y_1 | y_2, \mu, \Sigma, v$  has mean vector and covariance matrix

$$E(y_1 | y_2, \mu, \Sigma, v) = \mu_1 + \Sigma_{12}(\Sigma_{22})^{-1}(y_2 - \mu_2)$$

$$Var(y_1 | y_2, \mu, \Sigma, v) = \frac{v}{v-2} [\Sigma_{11} - \Sigma_{12}(\Sigma_{22})^{-1}\Sigma_{21}]$$

Marginal distribution of regression coefficients

Multivariate t

$$p(\beta_1, \beta_2 | y) \propto \int (\sigma^2)^{-\frac{(n-2)}{2}+1} \exp \left[ -\frac{S_e + S_\beta}{2\sigma^2} \right] d\sigma^2$$

$$\propto (S_e + S_\beta)^{-\frac{(n-2)}{2}} \propto \left[ 1 + \frac{S_\beta}{(n-2-p_1-p_2)\frac{S_e}{(n-2-p_1-p_2)}} \right]^{-(n-2-p_1-p_2+p_1+p_2)/2},$$

$$S_\beta = \begin{bmatrix} (\beta_1 - \hat{\beta}_1)' & (\beta_2 - \hat{\beta}_2)' \end{bmatrix} C \begin{bmatrix} \beta_1 - \hat{\beta}_1 \\ \beta_2 - \hat{\beta}_2 \end{bmatrix}$$

Degrees of freedom

$n > p_1 + p_2 + 2$

Mean vector

$$\hat{\beta} = [\hat{\beta}_1', \hat{\beta}_2']'$$

Covariance matrix

$$Var(\beta_1, \beta_2 | y) = \frac{S_e}{(n-p_1-p_2-4)} \begin{bmatrix} \mathbf{X}_1' \mathbf{X}_1 & \mathbf{X}_1' \mathbf{X}_2 \\ \mathbf{X}_2' \mathbf{X}_1 & \mathbf{X}_2' \mathbf{X}_2 \end{bmatrix}^{-1}$$

Marginal distribution of variance

$$\begin{aligned}
 p(\sigma^2 | \mathbf{y}) &\propto (\sigma^2)^{-\frac{n}{2}} \exp\left[-\frac{S_e}{2\sigma^2}\right] \iint \exp\left[-\frac{S_\beta}{2\sigma^2}\right] d\beta_1 d\beta_2 \\
 &\quad \iint \exp\left[-\frac{\left[\begin{array}{c} (\beta_1 - \hat{\beta}_1)' \\ (\beta_2 - \hat{\beta}_2)' \end{array}\right] \mathbf{C} \left[\begin{array}{c} \beta_1 - \hat{\beta}_1 \\ \beta_2 - \hat{\beta}_2 \end{array}\right]}{2\sigma^2}\right] d\beta_1 d\beta_2 \\
 &= (2\pi)^{\frac{p_1+p_2}{2}} |\mathbf{C}^{-1} \sigma^2|^{\frac{1}{2}}. \\
 &\quad \longrightarrow \\
 p(\sigma^2 | \mathbf{y}) &\propto (\sigma^2)^{-\left(\frac{n-p_1-p_2-2}{2}+1\right)} \exp\left(-\frac{S_e}{2\sigma^2}\right)
 \end{aligned}$$

$$\sigma^2 | \mathbf{y} \sim (n - p_1 - p_2 - 2) \frac{S_e}{(n - p_1 - p_2 - 2)} \chi_{n-p_1-p_2-2}^{-2}$$

$$E(\sigma^2 | \mathbf{y}) = \frac{S_e}{n - p_1 - p_2 - 4},$$

$$Var(\sigma^2 | \mathbf{y}) = \frac{2S_e^2}{(n-p_1-p_2-4)^2(n-p_1-p_2-6)}$$

Posterior distribution of residuals

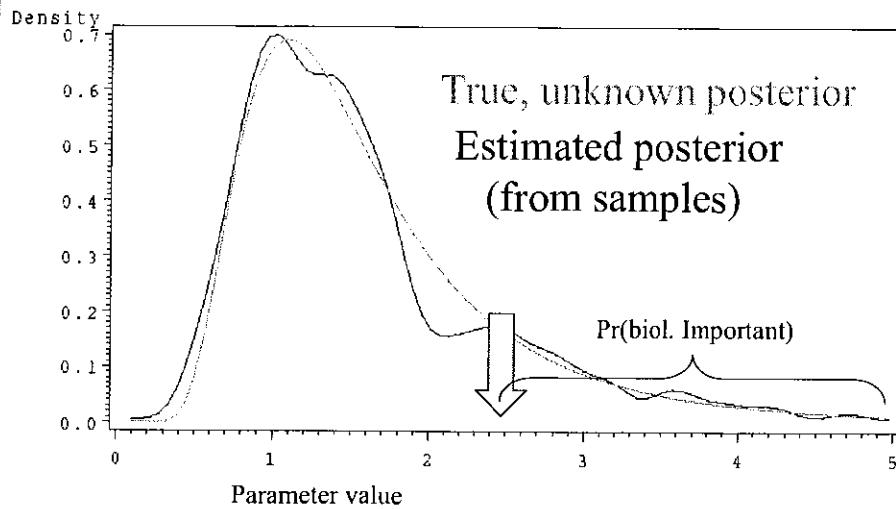
$$e_i = y_i - \mathbf{x}_i' \boldsymbol{\beta}$$

univariate-*t* on  $n - 2 - p_1 - p_2$  degrees of freedom

$$E(e_i|\mathbf{y}) = y_i - \mathbf{x}_i' \hat{\boldsymbol{\beta}}$$

$$Var(e_i|\mathbf{y}) = Var(\mathbf{x}_i' \boldsymbol{\beta}|\mathbf{y}) = \frac{S_e \mathbf{x}_i' \mathbf{C}^{-1} \mathbf{x}_i}{(n-p_1-p_2-4)}$$

Exact and estimated posterior densities  
(most of the time we will not be able to derive the posterior, but  
may be able to sample from it)



## Estimating a posterior expectation and variance from samples

Posterior Expectation:  $E(\theta|y) = \int \theta p(\theta|y)d\theta$

- ➡ May be posterior is unknown or integral impossible to compute
- ➡ Samples available from  $[\theta|y]$
- ➡  $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(S)}$
- ➡ Estimate integral as  $\hat{E}(\theta|y) = \frac{1}{S} \sum_{i=1}^S \theta^{(i)}$
- ➡ Monte Carlo Error =  $\hat{E}(\theta|y) - E(\theta|y)$       Goes to 0 as S tends to infinity<sub>49</sub>  
 $= \frac{1}{S} \sum_{i=1}^S \theta^{(i)} - E(\theta|y)$

### ➡ Monte Carlo Variance of estimate of posterior mean

Measures variability to be expected if repeated sampling (each time S samples drawn) is done from the posterior

$$Var(\text{Monte Carlo Error}) = Var_{\theta|y} [\hat{E}(\theta|y) - E(\theta|y)]$$

$$\begin{aligned} Var(\text{Monte Carlo Error}) &= Var_{\theta|y} \left[ \frac{1}{S} \sum_{i=1}^S \theta^{(i)} - E(\theta|y) \right] \\ &= Var_{\theta|y} \left[ \frac{1}{S} \sum_{i=1}^S \theta^{(i)} \right] \end{aligned}$$

$$\begin{aligned}
 Var(MCE) &= \frac{1}{S^2} \left[ \sum_{i=1}^S Var_{\theta|y}(\theta^{(i)}) + 2 \sum \sum_{i < j} Cov_{\theta|y}(\theta^{(i)}, \theta^{(j)}) \right] \\
 &= \frac{1}{S^2} \left[ \sum_{i=1}^S Var(\theta|y) + 2Var(\theta|y) \sum \sum_{i < j} \rho_{ij} \right] \\
 &= \frac{Var(\theta|y)}{S} \left( 1 + \frac{2}{S} \sum \sum_{i < j} \rho_{ij} \right)
 \end{aligned}$$

Null only if samples  
are independent



IF MARKOV CHAIN MONTE CARLO SAMPLING IS  
PRACTICED, SAMPLES ARE TYPICALLY SERIALLY  
CORRELATED



IMPORTANT TO EVALUATE AUTO-CORRELATIONS  
IN MCMC, TO ASSESS MONTE CARLO ERROR

51

## METROPOLIS-HASTINGS ALGORITHM

...and derivatives

52

### 1. FORM OF ALGORITHM

1. Generate candidate  $\theta^*$  from proposal density  $f(\theta^*|\theta^{[t-1]})$

2. Draw random number  $U(0,1)$

3. Compute ratio

$$R = \frac{g(\theta^*)/f(\theta^*|\theta^{[t-1]})}{g(\theta^{[t-1]})/f(\theta^{[t-1]}|\theta^*)} \quad \text{Posterior or conditional posterior}$$

4. If  $\begin{cases} U < \min(R, 1) \text{ set } \theta^{[t]} = \theta^* \\ \theta^{[t]} = \theta^{[t-1]} \end{cases}$  Important: sample not rejected.  
Chain value is just repeated

Integration constant is not needed

Ignore above formula

$$\begin{aligned} R &= \frac{cp(y|\theta^*)p(\theta^*)/f(\theta^*|\theta^{[t-1]})}{cp(y|\theta^{[t-1]})p(\theta^{[t-1]})/f(\theta^{[t-1]}|\theta^*)} \\ &= \frac{p(y|\theta^*)p(\theta^*)/f(\theta^*|\theta^{[t-1]})}{p(y|\theta^{[t-1]})p(\theta^{[t-1]})/f(\theta^{[t-1]}|\theta^*)} \end{aligned}$$

53

### 2. SPECIAL FORMS: USING THE POSTERIOR AS PROPOSAL

$$\begin{aligned} R &= \frac{p(y|\theta^*)p(\theta^*)/f(\theta^*|\theta^{[t-1]})}{p(y|\theta^{[t-1]})p(\theta^{[t-1]})/f(\theta^{[t-1]}|\theta^*)} \\ &= \frac{p(y|\theta^*)p(\theta^*)/[cp(y|\theta^*)p(\theta^*)]}{p(y|\theta^{[t-1]})p(\theta^{[t-1]})/[cp(y|\theta^{[t-1]})p(\theta^{[t-1]})]} = 1 \end{aligned}$$

If this were not so, one would have doubts...

54

### 3. SPECIAL FORMS: METROPOLIS ALGORITHM

Take a symmetric (in its arguments) proposal density:

$$f(\theta^* | \theta^{[t-1]}) = f(\theta^{[t-1]} | \theta^*)$$

Acceptance rate becomes

$$\begin{aligned} R &= \frac{p(y|\theta^*)p(\theta^*)/f(\theta^*|\theta^{[t-1]})}{p(y|\theta^{[t-1]})p(\theta^{[t-1]})/f(\theta^{[t-1]}|\theta^*)} \\ &= \boxed{\frac{p(y|\theta^*)p(\theta^*)}{p(y|\theta^{[t-1]})p(\theta^{[t-1]})}} \end{aligned}$$

 If  $\left\{ \begin{array}{l} U < \min(R, 1) \text{ set } \theta^{[t]} = \theta^* \\ \theta^{[t]} = \theta^{[t-1]} \end{array} \right.$

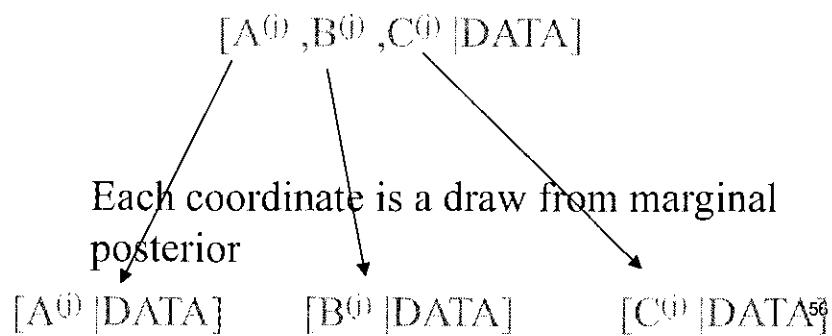
55

### GIBBS SAMPLING

Want to sample from joint posterior

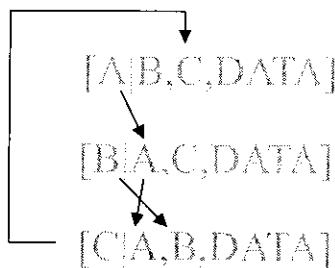
$$[A, B, C | \text{DATA}]$$

Sample is



Gibbs sampling works as follows:

- 1) Form all fully conditional posteriors
- 2) Draw and update successively
- 3) Repeat a number of times without storing samples (burn-in)
- 4) Collect all subsequent samples, and thin them if needed for storage purposes



57

At the end of process:

<u>j</u>	<u>A</u>	<u>B</u>	<u>C</u>	
1	$A^{(1)}$	$B^{(1)}$	$C^{(1)}$	
2	$A^{(2)}$	$B^{(2)}$	$C^{(2)}$	
.	.	.	.	
t	$A^{(t)}$	$B^{(t)}$	$B^{(t)}$	
t+1	$A^{(t+1)}$	$B^{(t+1)}$	$B^{(t+1)}$	
.	.	.	.	
t+m	$A^{(t+m)}$	$B^{(t+m)}$	$B^{(t+m)}$	

Discard first  $t$  samples as burn-in

Keep subsequent  $m$  samples for Posterior analysis

58

EXAMPLE: MH FOR A GLIM (Carlin and Louis, 2000)

Number of flour beetles killed after exposure to carbon disulphide

Dosage No. Killed No. Exposed

$w_i$	$y_i$	$n_i$
1.6097	6	59
1.7242	13	60
1.7552	18	62
1.7842	28	56
1.8113	52	63
1.8369	53	59
1.8610	61	62
1.8839	60	60

Generalized logit model

$$\Pr(death|w) = h(w) = \left[ \frac{\exp(x)}{1+\exp(x)} \right]^{m_1}$$

$w_i = \text{dose } i = 1, 2, \dots, k$

$$x = \frac{w - \mu}{\sigma} \quad \begin{matrix} \leftarrow \\ \text{Unknown parameters} \end{matrix}$$

Priors

$$m_1 \sim \text{Gamma}(a_0, b_0) \propto m_1^{a_0-1} \exp\left(-\frac{m_1}{b_0}\right) \quad \mu \sim N(c_0, d_0)$$

$$\sigma^2 \sim \text{Inverse Gamma}(e_0, f_0) \propto (\sigma^2)^{-(e_0+1)} \exp\left(-\frac{1}{f_0 \sigma^2}\right) \quad \begin{matrix} \leftarrow \\ 59 \end{matrix}$$

If posterior interests coincide  
• your prior knowledge present, it  
could be posterior be dependent.

Joint posterior

$$\begin{aligned} & p(\mu, \sigma^2, m_1 | \mathbf{y}, a_0, b_0, c_0, d_0, e_0, f_0) \\ & \propto \left\{ \prod_{i=1}^k [h(w_i)]^{y_i} [1 - h(w_i)]^{n_i - y_i} \right\} \exp\left[-\frac{(\mu - c_0)^2}{2d_0^2}\right] \\ & \quad \times (\sigma^2)^{-(e_0+1)} \exp\left(-\frac{1}{f_0 \sigma^2}\right) m_1^{a_0-1} \exp\left(-\frac{m_1}{b_0}\right) \\ & \propto \left\{ \prod_{i=1}^k [h(w_i)]^{y_i} [1 - h(w_i)]^{n_i - y_i} \right\} \frac{m_1^{a_0-1}}{(\sigma^2)^{(e_0+1)}} \exp\left[-\frac{(\mu - c_0)^2}{2d_0^2} - \frac{m_1}{b_0} - \frac{1}{f_0 \sigma^2}\right] \end{aligned}$$

Joint posterior is not recognizable...Use Metropolis-Hastings

Transform variables, to work on

$$\begin{bmatrix} \theta_1 = \mu \\ \theta_2 = \frac{1}{2} \log(\sigma^2) \\ \theta_3 = \log(m_1) \end{bmatrix} \Leftrightarrow \begin{bmatrix} \mu = \theta_1 \\ \sigma^2 = \exp(2\theta_2) \\ m_1 = \exp(\theta_3) \end{bmatrix}$$

$\Re^3$  so that Gaussian proposals can be used

$$J = \begin{bmatrix} \frac{\partial \mu}{\partial \theta_1} & \frac{\partial \mu}{\partial \theta_2} & \frac{\partial \mu}{\partial \theta_3} \\ \frac{\partial \sigma^2}{\partial \theta_1} & \frac{\partial \sigma^2}{\partial \theta_2} & \frac{\partial \sigma^2}{\partial \theta_3} \\ \frac{\partial m_1}{\partial \theta_1} & \frac{\partial m_1}{\partial \theta_2} & \frac{\partial m_1}{\partial \theta_3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 \exp(2\theta_2) & 0 \\ 0 & 0 & \exp(\theta_3) \end{bmatrix}$$

$\rightarrow |J| = 2 \exp(2\theta_2 + \theta_3)$

61

→ New density = old density (evaluated at transformed variables) times Jacobian

$$\begin{aligned} p(\theta_1, \theta_2, \theta_3 | \mathbf{y}, a_0, b_0, c_0, d_0, e_0, f_0) \\ \propto \left\{ \prod_{i=1}^k [h(w_i)]^{y_i} [1 - h(w_i)]^{n_i - y_i} \right\} \frac{[\exp(\theta_3)]^{a_0-1}}{(\exp(2\theta_2))^{(e_0+1)}} \\ \times \exp \left[ -\frac{(\theta_1 - c_0)^2}{2d_0^2} - \frac{\exp(\theta_3)}{b_0} - \frac{1}{f_0 \exp(2\theta_2)} \right] \exp(2\theta_2 + \theta_3) \end{aligned}$$

→ Collecting terms

$$\begin{aligned} p(\theta_1, \theta_2, \theta_3 | \mathbf{y}, a_0, b_0, c_0, d_0, e_0, f_0) \\ \propto \left\{ \prod_{i=1}^k [h(w_i)]^{y_i} [1 - h(w_i)]^{n_i - y_i} \right\} \exp(a_0\theta_3 - 2e_0\theta_2) \\ \times \exp \left[ -\frac{(\theta_1 - c_0)^2}{2d_0^2} - \frac{\exp(\theta_3)}{b_0} - \frac{1}{f_0 \exp(2\theta_2)} \right] \end{aligned}$$

POSTERIOR IS NOT RECOGNIZABLE...

62

Hyper-parameters:  $a_0 = .25$ ,  $b_0 = 4$ ,  $c_0 = 2$ ,  $d_0 = 10$ ,  $e_0 = 2.000004$ ,  $f_0 = 1000$

1) Metropolis-Hastings proposal distribution used

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \sim N\left(\begin{bmatrix} \theta_1^{[t-1]} \\ \theta_2^{[t-1]} \\ \theta_3^{[t-1]} \end{bmatrix}, \mathbf{D} = \begin{bmatrix} .00012 & 0 & 0 \\ 0 & .033 & 0 \\ 0 & 0 & .10 \end{bmatrix}\right)$$

$$R = \frac{p(y|\theta^*)p(\theta^*)/f(\theta^*|\theta^{[t-1]})}{p(y|\theta^{[t-1]})p(\theta^{[t-1]})/f(\theta^{[t-1]}|\theta^*)}$$

$$f(\theta^*|\theta^{[t-1]}) = \frac{1}{(2\pi)^3 |\mathbf{D}|} \exp\left[-\frac{1}{2}(\theta^* - \theta^{[t-1]})' \mathbf{D}^{-1} (\theta^* - \theta^{[t-1]})\right]$$

$$f(\theta^{[t-1]}|\theta^*) = \frac{1}{(2\pi)^3 |\mathbf{D}|} \exp\left[-\frac{1}{2}(\theta^{[t-1]} - \theta^*)' \mathbf{D}^{-1} (\theta^{[t-1]} - \theta^*)\right]$$

$$f(\theta^*|\theta^{[t-1]}) = f(\theta^{[t-1]}|\theta^*) \quad \text{Symmetric: use METROPOLIS RATIO } ^{63}$$

- Three parallel chains run each with 10,000 iterations
- Burn-in= 2,000 in each chain
- Histograms based on the  $(10,000-2,000)3 = 24,000$  sampled values
- Autocorrelations and inter-correlations estimated from chain 2

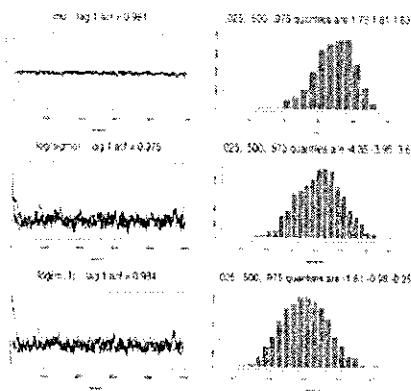


Figure 5.7: Metropolis analysis of the floor teeth mortality data using 2x3 matrix plots of density with a diagonal 3x3 matrix. Monitoring plots for three parallel chains and histograms for all samples following iteration 2000. Overall Metropolis is good in case 11.79.

- Chains mixed slowly (13.5% acceptance rate)
- High correlations between parameters:
- Makes sense to explore different proposal

$$\begin{bmatrix} 1 & -0.78 & -0.94 \\ -0.78 & 1 & 0.89 \\ -0.94 & 0.89 & 1 \end{bmatrix}$$

relation between parameter while we observed in case 11.79  
prior from data or proposal dist less likely

2) Metropolis-Hastings proposal distribution used

→ From first algorithm ,estimate posterior covariance matrix as  $\hat{\Sigma} = \frac{1}{m} \sum_{j=1}^m (\theta^{(j)} - \bar{\theta})(\theta^{(j)} - \bar{\theta})'$

→ Use Gaussian proposal with covariance matrix (gave acceptance rate 27.3%)

$$\Psi = 2\hat{\Sigma} = \begin{bmatrix} 0.000292 & -0.003546 & -0.007856 \\ -0.003546 & 0.074733 & 0.117809 \\ -0.007856 & 0.117809 & 0.241551 \end{bmatrix}$$

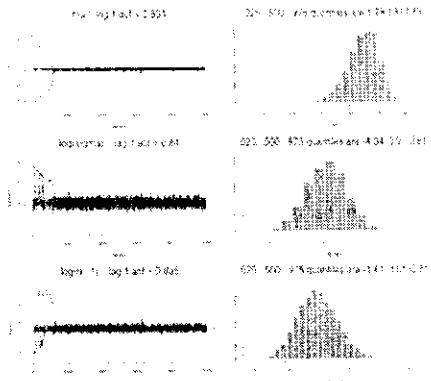


Figure 5.8: Metropolis analysis of the glue-brick viscosity data using a Gaussian proposal density with a standard normal center. Metropolis plots are their parallel chains, and histograms are all samples following iteration 2000. Overall Metropolis acceptance rate: 27.3%

65

## 4. GENOME-ENABLED PREDICTION GENOMIC BLUP, BAYES A, BAYES B, BAYESIAN LASSO

Standard analysis (fixed  $\mathbf{X}$ )

Genotypic value (signal from genome)

Assumption

$$y = f + e = \mathbf{X}\beta + e$$

Bayesian or Frequentist?  
(more later)  $\Rightarrow \beta | \sigma_\beta^2 \sim N(0, I\sigma_\beta^2)$

$$E(y|\mathbf{X}, \beta) = \mathbf{X}\beta$$
$$E(y|\mathbf{X}) = 0$$
$$Var(y|\mathbf{X}, \sigma_e^2, \sigma_\beta^2) = \mathbf{X}\mathbf{X}'\sigma_\beta^2 + I\sigma_e^2$$

**Example 1 (Ridge regression from Bayesian and frequentist points of view)**  
*Suppose the conditional prior of the regressions has the form*

$$\begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} \mid \sigma^2, H_\beta \sim N \left( \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}, \begin{bmatrix} I \frac{\sigma_{\beta_1}^2}{\sigma^2} & 0 \\ 0 & I \frac{\sigma_{\beta_2}^2}{\sigma^2} \end{bmatrix} \sigma^2 \right) \quad \text{Frequentist: random effects model}$$

so the two sets of coefficients are independent, *a priori*. Then the mean of the conditional posterior distribution of the regression coefficients, using (1.30) and (1.32), is

$$\begin{bmatrix} \bar{\beta}_1 \\ \bar{\beta}_2 \end{bmatrix} = \begin{bmatrix} X'_1 X_1 + I \frac{\sigma^2}{\sigma_{\beta_1}^2} & X'_1 X_2 \\ X'_2 X_1 & X'_2 X_2 + I \frac{\sigma^2}{\sigma_{\beta_2}^2} \end{bmatrix}^{-1} \begin{bmatrix} X'_1 y + m_1 \frac{\sigma^2}{\sigma_{\beta_1}^2} \\ X'_2 y + m_2 \frac{\sigma^2}{\sigma_{\beta_2}^2} \end{bmatrix} \quad \text{Frequentist: conditional distribution}$$

When there is a single set of regression coefficients and when the prior mean is a null vector, this reduces to

$$\bar{\beta} = (X'X + I\kappa)^{-1} X'y$$

Frequentist: mean of conditional distribution (BLUP here)  
 Bayesian: mean of conditional posterior distribution

where

$$k = \frac{\sigma^2}{\sigma_{\beta}^2} \quad \begin{array}{l} \text{Frequentist: estimate var. comp by, e.g., REM} \\ \text{Bayesian: use posterior distributions} \end{array}$$

Ridge regression is a special form w/ posterior under Gaussian normal assumption.

### Prediction of marker effects: BLUP (iid marker effects)

$$\begin{aligned} \left[ X'X + \frac{\sigma_e^2}{\sigma_{\beta}^2} I \right] \hat{\beta} &= X'y \\ \left[ I + \frac{\sigma_e^2}{\sigma_{\beta}^2} (X'X)^{-1} \right] \hat{\beta} &= (X'X)^{-1} X'y \quad \text{Assume inverse exists} \\ \hat{\beta} &= \left[ I + \frac{\sigma_e^2}{\sigma_{\beta}^2} (X'X)^{-1} \right]^{-1} \tilde{\beta}_{OLS} \Rightarrow \text{SHRINKAGE} \end{aligned}$$

### Prediction of signal ( $X\beta$ ) to phenotype

$$Var(X\beta|y) = XVar(\beta|y)X'$$

$$= X \left[ I + \frac{\sigma_e^2}{\sigma_{\beta}^2} (X'X)^{-1} \right]^{-1} X' \sigma_e^2 + \text{P.L.V.}$$

Prediction of future record

$$y^* = X^* \beta + e^*$$

future & current  
be different from now (communicating  
future)

$$E(X^* \beta + e^* | y, X, X^*) = X^* E(\beta | y, X)$$

$$= X^* \left[ I + \frac{\sigma_e^2}{\sigma_\beta^2} (X' X)^{-1} \right]^{-1} \tilde{\beta}_{OLS}$$

$$Var(X^* \beta + e^* | y, X, X^*) = X^* Var(\beta | y, X) X^* + I^* \sigma_e^2$$

1. Standard BLUP of signal ( $f$ )

$$y = f + e = X\beta + e$$

$X$  is fixed here

$$f \sim N(0, Var(f)) \quad Var(f) = XX' Var(\beta)$$

$$Var(y|X) = XX' Var(\beta) + I\sigma_e^2$$

$$\begin{aligned} BLUP(f) &= Cov(f, y')[XX' Var(\beta) + I\sigma_e^2]^{-1} y \\ &= XX' Var(\beta)[XX' Var(\beta) + I\sigma_e^2]^{-1} y \\ &= \left[ I + (XX')^{-1} \frac{\sigma_e^2}{Var(\beta)} \right]^{-1} y \end{aligned}$$

$$\left[ I + (XX')^{-1} \frac{\sigma_e^2}{Var(\beta)} \right] BLUP(f) = y$$

2. Morph into genomic BLUP a la Van Raden

$$G = \frac{(X - E(X))(X - E(X))'}{2 \sum_{j=1}^p p_j(1-p_j)} = \frac{X^* X'}{V_{M,HW}} \quad \text{Center using allelic frequency information}$$

$$\left[ I + G^{-1} \frac{\sigma_e^2}{Var(\beta)/V_{M,HW}} \right] \hat{g} = y$$

IS THIS G THE BEST ESTIMATE OF THE UNKNOWN  $G_M$ ? ARGUABLY NOT

## Get marker effects from G-BLUP models

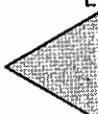
3. Estimate marker effects from genomic BLUP? Use standard BLUP theory under normality!

$$\begin{aligned}\hat{g} &= E(X\beta|y, \text{variance components}) \text{ under normality} \\ E(\beta|X\beta) &= E(\beta) + \text{Cov}(\beta, \beta'X')[\text{Var}(X\beta)^{-1}[X\beta - E(X\beta)]] \\ &= 0 + \sigma_\beta^2 X'(XX')^{-1} \frac{1}{\sigma_\beta^2} X\beta \\ E(\beta|y, \text{variance components}) &= E_{X\beta|y}[E(\beta|X\beta, y)] \\ &= E_{X\beta|y}[X'(XX')^{-1}X\beta|y] \\ &= X'(XX')^{-1} E_{X\beta|y}[X\beta|y] = X'(XX')^{-1} \hat{g}\end{aligned}$$

$$\hat{\beta} = E(\beta|y, \text{variance components}) = X'(XX')^{-1} \left[ I + (XX')^{-1} \frac{\sigma_e^2}{\sigma_\beta^2 / V_{M,HW}} \right]^{-1} y$$


[REMEMBER THIS]

BRUTE FORCE DEFINITION: BLUP is a conditional expectation under normality

$$\begin{aligned}\hat{\beta} &= E(\beta|y, \text{variance components}) = \text{Cov}(\beta, \beta'X') [XX'\sigma_\beta^2 + I\sigma_e^2]^{-1} y \\ &= \sigma_\beta^2 X'[XX'\sigma_\beta^2 + I\sigma_e^2]^{-1} y = \sigma_\beta^2 X'(XX')^{-1} [\sigma_\beta^2 + (XX')^{-1}\sigma_e^2]^{-1} \\ &= X'(XX')^{-1} \left[ I + (XX')^{-1} \frac{\sigma_e^2}{\sigma_\beta^2} \right]^{-1}\end{aligned}$$


[REMEMBER?] [REMEMBER?]

CAN GO BACK AND FORTH BETWEEN GENOMIC BLUP AND RIDGE REGRESSION ESTIMATES OF MARKER EFFECTS

$$\begin{aligned}\text{marker from G-BLUP} \quad \hat{\beta} &= X'(XX')^{-1} \hat{g} \\ \text{G-BLUP from marker} \quad \hat{g} &= X\hat{\beta}\end{aligned}$$

## GAUSSIAN PROCESS ANALYSIS (IID MARKER EFFECTS)

$$y = f + e = X\beta + e$$

$$\beta \sim N(0, I\sigma_\beta^2) \quad \leftarrow \quad \text{[Read Falconer and Mackay IQG]}$$

$$X \sim F \quad \leftarrow \quad \text{[Genotypes vary at random: population Genetics]}$$

$$E(y|X, \beta) = X\beta \quad \leftarrow$$

$$\begin{aligned} E(y|\beta) &= E_X E(y|X, \beta) = E(X)\beta \\ E(y) &= E_\beta [E(X)\beta] = E(X)E(\beta) = 0 \end{aligned} \quad \Bigg] \quad \text{Big assumption}$$

Are frequencies effect-dependent? Are effects frequency dependent?  
TURELLI, ZHANG&HILL, MACKAY WITH MARKERS AND

*while we assume that marker effects  
dist are independent from dist of marker genotype frequency.*



$$Var(y) = Var(f) + Var(e) = Var(f) + I\sigma_e^2$$

Covariance matrix of signal  $\rightarrow$

$$Var(f) = Var(X\beta)$$

$$= E_X(Var(X\beta|X) + Var_X[E(X\beta|X)])$$

$$= E_X[XVar(\beta)X'] + Var_X[XE(\beta)]$$

$$= E_X[XX'\sigma_\beta^2] + Var_X(0)$$

$$= \sigma_\beta^2 E_X[XX'] \quad \text{BP= "best predictor"  
(MULVN assumed)}$$

$$\hat{f} = BP(f)$$

$$\left[ \frac{1}{\sigma_e^2} I + Var^{-1}(f) \right] \hat{f} = \frac{1}{\sigma_e^2} y$$

$$\left[ I + \frac{\sigma_e^2}{\sigma_\beta^2} E_X^{-1}[XX'] \right] \hat{f} = y \quad \leftarrow$$

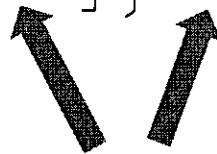
Looks like  
genomic BLUP  
(it is not)

$$E_X^{-1}[XX'] \left[ E_X[XX'] + \frac{\sigma_e^2}{\sigma_\beta^2} I \right] \hat{f} = y$$

$$\left[ E_X[XX'] + \frac{\sigma_e^2}{\sigma_\beta^2} I \right] \hat{f} = E_X[XX']y$$

### Under multivariate normality

$$\begin{aligned}
 Var(f|y) &= Var(f) - Cov(f,y)Var^{-1}(y)Cov'(f,y) \\
 &= Var(f) - Var(f)[Var(f) + I\sigma_e^2]^{-1}Var(f) \\
 &= \sigma_\beta^2 E_X[XX'] - \sigma_\beta^2 E_X[XX'] [\sigma_\beta^2 E_X[XX'] + I\sigma_e^2]^{-1} \sigma_\beta^2 E_X[XX'] \\
 &= \sigma_\beta^2 E_X[XX'] - \sigma_\beta^2 E_X[XX'] \frac{E_X^{-1}[XX']}{\sigma_\beta^2} \left[ I + \frac{\sigma_e^2}{\sigma_\beta^2} E_X[XX'] \right]^{-1} \sigma_\beta^2 E_X[XX'] \\
 &= \left\{ I - \left[ I + \frac{\sigma_e^2}{\sigma_\beta^2} E_X[XX'] \right]^{-1} \right\} \sigma_\beta^2 E_X[XX'].
 \end{aligned}$$



Proper assessment of posterior uncertainty requires knowledge  
of the genotypic distribution

$$\begin{aligned}
 X_{\text{ind,marker}} &= \begin{bmatrix} x_{11} & \dots & x_{1p} \\ x_{21} & \dots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{np} \end{bmatrix} \\
 XX' &= \begin{bmatrix} x_{11} & \dots & x_{1p} \\ x_{21} & \dots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{np} \end{bmatrix} \begin{bmatrix} x_{11} & x_{21} & \dots & x_{n1} \\ x_{12} & x_{22} & \dots & x_{n2} \\ \vdots & \ddots & \ddots & \vdots \\ x_{1p} & x_{2p} & \dots & x_{np} \end{bmatrix} \\
 &= \begin{bmatrix} \sum_{j=1}^p x_{1j}^2 & \sum_{j=1}^p x_{1j}x_{2j} & \sum_{j=1}^p x_{1j}x_{nj} \\ \sum_{j=1}^p x_{2j}^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots \\ \sum_{j=1}^p x_{nj}^2 & \sum_{j=1}^p x_{nj}x_{1j} & \sum_{j=1}^p x_{nj}x_{2j} \end{bmatrix}
 \end{aligned}$$

Genotypes (random variable  $W$  denotes genotype at a locus)



$$\begin{cases} W(aa) \rightarrow -1 \\ W(Aa) \rightarrow 0 \\ W(AA) \rightarrow 1 \end{cases}$$

$$\begin{aligned} E_{HW}(W) &= p^2 - q^2 = (p - q) = \mu \\ Var_{HW}(W) &= E(X^2) - E^2(X) \\ &= p^2 + q^2 - (p - q)^2 \\ &= 2pq \end{aligned}$$



$$\begin{cases} W(aa) \rightarrow 0 \\ W(Aa) \rightarrow 1 \\ W(AA) \rightarrow 2 \end{cases}$$

$$\begin{aligned} E_{HW}(W) &= 2p^2 + 2pq = 2p(p + q) = 2p \\ Var_{HW}(W) &= 4p^2 + 2pq - 4p^2 = 2pq \end{aligned}$$

Coding does not affect the variance of genotypes but mean shifts  $2p - (p - q) = 1$

DEVIATIONS FROM MEAN AND STANDARDIZED DEVS. ARE INVARIANT

$W - E(W)$ , coding 1

$$-1 - (p - q) = -1 - p + q = -2p$$

$$0 - (p - q) = q - p = 1 - 2p$$

$$1 - (p - q) = 1 - p + q = 2(1 - p)$$

$W - E(W)$ , coding 2

$$0 - 2p$$

$$1 - 2p$$

$$2 - 2p = 2(1 - p)$$



$$W^* = \frac{W - E(W)}{\sqrt{2pq}}$$

Under HW

$$\begin{aligned} E\left(\sum_{j=1}^p x_{ij}^2\right) &= \sum_{j=1}^p Var(x_{ij}) + \sum_{j=1}^p E^2(x_{ij}) \\ &= \sum_{j=1}^p 2p_j q_j + \sum_{j=1}^p (p_j - q_j)^2 \\ &= \sum_{j=1}^p (1 - 2p_j q_j) = p - \sum_{j=1}^p 2p_j q_j \end{aligned}$$

$$\begin{aligned} E\left(\sum_{j=1}^p x_{1j} x_{2j}\right) &= \sum_{j=1}^p Cov(x_{1j}, x_{2j}) + \sum_{j=1}^p E(x_{1j}) E(x_{2j}) \\ &= \sum_{j=1}^p 2\phi_{ij} p_j q_j + \sum_{j=1}^p (p_j - q_j)^2 \\ &= \sum_{j=1}^p p_j^2 + q_j^2 - 2p_j q_j (1 - \phi) \end{aligned}$$

$$\begin{aligned} Cov(x_{1j}, x_{2j}) &= p_j^2 + q_j^2 - 2p_j q_j (1 - \phi) - (p_j - q_j)^2 \\ &= 2pq\phi \end{aligned}$$

UNDER HARDY-WEINBERG AND IDEALIZED CONDITIONS

$$E(XX') = \begin{bmatrix} \sum_{j=1}^p 2p_j q_j + \sum_{j=1}^p (p_j - q_j)^2 & a_{12} \sum_{j=1}^p 2p_j q_j + \sum_{j=1}^p (p_j - q_j)^2 & \dots & a_{1n} \sum_{j=1}^p 2p_j q_j + \sum_{j=1}^p (p_j - q_j)^2 \\ \text{Symmetric} & \sum_{j=1}^p 2p_j q_j + \sum_{j=1}^p (p_j - q_j)^2 & a_{2n} \sum_{j=1}^p 2p_j q_j + \sum_{j=1}^p (p_j - q_j)^2 & \dots \\ & & \dots & \dots \\ & & & a_{n,n-1} \sum_{j=1}^p 2p_j q_j + \sum_{j=1}^p (p_j - q_j)^2 \\ & & & \sum_{j=1}^p 2p_j q_j + \sum_{j=1}^p (p_j - q_j)^2 \end{bmatrix}$$

Additive relationships

Likewise, if the x's are centered

$$E\{[X - E(X_{n \times p})][X - E(X_{n \times p})]'\} = \left(\sum_{j=1}^p 2p_j q_j\right) \begin{bmatrix} 1 & a_{12} & \dots & a_{1n} \\ a_{12} & 1 & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & 1 \end{bmatrix}$$

$$= A \left(\sum_{j=1}^p 2p_j q_j\right)$$

$$E\left[\frac{[X - E(X_{n \times p})][X - E(X_{n \times p})]'}{\left(\sum_{j=1}^p 2p_j q_j\right)}\right] = A$$

*A = n x n matrix of additive relationships*

Then, the "genomic" relationship matrix

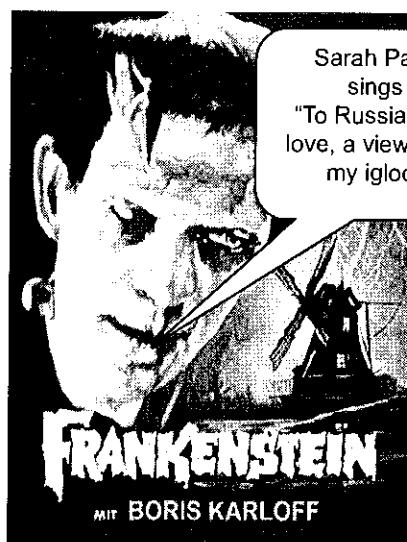
$$G = \frac{(X - E(X))(X - E(X))'}{p} = \frac{X^* X'^*}{V_{MHW}} \\ 2 \sum_{j=1}^p p_j(1-p_j)$$

Is the realization of a process. If this process is the HW process, then its expectation is

$$E \left[ \frac{[X - E(X_{nsp})][X - E(X_{nsp})]'}{\left( \sum_{j=1}^p 2p_j q_j \right)} \right] = A$$

For example: parent and offspring are expected to have a relationship=0.5  
but in reality it could be larger or smaller

## THE CURSE OF THE BAYESIAN ALPHABET



Featuring



Kim-Jong II,  
as "Bayes"



Halle Berry,  
as "A"



Scarlett Johansson  
as "B"



Herman Cain as  
"C-π - 9 - 9 - 9"

## BAYESIAN STATE OF KNOWLEDGE (in a finite sample)

Minimum → Prior

(that is why one gets data, "normative ignorant")

Maximum → Conditional posterior

(Know some things but not others)

Intermediate → Marginal posterior

(Have to use information to assess uncertainty about all unknowns)

## REASONABLE BAYESIAN MODEL

(for learning about state of nature)

- For any parameter, must be able to "kill" the prior asymptotically
- For any parameter, statistical distance between prior and posterior (and therefore conditional posterior) must go to infinity
- If this distance has a finite upper bound, it means that the prior is influential
- Must be able to reduce statistical entropy as conveyed by the prior by a sizable amount. If the reduction is tiny → prior very influential

# THE PROCESS OF DECONDITIONING (MARGINALIZATION CONSUMES INFORMATION ABOUT THE FOCAL POINT

Meaning: conditional posterior is  
the best world to live in

## Prediction of Total Genetic Value Using Genome-Wide Dense Marker Maps

T. H. E. Meuwissen,<sup>1,2</sup> B. J. Hayes<sup>1</sup> and M. E. Goddard<sup>1,3</sup>

Genetics 157: 1819–1829 (April 2001)

BAYES A + BAYES B

(as I understand these two methods)

Linear model proposed by Meuwissen et al. (2001)

Code for genotype  
of SNP j;  
 $x = -1, 0, 1$

SCALAR

$$y_i = \mu + \sum_{j=1}^p x_{ij} b_j + e_i,$$

$i = 1, 2, \dots, n; n \ll p$

Additive substitutio  
effect of  
SNP j

$$y_i | \mu, \mathbf{x}_i, \mathbf{b}, \sigma_e^2 \sim N\left(\mu + \sum_{j=1}^p x_{ij} b_j, \sigma_e^2\right)$$

MATRIX

$$\mathbf{y} = \mathbf{1}\mu + \mathbf{X}\mathbf{b} + \mathbf{e},$$

$$\mathbf{y} | \mu, \mathbf{X}, \mathbf{b} \sim N(\mathbf{1}\mu + \mathbf{X}\mathbf{b}, \mathbf{I}\sigma_e^2)$$

The priors

$$\mu \sim \text{uniform}$$

$$\sigma_e^2 \sim v_e S_e^2 \chi_{v_e}^{-2}$$

$$b_j | \mathbf{b} \sim N(0, \sigma_{b_j}^2); \quad j = 1, 2, \dots, p$$

$$\sigma_{b_j}^2 \sim v S^2 \chi_v^{-2} \text{ for all } j$$

Hyper-parameters, specified arbitrarily

## BAYES A (Meuwissen et al., 2001)

$$b_j | \sigma_j^2 \sim N(0, \sigma_j^2) \quad j=1, 2, \dots, p$$

$$\sigma_j^2 | v, S^2 \sim vS^2 \chi_v^{-2}$$

Note: each SNP has a variance  
(think of a sire model in which  
each sire effect has a variance)

Marginal prior

$$\begin{aligned} p(b_i | v, S^2) &= \int_0^\infty N(0, \sigma_j^2) p(vS^2 \chi_v^{-2}) d\sigma_j^2 \\ &\propto \int_0^\infty (\sigma_j^2)^{-\frac{1}{2}} \exp\left(-\frac{b_i^2}{\sigma_j^2}\right) (\sigma_j^2)^{-(\frac{v+1}{2})} \exp\left[-\frac{vS^2}{\sigma_j^2}\right] d\sigma_j^2 \\ &\propto \int_0^\infty (\sigma_j^2)^{-\frac{1+v+2}{2}} \exp\left(-\frac{b_i^2 + vS^2}{\sigma_j^2}\right) d\sigma_j^2 \\ &\propto \Gamma\left(\frac{1+v}{2}\right) (b_i^2 + vS^2)^{-\frac{v+1}{2}} \\ &\propto \left(1 + \frac{b_i^2}{vS^2}\right)^{-\frac{v+1}{2}} = t(0, v, S^2) \end{aligned}$$

These hyper-parameters will control the extent of shrinkage. Question: does their influence vanish asymptotically?

The prior of a marker effect is a  $t$ -distribution with known scale and df

MARGINALLY: IN BAYES A ALL MARKERS HAVE THE SAME VARIANCE

## Bayes B is Bayesianly “STRANGE”

Bayes B assumptions

$$b_j | \sigma_j^2 \sim \begin{cases} \text{point mass at some constant } k \text{ if } \sigma_j^2 = 0 \\ N(0, \sigma_j^2) \text{ if } \sigma_j^2 > 0 \end{cases}$$

$$\sigma_j^2 | \pi = \begin{cases} 0 \text{ with probability } \pi \\ vS^2 \chi_v^{-2} \text{ with probability } 1 - \pi \end{cases}$$

3. Recall: if a prior variance is 0, this means complete certainty

1. Meuwissen takes the constant = 0

2. Meuwissen assumes  $\pi$  is known, e.g., 0.95

Joint density:

$$p(b_j, \sigma_j^2 | \pi) = \begin{cases} b_j = k \text{ and } \sigma_j^2 = 0 \text{ with probability } \pi \\ N(0, \sigma_j^2) p(vS^2 \chi_v^{-2}) \text{ with probability } 1 - \pi \end{cases}$$

Marginal prior

$$p(b_j | \pi) = \begin{cases} b_j = k \text{ with probability } \pi \\ \int_0^\infty N(0, \sigma_j^2) p(vS^2 \chi_v^{-2}) d\sigma_j^2 \text{ with probability } 1 - \pi \end{cases}$$

Further

$$\begin{aligned} & \int_0^\infty (\sigma_j^2)^{-\frac{1}{2}} \exp\left(-\frac{b_j^2}{\sigma_j^2}\right) (\sigma_j^2)^{-(\frac{v+2}{2})} \exp\left[-\frac{vS^2}{\sigma_j^2}\right] d\sigma_j^2 \\ &= \int_0^\infty (\sigma_j^2)^{\frac{1+v+2}{2}} \exp\left(-\frac{b_j^2 + vS^2}{\sigma_j^2}\right) d\sigma_j^2 \\ &= \Gamma\left(\frac{1+v}{2}\right) (b_j^2 + vS^2)^{-\frac{v+1}{2}} \\ &\propto \left(1 + \frac{b_j^2}{vS^2}\right)^{-\frac{v+1}{2}} \Rightarrow t(0, v, S^2) \end{aligned}$$

Then:

PRIOR = MIXTURE OF A POINT MASS AND OF A  $t$ -DISTRIBUTION. BAYES B PUTS THE MASS AT 0 (IF NOT 0, THIS GETS ABSORBED INTO THE GENERAL MEAN)

$$p(b_j | \pi) = \begin{cases} b_j = k \text{ with probability } \pi \\ t(0, v, S^2) \text{ with probability } 1 - \pi \end{cases}$$

MARGINALLY: ALL MARKERS HAVE THE SAME DISTRIBUTION

Again, it is not true that variance is if means are different in Bayes B (prior)

Mean and variance of a mixture (e.g., Gianola et al. 2006, Genetics)

The first and second moments, and the variance of a finite mixture of  $K$  Gaussian distributions, with parameters  $\boldsymbol{\theta} = [P_1, \dots, P_K, \mu_1, \dots, \mu_K, \sigma_1^2, \dots, \sigma_K^2]'$ , where the mixture proportions  $P_k$  are such that  $\sum_{k=1}^K P_k = 1$ , are

$$\rightarrow E(y | \boldsymbol{\theta}) = \int y \left[ \sum_{k=1}^K P_k N(y | \mu_k, \sigma_k^2) \right] dy = \sum_{k=1}^K P_k \mu_k, \quad (\text{A1})$$

$$E(y^2 | \boldsymbol{\theta}) = \int y^2 \left[ \sum_{k=1}^K P_k N(y | \mu_k, \sigma_k^2) \right] dy = \sum_{k=1}^K P_k (\mu_k^2 + \sigma_k^2).$$

$$\rightarrow \text{Var}(y | \boldsymbol{\theta}) = \sum_{k=1}^K P_k \sigma_k^2 + \sum_{k=1}^K P_k \mu_k^2 - \left( \sum_{k=1}^K P_k \mu_k \right)^2.$$

In Bayes B:

$$\begin{aligned} E(b_j | \pi) &= \pi k + (1 - \pi)0 = \pi k \\ &\Rightarrow 0 \text{ if } k = 0 \end{aligned}$$

$$\begin{aligned} \text{Var}(b_j | \pi) &= \pi \times 0 + (1 - \pi) \frac{S^2 v}{v - 2} + \pi k^2 + (1 - \pi)0^2 - (\pi k)^2 \\ &= (1 - \pi) \frac{S^2 v}{v - 2} + \pi k^2 (1 - \pi) \\ &= (1 - \pi) \frac{S^2 v}{v - 2} \text{ if } k = 0 \end{aligned}$$

**ALL MARKERS HAVE THE SAME VARIANCE IN BAYES B!**

All markers have same mean and Var

for Bayes B

## BAYES A IS A SPECIAL CASE OF BAYES B ( $\pi=0$ )

Meaning: if Bayes A has an inferential flaw, this carries to Bayes B

HEURISTIC ARGUMENT:  
view form of Gibbs sampler  
for Bayes A

(element-wise sampling)

Note: the form of the implementation it is just an **algorithmic** matter: it is immaterial with respect to the issues

Sampling the mean

$$\mu | ELSE \sim N \left[ \frac{1}{n} \sum_{i=1}^n \left( y_i - \sum_{j=1}^p x_{ij} b_j \right), \frac{\sigma_e^2}{n} \right]$$

Flat prior for the mean (or for the fixed effects) is not influential

Sampling the residual variance

$$\sigma_e^2 | ELSE \sim n \left( 1 + \frac{v_e}{n} \right) \left( \sum_{i=1}^n \left( y_i - \mu - \sum_{j=1}^p x_{ij} b_j \right)^2 + v_e s_e^2 \right) \chi_{v_e+n}^{-2}$$

Goes to  $n$

The prior can be "killed" simply by increasing sample size

This will dominate the weighted average  
as  $n$  increases

Sampling the marker effects

$$b_j | ELSE \sim N \left[ \frac{\sum_{i=1}^n x_{ij} \left( y_i - \mu - \sum_{j'=1}^p x_{ij'} b_j \right)}{\sum_{i=1}^n x_{ij}^2 + \frac{\sigma_e^2}{\sigma_{b_j}^2}}, \frac{\sigma_e^2}{\sum_{i=1}^n x_{ij}^2 + \frac{\sigma_e^2}{\sigma_{b_j}^2}} \right]$$

$$j = 1, 2, \dots, p$$

Kill the prior simply by increasing sample size. The effect of the shrinkage ratio vanishes

$$\sum_{i=1}^n x_{ij}^2 + \frac{\sigma_e^2}{\sigma_{b_j}^2} \rightarrow \sum_{i=1}^n x_{ij}^2$$

Sampling the variance of the marker effects

$$\sigma_{b_j}^2 | ELSE \sim v \left( 1 + \frac{1}{v} \right) \left( \frac{b_j^2 + v S^2}{1 + v} \right) \chi_{v+1}^{-2}$$

Typically very small

Prior df: very influential

$$= v \left( 1 + \frac{1}{v} \right) S^2 \left( \left[ \left( \frac{b_j}{S} \right)^2 + v \frac{1}{1+v} \right] \right) \chi_{v+1}^{-2}$$

$$j = 1, 2, \dots, p$$

- **Prior cannot be killed here.** One can increase the number of data or of markers *ad nauseum* and gain only one degree of freedom, **always**
- Recall that, in the conditional posterior, all other parameters are known (i.e., they are assigned values)
- Since one must de-condition, actually the true posterior moves less than one degree of freedom away from the prior

## RECALL: STATE OF KNOWLEDGE

- Minimum → Prior
- Maximum → Conditional posterior
- Intermediate → Marginal posterior

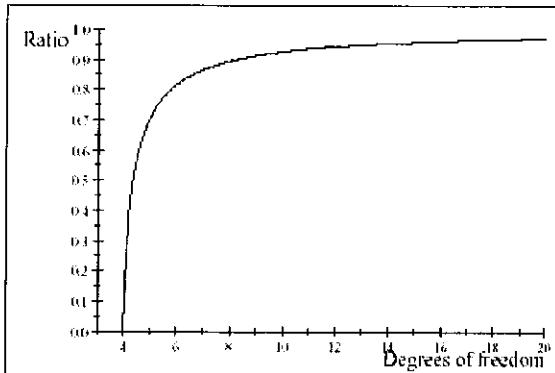


Figure 1. Ratio between coefficients of variation  $CV(\sigma^2_{MLE}) / CV(\sigma^2_p) = \sqrt{1 - \frac{1}{\nu - 3}}$  of the conditional posterior and prior distributions of the variance of the marker effect, as a function of the degrees of freedom  $\nu$  of the prior.

For  $\nu > 6$ , the relative variability of the posterior distribution of the variance of a SNP effect is essentially COPYING that of their prior distribution

## ENTROPY OF A DISTRIBUTION ("DISORDER")

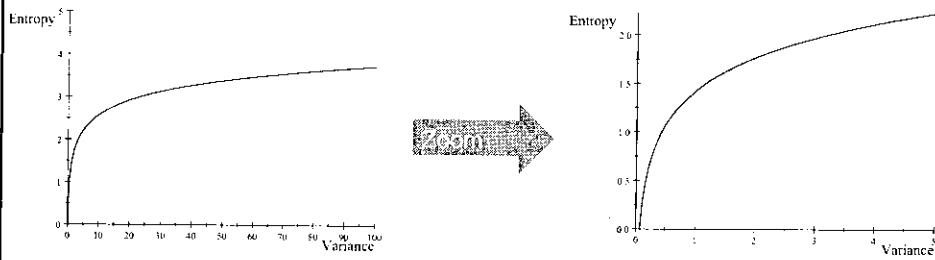
$$H(p(x|\theta)) = E_{x|\theta}[-\log p(x)]$$

Example: normal distribution

$$p(x|\mu = 0, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

$$\log[p(x|\mu = 0, \sigma^2)] = \log\left(\frac{1}{\sqrt{2\pi}}\right) - \frac{1}{2} \log(\sigma^2) - \frac{x^2}{2\sigma^2}$$

$$\begin{aligned} E\{-\log[p(x|\mu = 0, \sigma^2)]\} &= -\log\left(\frac{1}{\sqrt{2\pi}}\right) + \frac{1}{2}[\log(\sigma^2) + 1] \\ &= 0.91893853 + \frac{1}{2}[\log(\sigma^2) + 1] \end{aligned}$$



## ENTROPY CALCULATIONS

### Bayes A: variance of effect

Prior entropy

Variance of marker effect  
(sorry, change of notation)

$$\begin{aligned} H\{\sigma_{a_k}^2 | v, S^2\} &= - \int \log[p(\sigma_{a_k}^2 | v, S^2)] p(\sigma_{a_k}^2 | v, S^2) d\sigma_{a_k}^2 \\ &= -\frac{v}{2} - \log\left[\frac{vS^2}{2}\Gamma\left(\frac{v}{2}\right)\right] + \left(1 + \frac{v}{2}\right) \frac{d}{d\left(\frac{v}{2}\right)} \log\Gamma\left(\frac{v}{2}\right). \end{aligned}$$

Entropy of the conditional posterior

$$\begin{aligned} H\{\sigma_{a_k}^2 | ELSE\} &= - \int \log[p(\sigma_{a_k}^2 | ELSE)] p(\sigma_{a_k}^2 | ELSE) d\sigma_{a_k}^2 \\ &= -\frac{v+1}{2} - \log\left[\left(\frac{vS^2 + a_k^2}{2}\right)\Gamma\left(\frac{v+1}{2}\right)\right] + \left(1 + \frac{v+1}{2}\right) \frac{d}{d\left(\frac{v+1}{2}\right)} \log\Gamma\left(\frac{v+1}{2}\right). \end{aligned}$$

Learning from data: reduces entropy  
(cannot calculate entropy of posterior in closed form)

## Relative information gain

$$RIG = \frac{H\{\sigma_{a_k}^2 | v, S^2\} - H\{\sigma_{a_k}^2 | \text{ELSE}\}}{H\{\sigma_{a_k}^2 | v, S^2\}} \quad \begin{matrix} \text{Entropy} \\ \text{Difference} \end{matrix}$$

For  $a_k = 0, S = 1$  and  $v = 100, RIG = 9.60 \times 10^{-3}$

For  $a_k = 0, S = 1$  and  $v = 10, RIG = 6.51 \times 10^{-2}$

For  $a_k = 0, S = 1$  and  $v = 4, RIG = 0.125$

Metaphorically: the prior is totalitarian in Bayes A (B)



*minimum gain  
in Bayes A & B*

### STATISTICAL DISTANCE BETWEEN CONDITIONAL POSTERIOR AND PRIOR (KULLBACK-LEIBLER)

*most famous distance measure*

Specific distance at a given variance

$$KL[\text{conditional}, \text{prior}] = \int L(v, v + p, S^2, a_m) p(\sigma_{a_k}^2 | v, S^2) d\sigma_{a_k}^2.$$

where

$$L(v, v + p, S^2, a_m, \sigma_{a_k}^2) = \log \frac{p(\sigma_{a_k}^2 | v, S^2)}{p(\sigma_{a_k}^2 | \text{ELSE})},$$

- IF KL IS LARGE, THEN LEARNING BEYOND THE PRIOR HAS TAKEN PLACE.
- KL SHOULD GO TO INFINITY AS DATA ACCUMULATE IN ANY REASONABLE BAYESIAN MODEL

*Conditional dist. is not far from known  
prior in Bayes A & Bayes B*

## KULLBACK-LEIBLER DISTANCES BETWEEN CONDITIONAL POSTERIOR AND PRIOR (of variance of marker effect)

- 1)  $7.33 \times 10^{-2}$  for  $v = 4, S = 1, p = 1$  and  $a_k = 0$
- 2)  $2.64 \times 10^{-2}$  for  $v = 10, S = 1, p = 1$  and  $a_k = 0$
- 3)  $2.52 \times 10^{-3}$  for  $v = 100, S = 1, p = 1$  and  $a_k = 0$

If 10 markers are allowed to share the same variance, KL= 4.47  
Relative to (1), KL distance increases 61 times...

*If you group markers*       $\Rightarrow \uparrow$

## BAYES A (B)

- The prior always matters
- The effect of the prior is via the extent of shrinkage of marker effects
- Cannot learn about variance of marker effect
- Statistically greedy models (same will apply for any model assigning marker-specific variances)
- May have good predictive ability

## SIMULATION

(never take a simulation too seriously)

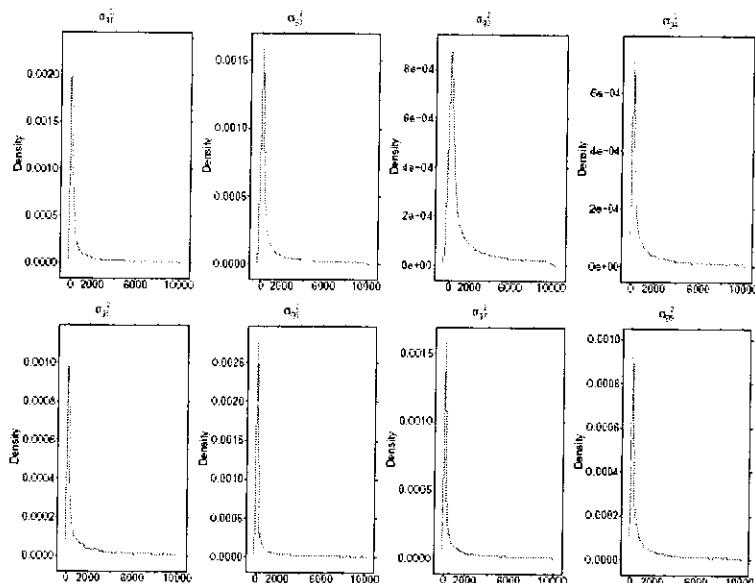
## RESURRECTION OF BAYES A

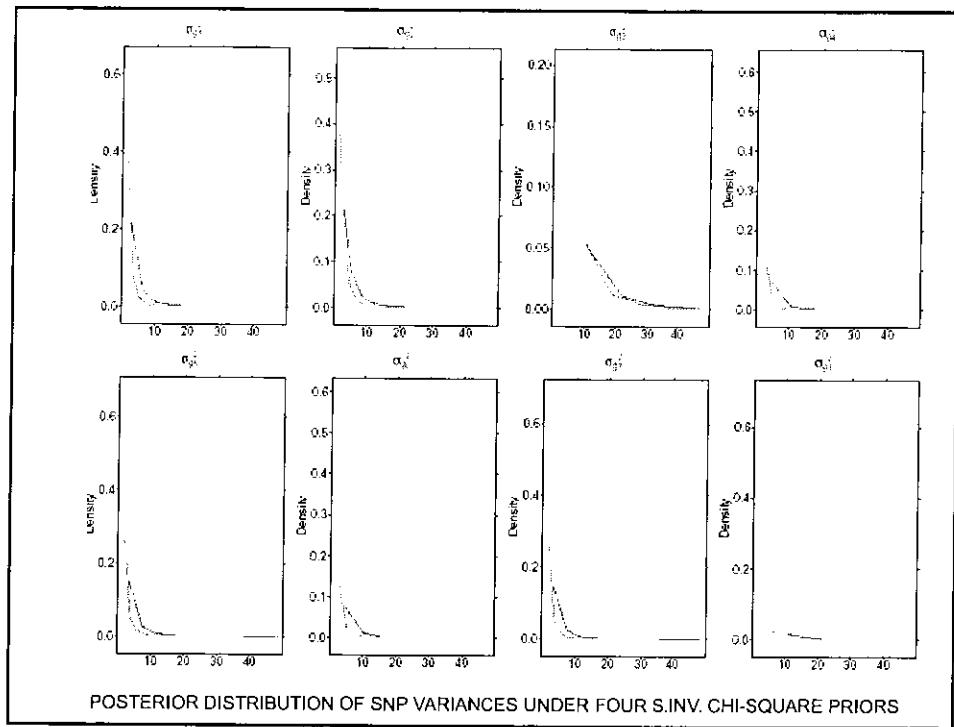
(If additive model holds, it may give sensible inferences about marker effects)

### Description for slides 1, 2 and 3

- Bayes A was fitted on a simulated data of 50 observations.
  - True linear relationship between response and SNP ( $x_1, x_2, x_3$ ) effects
    - $Y = w_1 + 2'w_2 + x_1 - 2'x_2 + 5'x_3 + \text{error} (\sim N(0, \text{sd}=1.2))$
  - Model fitted.
    - $Y = Wg + \text{error}$
    - $W$  is incidence matrix for two nuisance parameters.
    - $X$  is incidence matrix for SNP effects. Besides  $x_1, x_2$  and  $x_3$ , five additional irrelevant SNPs ( $x_4$  to  $x_8$ ) added. SNP value is allele copy numbers, i.e., 0, 1 or 2
- Slide 1—Posterior distributions of SNP effects  $g_i$  ( $i = 1, 2, \dots, 8$ ) when using five different priors on  $\sigma_{g_i}^2$ , scale determined by estimated residual variance (1.5)
  - Black:  $\sigma_{g_i}^2 \sim \text{unif}(0, 100)$
  - Red:  $\sigma_{g_i}^2 \sim \text{scaled inverse } \chi^2(\text{df}=4, \text{scale}=1.5)$
  - Blue:  $\sigma_{g_i}^2 \sim \text{scaled inverse } \chi^2(\text{df}=4, \text{scale}=3)$
  - Green:  $\sigma_{g_i}^2 \sim \text{scaled inverse } \chi^2(\text{df}=8, \text{scale}=1.5)$
  - Pink:  $\sigma_{g_i}^2 \sim \text{scaled inverse } \chi^2(\text{df}=8, \text{scale}=3)$
- Slides 2 and 3—Posterior distributions of 8 SNP specific variances under above five priors. Because the uniform prior leads to a very different posterior of SNP variance as compared to the other four priors, it was plotted separately (slide 2). Slide 3 is for the four scaled inverse chi-square priors, with same color representations in slide 1.

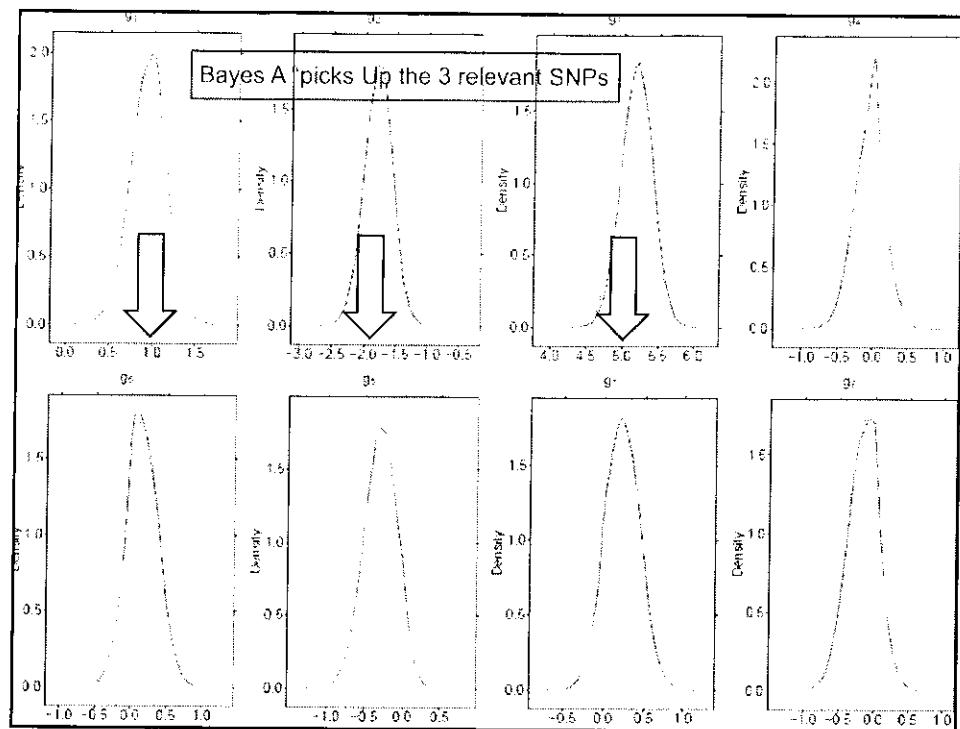
POSTERIOR DISTRIBUTION OF SNP VARIANCES UNDER UNIFORM PRIOR





## THE GOOD NEWS

Posterior distribution of SNP effects

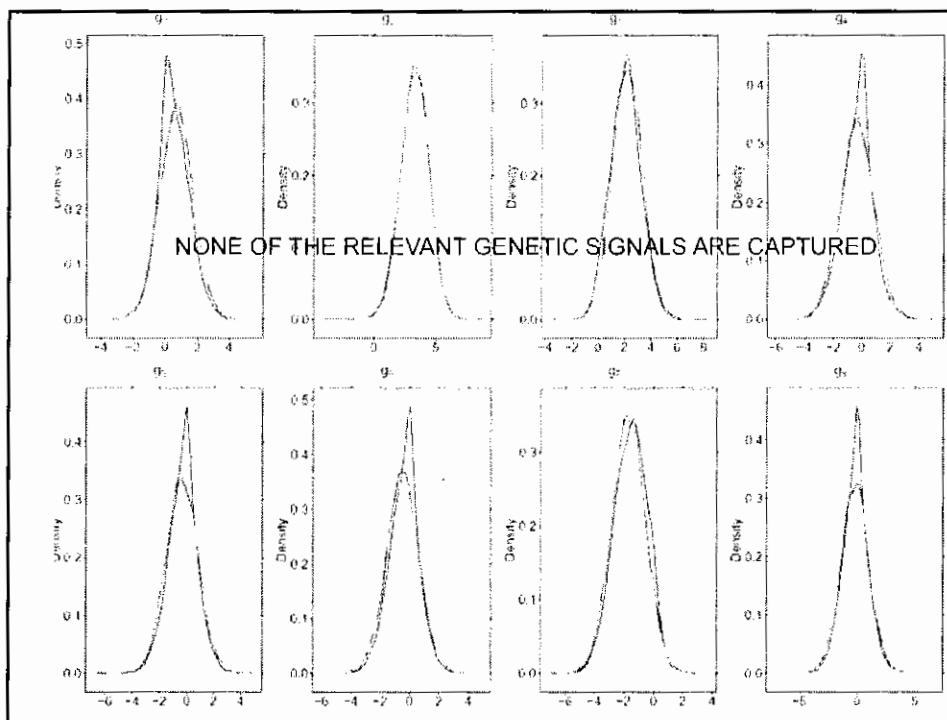


## DEATH-RESURRECTION-DEATH

Bayes A may give a distorted picture if there is non-linearity or non-additivity

### Description for slides 5, 6 and 7

- Bayes A was fitted on a simulated data of 50 observations.
  - True ~~biological~~ relationship between response and SNP ( $x_1 x_2 x_3$ ) effects
    - $Y = w_1 + 2'w_2 + \exp(x_1) \sin(x_2 - 0.5)'x_3^2 + \text{error} (\sim N(0, \text{sd}=0.25))$
  - Model fitted.
    - $Y = W\beta + Xg + \text{error}$
    - $W$  is incidence matrix for two nuisance parameters.
    - $X$  is incidence matrix for SNP effects. Besides  $x_1, x_2$  and  $x_3$ , five additional irrelevant SNPs ( $x_4$  to  $x_8$ ) added. SNP value is allele copy numbers, i.e., 0, 1 or 2
- Slide 5—Posterior distributions of SNP effects  $g_i$  ( $i = 1, 2, \dots, 8$ ) when using five different priors on  $\sigma_{g_i}^2$ , scale determined by estimated residual variance (42)
  - Black:  $\sigma_{g_i}^2 \sim \text{unif}(0, 100)$
  - Red:  $\sigma_{g_i}^2 \sim \text{scaled inverse } \chi^2(\text{df}=4, \text{scale}=42)$
  - Blue:  $\sigma_{g_i}^2 \sim \text{scaled inverse } \chi^2(\text{df}=4, \text{scale}=84)$
  - Green:  $\sigma_{g_i}^2 \sim \text{scaled inverse } \chi^2(\text{df}=8, \text{scale}=42)$
  - Pink:  $\sigma_{g_i}^2 \sim \text{scaled inverse } \chi^2(\text{df}=8, \text{scale}=84)$
- Slides 6 and 7—Posterior distributions of 8 SNP specific variances under the above five priors. Because the uniform prior leads to a very different posterior of SNP variance as compared to the other four priors, it was plotted separately (slide 6). Slide 7 is for the four scaled inverse chi-square priors, with same color representations in slide 5.



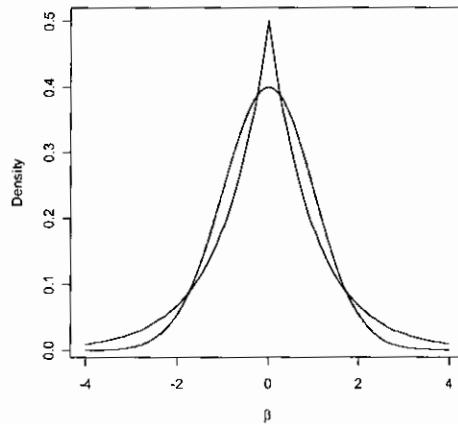
## BAYES A vs. BAYES L

(Bayes L= Bayesian Lasso)

Again Bayes L assign some variance in prior

In the Bayesian Lasso, marker effects are assigned double exponential distributions

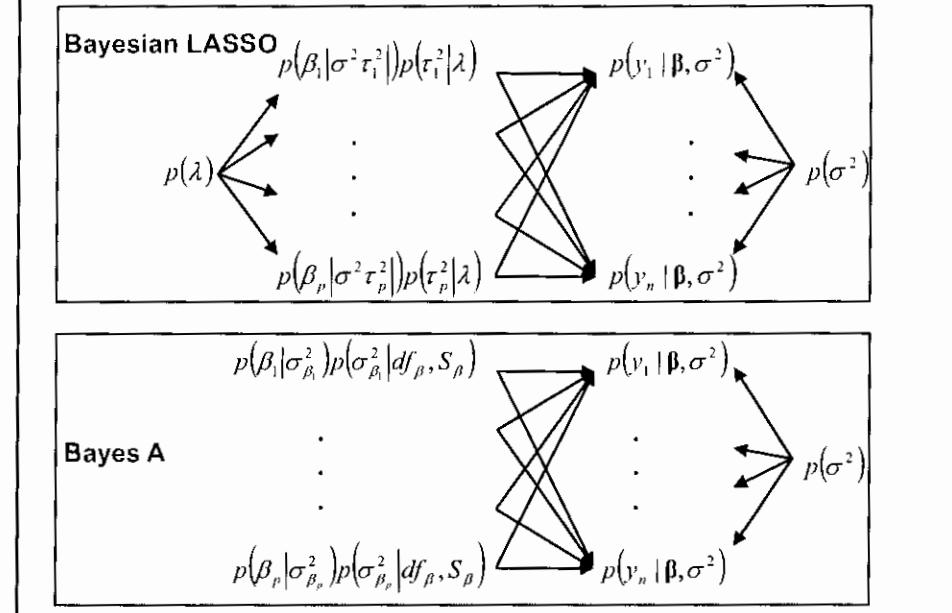
$$p(\beta) = \prod_{j=1}^p \frac{1}{2} \lambda e^{-\lambda |\beta_j|} \quad \text{EACH MARKER HAS THE SAME D.E DISTRIBUTION: NO HETEROGENEOUS VARIANCE EITHER}$$



Density of a Normal and of a Double-Exponential Distribution

in ASIO you assign > weights

Graphical Representation of the hierarchical structure of the Bayesian LASSO and Bayes A



Don't use uniform prior in Bayesian L

## ANOTHER SIMULATION (learning of marker effects versus learning signal)

(never take simulation too seriously,  
although it is great for checking ideas  
and code)

DE LOS CAMPOS ET AL. (2009)

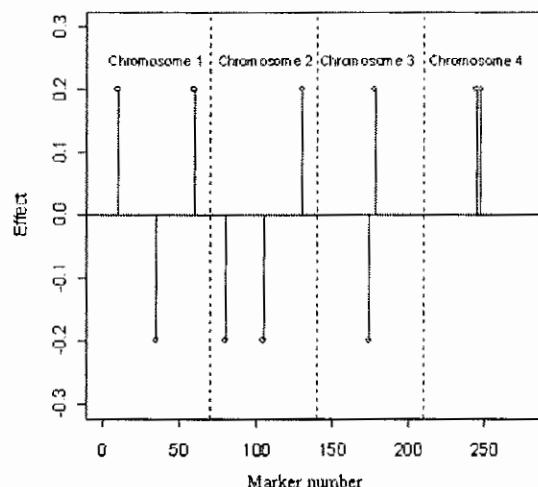
$$y_i = \sum_{j=1}^{280} x_{ij} \beta_j + \varepsilon_i \quad i = 1, \dots, 300.$$

280 markers. Residuals assumed  $N(0,1)$

**Pearson's correlation between marker genotypes  
(average across markers and 100 Monte-Carlo simulations)  
by scenario ( $X_0$ : low LD;  $X_1$  high LD).**

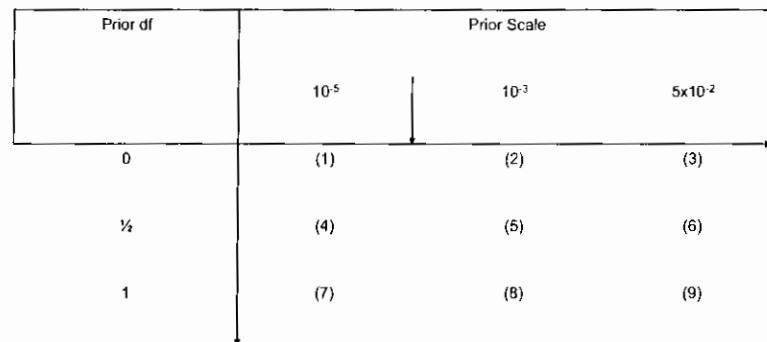
		Adjacency between markers			
		1	2	3	4
Scenario	$X_0$	0.007	0.002	-0.002	0.013
	$X_1$	0.722	0.567	0.450	0.356

Only 10 markers had effects → 270 had no effect on the trait simulated



**Positions (chromosome and marker number) and effects of markers  
(there were 280 markers, with 270 with no effect)**

# NINE SPECIFICATIONS OF BAYES A



PRIORS 1, 2, 3 ARE IMPROPER  
 PRIORS 7, 8, 9 WOULD LEAD TO CAUCHY PRIOR DISTRIBUTION OF  
 MARKER EFFECTS IF SCALE WERE 1

Table 3. Posterior estimates of residual variance ( $\sigma^2$ ) and correlation between the true and estimated value for several items ( $y$ , phenotypes;  $X\beta$ , true genomic value;  $\beta$ , marker effects; all quantities averaged of 100 MC replicates).

	$\sigma^2$		$Corr(y, X\hat{\beta})$		$Corr(X\beta, X\hat{\beta})$		$Corr(\beta, \hat{\beta})$	
	Mean <sup>1</sup>	SD <sup>2</sup>	Mean <sup>1</sup>	SD <sup>2</sup>	Mean <sup>1</sup>	SD <sup>2</sup>	Mean <sup>1</sup>	SD <sup>2</sup>
Low linkage disequilibrium between markers ( $X_0$ )								
Bayes A:								
(1)	0.518	0.062	0.839	0.027	0.580	0.063	0.102	0.048
(2)	0.941	0.089	0.577	0.028	0.721	0.092	0.200	0.022
(3)	1.074	0.105	0.496	0.032	0.701	0.106	0.199	0.020
(4)	0.394	0.053	0.895	0.022	0.531	0.060	0.079	0.051
(5)	0.824	0.077	0.652	0.025	0.699	0.079	0.183	0.028
(6)	0.950	0.089	0.578	0.027	0.722	0.088	0.201	0.021
(7)	0.173	0.053	0.960	0.015	0.455	0.057	0.042	0.043
(8)	0.575	0.056	0.813	0.019	0.606	0.066	0.116	0.044
(9)	0.710	0.066	0.728	0.020	0.659	0.072	0.152	0.037
BL	0.886	0.080	0.623	0.028	0.708	0.081	0.191	0.024

<sup>1</sup>: Mean (across 100 MC replicates) of the posterior mean. <sup>2</sup>: Between-replicate standard deviation of the estimate. <sup>3</sup>: Mean (across MC replicates) of the correlation evaluated at the posterior mean of  $\beta$ .

Kill variant over fitting      i worst  
 the model that fit  
 data too much is danger  
 because it capture noise as well

Table 3. Posterior estimates of residual variance ( $\sigma^2$ ) and correlation between the true and estimated value for several items (y, phenotypes; X $\beta$ , true genomic value; B, marker effects; all quantities averaged of 100 MC replicates).

	$\sigma^2$ Mean <sup>1</sup>	$\sigma^2$ SD <sup>2</sup>	$Corr(y, XB)$ Mean <sup>3</sup>	$Corr(y, XB)$ SD <sup>2</sup>	$Corr(XB, XB)$ Mean <sup>1</sup>	$Corr(XB, XB)$ SD <sup>2</sup>	$Corr(B, B)$ Mean <sup>3</sup>	$Corr(B, B)$ SD <sup>2</sup>
High linkage disequilibrium between markers (X <sub>1</sub> )								
Bayes A:								
(1)	0.535	0.069	0.824	0.029	0.580	0.070	0.121	0.045
(2)	0.938	0.076	0.609	0.033	0.677	0.083	0.210	0.026
(3)	1.093	0.085	0.528	0.034	0.650	0.086	0.211	0.025
(4)	0.404	0.067	0.888	0.025	0.533	0.067	0.094	0.048
(5)	0.809	0.069	0.670	0.030	0.659	0.076	0.200	0.030
(6)	0.948	0.075	0.616	0.031	0.676	0.081	0.211	0.026
(7)	0.195	0.056	0.960	0.015	0.462	0.060	0.062	0.048
(8)	0.560	0.058	0.809	0.021	0.593	0.070	0.132	0.042
(9)	0.689	0.062	0.734	0.024	0.629	0.072	0.173	0.036
BL	1.004	0.088	0.610	0.042	0.668	0.079	0.211	0.025

<sup>1</sup>: Mean (across 100 MC replicates) of the posterior mean. <sup>2</sup>: Between-replicate standard deviation of the estimate. <sup>3</sup>: Mean (across MC replicates) of the correlation evaluated at the posterior mean of B.

## Simple fixes of Bayes A

- Assign the same variance to all markers (trivial Bayesian regression problem)
- Assign the same variance to groups of markers (e.g., chromosomes or genomic regions): model comparison issue
- Assign non-informative priors to S and to the degrees of freedom y  
→ can be done. Just an algorithmic matter

## Issues and questions

- Bayes A can be “fixed”, but may not be the best thing to do. Open question...
- Bayes A, as is, may still have a good predictive (out of sample) behavior, even though it is not completely defensible
- Bayes B is Bayesianly ill-posed. If you do not believe me, check with local Bayesian statisticians...
- More reasonable: mixture at the level of the effects. This is done in Bayes C (Habier et al. 2010), for example [ASK OUR IOWA STATE HOSTS]
- Bayes B may have good predictive ability though



4. Dealing with epistatic  
interactions and non-linearities  
gene x gene  
gene x gene x gene  
gene x gene x gene x gene



.....  
(Alice in Wonderland)



## Statistical Interaction (fixed effects models)

$$y_{ijk} = \mu + A_i + B_j + AB_{ij} + e_{ijk}$$

$$E(y_{ijk}|A_i, B_j, AB_{ij}) = \mu + A_i + B_j + AB_{ij}$$

$$\begin{aligned} E(y_{ijk} - y_{ij'k'}|A_i, B_j, AB_{ij}, A_{i'}, B_j, AB_{i'j'}) &= \mu + A_i + B_j + AB_{ij} \\ &\quad - (\mu + A_{i'} + B_j + AB_{i'j'}) \\ &= A_i - A_{i'} + AB_{ij} - AB_{i'j'} \end{aligned}$$

Difference between levels of factor A depends on level of B

If factor A has  $a$  levels and factor B has  $b$  levels, the degrees of freedom are:

- $(a-1)$
- $(b-1)$
- $(a-1)(b-1)$  [assuming no-empty cells]

## Multi-SNP Fixed effects models? (unraveling “physiological epistasis” a la Cheverud)

- Lots of “main effects”
- Splendid non-orthogonality
- Lots of 2-factor interactions
- Lots of 3-factor interactions
- Lots of non-estimability
- Lots of uninterpretable high-order interactions
- Run out of “degrees of freedom”

Dealing with interactions (“statistical epistasis”): much of this took place in inspiring Iowan landscapes...



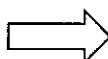
$$\Sigma_i \Sigma_j \Sigma_k \Sigma_l \text{pig}_{ijkl}^2 - (\Sigma_i \Sigma_j \Sigma_k \Sigma_l \text{pig}_{ijkl})^2 / \text{as many pigs as you got}$$

**RANDOM EFFECTS MODELS**  
FOR ASSESSING EPISTASIS REST ON:  
Cockerham (1954) and Kempthorne (1954)

--Orthogonal partition of genetic variance into additive, dominance, additive x additive, etc. ONLY if



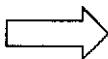
- No selection
- No inbreeding
- No assortative mating
- No mutation
- No migration
- Linkage equilibrium



A standard decomposition of phenotypic value in quantitative genetics (Falconer & Mackay, 1996) is

$$y = \mu + a + d + i + e,$$

where  $a$ ,  $d$  and  $i$  are additive, dominance and epistatic effects, respectively, and  $e$  is a residual, reflecting environmental variability. This leaves an



The  $i$  effect can be decomposed into additive x additive, additive x dominance, dominance x dominance, etc.. deviates. In what has been termed 'statistical epistasis' (Cheverud & Routman, 1995), these deviates are assumed to be random draws from some distributions

The degrees of freedom of the distribution are NOT GIVEN by the number of levels.

There is now 1 df for each type of genetic effect.

$$\begin{aligned} & N(0, \sigma_a^2) \\ & N(0, \sigma_d^2) \\ & N(0, \sigma_{aa}^2) \\ & N(0, \sigma_{ad}^2) \\ & N(0, \sigma_{dd}^2) \\ & \dots \\ & N(0, \sigma_{ddd\dots d}^2) \end{aligned}$$

Matrix representation →

$$\begin{aligned} \mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + (\mathbf{a} + \mathbf{d} + \mathbf{i}_{aa} + \mathbf{i}_{ad} + \mathbf{i}_{dd}) + \mathbf{e} \\ &= \mathbf{X}\boldsymbol{\beta} + \mathbf{g} + \mathbf{e}, \end{aligned} \quad (1)$$

where  $\boldsymbol{\beta}$  is some nuisance location vector (equal to  $\mu$  if it contains a single element);  $\mathbf{X}$  is a known incidence matrix;  $\mathbf{a}$  and  $\mathbf{d}$  are vectors of additive and dominance effects, respectively;  $\mathbf{i}_{aa}$ ,  $\mathbf{i}_{ad}$  and  $\mathbf{i}_{dd}$  are epistatic effects, and  $\mathbf{g} = \mathbf{a} + \mathbf{d} + \mathbf{i}_{aa} + \mathbf{i}_{ad} + \mathbf{i}_{dd}$  is the 'total' genetic value. Assuming that  $\mathbf{g}$  and  $\mathbf{e}$  are uncorrelated, the variance-covariance decomposition is

Variance-covariance →

$$\mathbf{V}_y = \mathbf{V}_g + \mathbf{V}_e, \quad (2)$$

where  $\mathbf{V}_y$ ,  $\mathbf{V}_g$  and  $\mathbf{V}_e$  are the phenotypic, genetic and residual variance covariance matrices, respectively. Further,

Decomposition →

$$\mathbf{V}_g = \mathbf{A}\sigma_a^2 + \mathbf{D}\sigma_d^2 + (\mathbf{A} \# \mathbf{A})\sigma_{aa}^2 + (\mathbf{A} \# \mathbf{D})\sigma_{ad}^2 + (\mathbf{D} \# \mathbf{D})\sigma_{dd}^2. \quad (3)$$

Here,  $\mathbf{A}$  is the numerator relationship matrix;  $\mathbf{D}$  is a matrix due to dominance relationships which can be computed from entries in  $\mathbf{A}$ , and the remaining matrices involve Hadamard (element by element) products of matrices  $\mathbf{A}$  or  $\mathbf{D}$ . Thus, under CK, all



DO THESE ASSUMPTIONS HOLD?

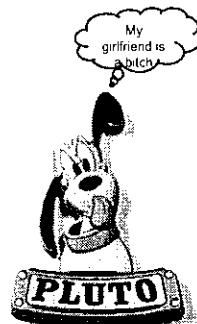
**RANDOM EFFECTS MODELS**  
FOR ASSESSING EPISTASIS REST ON:  
Cockerham (1954) and Kempthorne (1954)

--Orthogonal partition of genetic variance into additive, dominance, additive x additive, etc. **ONLY** if

- No selection
- No inbreeding
- No assortative mating
- No mutation
- No migration
- Linkage equilibrium

ALL  
ASSUMPTIONS  
VIOLATED!

Just consider  
Linkage disequilibrium



## GAMETIC DISEQUILIBRIUM (variances, covariances, correlations)

Gamete at locus	b (0)	B (1)	Marginals
a (0)	$P_{00} = \Pr(X = 0, Y = 0)$	$P_{01} = \Pr(X = 0, Y = 1)$	$P_{00} + P_{01} = p_{0+}$
A (1)	$P_{10} = \Pr(X = 1, Y = 0)$	$P_{11} = \Pr(X = 1, Y = 1)$	$P_{10} + P_{11} = p_{+1}$
Marginals	$P_{00} + P_{10} = p_{+0}$	$P_{01} + P_{11} = p_{+1}$	$P_{00} + P_{01} + P_{10} + P_{11} = 1$

$$\begin{aligned}E(X) &= 0 \times (P_{00} + P_{01}) + 1 \times (P_{10} + P_{11}) = p_{+1} \\E(X^2) &= 0^2(P_{00} + P_{01}) + 1^2(P_{10} + P_{11}) = p_{++} \\Var(X) &= E(X^2) - E^2(X) = p_{++} - p_{+1}^2 \\&= p_{+1}(1 - p_{+1})\end{aligned}$$

$$\begin{aligned}E(Y) &= 0 \times (P_{00} + P_{10}) + 1 \times (P_{01} + P_{11}) = p_{+1} \\E(Y^2) &= 0^2(P_{00} + P_{10}) + 1^2(P_{01} + P_{11}) = p_{++} \\Var(Y) &= E(Y^2) - E^2(Y) = p_{++} - p_{+1}^2 \\&= p_{+1}(1 - p_{+1})\end{aligned}$$

	Gamete at locus	b (0)	B (1)	Marginals
Parameterization 1 (3 probabilities)	a (0)	$P_{00} = \Pr(X = 0, Y = 0)$	$P_{01} = \Pr(X = 0, Y = 1)$	$P_{00} + P_{01} = p_0$
	A (1)	$P_{10} = \Pr(X = 1, Y = 0)$	$P_{11} = \Pr(X = 1, Y = 1)$	$P_{10} + P_{11} = p_1$
	Marginals	$P_{00} + P_{10} = p_{+0}$	$P_{01} + P_{11} = p_{+1}$	$P_{00} + P_{01} + P_{10} + P_{11} = 1$

$$\begin{aligned} E(XY) &= 0 \times 0 \times P_{00} + 0 \times 1 \times P_{01} \\ &\quad + 1 \times 0 \times P_{10} + 1 \times 1 \times P_{11} \end{aligned}$$

Disequilibrium parameter 

$$\begin{aligned} D &= \text{Cov}(X, Y) = E(XY) - E(X)E(Y) \\ &= P_{11} - p_{+1}p_{+1} \end{aligned}$$

If  $D > 0 \Rightarrow$  there is "positive" disequilibrium

If  $D = 0 \Rightarrow$  the two loci segregate independently,  $P_{11} = p_{+1}p_{+1}$  (stochastic independence)

If  $D < 0 \Rightarrow$  there is "negative disequilibrium"

	Gamete at locus	b (0)	B (1)	Marginals
Parameterization 2 (2 marginals, D)	a (0)	$p_{0+}p_{+0} + D$	$p_{+1}(1 - p_{1+}) - D$	$P_{00} + P_{01} = p_{+0}$
	A (1)	$p_{+0}(1 - p_{0+}) - D$	$p_{1+}p_{+1} + D$	$P_{10} + P_{11} = p_{1+}$
	Marginals	$P_{00} + P_{10} = p_{+0}$	$P_{01} + P_{11} = p_{+1}$	$P_{00} + P_{01} + P_{10} + P_{11} = 1$

Under the hypothesis of no gametic disequilibrium

$N$  = sample size (large)

$O_{ij}$  = #observed in cell  $(i, j)$

$E_{ij}$  = #expected in cell  $(i, j)$

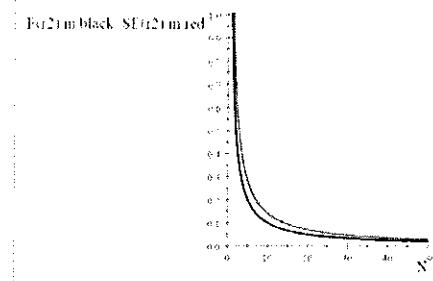
$$\begin{aligned} \chi^2 &= \sum_i \sum_j \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \\ &= \sum_i \sum_j \frac{(NP_{ij} - Np_{i+}p_{+j})^2}{Np_{i+}p_{+j}} \\ &= N \sum_i \sum_j \frac{D^2}{p_{i+}p_{+j}} = ND^2 \left( \frac{1}{p_{0+}p_{+0}} + \frac{1}{p_{0+}p_{+1}} + \frac{1}{p_{1+}p_{+0}} + \frac{1}{p_{1+}p_{+1}} \right) \\ &= ND^2 \frac{p_{1+}p_{-1} + p_{+0}p_{1+} + p_{0+}p_{+1} + p_{0+}p_{+0}}{p_{0+}p_{+0}p_{1+}p_{+1}} \\ &= ND^2 \frac{(p_{+1} + p_{+0})p_{1+} + p_{0+}(p_{+1} + p_{+0})}{p_{0+}p_{1+}p_{+0}p_{+1}} = ND^2 \frac{p_{1+} + p_{0+}}{p_{0+}p_{1+}p_{+0}p_{+1}} \\ &= N \frac{D^2}{p_{0+}p_{1+}p_{+0}p_{+1}} \\ \frac{D^2}{p_{0+}p_{1+}p_{+0}p_{+1}} &\approx \frac{\chi^2}{N} \text{ (for large } N\text{)} \end{aligned}$$

Now  $\rightarrow r = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}} = \frac{P_{11} - p_{+1}p_{1+}}{\sqrt{p_{1+}(1-p_{1+})p_{+1}(1-p_{+1})}}$

$$r^2 = \frac{(P_{11} - p_{+1}p_{1+})^2}{p_{1+}(1-p_{1+})p_{+1}(1-p_{+1})}$$

$$r^2 = \frac{D^2}{p_{1+}(1-p_{1+})p_{+1}(1-p_{+1})} \sim \frac{\chi_1^2}{N} \quad \text{Most commonly used metric for LD}$$

Under the null hypothesis  $\rightarrow E\left(\frac{\chi_1^2}{N}\right) \approx \frac{1}{N}; Var\left(\frac{\chi_1^2}{N}\right) = \frac{2}{N^2}; \sqrt{Var\left(\frac{\chi_1^2}{N}\right)} = \frac{\sqrt{2}}{N}$



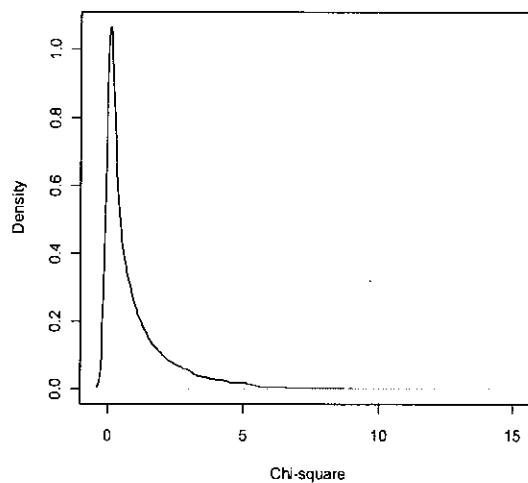
Some correlation will be found Under the null for small samples. However, the expected "studentized" value will be

$$t = \frac{\frac{1}{N}}{\frac{\sqrt{2}}{N}} = \frac{1}{\sqrt{2}} = 0.7071$$

Note  $\rightarrow$  Under  $H_0 \quad Nr^2 \sim \chi_1^2$

$$E(Nr^2) \sim E(\chi_1^2) = 1 \Leftrightarrow Var(Nr^2) = 2$$

Empirical distribution of 10,000 chi-square variates, df=1



# METRICS FOR GAMETIC DISEQUILIBRIUM

1. Squared correlation (conceals negative disequilibrium: values in 0-1)

$$r^2 = \frac{D^2}{p_{1+}(1-p_{1+})p_{+1}(1-p_{+1})}$$

2. Lewontin's  $D'$

$$P_{10} + P_{11} = p_{1+} \Rightarrow P_{11} \leq p_{1+}$$

$$P_{01} + P_{11} = p_{+1} \Rightarrow P_{11} \leq p_{+1}$$

$$D = P_{11} - p_{1+}p_{+1}$$

IF  $D > 0$

$$D_{\max} = p_{1+} - p_{1+}p_{+1}$$

$$= p_{1+}(1 - p_{+1}) > 0 \text{ [actually 0.5]}$$

$$D_{\max} = p_{+1} - p_{1+}p_{+1}$$

$$= p_{+1}(1 - p_{1+}) > 0 \text{ [actually 0.5]}$$

$$D_{\max} = \min[p_{1+}(1 - p_{+1}), p_{+1}(1 - p_{1+})]$$

Positive  
disequilibrium

Negative  
disequilibrium

$$D = P_{11} - p_{1+}p_{+1}$$

IF  $D < 0$

$$|D_{\max}| = |0 - p_{1+}p_{+1}| = p_{1+}p_{+1}$$

$$|D_{\max}| = |0 - p_{0+}p_{+0}| = p_{0+}p_{+0}$$

$$|D_{\max}| = \min(p_{0+}p_{+0}, p_{1+}p_{+1})$$

Lewontin's  $D'$

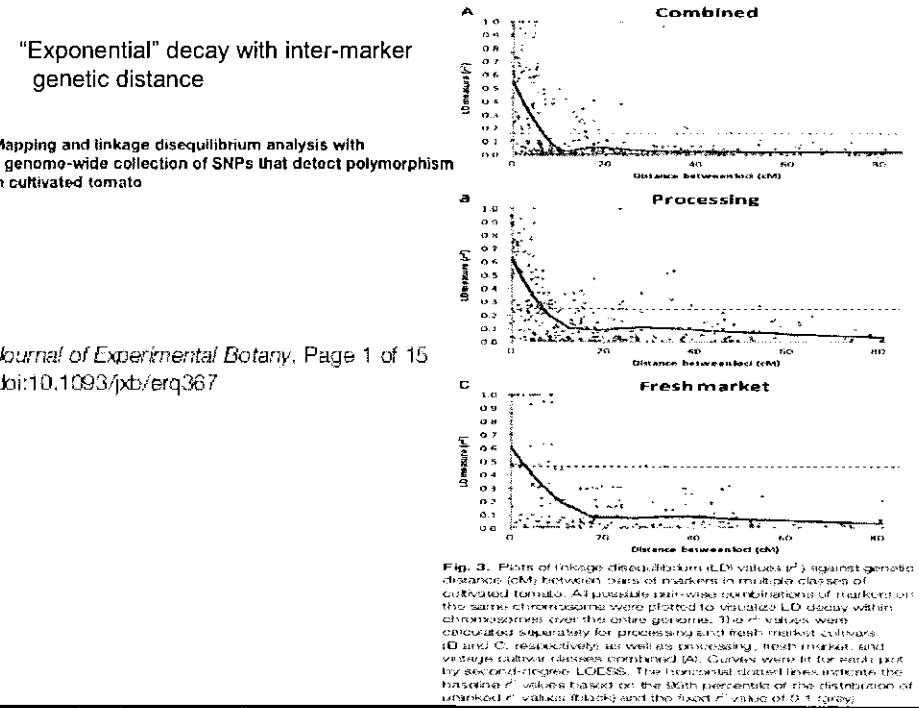
$$D' = \frac{D}{|D_{\max}|}$$

$$|D_{\max}| = \min[p_{1+}(1 - p_{+1}), p_{+1}(1 - p_{1+})] \text{ if } D > 0$$

$$|D_{\max}| = \min(p_{0+}p_{+0}, p_{1+}p_{+1}) \text{ if } D < 0$$

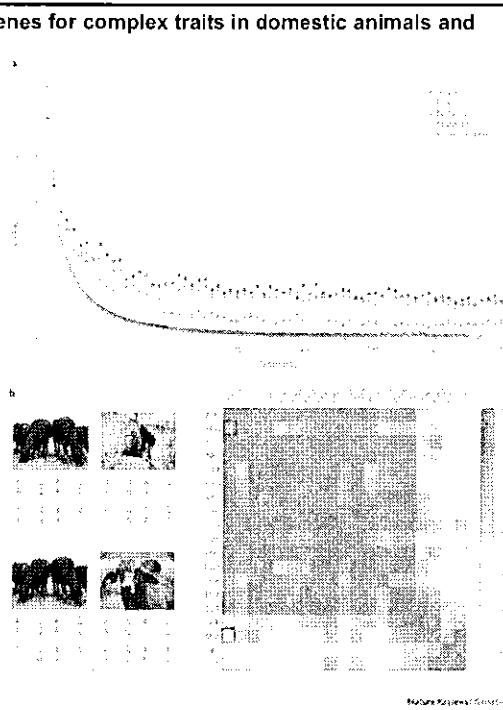
You can not compare LD between two populations  
because it is dependent to frequency (it is  
the same as  $\chi^2$  for threshold traits)

# VISUAL DISPLAY OF LD



Goddard ME, Hayes BJ. 2009. Mapping genes for complex traits in domestic animals and their use in breeding programmes  
Nature Reviews Genetics 10, 381-391

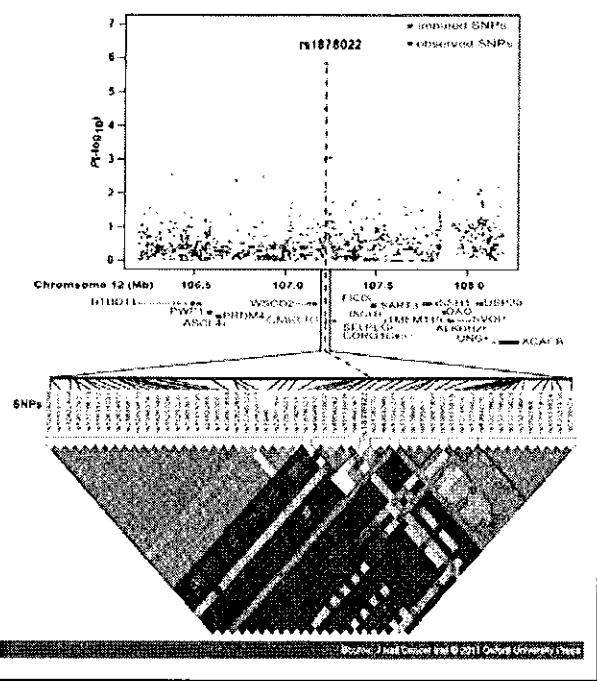
a | Decline of LD with distance between pairs of SNPs as measured by LD within breeds of cattle (derived from approximately 35,000 SNPs, and a human population with northern and western European ancestry (CEPH cohort))<sup>11</sup>. b | LD between breeds of cattle. The heat map shows the correlation between LD in different breeds for SNPs within 10 kb of each other. For two closely related breeds (Angus and Red Angus) the correlation is high, as shown in a hypothetical example in which a-q and A-Q chromosomes are common in both breeds (upper box). However, when Angus is compared with Brahman (a distantly related breed) the correlation is low and, in the hypothetical example, Brahman chromosomes often carry a-Q, which is a rare haplotype in Angus (lower box). In fact, the correlation is low for any combination of a Bos indicus breed and a Bos taurus breed.  
 ANG, Angus; BMA, Beefmaster; BRM, Brahman; BSW, Brown Swiss; CHL, Charolais; GIR, Gir; GNS, Guernsey; HFD, Hereford; HOL, Holstein; JER, Jersey; LIM, Limousin; NDA, N'Dama; NEL, Nelore; NRC, Norwegian Red; PMT, Piedmontese; RGM, Romagnola; RGU, Red Angus; SGT, Santa Getrudis; SHK, Sheko.



### Genome-Wide Association Study of Survival in NSCLC

Journal of the National Cancer Institute. 2011. 103(10): 817-825

**Figure 3.** Linkage disequilibrium structure and association of observed and imputed single-nucleotide polymorphisms (SNPs) surrounding rs1878022 on chromosome 12. The linkage disequilibrium structure was created with the GOLD heat map Haplovew 4.0 color scheme using the standardized disequilibrium coefficient, D', with associations expressed as  $-\log_{10}(P)$  and calculated by the multivariable Cox

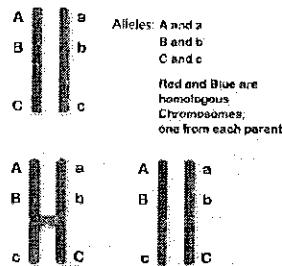


# EVOLUTION OF LD FOLLOWED BY RANDOM MATING

Suppose we start with the following gametic array and that loci are physically linked

Gamete at locus	b (0)	B (1)	Marginals
a (0)	$p_{0+}p_{-0} + D_{[0]}$	$p_{+1}(1 - p_{1-}) - D_{[0]}$	$P_{00} + P_{01} = p_{0+}$
A (1)	$p_{+0}(1 - p_{-0}) - D_{[0]}$	$p_{1+}p_{+1} + D_{[0]}$	$P_{10} + P_{11} = p_{1+}$
Marginals	$P_{00} + P_{10} = p_{+0}$	$P_{01} + P_{11} = p_{+1}$	$P_{00} + P_{01} + P_{10} + P_{11} = 1$

Recombination between Homologous Chromosomes



Let  $\Pr(\text{recombination}) = c$

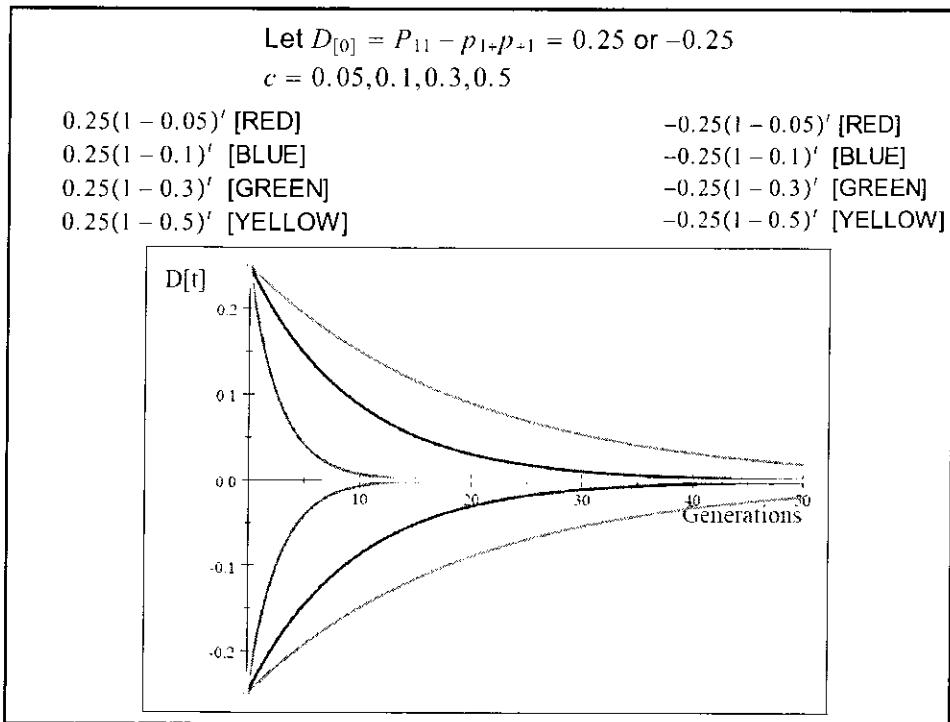
$$\begin{aligned}
 P_{11[1]} &= \Pr(AB|_{[1]}) = \Pr(AB|\text{recombination})\Pr(\text{recombination}) \\
 &\quad + \Pr(AB|\text{no recombination})_{[0]}\Pr(\text{no recombination}) \\
 &= p_{1+}p_{+1}c + P_{11[0]}(1 - c) \\
 D_{[1]} &= P_{11[1]} - p_{1+}p_{+1} \\
 &= p_{1+}p_{+1}c + P_{11[0]}(1 - c) - p_{1+}p_{+1} \\
 &= P_{11[0]} - p_{1+}p_{+1} - c(P_{11[0]} - p_{1+}p_{+1}) \\
 &= (1 - c)D_{[0]}
 \end{aligned}$$

$$\begin{aligned}
 P_{11[2]} &= \Pr(AB|_{[2]}) = \Pr(AB|\text{recombination})\Pr(\text{recombination}) \\
 &\quad + \Pr(AB|\text{no recombination})_{[1]}\Pr(\text{no recombination}) \\
 &= p_{1+}p_{+1}c + P_{11[1]}(1 - c)
 \end{aligned}$$

$$\begin{aligned}
 D_{[2]} &= p_{1+}p_{+1}c + P_{11[1]}(1 - c) - p_{1+}p_{+1} \\
 &= p_{1+}p_{+1}c - P_{11[1]}c + D_{[1]} \\
 &= -cD_{[1]} + D_{[1]} = D_{[1]}(1 - c) \\
 &= (1 - c)^2 D_{[0]}
 \end{aligned}$$

In general  $\rightarrow D_{[t]} = (1 - c)^t D_{[0]}$

Recall  $\rightarrow |D_{\max}| = \min[p_{1+}(1 - p_{+1}), p_{+1}(1 - p_{1+})] \text{ if } D > 0$   
 $|D_{\max}| = \min(p_{0+}p_{+0}, p_{1+}p_{+1}) \text{ if } D < 0$



## POPULATION ADMIXTURE

Often populations have a hidden structure and LD can be due to admixture or hidden heterogeneity

Haplotype	Probability	Sub-population 1	Sub-population 2	Mixture (50:50)
AB	$P_{11}$	0.0025	0.9025	0.4525
Ab	$P_{10}$	0.0475	0.0475	0.0475
bA	$P_{01}$	0.0475	0.0475	0.0475
ab	$P_{00}$	0.9025	0.0025	0.4525
	$D = P_{11}P_{00} - P_{10}P_{01}$	0	0	0.2025

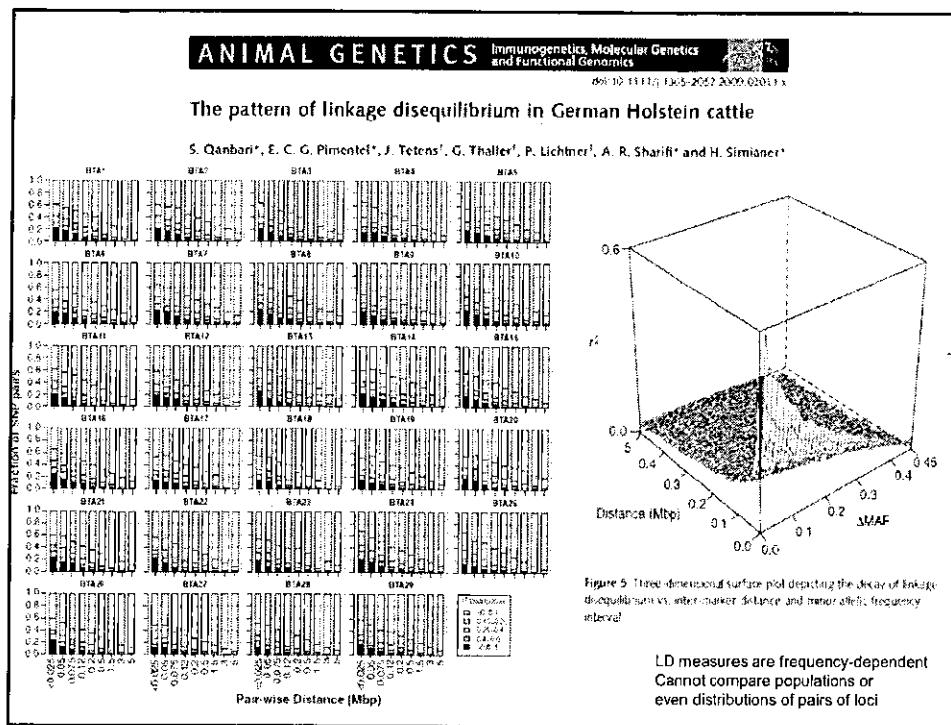
Conceivably, if genotypic frequencies vary over groups with LD=0, mixing these groups results in LD

If combination of "A" and "B" favored, it creates LD.

Two populations in LD:  
one in positive LD; the other in negative LD  
Mixed at 50:50 yields

Haplotype	Probability	Sub-population 1	Sub-population 2	Mixture (50:50)
AB	$P_{11}$	0.4525	0.0475	0.25
Ab	$P_{10}$	0.0475	0.4525	0.25
bA	$P_{01}$	0.0475	0.4525	0.25
ab	$P_{00}$	0.4525	0.0475	0.25
	$D = P_{11}P_{00} - P_{01}P_{10}$	0.2025	-0.2025	0

Mixing populations can also eliminate LD



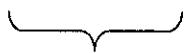
$r^2 \rightarrow$  frequency dependent.

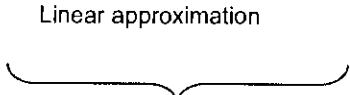
# A VIEW OF LINEAR MODELS

## (as employed in q. genetics)

Mathematically, can be viewed as a "local" approximation of a complex process

$$f(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \frac{f'''(a)}{3!}(x - a)^3 + \dots + \frac{f^{(n)}(a)}{n!}(x - a)^n + \dots$$

 Linear approximation

 Quadratic approximation

 n<sup>th</sup> order approximation

LINEAR MODELS ARE "LOCAL"  
(Feldman and Lewontin, 1974)

### Example

$$y = g(x) + e$$

Response variate      Model residual  
Some function of a covariate  $x$

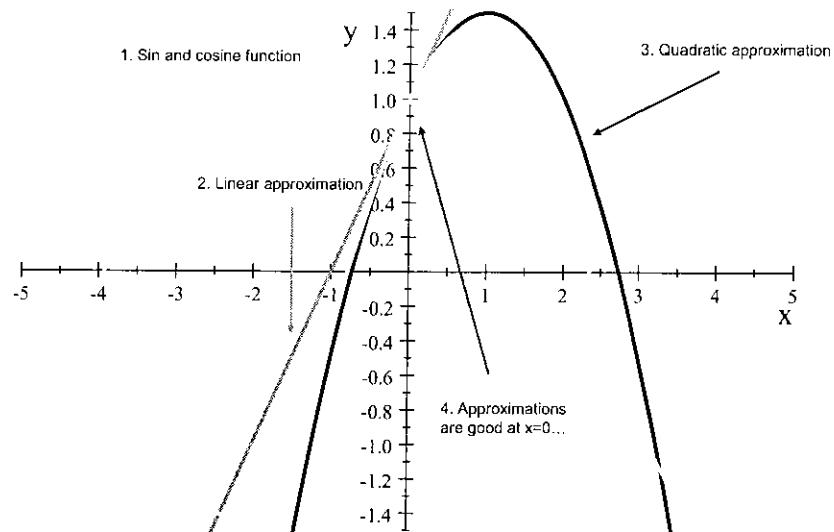
Suppose  $g(x) = \sin(x) + \cos(x)$

$$\begin{aligned}\frac{d}{dx}[\sin(x)] &= [\cos x] \\ \frac{d}{dx}[\cos(x)] &= [-\sin x] \\ \frac{d}{dx}[\sin(x) + \cos(x)] &= [\cos x - \sin x] \\ \frac{d}{dx}[\cos x - \sin x] &= [-\cos x - \sin x] \\ \frac{d^2}{dx^2}[\sin(x) + \cos(x)] &= [-\cos x - \sin x]\end{aligned}$$

Second-order Taylor series expansion about 0

$$\begin{aligned}[\sin(x) + \cos(x)] &\approx [\sin(0) + \cos(0)] + [\cos 0 - \sin 0](x - 0) + \frac{1}{2}[-\cos 0 - \sin 0](x - 0)^2 \\ &= 1 + x - \frac{x^2}{2}\end{aligned}$$

How good are the linear and quadratic approximations? Recall that a Taylor series provides a local approximation only...



### Finding structure from noisy data without models we have measurement noise....

evaluate function  $\sin(x)+\cos(x)$  at  $x=0, 0.5$  and  $1$

True values are:

```
> sin(0)+cos(0)
[1] 1
> sin(0.5)+cos(0.5)
[1] 1.357008
> sin(1)+cos(1)
[1] 1.381773
```

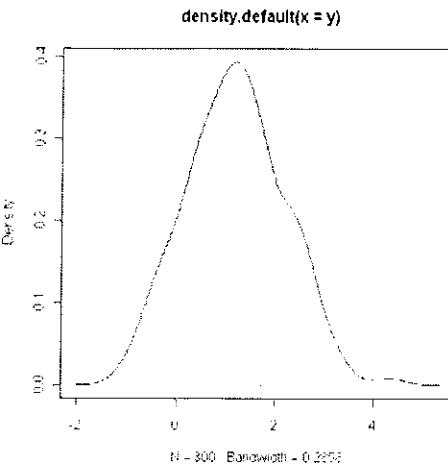
VERY CLOSE TO EACH OTHER  
NOISE CAN MASK SIGNALS!

Create an R data set (N=300) from adding 100 N(0,1) residuals to each of the 3 values

```
> y0<-sin(0)+cos(0) +rnorm(100,0,1)
> y05<-sin(0.5)+cos(0.5)+rnorm(100,0,1)
> y1<-sin(1)+cos(1) +rnorm(100,0,1)
> y<-c(y0,y05,y1)
```

### MEASURING MACHINE 1

This is arrived at by using kernel estimation



From a finite sample, provide estimate for each of an infinite number of points

$$\hat{f}_h = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$

$$= \frac{1}{n} \sum_{i=1}^n \frac{1}{h} K\left(\frac{x - x_i}{h}\right)$$

$$K(z) \geq 0$$

$$\int K(z) dz = 1$$

$$\int z K(z) dz = 0$$

$$\int z^2 K(z) dz \geq 0$$

Wasserman L. 2004. All of Statistics. Springer

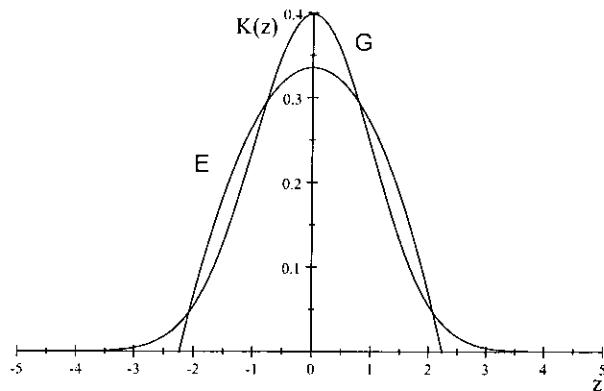
Some kernels

Epanechnikov

$$K(z) = \begin{cases} \frac{3}{4}(1 - \frac{z^2}{5})/\sqrt{5} & \text{for } |z| \leq 5 \\ 0 & \text{otherwise} \end{cases}$$

Gaussian

$$K(z) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{z^2}{2})$$

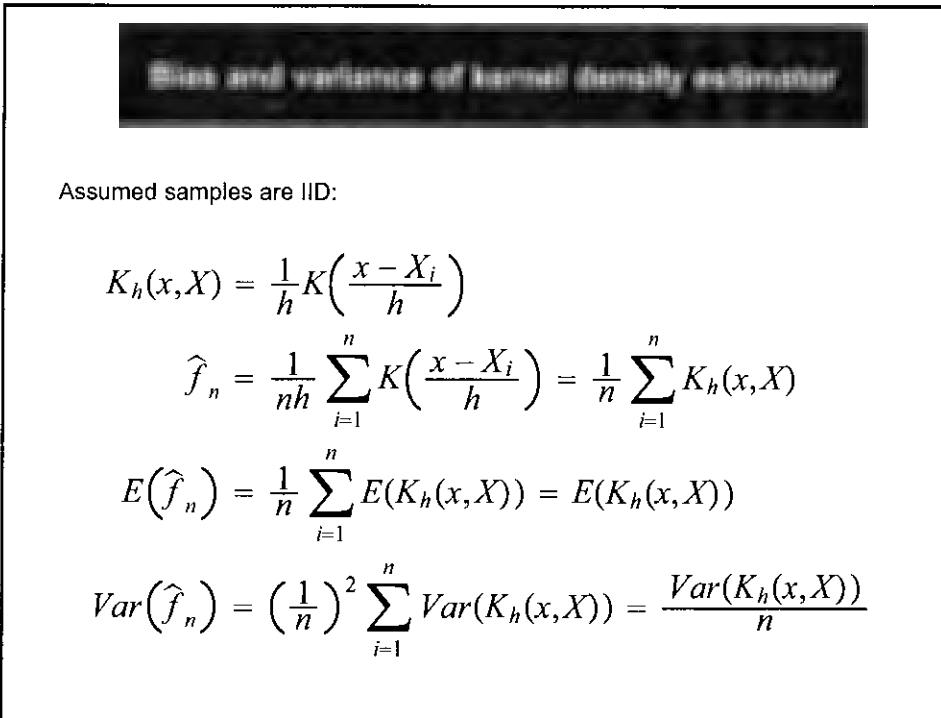
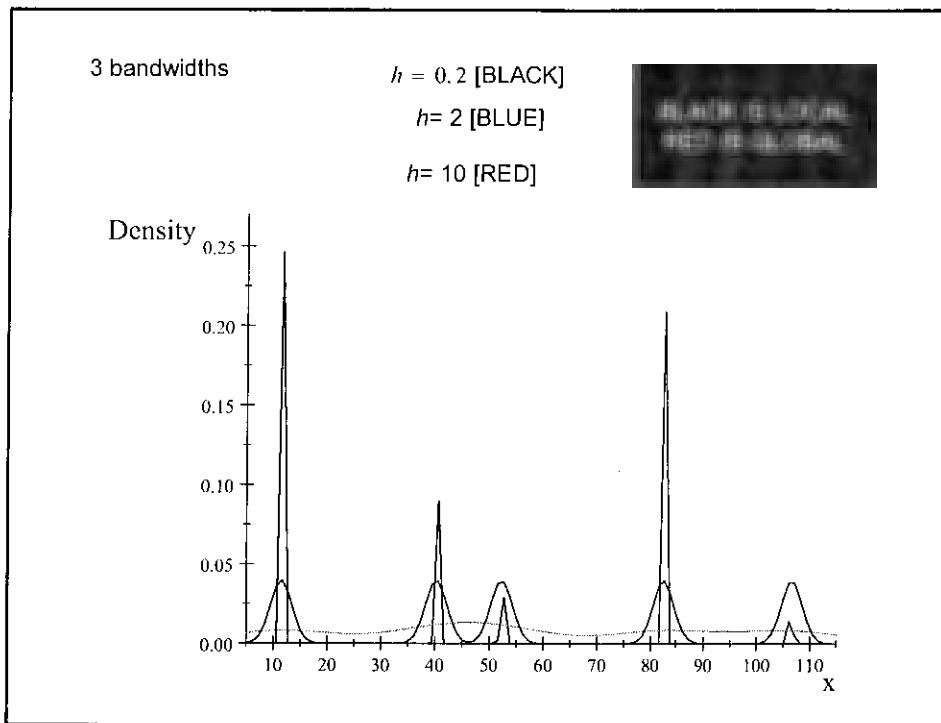


**Example:** 5 draws from exponential distribution with parameter 40

52, 32, 82, 55, 11, 47, 106.63, 40, 21

Gaussian kernel with bandwidth h

$$\begin{aligned}\hat{f}_h &= \frac{1}{5h} \sum_{i=1}^5 \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-x_i}{h}\right)^2\right) \\ &= \frac{1}{5h\sqrt{2\pi}} \left[ \exp\left(-\frac{1}{2}\left(\frac{x-52.32}{h}\right)^2\right) + \exp\left(-\frac{1}{2}\left(\frac{x-82.55}{h}\right)^2\right) \right. \\ &\quad \left. + \exp\left(-\frac{1}{2}\left(\frac{x-11.47}{h}\right)^2\right) + \exp\left(-\frac{1}{2}\left(\frac{x-106.63}{h}\right)^2\right) + \exp\left(-\frac{1}{2}\left(\frac{x-40.21}{h}\right)^2\right) \right]\end{aligned}$$



$$E(K_h(x, X)) = \int \frac{1}{h} K\left(\frac{x-t}{h}\right) f(t) dt$$

**Change variables** → Let  $u = \frac{x-t}{h} \Rightarrow du = -\frac{dt}{h} \Rightarrow \left| \frac{dt}{du} \right| = h$

$$E(K_h(x, X)) = \int \frac{1}{h} K\left(\frac{x-t}{h}\right) f(t) dt$$

$$= \int K(u) f(x - hu) du$$

Expanding at  $u = 0$

$$f(x - hu) \approx f(x) + f'(x - hu)|_{u=0}(x - hu - x) + \frac{1}{2} f''(x - hu)|_{u=0}(x - hu - x)^2 + \dots$$

$$\approx f(x) - f'(x)hu + \frac{1}{2} f''(x)h^2 u^2$$

$$E(K_h(x, X)) \approx \int K(u) \left[ f(x) - f'(x)hu + \frac{1}{2} f''(x)h^2 u^2 \right] du$$

$$= f(x) \int K(u) du - f'(x)h \int u K(u) du + \frac{1}{2} f''(x)h^2 \int u^2 K(u) du$$

$$E(K_h(x, X)) \approx \int K(u) \left[ f(x) - f'(x)hu + \frac{1}{2} f''(x)h^2 u^2 \right] du$$

$$= f(x) \int K(u) du - f'(x)h \int u K(u) du + \frac{1}{2} f''(x)h^2 \int u^2 K(u) du$$

Using properties of the kernel function (see above)  
 $\int K(u) du = 1$ ;  $\int u K(u) du = 0$ ;  $\sigma_K^2 = \int u^2 K(u) du$

$$E(K_h(x, X)) \approx f(x) + \frac{1}{2} f''(x)h^2 \sigma_K^2$$

Bias is

$$E(K_h(x, X)) - f(x) \approx \frac{1}{2} f''(x)h^2 \sigma_K^2$$

$$E(\hat{f}_n) \approx f(x) + \frac{1}{2} f''(x)h^2 \sigma_K^2$$

Similarly

$$Var(K_h(x, X)) = E(K_h(x, X))^2 - E^2(K_h(x, X))$$

$$\begin{aligned} E(K_h(x, X))^2 &= \int \left(\frac{1}{h}\right)^2 K^2\left(\frac{x-t}{h}\right) f(t) dt \\ &= \int \left(\frac{1}{h}\right)^2 K^2(u) f(u) h du \\ &= \frac{1}{h} \int K^2(u) \left[ f(x) - f'(x)hu + \frac{1}{2}f''(x)h^2u^2 \right] du \\ &\approx \frac{f(x)}{h} \int K^2(u) du \end{aligned}$$

$$\rightarrow Var(\hat{f}_n) = \frac{Var(K_h(x, X))}{n} \approx \frac{f(x)}{nh} \int K^2(u) du$$

The conditional risk (mean squared error= variance+ squared bias) is

$$R(f, \hat{f}_n | x) = \frac{f(x)}{nh} \int K^2(u) du + \frac{1}{4}(f''(x))^2 h^4 \sigma_K^4$$

The integrated risk is

$$\begin{aligned} R(f, \hat{f}_n) &= \int \left[ \frac{1}{nh} \int K^2(u) du \right] f(x) dx \\ &\quad + \int \left[ \frac{1}{4}(f''(x))^2 h^4 \sigma_K^4 \right] dx \\ &= \frac{\int K^2(u) du}{nh} \int f(x) dx + \frac{h^4 \sigma_K^4}{4} \int (f''(x))^2 dx \\ &= \frac{\int K^2(u) du}{nh} + \frac{h^4 \sigma_K^4}{4} \int (f''(x))^2 dx \end{aligned}$$

$$\frac{dR(f, \hat{f}_n)}{dh} = -\frac{\int K^2(u)du}{nh^2} + \frac{h^3 \sigma_K^4}{4} \int (f''(x))^2 dx$$

**Set to 0**

$$\frac{\int K^2(u)du}{nh^2} = \frac{h^3 \sigma_K^4}{4} \int (f''(x))^2 dx$$

$$h^5 = \frac{\frac{4 \int K^2(u)du}{n}}{\sigma_K^4 \int (f''(x))^2}$$

$$h = \frac{1}{(n)^{\frac{4}{5}}} \sqrt{\frac{4 \int K^2(u)du}{n \sigma_K^4 \int (f''(x))^2}}$$

Not very useful because it depends on unknown  $f(x)$  through second derivatives  $f''(x)$



$$\mathbf{x}' = [x_{i1}, x_{i2}, \dots, x_{ip}]$$

$$\Rightarrow \hat{f}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h_1 h_2 \dots h_p} \prod_{j=1}^p K\left(\frac{x_{ij} - X_{ij}}{h_j}\right)$$

Sample size required to obtain  $MSE < 0.1$  for multivariate normal density and optimal  $h$  selected.

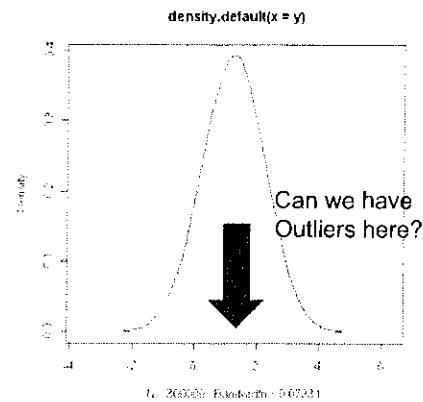
Dimension	$n$
1	4
2	19
3	67
4	223
5	768
6	2790
7	10700
8	43700
9	187000
10	842000



Create a larger R data set (N=300000) by adding 100000 N(0,1) residuals to each of the 3 values

```
> y0<-sin(0)+cos(0)      +rnorm(100000,0,1)
> y05<-sin(0.5)+cos(0.5) +rnorm(100000,0,1)
> y1<-sin(1)+cos(1)      +rnorm(100000,0,1)
> y<-c(y0,y05,y1)
```

CANNOT SEE UNDERLYING STRUCTURE.  
LARGE NOISE (ERROR VARIANCE)

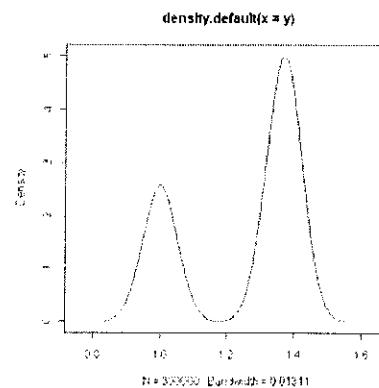


Now we get a more precise measuring instrument with variance 0.05

```
> y0<-sin(0)+cos(0)      +rnorm(100000,0,.05)
> y1<-sin(1)+cos(1)      +rnorm(100000,0,.05)
> y05<-sin(0.5)+cos(0.5) +rnorm(100000,0,.05)
```

#### MEASURING MACHINE 2

STRUCTURE IS REVEALED BUT  
WE CANNOT DIFFERENTIATE  
BETWEEN TWO OF THE UNDERLYING  
VALUES



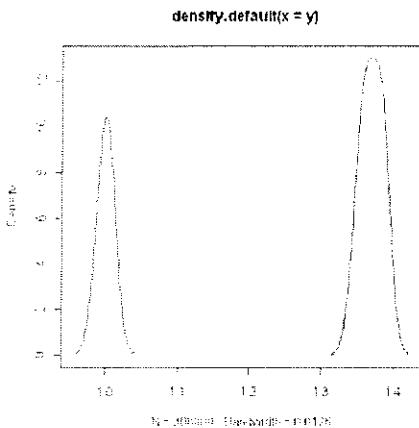
...SO WE BUY ANOTHER INSTRUMENT WITH VARIANCE 0.001!

```
> y0<-sin(0)+cos(0)      +rnorm(100000,0,.001)
> y1<-sin(1)+cos(1)      +rnorm(100000,0,.001)
> y05<-sin(0.5)+cos(0.5) +rnorm(100000,0,.001)
> y<-c(y0,y05,y1)
```

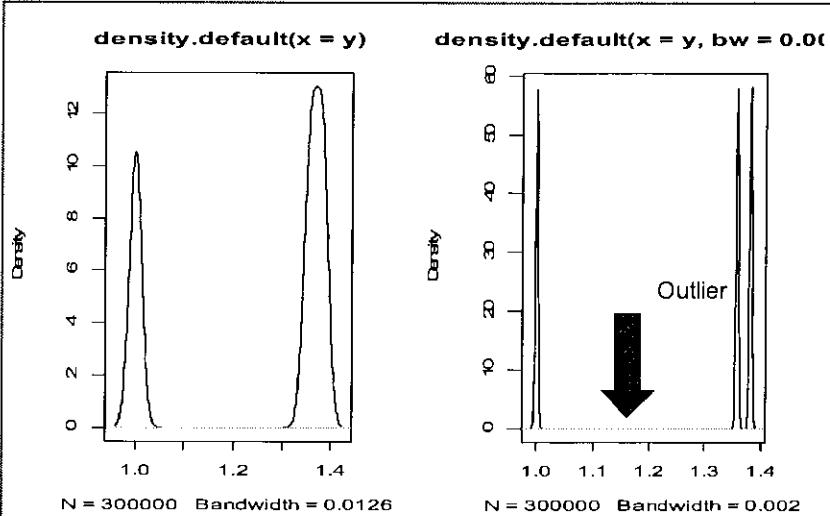
### MEASUREMENT MACHINE 3

STILL CANNOT DIFFERENTIATE  
BETWEEN THE

```
> sin(0.5)+cos(0.5)
[1] 1.357008
> sin(1)+cos(1)
[1] 1.381773
```



HOWEVER, NON-PARAMETRIC DENSITY ESTIMATES DEPEND ON SOME BANDWIDTH PARAMETER. BY REDUCING IT, WE CAN SEE THE ENTIRE STRUCTURE OF THE PROBLEM...



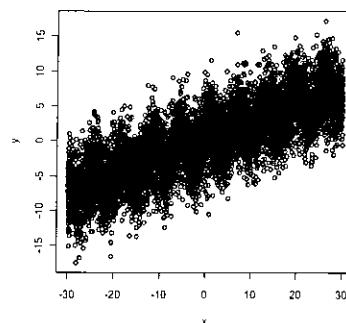
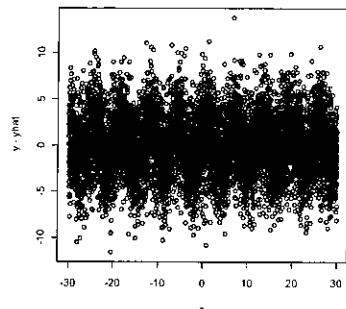
FINDING "STRUCTURE" WITH A LINEAR MODEL  
 (some call this "learning architecture")

We are given (x,y) data ( $n=10,000$ ). It looks like this and we run a linear regression

$$y\hat{=} 0.07936 + 0.24814 \cdot x$$

```
> cor(x,y)
[1] 0.8064256
```

```
> cor(y,yhat)
[1] 0.8064256
```



← RESIDUALS DISPLAY SINUSOIDAL BEHAVIOR

TRUE MODEL

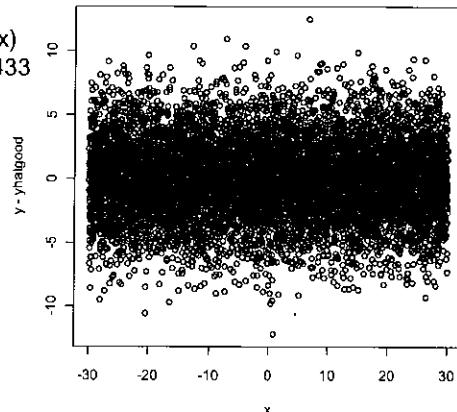
```
> e<-rnorm(10000,0,sqrt(9))
> x<-runif(10000,-30,30)
> a<-0.10
> b<-0.25
> y<-a+b*x+sin(x)+cos(x)+e

> model<-lm(y~x+sin(x)+cos(x))
```

> Coefficients:

	(Intercept)	x	sin(x)	cos(x)
>	0.1030	0.2489	0.9518	0.9433

RESIDUALS LOOK RANDOM



WE GENERATE A NEW SAMPLE AT THE SAME VALUES OF X

```
> enew<-rnorm(10000,0,sqrt(9))
>ynew<-a+b*x+sin(x)+cos(x)+enew
```

CALCULATE PREDICTIVE MEAN SQUARED ERROR

```
> msepredbadmodel<-sum((ynew-yhat)**2/10000)
> msepredbadmodel
[1] 9.725709
```

```
> msepredgoodmodel<-sum((ynew-yhatgood)**2/10000)
> msepredgoodmodel
[1] 8.729272
```

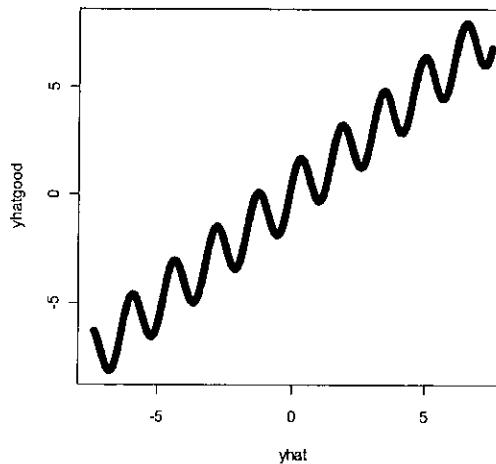
CALCULATE PREDICTIVE CORRELATIONS

```
> cor(yhat,ynew)
[1] 0.8070097
> cor(yhatgood,ynew)
[1] 0.828854
```

MSE(Good)/MSE(Bad)=0.8975  
MSE(Bad)/MSE(Good)=1.1141  
  
Cor(BAD)/Cor(GOOD)=0.9736

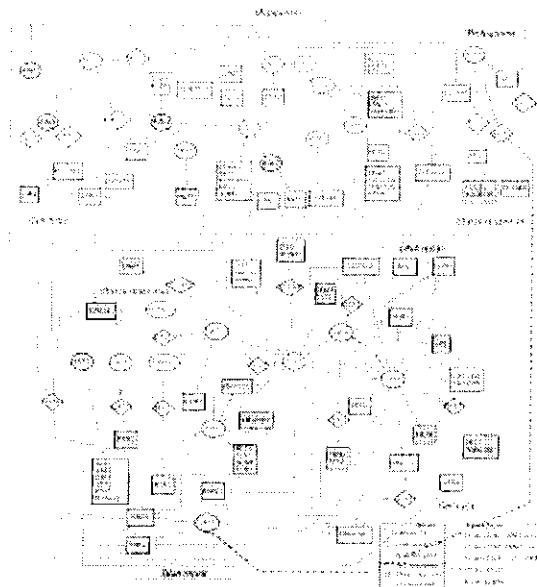
```
> lm(yhat~yhatgood)
Coefficients:
(Intercept)  yhatgood
 0.005653   0.953468
```

**DO NOT TRUST CORRELATIONS!**



## **RECALLING COMPLEXITY...**

How one  
Would model  
something like this?



## **Heal Thyself: Systems Biology Model Reveals How Cells Avoid Becoming Cancerous.** ScienceDaily (May 21, 2006)

## What to do in genomic-assisted analysis of complex genetic signals?

- Include all markers, model all possible interactions?  
Unrealistic...
  - Select sets of influential markers via model selection
    - Huge search space
    - Frequentist methods “err” probabilistically
    - Bayesian model selection (RJMC) difficult to tune
  - Use LASSO (least absolute shrinkage and selection operator): Tibshirani (1996). What about interactions?
  - Explore model-free techniques that have been used successfully in many domains
    - **semi-parametric regression**
    - **machine learning**: focus on prediction, learning mappings from inputs to outputs

## DEFINITION OF MACHINE LEARNING (Wikipedia)

**Machine learning:** subfield of artificial intelligence concerned with design and development of algorithms that allow computers (machines) to improve their performance over time (to learn) based on data,

A major focus of machine learning research is to automatically produce (induce) models, such as rules and patterns, from data. Hence, machine learning is closely related to fields such as data mining, statistics, inductive reasoning, pattern recognition, and theoretical computer science.

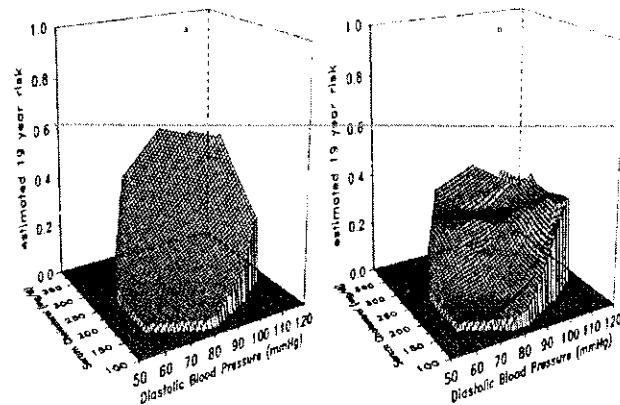
## 5. Introduction to non-parametric curve fitting:

Loess, kernel regression,  
reproducing kernel methods,  
neural networks

### Distinctive aspects of non-parametric fitting

- **Objectives:** investigate patterns free of strictures imposed by parametric models
- Can produce surprising results
- Regression coefficients appear but (typically) do not have an obvious interpretation
- Often have very good predictive performance in cross-validation
- Tuning methods similar to those for parametric methods

## Example: thin-plate splines



$$f(\mathbf{x}_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \sum_{j=1}^N \alpha_j \left[ (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 \right] \log \left[ (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 \right]$$

Risk of heart attack after 19 years as a function of cholesterol level and blood pressure.  
Left: logistic regression model. Right: thin plate spline fit. Wahba (2007)

## LOESS REGRESSION:

Non-parametric exploration  
of inbreeding depression for  
yield and somatic cell count  
in Jersey cattle

# AN OVERVIEW OF LOWESS REGRESSION

1) DATA POINTS  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$

2) SPANNING PARAMETER  $f$ :  $0 < f < 1$

$$k = fn; k = \text{LARGEST INTEGER} \leq fn$$

3) FOR EACH  $x_0$  FIND  $k$  POINTS  $x_i$  "CLOSEST" TO  $x_0$   
 $N(x_0)$  = NEIGHBORHOOD OF  $k$  POINTS

4) COMPUTE  $\Delta(x_0) = \max_{x_i \in N(x_0)} |x_0 - x_i|$

5) TO EACH  $(x_i, y_i)$ ,  $x_i \in N(x_0)$  ASSIGN WEIGHT

$$w_i(x_0) = \left\{ 1 + \left[ \frac{|x_0 - x_i|}{\Delta(x_0)} \right]^2 \right\}^{-1}$$

6) FIT BY WEIGHTED LEAST-SQUARES

$$\sum_{i=1}^k w_i(x_0) (y_i - \beta_0 - \beta_1 x_i - \beta_2 x_i^2)^2$$

$$\text{RETURN } \hat{y}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2$$

7) REPEAT FOR EACH OF THE  $x_0$

# ROBUST LOWESS

- STANDARD LOWESS NOT ROBUST
  - ⇒ BASED ON LEAST-SQUARES WEIGHTS
- BI-SQUARE LOWESS
  - ⇒ RE-WEIGHT POINTS ACCORDING TO RESIDUAL
  - ⇒ IF RESIDUAL LARGE, WEIGHT IS DECREASED

1) FIT DATA USING STANDARD LOESS

2) CALCULATE LOESS RESIDUALS  $y_i - \hat{y}_i$

3) COMPUTE  $\hat{q}_{1/2} = median[y_i - \hat{y}_i]$

4) CALCULATE BI-SQUARE ROBUST WEIGHTS

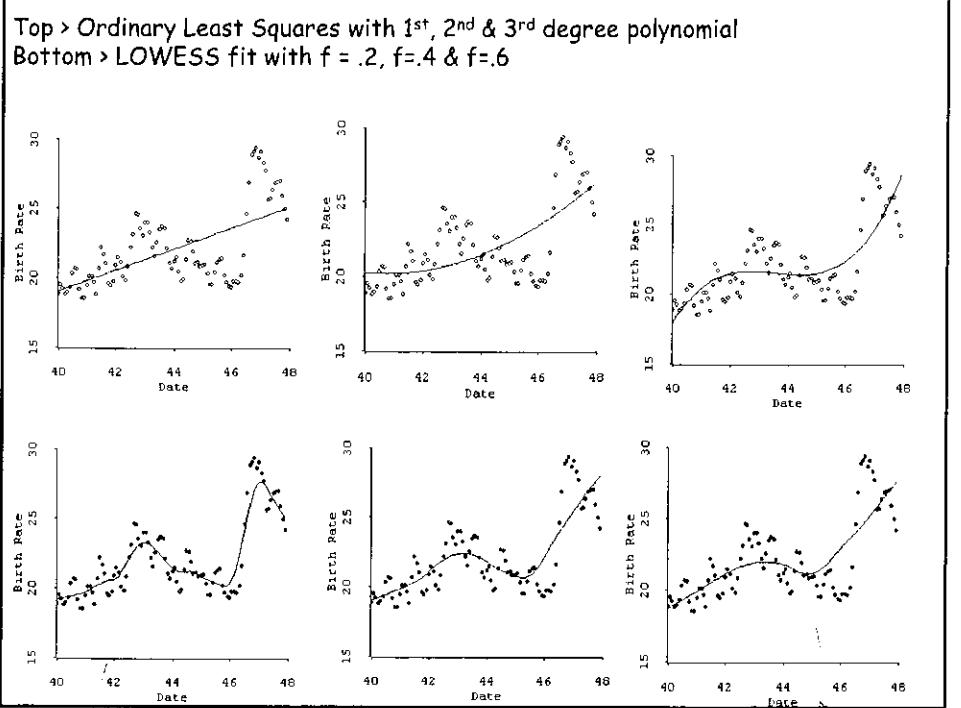
$$r_i = \left\{ 1 + \left[ \frac{|y_i - \hat{y}_i|}{6\hat{q}_{1/2}} \right]^2 \right\}^{-2}$$

5) REPEAT LOESS WITH WEIGHTS  $r_i w_i(x_0)$

6) REPEAT 2-5 UNTIL LOESS CURVE "CONVERGES"

## Example

- Birth rate in US population  
(U. S. Department of Health, Education and Welfare)
- $n=96$
- births per 1000 US population
- during 1940-47



less bias  
high VAR

less VAR  
high bias

# GALTON'S BEND

(Wachsmuth et al. 2003, Am. Stat.)

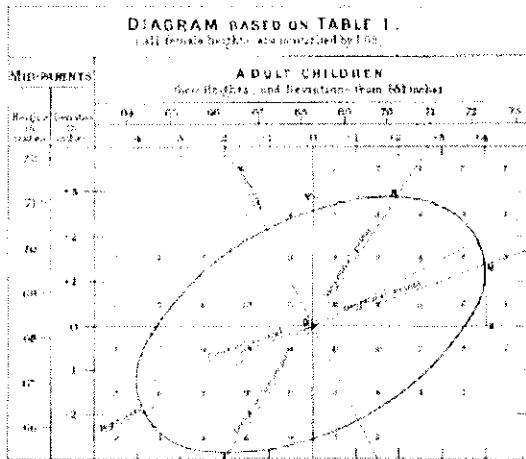
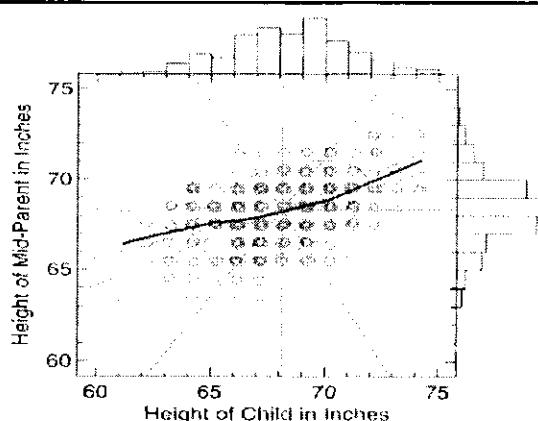


Figure 1. Galton's fitted regression model.



A possibility is that Galton ignored concealed heterogeneity

Figure 2. SYSTAT plot of Galton's Data with loess fit.

The dark curve in the center of the plot is a *loess* smoother (Cleveland and Devlin 1988). The smoother suggests that the relation between parent and child stature is not linear. There is a bend in the curve somewhere around the average height of approximately 68 inches for parents and children. A two-stage piecewise linear regression (Hinkley 1971) identifies a breakpoint at around 70 and finds it highly significant ( $p < .0001$ ).

Does the bend disappear by disaggregation of the sample?  
Analysis of data from Pearson and Lee (1903)

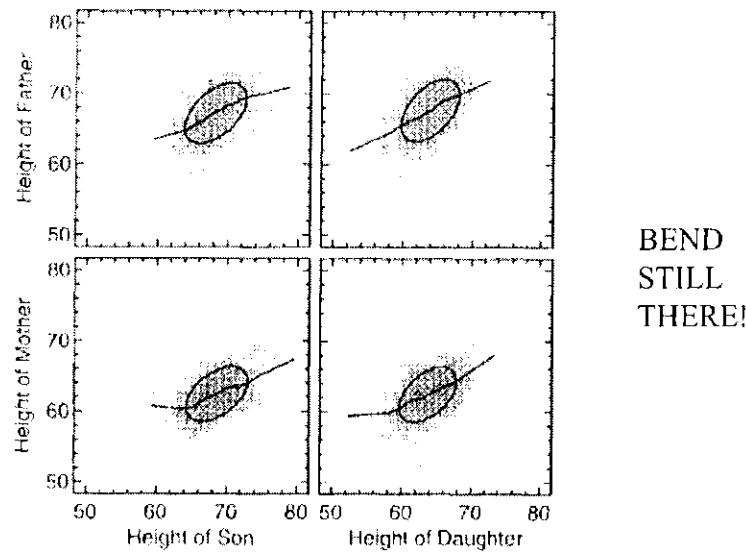


Figure 3. Pearson's data.

Wachsmuth et al. (2003) write:

In their search for universal hereditary laws, Galton and Pearson were driven by the linear model and the normal distribution because the associated parameters had scientific meaning for them that went beyond mere description.

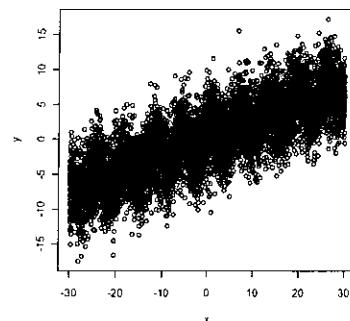
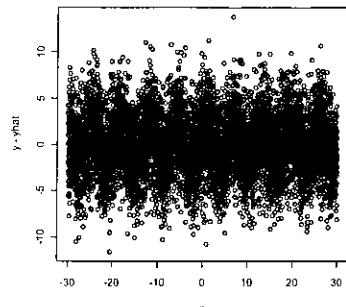
**RECALL: FINDING "STRUCTURE" WITH A LINEAR MODEL  
(some call this "learning architecture")**

We were given  $(x,y)$  data ( $n=10,000$ ). It looked like this and we run a linear regression

$$\hat{y} = 0.07936 + 0.24814 \cdot x$$

```
> cor(x,y)
[1] 0.8064256
```

```
> cor(y,yhat)
[1] 0.8064256
```



**RESIDUALS DISPLAY SINUSOIDAL BEHAVIOR**

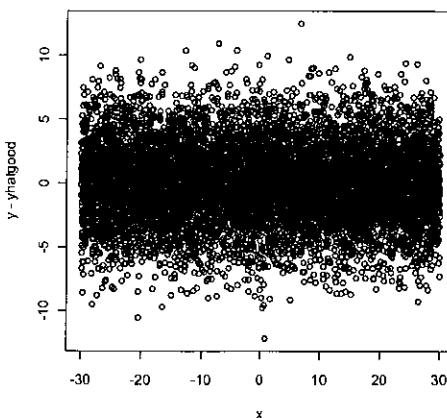
**TRUE MODEL**

```
> c<-rnorm(10000,0,sqrt(9))
> x<-runif(10000,-30,30)
> a<-0.10
> b<-0.25
> y<-a+b*x+sin(x)+cos(x)+c

> model<-lm(y~x+sin(x)+cos(x))

> Coefficients:
>(Intercept)      x    sin(x)    cos(x)
> 0.1030     0.2489   0.9518   0.9433
```

**RESIDUALS LOOK RANDOM**



WE GENERATED A NEW SAMPLE AT THE SAME VALUES OF X

```
> enew<-rnorm(10000,0,sqrt(9))
>ynew<-a+b*x+sin(x)+cos(x)+enew
```

CALCULATED PREDICTIVE MEAN SQUARED ERROR

```
>msepredbadmodel<-sum((ynew-yhat)**2/10000)
>msepredbadmodel
[1] 9.725709
```

```
>msepredgoodmodel<-sum((ynew-yhatgood)**2/10000)
>msepredgoodmodel
[1] 8.729272
```

CALCULATED PREDICTIVE CORRELATIONS

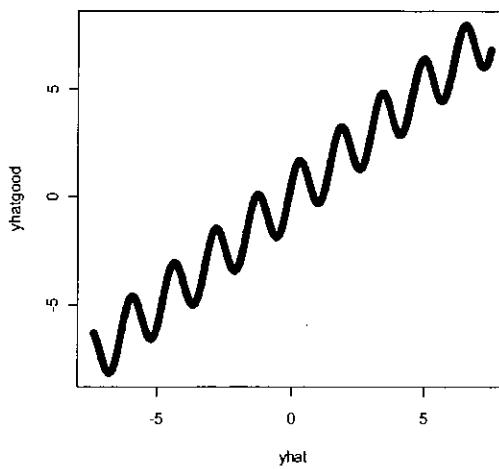
```
> cor(yhat,ynew)
[1] 0.8070097
> cor(yhatgood,ynew)
[1] 0.828854
```

MSE(Good)/MSE(Bad)=0.8975  
MSE(Bad)/MSE(Good)=1.1141

Cor(BAD)/Cor(GOOD)=0.9736

```
> lm(yhat~yhatgood)
Coefficients:
(Intercept)  yhatgood
0.005653    0.953468
```

FOUND: DO NOT TRUST CORRELATIONS!



## NEW TRAINING SAMPLE

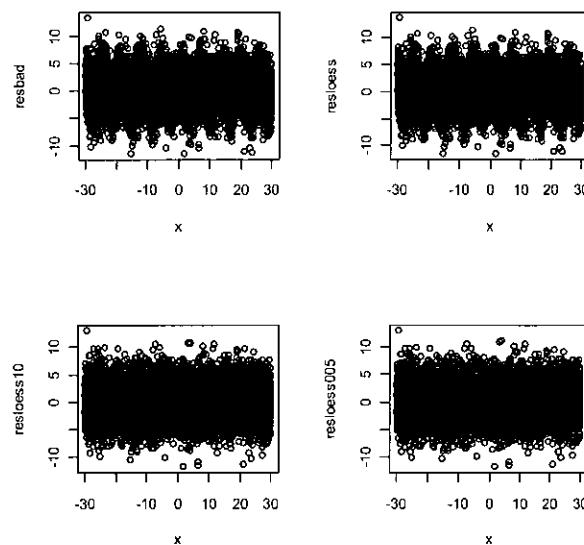


```

e<-rnorm(10000,0,sqrt(9))
x<-runif(10000,-30,30)
a<-0.10
b<-0.25
y<-a+b*x+sin(x)+cos(x)+e
#####TRAIN USING PARAMETRIC MODEL
modelgood<-lm(y~x+sin(x)+cos(x))
modelgood
yhatgood<-fitted(modelgood)
resgood<-y-yhatgood
modelbad<-lm(y~x)
yhatbad<-fitted(modelbad)
resbad<-y-yhatbad
#####TRAIN USING LOESS SPAN 0.50
yloess<-loess(y~x,span=0.50,degree=2)
yhatloess<-predict(yloess)
resloess<-y-yhatloess
#####TRAIN USING LOESS SPAN 0.10
yloess10<-loess(y~x,span=0.10,degree=2)
yhatloess10<-predict(yloess10)
resloess10<-y-yhatloess10
#####TRAIN USING LOESS SPAN 0.05
yloess005<-loess(y~x,span=0.05,degree=2)
yhatloess005<-predict(yloess005)
resloess005<-y-yhatloess005
par(mfrow=c(2,2))
plot(x,resbad)
plot(x,resloess)
plot(x,resloess10)
plot(x,resloess005)
par(mfrow=c(1,1))

```

TRAINING SAMPLE RESIDUALS: SINUSOIDAL BEHAVIOR LESS OBVIOUS  
IN BOTTOM PLOTS



```

##GENERATE A NEW SAMPLE AT THE SAME VALUES OF X
enew<-rnorm(10000,0,sqr(9))
ynew<-a+b*x+sin(x)+cos(x)+enew

#####TRAINING MEAN SQUARED ERROR
msebadmodel<-sum((ynew-yhatbad)**2/10000)

msegoodmodel<-sum((ynew-yhatgood)**2/10000)

mseloess<-sum((ynew-yhatloess)**2/10000)
mseloess10<-sum((ynew-yhatloess10)**2/10000)
mseloess005<-sum((ynew-yhatloess005)**2/10000)

###EVALUATION OF TRAINING PERFORMANCE
msebadmodel
msegoodmodel
mseloess
mseloess10
mseloess005

```

PROPERLY TUNED LOESS  
HAS ALMOST AS GOOD  
PERFORMANCE AS 'TRUE'  
MODEL



```

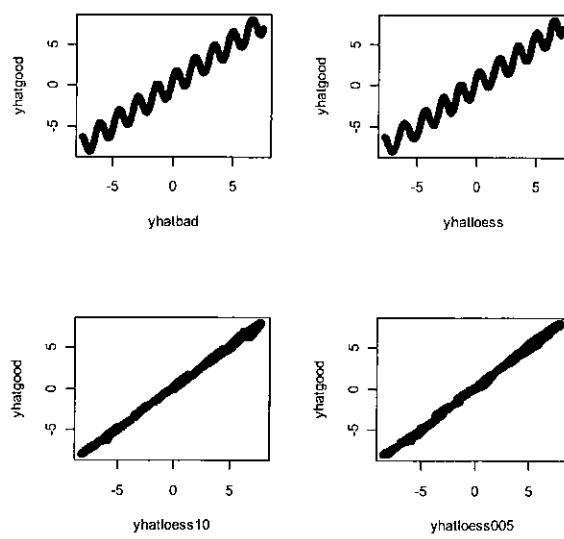
> msebadmodel
[1] 10.0463
>
> msegoodmodel
[1] 9.069427
>
> mseloess
[1] 10.03659
>
> mseloess10
[1] 9.109832
>
> mseloess005
[1] 9.111001
>

```

```

par(mfrow=c(2,2))
plot(yhatbad,yhatgood)
plot(yhalloess,yhatgood)
plot(yhatloess10,yhatgood)
plot(yhatloess005,yhatgood)
par(mfrow=c(1,1))

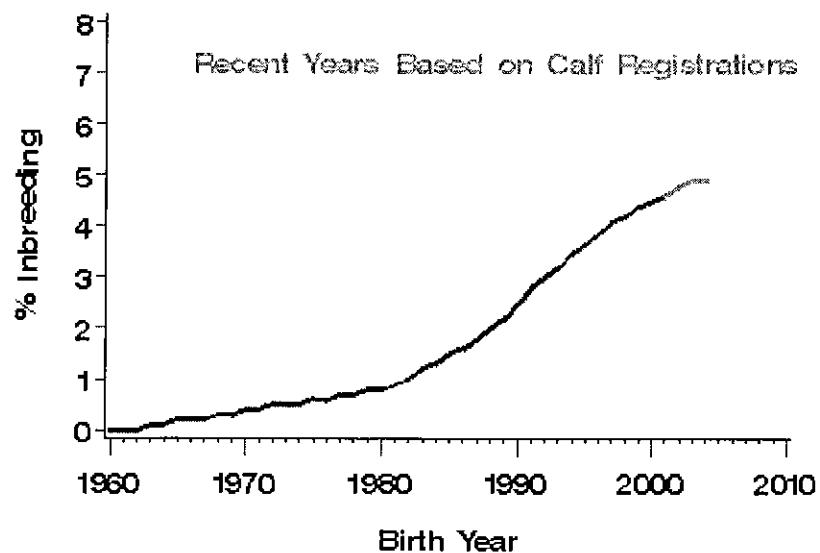
```



## INBREEDING DEPRESSION

- Examine relationships of yield (milk, protein, fat) and somatic cell score (SCS) with inbreeding coefficient ( $F$ ) using field data from US Jerseys
- Use REML, BLUP and "local regression" method (LOESS) for this purpose

### LEVEL OF INBREEDING IN HOLSTEINS, USA



- Relationship between mean value of a quantitative trait and inbreeding coefficient ( $F$ ) expected to be linear under dominance
- Not so if epistatic interactions between dominance effects exist

(Crow & Kimura, 1970)

## ONE-LOCUS MODEL

GENOTYPE ( $A$ )	$A_1A_1$	$A_1A_2$	$A_2A_2$
FREQUENCY	$p_1^2(1-F) + p_1F$	$2p_1p_2(1-F)$	$p_2^2(1-F) + p_2F$
PHENOTYPE	$\mu - A$	$\mu + D$	$\mu + A$

$$\begin{aligned}
 E(X) &= \mu + A(p_2 - p_1) + 2p_1p_2D - 2p_1p_2DF \\
 &= \alpha + \beta F \\
 &= \alpha - \beta(1 - F - 1) \\
 &= (\alpha + \beta) - \beta(\% \text{ Heterozygosity})
 \end{aligned}$$

ADDITIONAL MODEL WITH  $F$  (or  $H$ ) AS COVARIATE  $\rightarrow$  CONTRADICTORY

## TWO (UNLINKED) LOCI: NO EPISTASIS

Joint frequencies are product of marginal frequencies

GENOTYPE	$A_1A_1$	$A_1A_2$	$A_2A_2$
FREQUENCY	$p_1^2(1-F) + p_1F$	$2p_1p_2(1-F)$	$p_2^2(1-F) + p_2F$
$B_1B_1$	$r_1^2(1-F) + r_1F$	$\mu + A + B + I$	$\mu + D_A + B - L$
$B_1B_2$	$2r_1r_2(1-F)$	$\mu + A + D_B - K$	$\mu + D_A + D_B + J$
$B_2B_2$	$r_2^2(1-F) + r_2F$	$\mu + A + B - I$	$\mu + D_A + B + L$

$$E(X) = \mu + A(p_1 - p_2) + B(r_1 - r_2) \\ + 2p_1p_2D_A + 2r_1r_2D_B \\ - 2(p_1p_2D_A + r_1r_2D_B)F \\ = \alpha' + \beta'F$$

## TWO (UNLINKED) LOCI: EPISTASIS

GENOTYPE		$A_1A_1$	$A_1A_2$	$A_2A_2$
FREQUENCY		$p_1^2(1-F) + p_1F$	$2p_1p_2(1-F)$	$p_2^2(1-F) + p_2F$
$B_1B_1$	$r_1^2(1-F) + r_1F$	$\mu + A + B + I$	$\mu + D_A + B - L$	$\mu + A + B - I$
$B_1B_2$	$2r_1r_2(1-F)$	$\mu + A + D_B - K$	$\mu + D_A + D_B + J$	$\mu + A + D_B + K$
$B_2B_2$	$r_2^2(1-F) + r_2F$	$\mu + A + B - I$	$\mu + D_A + B + L$	$\mu + A + B + I$

\* ALL alleles AT A and B LOCI SAME SUBSCRIPT  $\rightarrow$  ADD I  
(ADDITIVE X ADDITIVE)

\* HOMOZYGOUS AT A HETEROZYGOUS AT B  $\rightarrow$  SUBTRACT AND ADD K  
HOMOZYGOUS AT B HETEROZYGOUS AT A  $\rightarrow$  SUBTRACT AND ADD I.  
(ADDITIVE X DOMINANCE)

\* HETEROZYGOUS AT A AND B  $\rightarrow$  ADD J  
(DOMINANCE X DOMINANCE)

I,J,K,L: parameters (4 d. freedom)

## Mean value under dominance x dominance epistasis

$$\begin{aligned}
 E(\lambda) = & \mu + \delta(p_2 - p_1) + \beta(r_2 - r_1) - 2p_1p_2D_A + 2r_1r_2D_R \\
 & + l(p_1 - p_2)(r_1 - r_2) + 2l p_1 p_2 (r_1 - r_2) + 2K r_1 r_2 (p_1 - p_2) \\
 & - 4l p_1 p_2 r_1 r_2 \\
 & - 2[p_1 p_2 D_A + r_1 r_2 D_R + l p_1 p_2 (r_1 - r_2) + K r_1 r_2 (p_1 - p_2) + 4l p_1 p_2 r_1 r_2] F^2 \\
 & + (4l p_1 p_2 r_1 r_2) F^3 \\
 = & \alpha^k + \beta^k F + \gamma^k F^3
 \end{aligned}$$

- \* Dominance, additive x dominance, and dominance x dominance intervene in **linear** regression
- \* Epistasis without dominance does not enter into mean-F relationship
- \* Dominance x dominance intervenes in second-order regression

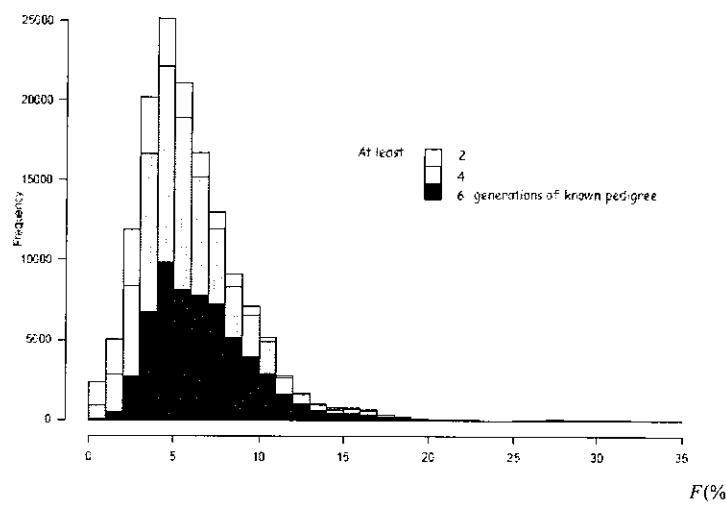
## DATA

- First lactation records (herds) on 59,778 (1,142) Jersey cows
- 6 generations of known pedigree
- First calving between 1995 and 2000

## Distribution of F

- F calculated from all known pedigree information
- F ranged between 0 and 34%
- Median F = 6.25%

Histogram of F values



## Procedures

- Fit linear models without F as covariate
- Compute EBLUP residuals from these models
- Fit nonparametric regression to EBLUP residuals in order to obtain nonparametric lines describing relationship between performance and inbreeding level

## Linear Models

Model

$$y_{ijk} = HYS_i + AGE_j + \beta_1(D_{ijk} - \bar{D}) + a_k + e_{ijk}$$

$y_{ijk}$  = somatic cell score (SCS), milk, protein, or fat yield;

$HYS_i$  = fixed effect of herd-year-season ( $i = 1, 2, \dots, 12276$  for DS2; 11158 for DS4 or 6406 for DS6, with seasons classes January-April, May-August, September-December);

$AGE_j$  = fixed effect of age at calving class;  $j = 1, 2, \dots, 6$

(< 617, 617-716, 717-816, 817-916, 917-1016, or >1016 days of age);

$\beta_1$  = fixed regression coefficient of performance on days in milk;

$D_{ijk}$  = days in milk for animal  $k$  in herd-year-season  $i$  and age of calving class  $j$ ;

$D = 263$ ;

$a_k$  = random additive genetic effect of animal  $k$ , and

$e_{ijk}$  = random residual.

## Linear Model Assumptions

- Genetic and residual effects assumed mutually independent, with  $e \sim N(0, I\sigma^2_e)$  and  $a \sim N(0, A\sigma^2_a)$ , where  $A$  is the additive relationship matrix ( $1 + F_k$  in the  $k^{\text{th}}$  diagonal position,  $F_k$  is the inbreeding coefficient of animal  $k$ )

## Nonparametric regression

- Fit LOESS regression to BLUP residuals with  $F$  as covariate
- Vary spanning parameter & degree of local polynomial
- Plot fitted values of residuals against  $F$

## LOESS

(Fitting done by locally weighted least squares)

- $\tilde{\varepsilon}_{ij}$  is LOESS fit using only residuals in the neighborhood of  $F_i$ ,  $i=1,2,\dots,n$   
( $i=1,2,\dots,n$  animals;  $j=1,\dots,4$  traits)
- Size of neighborhood determined by  $f = \frac{q}{n}$   
 $q$  = number of points in neighborhood  
 $n$  = total number of points

## "Robust" LOESS

Weights assigned to  $\hat{\varepsilon}_{ijk}$ :

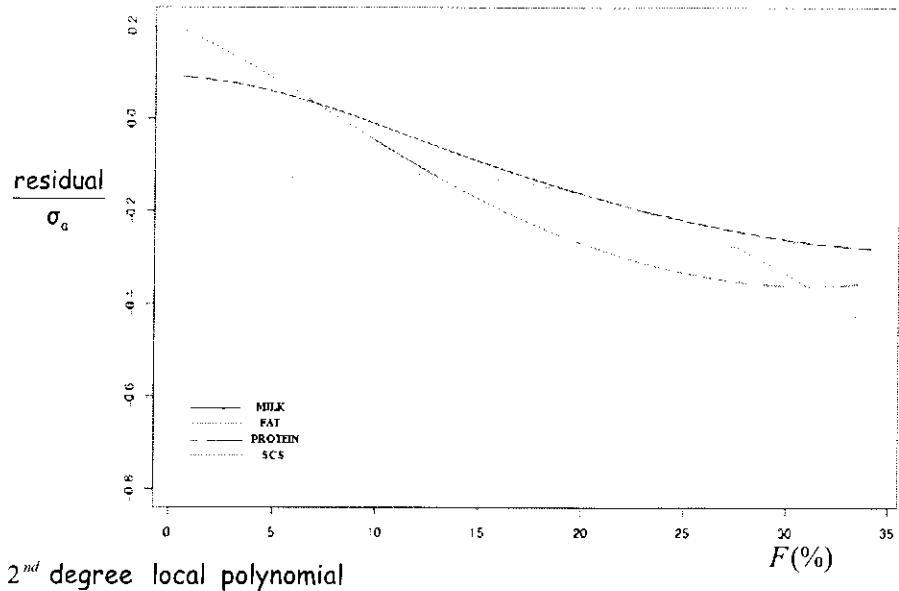
$$\Rightarrow w_{ijk}^{[t+1]} = w_{ijk}^{[t]} \cdot \delta_{ijk}^{[t]} \quad t=1,2,3,4$$

$$\text{I)} \quad w_{ijk}^{[1]} = [1 - (\frac{F_k - F_i}{\max(F_l - F_i)})^3]^3 \quad l=1,2,\dots,q$$

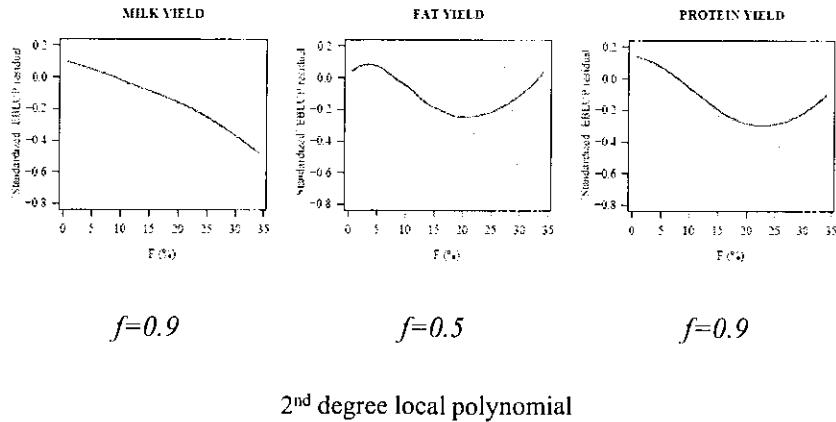
$$\text{II)} \quad \delta_{ijk}^{[t]} = [1 - (\frac{\tilde{\varepsilon}_{ijk} - \hat{\varepsilon}_{ijk}}{6 \cdot \text{med}})^2]^2$$

med = median of all  $(\tilde{\varepsilon}_{ijk} - \hat{\varepsilon}_{ijk})$

Cows with at least 6 generations of known pedigree  $f=1$



“Robust” original (black) with bootstrap (light blue) LOESS curves of yields for US Jerscys with at least 6 generations of known pedigree,  
based on medians of EBLUP residuals (y-axis =  $\hat{e}_{ijk} / \hat{\sigma}_a$ )



## Conclusions

- LOESS analysis suggested local relationships.
- Effects of inbreeding seem nil, until for F values up to ~7%
- Effects of inbreeding not accounted well by additive models
- Results may be confounded by effects of selection that are unaccounted for

## Kernel Regression

$$y_i = g(\mathbf{x}_i) + e_i; i = 1, 2, \dots, n$$

where:

- $y_i$  is the measurement taken on individual  $i$
- $\mathbf{x}_i$  is a  $p \times 1$  vector of observed SNP genotypes
- $g(\cdot)$  is some unknown function relating genotypes to phenotypes.
- Set  $g(\mathbf{x}_i) = E(y_i | \mathbf{x}_i)$  = conditional expectation function
- $e_i \sim (0, \sigma^2)$  is a random residual



Conditional expectation function

$$\begin{aligned} g(\mathbf{x}) &= \int y \frac{p(\mathbf{x}, y)}{p(\mathbf{x})} dy \\ &= \frac{\int y p(\mathbf{x}, y) dy}{p(\mathbf{x})}. \end{aligned}$$



Non-parametric estimator of density of  $\mathbf{x}$

$$\hat{p}(\mathbf{x}) = \frac{1}{nh^p} \sum_{i=1}^n K\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right)$$

"Focal point"

"Kernel", possibly a probability density function with some bandwidth parameter  $h$

We would like:

$$\int_{-\infty}^{\infty} \hat{p}(\mathbf{x}) d\mathbf{x} = \frac{1}{nh^p} \sum_{i=1}^n \int_{-\infty}^{\infty} K\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right) d\mathbf{x} = 1$$

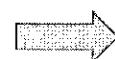
Implying  $\rightarrow$

$$\int_{-\infty}^{\infty} \frac{1}{h^p} K\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right) d\mathbf{x} = 1$$



Similarly, can form non-parametric estimator of joint density

$$\hat{p}(\mathbf{x}, y) = \frac{1}{nh^{p+1}} \sum_{i=1}^n K\left(\frac{y_i - y}{h}\right) K\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right)$$



Recall

$$g(\mathbf{x}) = \int_{\mathcal{Y}} \frac{p(\mathbf{x}, y)}{p(\mathbf{x})} dy$$

$$= \frac{\int_{\mathcal{Y}} p(\mathbf{x}, y) dy}{p(\mathbf{x})}$$

ESTIMATE NUMERATOR      ESTIMATE DENOMINATOR

Estimate numerator

$$\begin{aligned}\int y \hat{p}(\mathbf{x}, y) dy &= \int y \frac{1}{nh^{p+1}} \sum_{i=1}^n K\left(\frac{y_i - y}{h}\right) K\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right) dy \\ &= \frac{1}{nh^p} \sum_{i=1}^n \left[ \frac{1}{h} \int y K\left(\frac{y_i - y}{h}\right) dy \right] K\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right).\end{aligned}$$

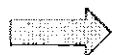
Let  $z = \frac{y-y_i}{h}$ , so that  $dy = h dz$  and

$$\begin{aligned}\frac{1}{h} \int y K\left(\frac{y_i - y}{h}\right) dy &= \frac{1}{h} \int (y_i + hz) K(z) h dz \\ &= \int (y_i + hz) K(z) dz \\ &= \int y_i K(z) dz + h \int z K(z) dz \\ &= y_i \int K(z) dz + h E(z).\end{aligned}$$

$K(\cdot)$  can be constructed such that:

$$\int K(z) dz = 1 \text{ and } E(z) = \int z K(z) dz = 0$$

$$\text{Then: } \frac{1}{h} \int y K\left(\frac{y_i - y}{h}\right) dy = y_i$$



Estimator of numerator is

$$\int y \hat{p}(\mathbf{x}, y) dy = \frac{1}{nh^p} \sum_{i=1}^n y_i K\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right)$$



Forming non-parametric estimator of conditional expectation

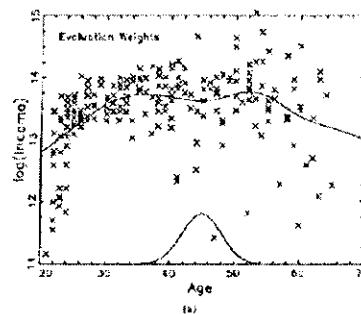
$$\hat{E}(Y \mid \mathbf{x}) = \hat{g}(\mathbf{x}) = \frac{\int y \hat{p}(\mathbf{x}, y) dy}{\hat{p}(\mathbf{x})}$$

$$\begin{aligned}\hat{E}(Y \mid \mathbf{x}) &= \hat{g}(\mathbf{x}) = \frac{\frac{1}{nh^p} \sum_{i=1}^n y_i K\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right)}{\frac{1}{nh^p} \sum_{i=1}^n K\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right)} \\ &= \frac{\sum_{i=1}^n y_i K\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right)}{\sum_{i=1}^n K\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right)} = \sum_{i=1}^n w_i(\mathbf{x}) y_i\end{aligned}$$

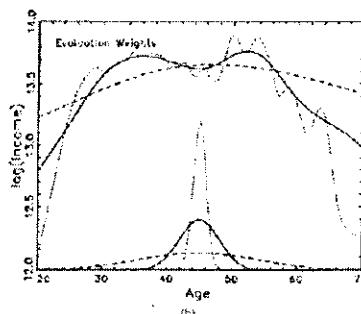
Nadaraya-Watson estimator  
(weighted average)

$w_i(\mathbf{x}) = \frac{K\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right)}{\sum_{i=1}^n K\left(\frac{\mathbf{x}_i - \mathbf{x}}{h}\right)}$

Relationship between Income and age  
(Chu and Marron, 1991)



$h=9$  local features  
Disappear (dashes)



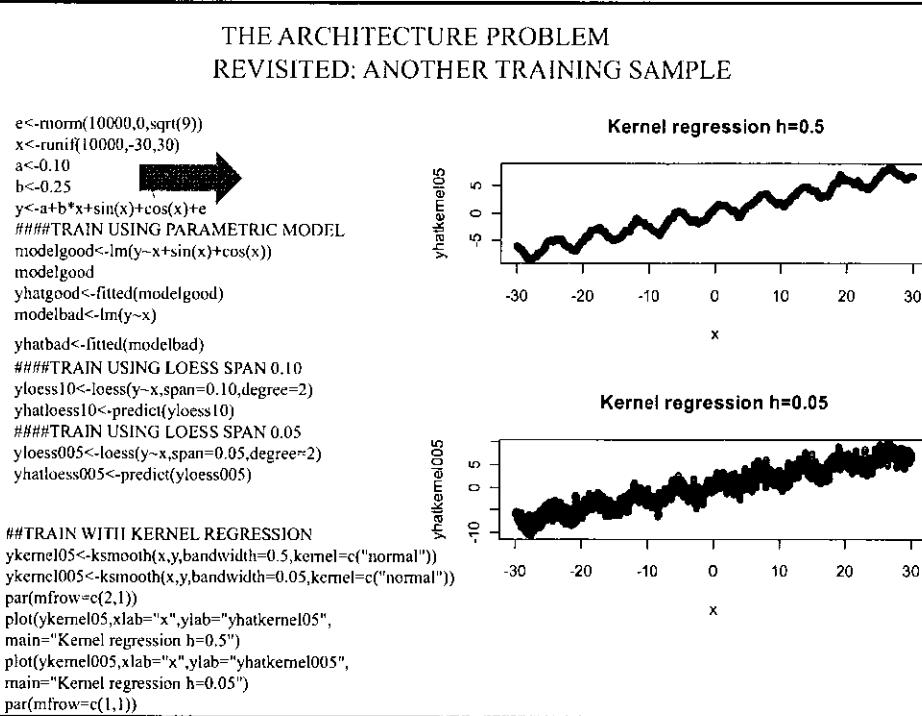
$h=1$  lots of variation  
(dots)

FIG. 1. Scatter plot and smooths for earning power data. Kernel is  $N(0, 1)$ . Median values are represented by curves at the bottom: solid curve  $h = 9$ , dashed curve  $h = 3$ , dotted curve  $h = 1$ .

Bandwidth can be gauged by, e.g., cross-validation

$$CV(h) = \frac{\sum_{i=1}^n [y_i - \hat{g}_{i,-i}(x_i|h)]^2}{n}$$

- Create a grid of  $h$  values
- For each value compute the CV mean squared error  
(above is leave-one-out, but this may not be best)
- Use the  $h$  value which minimizes  $CV(\cdot)$



## PENALIZED METHODS for functional inference

- The idea of “penalty” is ad-hoc
- It does not arise “naturally” in classical inference
- It appears very naturally in Bayesian inference
  - ➔  $L_2$  penalty: equivalent to Gaussian prior
  - ➔  $L_1$  penalty: equivalent to double exponential prior

### The concept of penalized likelihood (example in the mixed linear model)

$$\begin{aligned} \mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \mathbf{e} \\ \mathbf{y} | \boldsymbol{\beta}, \mathbf{u}, \mathbf{R} &\sim N(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u}, \mathbf{R}) \\ \mathbf{u} &\sim N(\mathbf{0}, \mathbf{G}) \end{aligned}$$

$$\begin{aligned} p(\mathbf{y} | \boldsymbol{\beta}, \mathbf{u}, \mathbf{R}) &= \frac{1}{(2\pi)^{\frac{N}{2}} |\mathbf{R}|^{\frac{1}{2}}} \exp \left[ -\frac{1}{2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{u})' \mathbf{R}^{-1} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\mathbf{u}) \right] \\ p(\mathbf{u} | \mathbf{G}) &= \frac{1}{(2\pi)^{\frac{q}{2}} |\mathbf{G}|^{\frac{1}{2}}} \exp \left[ -\frac{1}{2} \mathbf{u}' \mathbf{G}^{-1} \mathbf{u} \right] \end{aligned}$$

Assuming known variance components, the log of the joint density of the data and random effects is termed "penalized likelihood"

$$\rightarrow l(\beta, u | y, R, G) = K - \frac{1}{2} (y - X\beta - Zu)' R^{-1} (y - X\beta - Zu) - \frac{1}{2} u' G^{-1} u$$

$$-2l(\beta, u | y, R, G) = K + (y - X\beta - Zu)' (y - X\beta - Zu) + u' G^{-1} u \quad \text{Penalized SS}$$

$$\rightarrow \frac{\partial l(\beta, u | y, R, G)}{\partial \beta} = X' R^{-1} (y - X\beta - Zu)$$

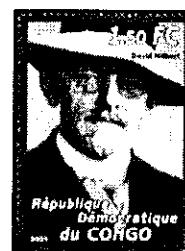
$$\frac{\partial l(\beta, u | y, R, G)}{\partial u} = Z' R^{-1} (y - X\beta - Zu) - G^{-1} u$$

Setting the derivatives to 0 yields

$$\begin{bmatrix} X' R^{-1} X & X' R^{-1} Z \\ Z' R^{-1} X & Z' R^{-1} Z + G^{-1} \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} X' R^{-1} y \\ Z' R^{-1} y \end{bmatrix}$$

- The solution to these equations produces the "maximum penalized likelihood" estimates of  $\beta$  and  $u$
- These solutions are also the  $\text{BLUE}(\beta)$  and  $\text{BLUP}(u)$

## 8. Reproducing Kernel Hilbert spaces mixed model



Function of molecular information  $x$  (vector of SNP variables)

$$SS[g(x), \lambda] = \sum_{i=1}^n [y_i - w_i' \beta - z_i' u - g(x_i)]^2 + \lambda \|g(x)\|_H^2$$

"Penalized sum of squares"

Smoothing parameter ( $\lambda$ )

Some norm under  
Hilbert space ( $H$ ) of  
functions

Variational problem: find  $g(x)$  over entire space of functions minimizing  $SS(\cdot)$

**Solution to variational problem: linear function**

$$g(\cdot) = \alpha_0 + \sum_{j=1}^n \alpha_j K(\cdot, \mathbf{x}_j)$$

Regression coefficient                          Reproducing kernel

No. individuals with molecular data  
reduction of dimension  
 $p \# \text{SNPs} \gg n \# \text{indiv}$

Example of reproducing kernel:

$$K_h(\mathbf{x}, \mathbf{x}_j) = \exp\left[-\frac{(\mathbf{x}-\mathbf{x}_j)'(\mathbf{x}-\mathbf{x}_j)}{h}\right]$$

→ Definition of positive-definite kernel (the theory deals with "reproducing kernels" function

$$\int k(\mathbf{x}, \mathbf{t}) g(\mathbf{x}) g(\mathbf{t}) p(\mathbf{x}, \mathbf{t}) d\mathbf{x} dt > 0$$

→ Positive-definite kernel matrix; symmetric, with  $k(i,j,h)=k(j,i,h)$

$$\mathbf{K}_h = \begin{bmatrix} k(1,1,\mathbf{h}) & k(1,2,\mathbf{h}) & \dots & \dots & k(1,n,\mathbf{h}) \\ k(2,1,\mathbf{h}) & k(2,2,\mathbf{h}) & \dots & \dots & k(2,n,\mathbf{h}) \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ k(n,1,\mathbf{h}) & k(n,2,\mathbf{h}) & \dots & \dots & k(n,n,\mathbf{h}) \end{bmatrix}$$

$\mathbf{h}$  = scalar or vector of bandwidth parameters

# MEASURES OF DISTANCE THAT CAN BE USED IN KERNELS

Euclidean

$$d(x, y) = \|x - y\| = \sqrt{\sum_{k=1}^p (x_k - y_k)^2}$$

Manhattan

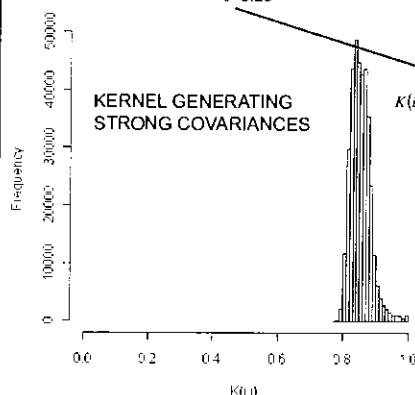
$$d(x, y) = \sum_{k=1}^p |x_k - y_k|,$$

Bray-Curtis

$$d_{ij} = (\sum_k |x_{ik} - x_{jk}|) / (\sum_k x_{ik} + x_{jk})$$

$\theta=0.25$

KERNEL GENERATING STRONG COVARIANCES



$$K(i, j) = \exp\{-\theta k^{-1} d_{ij}\}$$

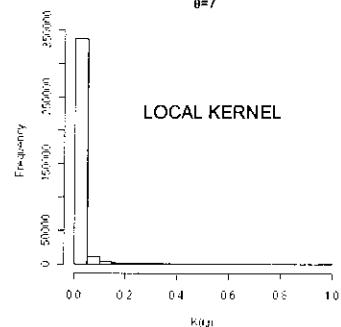
$$\mathbf{x}_i = (x_{i1}, \dots, x_{ip})'$$

$$d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$$

$$k = \max_{i,j} \{\|\mathbf{x}_i - \mathbf{x}_j\|^2\}$$

$\theta=7$

LOCAL KERNEL



Histogram of evaluations of Gaussian kernel by value of bandwidth parameter

Multivariate-t kernel (Gianola, 2012) for an  $S \times 1$  vector

$$k_{v,\Sigma}(x_i, x_j) = \left[ 1 + \frac{(x_i - x_j)' \sum^{-1} (x_i - x_j)}{v} \right]^{-\left(\frac{S+v}{2}\right)}$$

$$\Sigma^{-1} = \text{Diag}(2p_k q_k),$$

$$\Sigma = \text{Diag}(2p_k q_k)$$

$\Sigma^{-1} = R$  where  $R$  is a matrix containing  $r^2$  from LD

$$\Sigma = R$$

Let  $S = 5$   $d = \sum_{k=1}^5 (x_{ik} - x_{jk})^2$

For Gaussian kernel suppose  $\theta = \frac{1}{h} = 1, 2, 3$

For  $t$ -kernel  $v = 2, 4, 8$ .

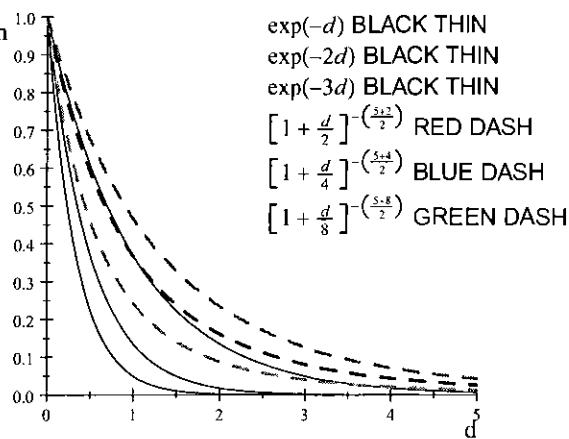
We will have



Gaussian =  $\exp(-\theta d)$

$$t = \left[ 1 + \frac{d}{v} \right]^{-\left(\frac{S+v}{2}\right)}$$

Kernel evaluation



### Mixed model representation (enhancing pedigrees...)

$$y_i = \mathbf{w}'_i \beta + \mathbf{z}'_i \mathbf{u} + \sum_{j=1}^n \exp\left[-\frac{(\mathbf{x}_i - \mathbf{x}_j)'(\mathbf{x}_i - \mathbf{x}_j)}{h}\right] \alpha_j + e_i$$

Define row vector

$$\mathbf{t}'_i(h) = \left\{ \exp\left[-\frac{(\mathbf{x}_i - \mathbf{x}_j)'(\mathbf{x}_i - \mathbf{x}_j)}{h}\right] \right\}$$

$$\mathbf{T}(h) = \begin{bmatrix} \mathbf{t}'_1(h) \\ \mathbf{t}'_2(h) \\ \vdots \\ \mathbf{t}'_n(h) \end{bmatrix}$$

$$\mathbf{t}'_i(h) = \mathbf{K}'_i(h)$$

$$\mathbf{T}(h) = \mathbf{K}(h)$$

Then:

$$\mathbf{y} = \mathbf{W}\beta + \mathbf{Z}\mathbf{u} + \mathbf{T}(h)\alpha + \mathbf{e}$$

Bandwidth parameter

Do:

$$\alpha \sim \mathbf{N}(\mathbf{0}, \mathbf{T}^{-1}(h)\sigma_\alpha^2)$$

$$\sigma_\alpha^2 = \frac{1}{\lambda}$$

Smoothing parameter

$$\begin{bmatrix} \mathbf{W}'\mathbf{W} & \mathbf{W}'\mathbf{Z} & \mathbf{W}'\mathbf{T}(h) \\ \mathbf{Z}'\mathbf{W} & \mathbf{Z}'\mathbf{Z} + \mathbf{A}^{-1} \frac{\sigma_e^2}{\sigma_u^2} & \mathbf{Z}'\mathbf{T}(h) \\ \mathbf{T}'(h)\mathbf{W} & \mathbf{T}'(h)\mathbf{Z} & \mathbf{T}'(h)\mathbf{T}(h) + \mathbf{T}(h) \frac{\sigma_e^2}{\sigma_\alpha^2} \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \hat{\mathbf{u}} \\ \hat{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{W}'\mathbf{y} \\ \mathbf{Z}'\mathbf{y} \\ \mathbf{T}'(h)\mathbf{y} \end{bmatrix}$$

$h$  assumed known here

If you apply  $A$  matrix as kernel matrix solution would be

THE "ANIMAL MODEL" IS A PARTICULAR CASE OF RKHS

$$\begin{aligned} y &= A\alpha + e \\ \alpha &\sim N(0, A^{-1}\sigma_a^2) \\ e &\sim N(0, I\sigma_e^2) \\ \Rightarrow u &= A\alpha \sim N(0, A\sigma_a^2) \end{aligned}$$

$$\left( A'A + A \frac{\sigma_e^2}{\sigma_a^2} \right) \hat{\alpha} = A'y$$

$$A \left( A + I \frac{\sigma_e^2}{\sigma_a^2} \right) \hat{\alpha} = Ay$$

$$\hat{\alpha} = \left( A + I \frac{\sigma_e^2}{\sigma_a^2} \right)^{-1} y$$

Predicted Genetic signal  $\rightarrow A\hat{\alpha} = \left( I + A^{-1} \frac{\sigma_e^2}{\sigma_a^2} \right)^{-1} y = \text{BLUP(additive effects)}$

BLUP

Penalized estimation

$$\hat{\alpha} = \arg \min_{\alpha} \left\{ (y - K\alpha)'(y - K\alpha) + \lambda \alpha' K \alpha \right\}$$

Bayesian View

$$\begin{cases} y = K\alpha + \epsilon \\ p(\epsilon, \alpha) = N(\epsilon | 0, I\sigma_e^2)N(\alpha | 0, K^{-1}\sigma_\alpha^2) \end{cases}$$

[1] Kimeldorf, G.S. & Wahba, G. (1970).

Genomic BLUP is a special case of RKH where  
 $\mathbf{X}\mathbf{X}'$  is the Kernel matrix.

#### GENOMIC BLUP IS A PARTICULAR CASE OF RKHS

$$y = \mathbf{X}\mathbf{X}'\boldsymbol{\alpha} + e$$

$$\boldsymbol{\alpha} \sim N(0, (\mathbf{X}\mathbf{X}')^{-1} \sigma_\beta^2)$$

$$e \sim N(0, I\sigma_e^2)$$

$$\Rightarrow u = \mathbf{X}\mathbf{X}'\boldsymbol{\alpha} \sim N(0, \mathbf{X}\mathbf{X}'\sigma_\beta^2)$$

$$\left( \mathbf{X}\mathbf{X}'\mathbf{X}\mathbf{X}' + \mathbf{X}\mathbf{X}' \frac{\sigma_e^2}{\sigma_\beta^2} \right) \hat{\boldsymbol{\alpha}} = \mathbf{X}\mathbf{X}'y$$

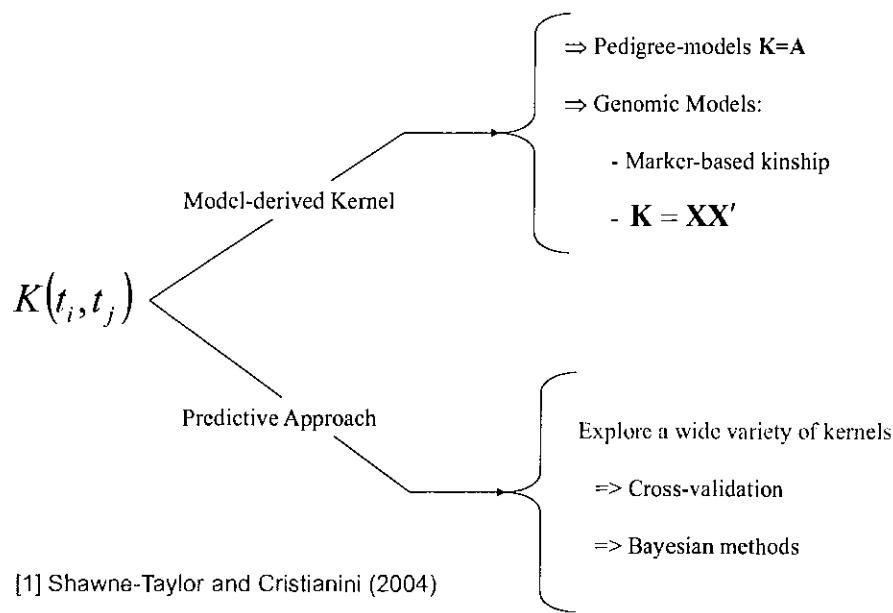
$$(\mathbf{X}\mathbf{X}') \left( \mathbf{X}\mathbf{X}' + I \frac{\sigma_e^2}{\sigma_\beta^2} \right) \hat{\boldsymbol{\alpha}} = \mathbf{X}\mathbf{X}'y$$

$$\hat{\boldsymbol{\alpha}} = \left( \mathbf{X}\mathbf{X}' + I \frac{\sigma_e^2}{\sigma_\beta^2} \right)^{-1} y$$

Predicted Genetic signal  $\rightarrow \mathbf{X}\mathbf{X}'\hat{\boldsymbol{\alpha}} = \mathbf{X}\mathbf{X}' \left( \mathbf{X}\mathbf{X}' + I \frac{\sigma_e^2}{\sigma_\beta^2} \right)^{-1} y$

$$\hat{u} = \left( I + (\mathbf{X}\mathbf{X}')^{-1} \frac{\sigma_e^2}{\sigma_\beta^2} \right)^{-1} y = \text{"GENOMIC BLUP"}$$

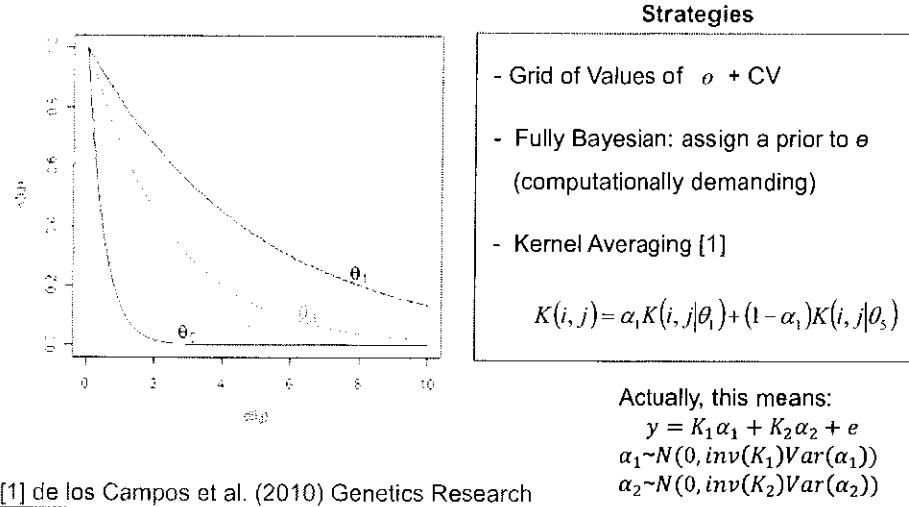
#### How to Choose the Reproducing Kernel? [1]



## Choosing the RK based on predictive ability

$$d(\mathbf{x}_i, \mathbf{x}_j) : \quad \Rightarrow \quad K(i, j | \theta) = \text{Exp} \left\{ -\theta \times d(\mathbf{x}_i, \mathbf{x}_j) \right\}$$

(genetic) distance between individuals



## Example 1 of RKHS

$$\begin{bmatrix} y_2 = 5 \\ y_3 = 3 \\ y_4 = 7 \\ y_5 = 8 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 1 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \left( \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{bmatrix} \right) + \begin{bmatrix} e_2 \\ e_3 \\ e_4 \\ e_5 \end{bmatrix}$$

=  $\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}(\mathbf{a} + \mathbf{d}) + \mathbf{e}$ .

Additive      Dominance

Henderson (1985) assumed  $\sigma_a^2 = 5, \sigma_d^2 = 4$  and  $\sigma_e^2 = 20$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{4} & \frac{1}{4} & 1 \end{bmatrix} \text{ and } \mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Application of BLUP paradigm leads to

$$\hat{\beta}' = \begin{bmatrix} 5.145 & 0.241 \end{bmatrix},$$

$$\hat{\alpha}' = \begin{bmatrix} 0.045 & -0.192 & -0.343 & 0.096 & 0.242 \end{bmatrix},$$

$$\hat{\mathbf{d}}' = \begin{bmatrix} 0 & -0.073 & -0.365 & 0.162 & 0.234 \end{bmatrix}.$$

$$\hat{\mathbf{g}} = \hat{\alpha} + \hat{\mathbf{d}} = \begin{bmatrix} 0.045 & -0.265 & -0.708 & 0.259 & 0.477 \end{bmatrix}$$

Next, do RKHS with  $\mathbf{K}=\mathbf{A}+\mathbf{D}$  as positive-definite kernel matrix

$$\mathbf{K} = \mathbf{A} + \mathbf{D} = \begin{bmatrix} 2 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & 2 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 2 & \frac{3}{4} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & \frac{3}{4} & 2 & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{4} & \frac{1}{4} & 2 \end{bmatrix}$$

$$\begin{bmatrix} y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 1 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} 2 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 2 & \frac{3}{4} & \frac{1}{4} \\ \frac{1}{2} & \frac{3}{4} & 2 & \frac{1}{4} \\ 0 & \frac{1}{4} & \frac{1}{4} & 2 \end{bmatrix} \begin{bmatrix} \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{bmatrix} + \begin{bmatrix} e_2 \\ e_3 \\ e_4 \\ e_5 \end{bmatrix}$$

$$= \mathbf{X}\hat{\beta} + \mathbf{K}\hat{\alpha} + \mathbf{e}.$$

$$\sigma_a^2 = \sigma_d^2 + \sigma_e^2 = 9 \rightarrow \text{This is } 1/\lambda$$

$$\left[ \hat{\beta}_0 = 5.289 \quad \hat{\beta}_1 = 0.200 \quad \hat{\alpha}_2 = -0.128 \quad \hat{\alpha}_3 = -0.781 \quad \hat{\alpha}_4 = 0.487 \quad \hat{\alpha}_5 = 0.422 \right]$$

$$\begin{bmatrix} \hat{g}_{K,1} \\ \hat{g}_{K,2} \\ \hat{g}_{K,3} \\ \hat{g}_{K,4} \\ \hat{g}_{K,5} \end{bmatrix} = \begin{bmatrix} 0.036 \\ -0.210 \\ -0.569 \\ 0.206 \\ 0.382 \end{bmatrix}$$

COMPARED WITH

$$\hat{\mathbf{g}} = \hat{\mathbf{a}} + \hat{\mathbf{d}} = \begin{bmatrix} 0.045 & -0.265 & -0.708 & 0.259 & 0.477 \end{bmatrix}$$

PREDICTING FUTURE RECORDS UNDER THE SAME ENVIRONMENTAL CONDITIONS; PARAMETRICALLY

$$\begin{bmatrix} y_1^f \\ y_2^f \\ y_3^f \\ y_4^f \\ y_5^f \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 3 \\ 1 & 1 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{bmatrix} + \begin{bmatrix} e_1^f \\ e_2^f \\ e_3^f \\ e_4^f \\ e_5^f \end{bmatrix}$$

$$= \mathbf{M}_P \boldsymbol{\theta}_P + \mathbf{e}^f,$$

PREDICTION OF FUTURE RECORDS NON-PARAMETRICALLY

$$\begin{bmatrix} y_1^f \\ y_2^f \\ y_3^f \\ y_4^f \\ y_5^f \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 3 \\ 1 & 1 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 2 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 2 & \frac{3}{4} & \frac{1}{4} \\ \frac{1}{2} & \frac{3}{4} & 2 & \frac{1}{4} \\ 0 & \frac{1}{4} & \frac{1}{4} & 2 \end{bmatrix} \begin{bmatrix} \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{bmatrix} + \begin{bmatrix} e_1^f \\ e_2^f \\ e_3^f \\ e_4^f \\ e_5^f \end{bmatrix}$$

$$= \mathbf{M}_K \boldsymbol{\theta}_K + \mathbf{e}^f.$$

FOR BOTH APPROACHES THE PREDICTIVE DISTRIBUTION IS

$$\begin{bmatrix} y_1^f \\ y_2^f \\ y_3^f \\ y_4^f \\ y_5^f \end{bmatrix} \sim \begin{bmatrix} y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}, \text{ dispersion (smoothing) parameters}$$

*All predictive future*

$$\sim (\mathbf{M} \hat{\boldsymbol{\theta}}, (\mathbf{M} \mathbf{C}^{-1} \mathbf{M}' + \mathbf{I}_f) \sigma_e^2),$$

*S.E.V for random  
b.v for fit*

For the two procedures the mean and SD of the predictive distributions are:

$$P = \begin{bmatrix} 5.674 \pm 6.020 \\ 5.364 \pm 5.460 \\ 5.162 \pm 5.353 \\ 5.646 \pm 5.834 \\ 6.828 \pm 6.115 \end{bmatrix}; K = \begin{bmatrix} 5.754 \pm 5.576 \\ 5.286 \pm 5.659 \\ 4.735 \pm 5.561 \\ 5.919 \pm 5.940 \\ 7.061 \pm 6.157 \end{bmatrix}$$

## Example 2 of RKHS

$$E(y | \alpha_i, \alpha_j, \beta_i, \beta_j) = \alpha_i + \alpha_j + \beta_i \beta_j + \alpha_i \alpha_j \sqrt{\beta_i \beta_j}, \quad (21)$$

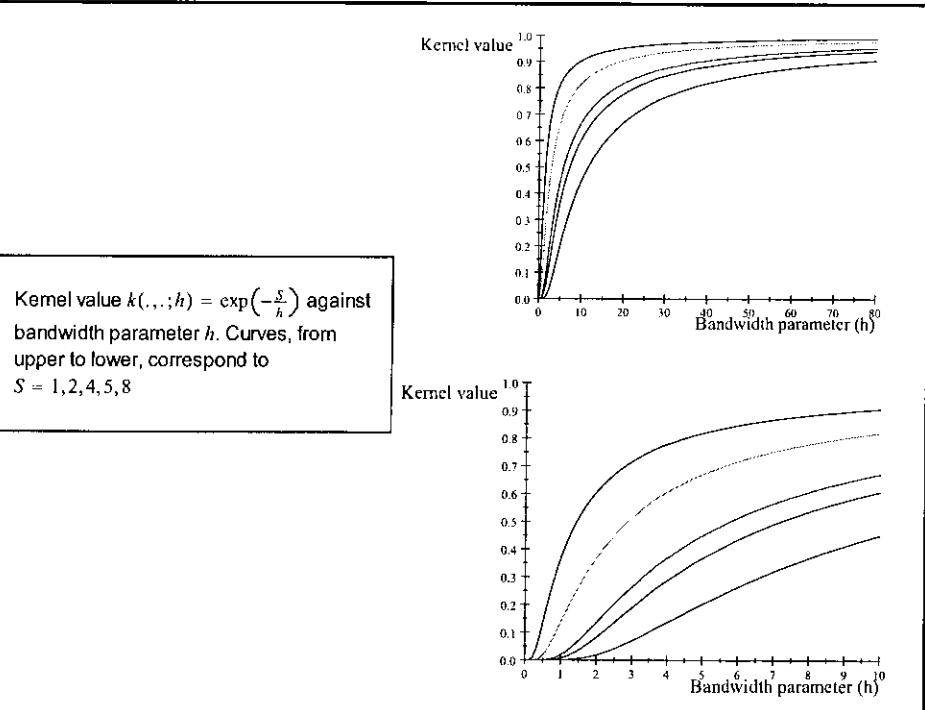
where  $\alpha_i$  ( $\beta_i$ ) and  $\alpha_j$  ( $\beta_j$ ) are effects of alleles  $i$  and  $j$  at the  $\alpha$  ( $\beta$ ) locus. The system is non-linear on allelic effects, as indicated by the first derivatives of the conditional expectation function with respect to the  $\alpha$ 's or  $\beta$ 's. For instance

$$\frac{\partial E(.)}{\partial \alpha_i} = 1 + \alpha_i \sqrt{\beta_i \beta_j}; \quad \frac{\partial E(.)}{\partial \beta_j} = \beta_j + \frac{1}{2} \alpha_i \alpha_j \sqrt{\frac{\beta_i}{\beta_j}},$$

Arbitrary Gaussian kernel adopted for the RKHS regression using as covariate a  $2 \times 1$  vector: number of alleles at each of the two loci, e.g.,  $x_{AA} = 2, x_{Aa} = 1$  and  $x_{aa} = 0$ . For example, the kernel entry  $AABB$  and  $AAbb$  is

$$k(\mathbf{x}_{AABB}, \mathbf{x}_{AAbb}, h) = \exp\left[-\frac{(2-2)^2 + (2-0)^2}{h}\right] = \exp\left[-\frac{4}{h}\right],$$

	$AABB$	$AABb$	$AAbb$	$AaBB$	$AaBb$	$Aabb$	$aaBB$	$aaBb$	$aabb$
$K_h =$	$AABB$	1	$e^{-\frac{1}{h}}$	$e^{-\frac{4}{h}}$	$e^{-\frac{1}{h}}$	$e^{-\frac{2}{h}}$	$e^{-\frac{5}{h}}$	$e^{-\frac{4}{h}}$	$e^{-\frac{5}{h}}$
	$AABb$	$e^{-\frac{1}{h}}$	1	$e^{-\frac{1}{h}}$	$e^{-\frac{3}{h}}$	$e^{-\frac{1}{h}}$	$e^{-\frac{2}{h}}$	$e^{-\frac{5}{h}}$	$e^{-\frac{4}{h}}$
	$AAbb$	$e^{-\frac{4}{h}}$	$e^{-\frac{1}{h}}$	1	$e^{-\frac{5}{h}}$	$e^{-\frac{2}{h}}$	$e^{-\frac{1}{h}}$	$e^{-\frac{5}{h}}$	$e^{-\frac{4}{h}}$
	$AaBB$	$e^{-\frac{1}{h}}$	$e^{-\frac{2}{h}}$	$e^{-\frac{5}{h}}$	1	$e^{-\frac{1}{h}}$	$e^{-\frac{4}{h}}$	$e^{-\frac{1}{h}}$	$e^{-\frac{5}{h}}$
	$AaBb$	$e^{-\frac{2}{h}}$	$e^{-\frac{1}{h}}$	$e^{-\frac{2}{h}}$	$e^{-\frac{1}{h}}$	1	$e^{-\frac{1}{h}}$	$e^{-\frac{2}{h}}$	$e^{-\frac{2}{h}}$
	$Aabb$	$e^{-\frac{5}{h}}$	$e^{-\frac{2}{h}}$	$e^{-\frac{1}{h}}$	$e^{-\frac{4}{h}}$	$e^{-\frac{1}{h}}$	1	$e^{-\frac{5}{h}}$	$e^{-\frac{2}{h}}$
	$aaBB$	$e^{-\frac{4}{h}}$	$e^{-\frac{5}{h}}$	$e^{-\frac{1}{h}}$	$e^{-\frac{1}{h}}$	$e^{-\frac{2}{h}}$	$e^{-\frac{5}{h}}$	1	$e^{-\frac{1}{h}}$
	$aaBb$	$e^{-\frac{5}{h}}$	$e^{-\frac{4}{h}}$	$e^{-\frac{5}{h}}$	$e^{-\frac{2}{h}}$	$e^{-\frac{1}{h}}$	$e^{-\frac{2}{h}}$	$e^{-\frac{1}{h}}$	1
	$aabb$	$e^{-\frac{5}{h}}$	$e^{-\frac{5}{h}}$	$e^{-\frac{4}{h}}$	$e^{-\frac{5}{h}}$	$e^{-\frac{2}{h}}$	$e^{-\frac{1}{h}}$	$e^{-\frac{4}{h}}$	1



$h = 1.75$  as bandwidth parameter  
6 unique entries in the  $K$  matrix:  
1.0 (diagonal elements, the two individuals have identical genotypes)  
0.565 (3 alleles in common in a pair of individuals)  
0.319 (2 alleles in common, 1 per locus)  
0.102 (2 alleles in common at only one locus)  
0.06 (1 allele in common)  
0.01 (no alleles shared).

### Training set

Residuals were drawn from the normal distribution  $N(0, 20)$ , and added to (21) to form phenotypes. The resulting phenotypic distribution is unknown, because  $y$  is a non-linear function of exponential and Weibull variates, plus of an additive normally distributed residual. There were 5 individuals with records for each of the  $AABB, AABb, AAbb$  genotypes; 20 for each of  $AaBB, AaBb$  and  $Aabb$ , and 5 of each of  $aaBB, aaBb$  and  $aabb$ . Thus, there were 90 individuals with phenotypic records, in total.

### Testing set

100  
  
A more important issue, at least from the perspective taken in this paper, is "out of sample" predictive ability. To examine this, 3 new (independent) samples of phenotypes were generated, assuming the residual distribution  $N(0, 20)$ , as before, and with 5 individuals per genotype, i.e., there were 45 subjects in each sample. The predictive

IMPORTANT ISSUE TO DISCUSS HERE

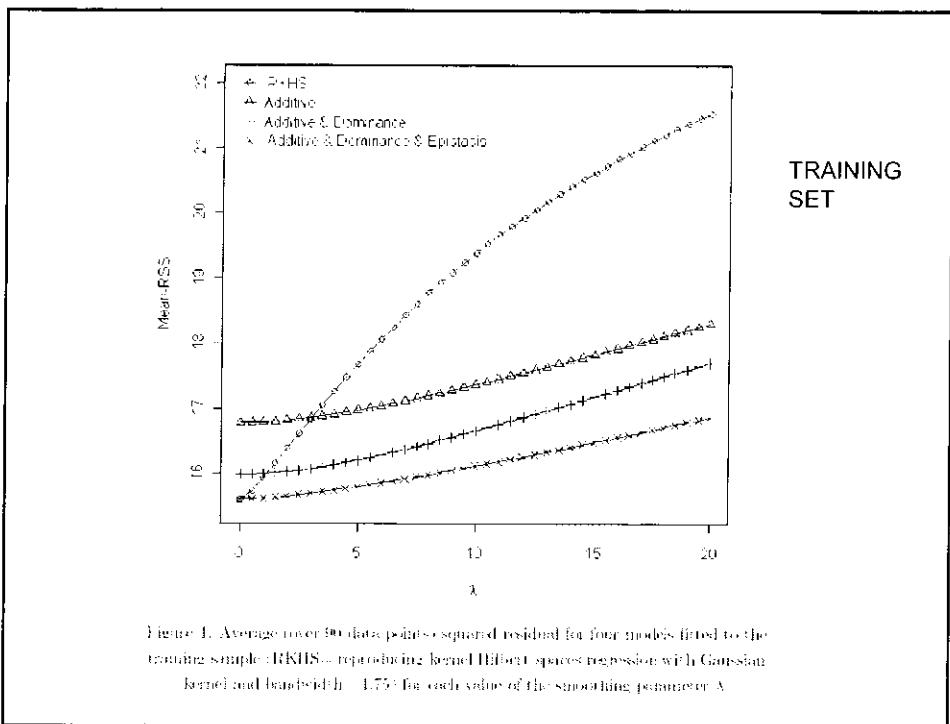


Figure 4. Average over 50 data points squared residual for four models fitted to the training sample (RKHS= reproducing kernel Hilbert spaces regression with Gaussian kernel and bandwidth = 1.75) for each value of the smoothing parameter  $\lambda$ .

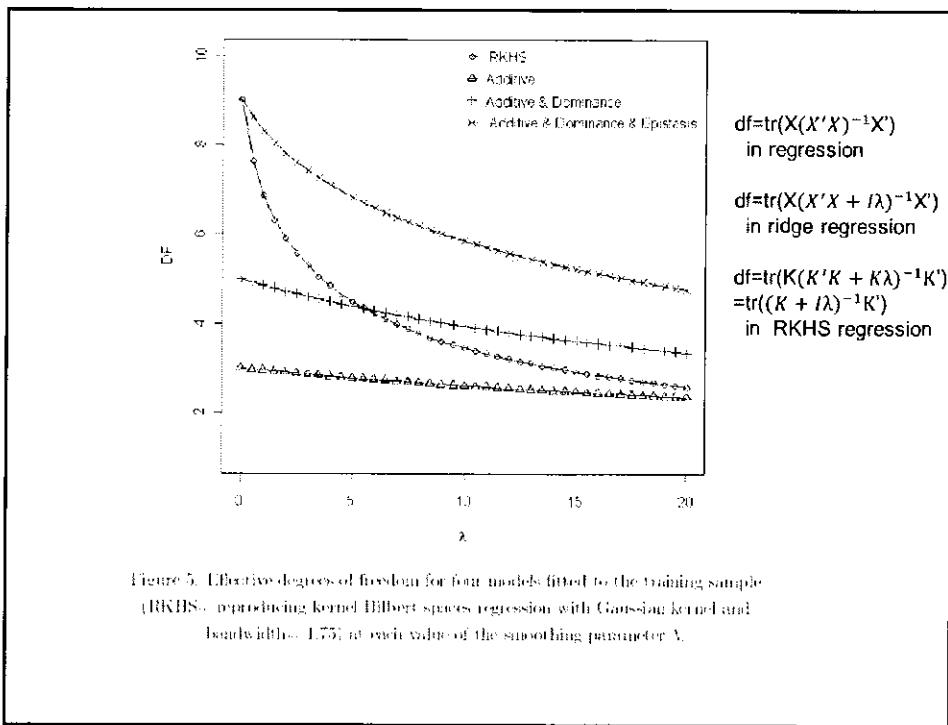
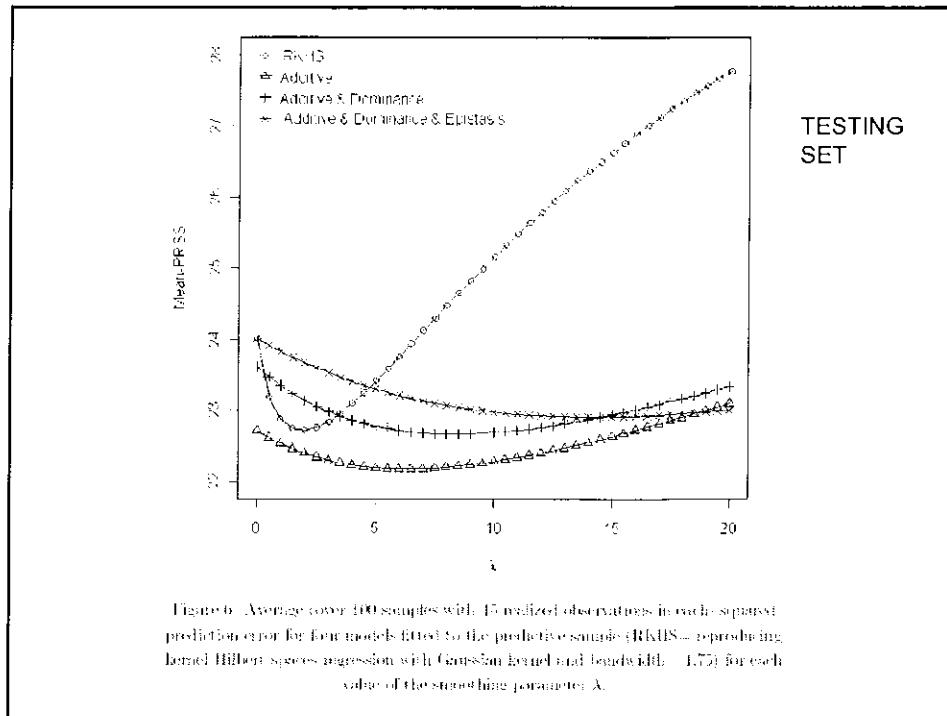


Figure 5. Effective degrees of freedom for four models fitted to the training sample (RKHS= reproducing kernel Hilbert spaces regression with Gaussian kernel and bandwidth = 1.75) at each value of the smoothing parameter  $\lambda$ .

Additive model performed better than rest models.



## Explanation of results

How does one explain the paradox that a simple additive model had better predictive performance when gene action was non-linear, as simulated here? In order to address this question, consider the "true" mean value of the 9 genotypes simulated:

	<i>BB</i>	<i>Bb</i>	<i>bb</i>
<i>AA</i>	11.933	8.000	6.117
<i>Aa</i>	3.626	2.919	2.757
<i>aa</i>	0.916	0.304	0.185

The "corrected" sum of squares among these means is 125.23. A fixed effects analysis of variance of these "true" values (assuming genotypes were equally frequent) gives the following partition of sequential sum of squares, apart from rounding errors: 1) additive effect of locus *A* : 82.8%; 2) additive effect of locus *B* after accounting for *A* : 7.06%; 3) dominance effects of loci *A* and *B* : 4.2%, and 3) epistasis: 6.2%. Thus, even though the genetic system was non-linear, most of the variation among genotypic means can be accounted for with a linear model on additive effects. The additive model had the worst fit to the data (even worse than the models that assume dominance and epistasis) and, yet, it had the best predictive ability, followed by RKHS for (roughly)  $0.5 < \lambda < 3$ . !!

**Example  
Of RKHS 2**

		<i>C<sub>A</sub></i>	<i>C<sub>B</sub></i>	<i>C<sub>C</sub></i>
<i>AA</i>	<i>BB</i>	3	0	3
<i>AA</i>	<i>Bb</i>	0	6	0
<i>AA</i>	<i>bb</i>	3	1	3
<i>Aa</i>	<i>BB</i>	1	2	3
<i>Aa</i>	<i>Bb</i>	3	2	1
<i>Aa</i>	<i>bb</i>	2	2	2
<i>aa</i>	<i>BB</i>	2	2	2
<i>aa</i>	<i>Bb</i>	2	2	2
<i>aa</i>	<i>bb</i>	2	2	2
<i>L(AA)</i>	$= (3 + 3 + 6 + 3 + 3)/9 = 2$			
<i>L(Aa)</i>	$= (1 + 2 + 3 + 2 + 1 + 2 + 2 + 2)/9 = 2$			
<i>L(aa)</i>	$= (2 + 0)/9 = 2$			
<i>L(BB)</i>	$= (3 + 0 + 3 + 1 + 2 + 3 + 2 + 2 + 2)/9 = 2$			
<i>L(Bb)</i>	$= (1 + 6 + 0 + 3 + 2 + 1 + 2 + 2 + 2)/9 = 2$			
<i>L(bb)</i>	$= (3 + 0 + 3 + 2 + 2 + 2 + 2 + 2 + 2)/9 = 2$			
<i>L(CC)</i>	$= (3 + 0 + 3 + 1 + 3 + 2 + 2 + 2 + 2)/9 = 2$			
<i>L(Cc)</i>	$= (0 + 6 + 0 + 2 + 2 + 2 + 2 + 2 + 2)/9 = 2$			
<i>L(cc)</i>	$= (3 + 0 + 3 + 1 + 2 + 2 + 2 + 2 + 2)/9 = 2$			

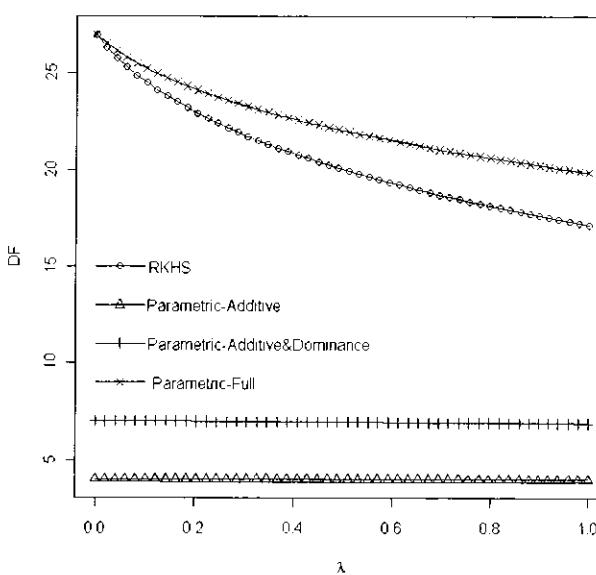
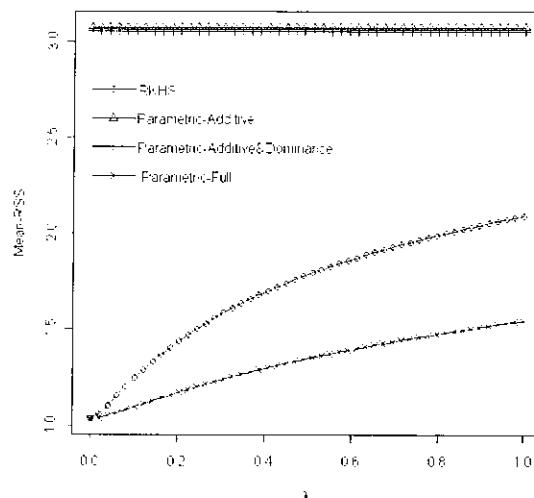
- There is no additive variability at any of the three loci, since adding or removing a "large" allele does not affect mean values.
- There is no dominance at any of the three loci, as indicated by a zero difference between heterozygotes and the average of the homozygotes.
- There is considerable interaction. If genotypes are *AA*, there is pure dominance at each of the *B* and *C* loci. In *AaBB* individuals, removing the *C* allele increases the mean, with the opposite being true in *AaBb*. In *Aabb* individuals the *C* locus genotype is immaterial. In *aa* genotypes, nothing happens.

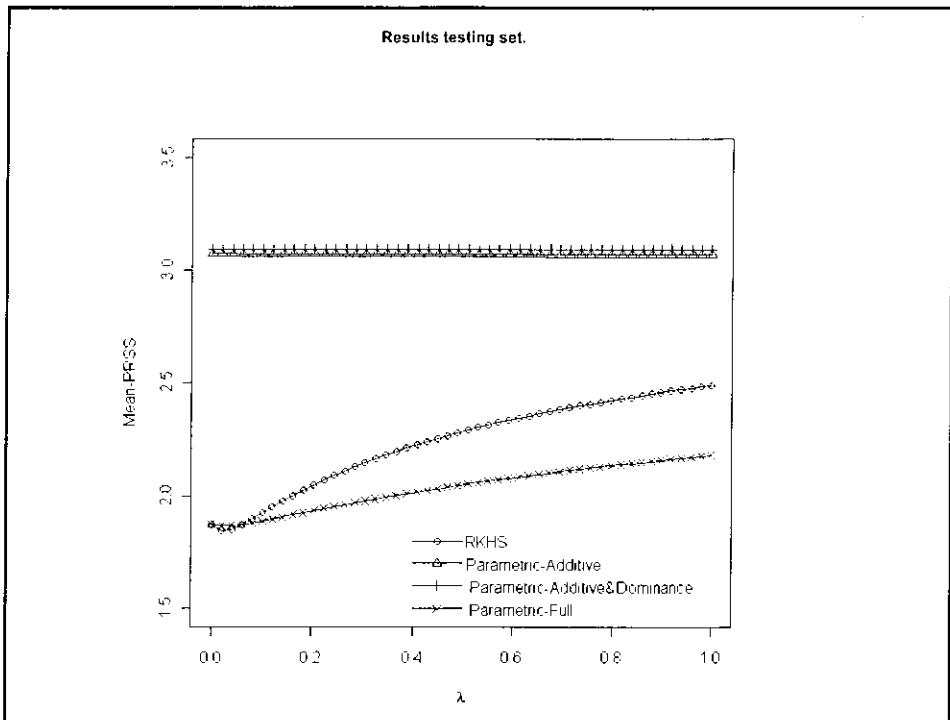
Source	DF	Anova SS	Mean Square	F Value	Pr > F
a	2	0.00000000	0.00000000	0.00	1.0000
b	2	0.00000000	0.00000000	0.00	1.0000
c	2	0.00000000	0.00000000	0.00	1.0000
a*b	4	0.00000000	0.00000000	0.00	1.0000
a*c	4	0.00000000	0.00000000	0.00	1.0000
b*c	4	13.33333333	3.33333333	1.00	0.4609
Error (a*b*c)	8	26.66666667	3.33333333		

Variation between genotypic values is pure interaction

**Training set:**  
 - 27 genotypes,  
 - 5 replicates per genotype,  
 - residual variance 1.5  
**Testing set:** 50 MC replicates, each as the training set.

## Results in training set



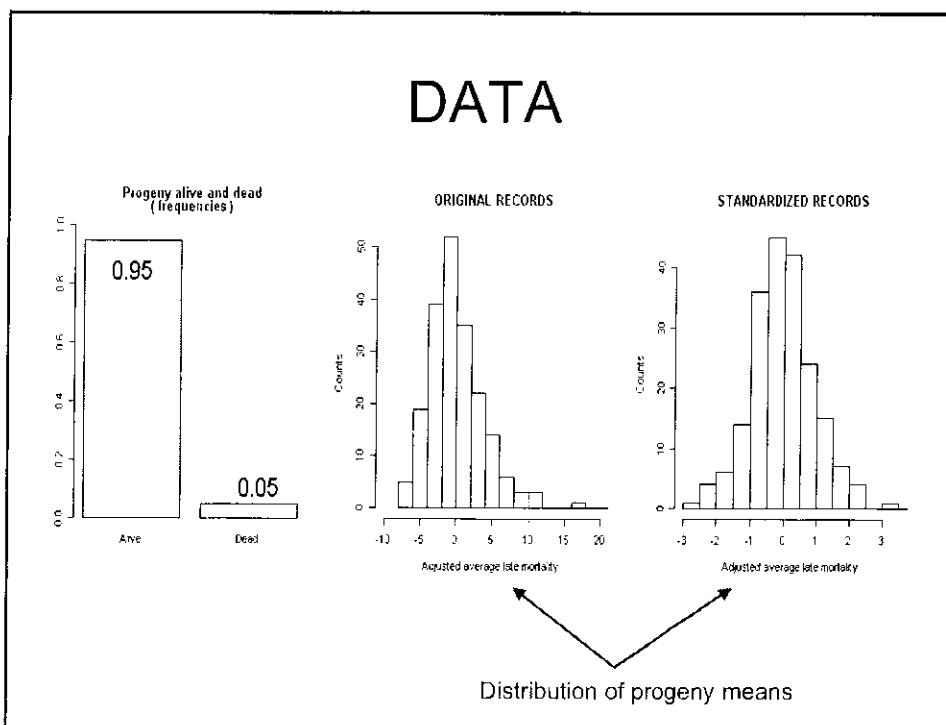


Amagom

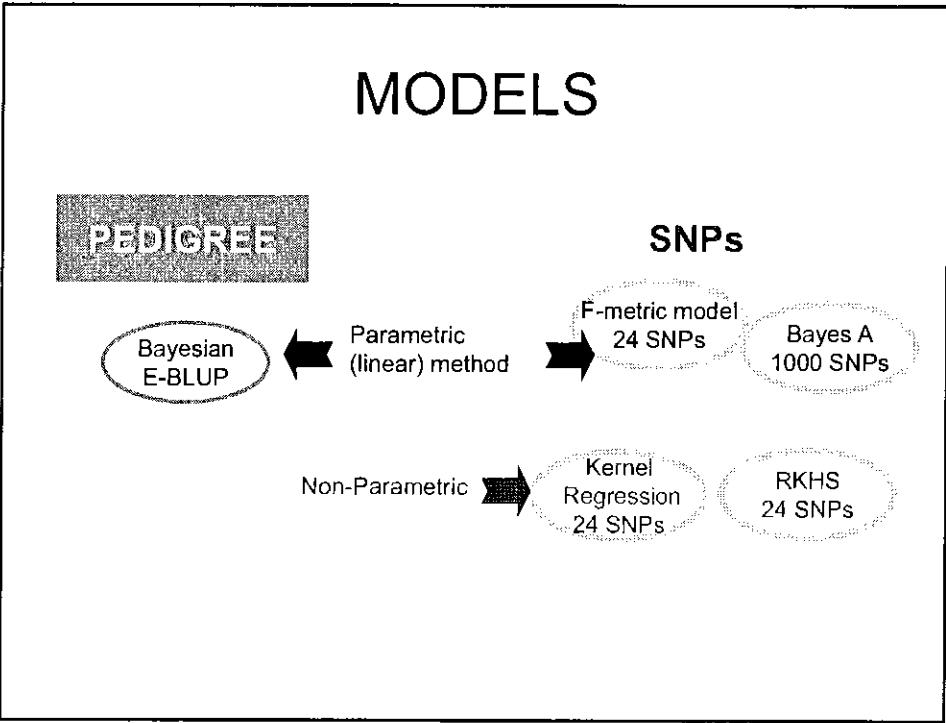
### EXAMPLE 3: CHICKENDATA

- Average progeny “late mortality” ( $l_m$ ) in low hygiene environment for 200 sires of line29 (12,167 progenies).
  - Pre-corrected for hatch, age of dam and dam,
  - Standardized log-transformed means
- SNPs: filter and wrapper strategy (Long et al., 2007)
  - 24 SNPs selected out of over 5000 genotyped on sires

# DATA



# MODELS



Dynamic programming algorithms

Similarity between two DNA sequences

Adapted to SNP sequences

$$K_h(\mathbf{x} - \mathbf{x}_i) = \exp[-Score(\mathbf{x} - \mathbf{x}_i)]$$

No need to tune  $h$

(Delcher et al., 1999, 2002)

## Variance component & parameter estimates

Parameter	Posterior features	E-BLUP	F-metric	RKHS	BR (Xu's)
$\sigma_e^2$	$\mu$ (s.d)	24.38 (3.88)	29.72 (3.56)	17.07 (3.02)	20.75 (2.91)
	HPD (95%)	16.88-32.04	23.60-37.51	11.78-23.64	15.62-27.09
$\sigma_u^2$	$\mu$ (s.d)	0.10 (0.06)	...	...	1.03 (0.71)
	HPD (95%)	0.03-0.24	...	...	0.67-1.95
$\sigma_\alpha^2$	$\mu$ (s.d)	...	...	0.40 (0.07)	...
	HPD (95%)	...	...	0.28-0.55	...
$h^2$	$\mu$ (s.d)	0.02 (0.01)	...	...	...
	HPD (95%)	0.004-0.050	...	...	...

Sum of posterior means of variances of the 1000 markers

- Spearman (above diagonal) and Pearson correlations (below diagonal) between posterior means of sire effects

	E-BLUP	t-metric	Kernel	RKHS	BR
E-BLUP		0.52	0.77	0.84	0.91
F-metric					
Kernel	0.66		...	0.93	0.76
RKHS	0.84		0.79	...	0.84
BR	0.92		0.58	0.80	...

- E-BLUP & Bayes A very similar.

## MODEL FIT

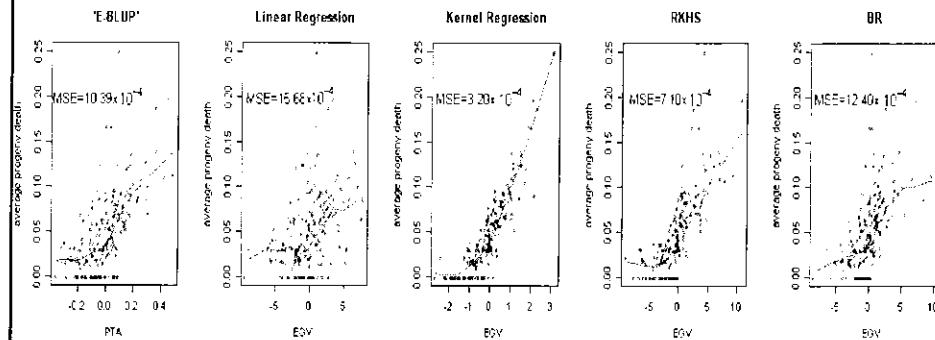
-Compute deviance measurement based on mean squared errors:

- A) Regression of adjusted average progeny on sire's PTA or EGV
- B) Regression of raw average progeny on sire's PTA or EGV

-Lowess regression  
(Non-parametric locally weighted regression)

## MODEL FIT

- Regression of adjusted raw progeny LM on sire's PTA or EGV



## MODEL FIT

- Less dispersion in non-parametric models
- Lower MSE for kernel regression
- Worst for Linear regression (F-metric model)

Still....which model predicts the data best ?

## Predictive ability

- Cross validation
  1. 5 subsets, letting 20% sire means missing each time at random
  2. Calculate correlations between actual and inferred average progeny, for each method within subset.

## Predictive ability

Subset	E-BLUP	F-metric	Kernel	RKHS	BR
1 <sup>st</sup>	0.03	<b>0.27</b>	0.05	<b>0.27</b>	0.13
2 <sup>nd</sup>	0.18	0.19	0.28	<b>0.37</b>	0.12
3 <sup>rd</sup>	<b>0.18</b>	0.08	0.06	-0.01	0.17
4 <sup>th</sup>	-0.04	0.07	0.13	<b>0.28</b>	0.15
5 <sup>th</sup>	0.17	-0.12	0.23	0.15	<b>0.25</b>
GLOBAL	0.10	0.06	0.14	<b>0.20</b>	0.16

- RKHS showed better predictive ability
  - 25% higher reliability than Xu's method
  - 100% higher reliability than E-BLUP
  - 233% higher reliability than F-metric (linear regression on markers)
- RKHS better than fixed or random regression on markers and E-BLUP.

EXAMPLE 4: CHICKEN DATA

Genomic-assisted prediction of a quantitative trait in parents and progeny: application to food conversion rate in chickens

FCR measured on progeny of 333 sires with 3481 SNPs  
 FCR measured on progeny of 61 birds (sons of the above sires)

→ 2-generation data set

BAYES A --all markers  
 RKHS --all markers  
 RKHS --400 markers filtered using different INFOGAINS  
 BLUP (Bayes) –pedigree information

Training set: 333 sires of sons

Predictive set: 61 sons of sires

**Table 4:** Means, standard deviation (s.d.) and 95% confidence intervals (CI) of the Bootstrap distribution of Spearman correlations between predicted and observed phenotypes in the testing set (E-BLUP: Bayesian linear model; Bayes A: Bayesian regression on SNP; RKHS: reproducing kernel Hilbert spaces regression).

Whole genome methods

method	mean	s.d.	CI(95%)
E-BLUP	0.11	0.13	(-0.13, 0.35)
Bayes A	0.27	0.12	(0.04, 0.49)
RKHS	0.27	0.12	(0.03, 0.50)

Information gain using 2 classes (400 pre-selected SNPs)  
 + RKHS

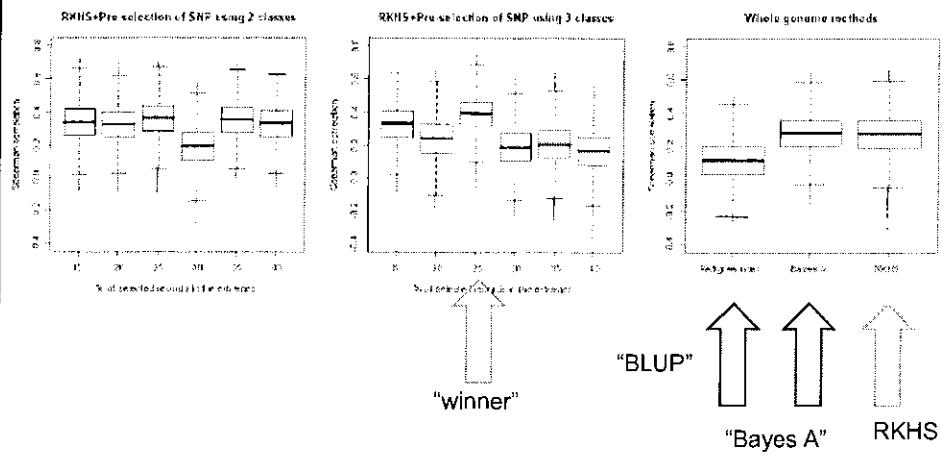
percentile	mean	s.d.	CI(95%)
0.15	0.33	0.12	(0.09, 0.56)
0.20	0.32	0.11	(0.10, 0.53)
0.25	0.36	0.11	(0.13, 0.57)
0.30	0.19	0.12	(-0.05, 0.42)
0.35	0.35	0.11	(0.12, 0.55)
0.40	0.33	0.11	(0.10, 0.53)

Information gain using 3 classes (400 pre-selected SNPs)  
 + RKHS

percentile	mean	s.d.	CI(95%)	percentile	mean	s.d.	CI(95%)
0.15	0.32	0.11	(0.10, 0.54)	0.30	0.19	0.12	(-0.05, 0.42)
0.20	0.24	0.13	(-0.01, 0.48)	0.35	0.20	0.12	(-0.04, 0.43)
0.25	0.39	0.11	(0.16, 0.59)	0.40	0.16	0.12	(-0.08, 0.40)

Note that the confidence bands of the predictive correlations are wide

**Figure 2.** Box plots for the bootstrap distribution of Spearman correlations between predicted and observed phenotype in the testing set (progeny) obtained with: RKHS on 400 pre-selected SNPs using 2 or 3 classes to classify sires with different percentiles (left and middle panels, respectively) and methods using pedigree or all available SNPs (right panel).



### EXAMPLE 5: Application to US Jersey data

#### ⇒ US Jersey

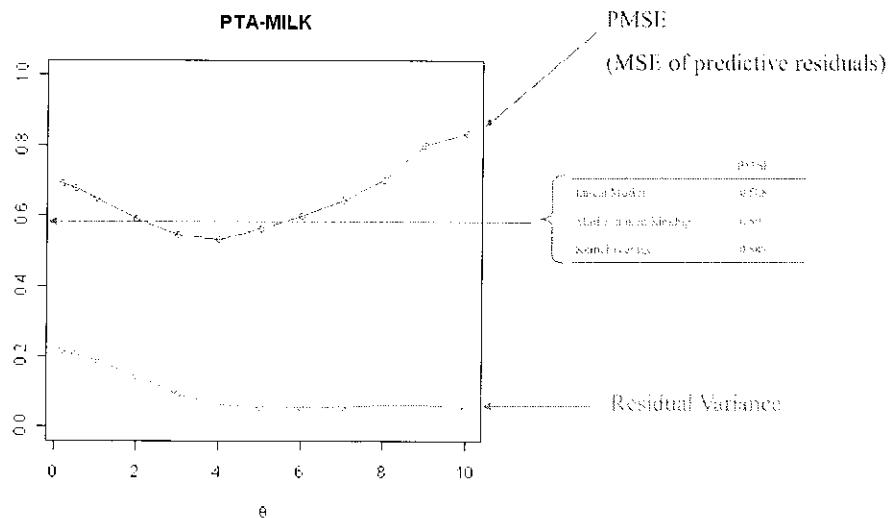
- N= 1,762 sires (n=1446, training n=1130 ; testing, n=316 ).
- **Markers:** BovineSNP50 BeadChip (50k).
- **Traits:** PTAs for Milk, Protein Content and Daughter Pregnancy Rate

#### ⇒ Models:

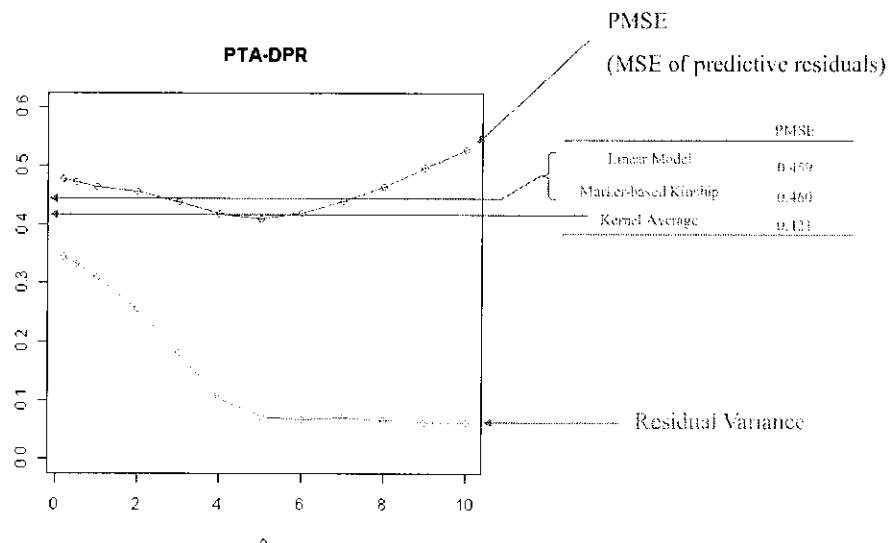
- Linear model       $\mathbf{K} = \mathbf{XX}'$
- Genomic-based kinship     $\mathbf{K} = \mathbf{G}$       [1]
- Gaussian Kernel     $K(i, j | \theta) = \text{Exp}\left\{ -\theta \times d(\mathbf{x}_i, \mathbf{x}_j) \right\}$ 
  - Fixed over a grid of values
  - Kernel averaging:

[1] Hayes and Goddard (2008) Journal of Animal Science.

### Application to US Jersey data



### Application to US Jersey data



### Predictive ability of models for genomic selection in Wheat [1]

Environment	Predictive Correlation		Difference (%)
	BL	RKHS	
E1	0.518	0.601	+16%
E2	0.493	0.494	0%
E3	0.403	0.445	+10%
E4	0.457	0.524	+15%

N= 599;

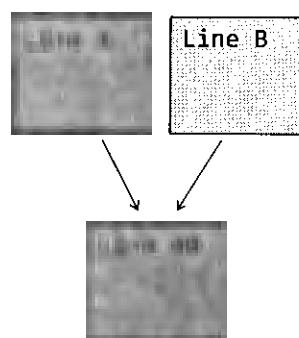
Trait: Grain Yield (4 environments);

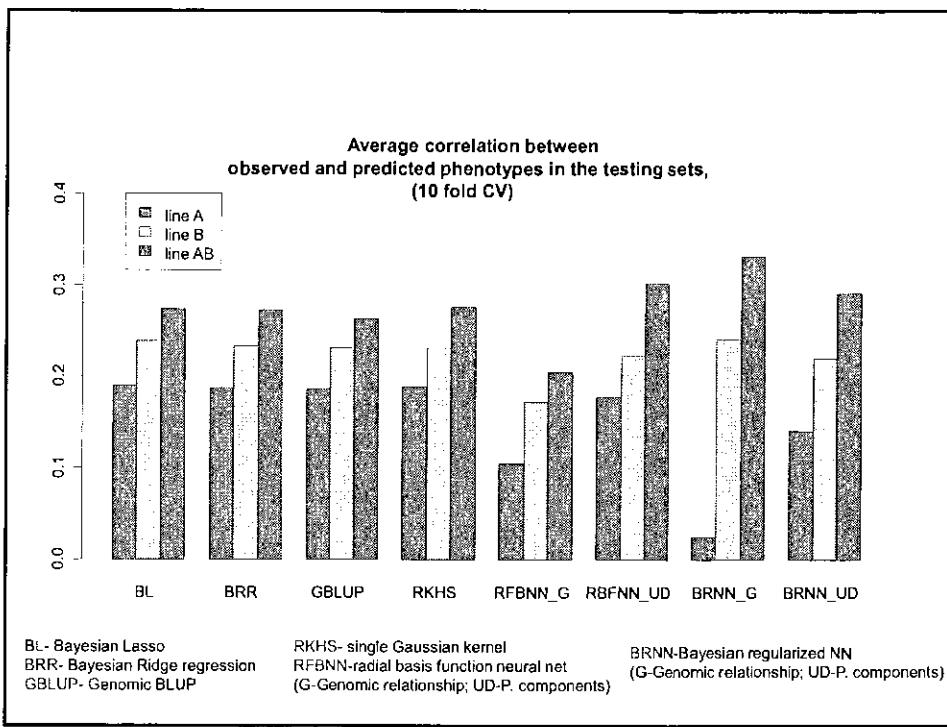
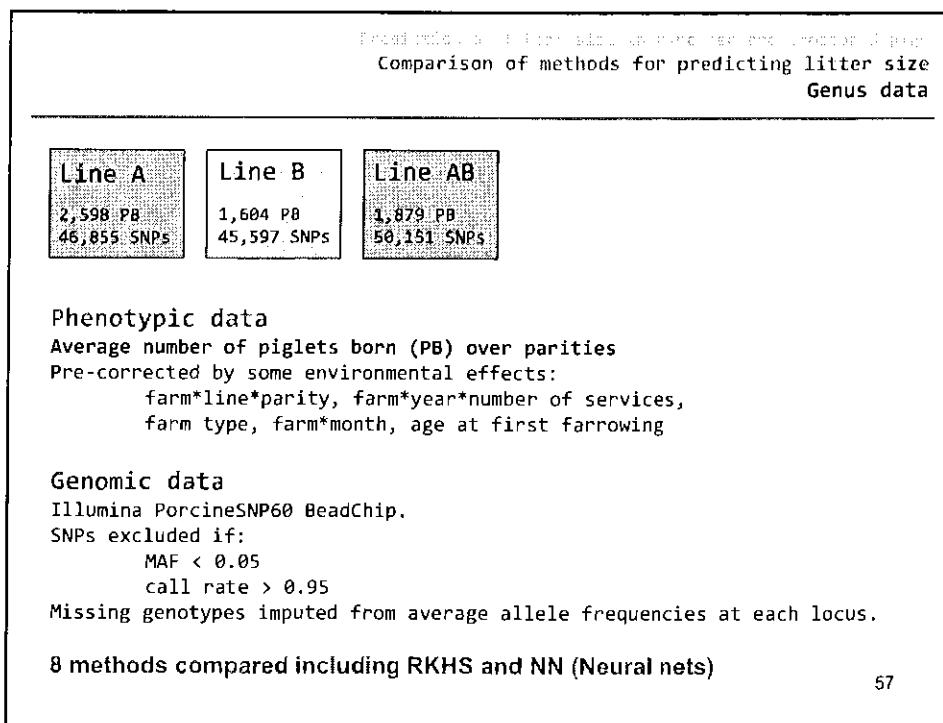
Models: RKHS and Bayesian LASSO (BL)

[1] Crossa *et al.* (2010) Genetics.

### Case study: pigs

#### \*Prediction of litter size in purebred and crossbred pigs





## RKHS-KERNEL AVERAGING-MODEL AVERAGING

http://.../...

- It is **theoretically** possible to enhance ANY predictive model by using Bayesian Model Averaging:

"Predictions obtained by averaging over models are better, on average, than predictions from single model, even the "best"."

### WELL KNOWN THEORETICAL RESULT IN BAYESIAN MODEL AVERAGING

- Example: with 3 bandwidths for Gaussian kernels, we can have predictions based on the following models:

- 1 : RKHS with K<sub>1</sub>
- 2 : RKHS with K<sub>2</sub>
- 3 : RKHS with K<sub>3</sub>
- 4 : RKHS-KA with K<sub>1</sub>, K<sub>2</sub>
- 5 : RKHS-KA with K<sub>1</sub>, K<sub>3</sub>
- 6 : RKHS-KA with K<sub>2</sub>, K<sub>3</sub>
- 7 : RKHS- KA with K<sub>1</sub>, K<sub>2</sub>, K<sub>3</sub>

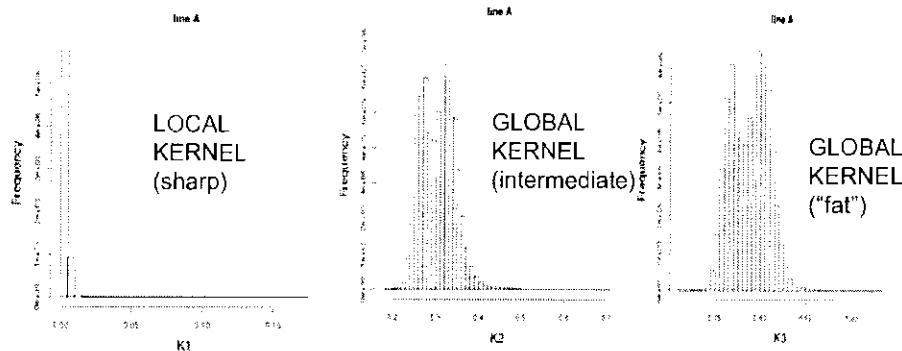
8 : Average of predictions from models 1 to 7

8\*: Weighted average from model 1 to 7 according to harmonic mean of  $\log \left[ \hat{p}(\mathbf{y}|M_i) \right]$

(See Sorensen & Gianola, 2002.)

**Continued...**

### PIC dataset for line A (litter size): GAUSSIAN KERNEL AT 3 BANDWIDTHS



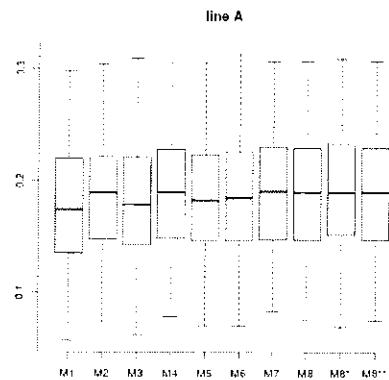
**Figure 6:** histograms of the entries of  $K = \{K(\mathbf{x}_i, \mathbf{x}_j)\}_{i,j}$ .

60

Continued...

Predictive ability:

- 50 random partitions with 90% of observations in training and 10% in testing and
- Correlations between observed and predicted phenotypes.



**Figure 6:** Distribution of correlations between observed and predicted phenotypes.

64

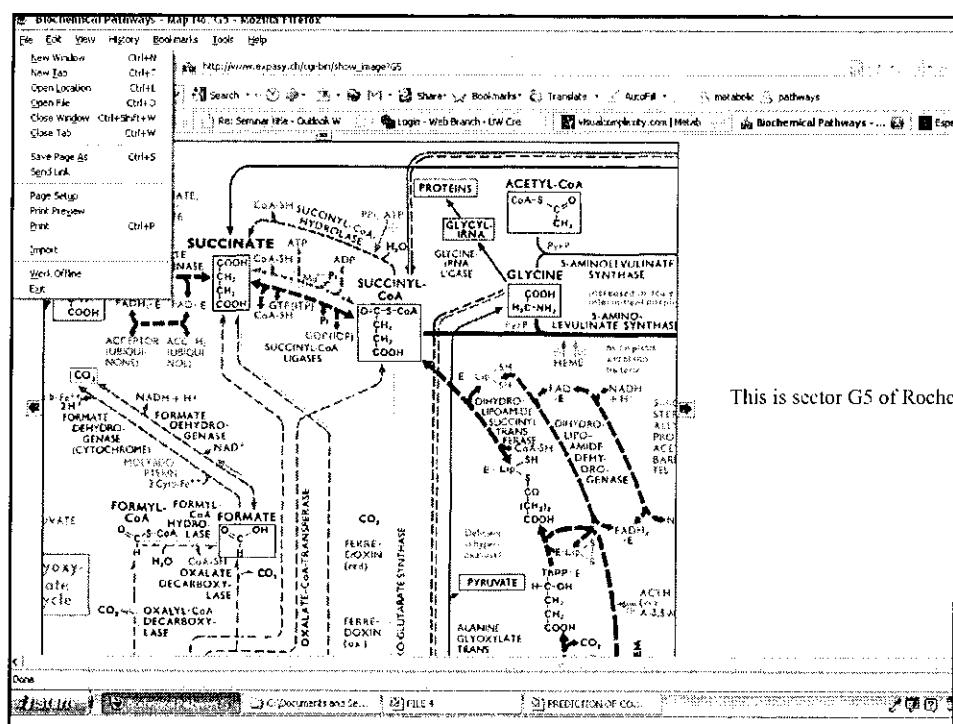
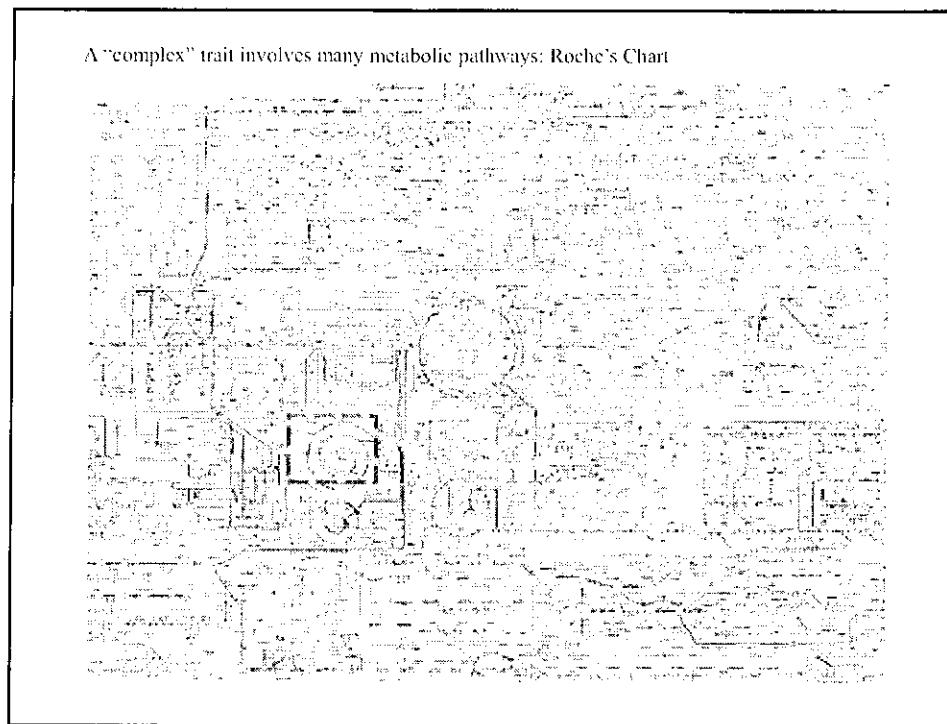
AVERAGING PREDICTIONS NOT WORSE THAN BMA; NOISY DATA

## **Neural networks applied to pedigree or genomic-enabled prediction**

### **Proposition 1**

It must be true that quantitative traits  
are “complex”, in any sense of the  
word.  
Why?

A "complex" trait involves many metabolic pathways: Roehc's Chart

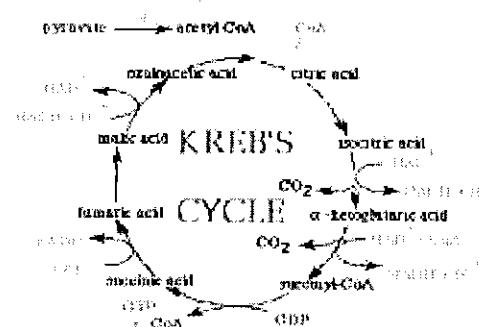


This is sector G5 of Roche

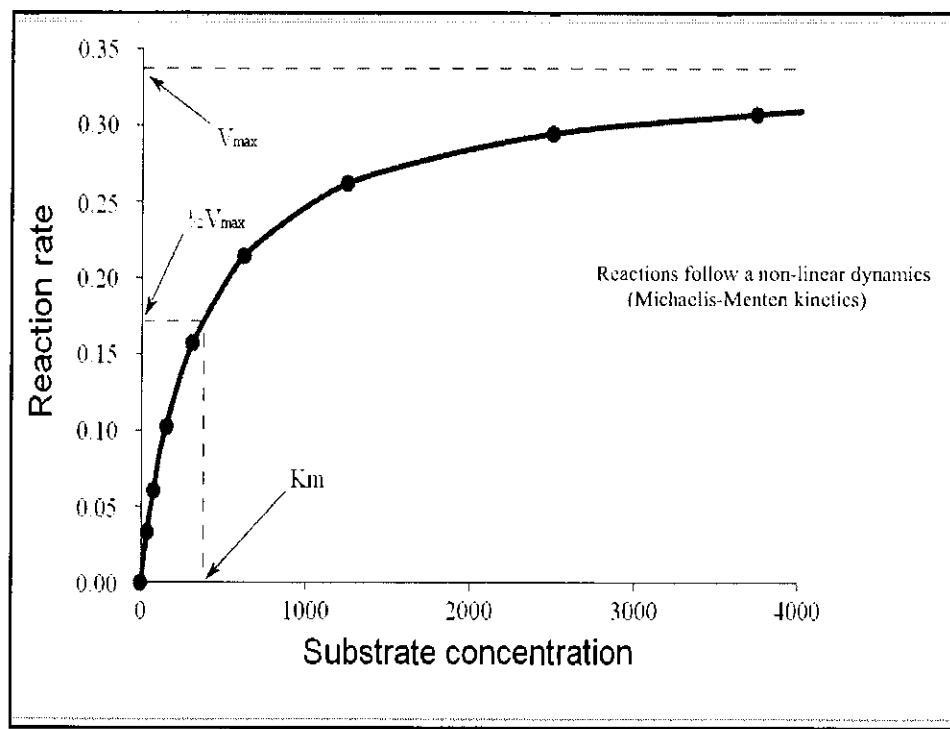
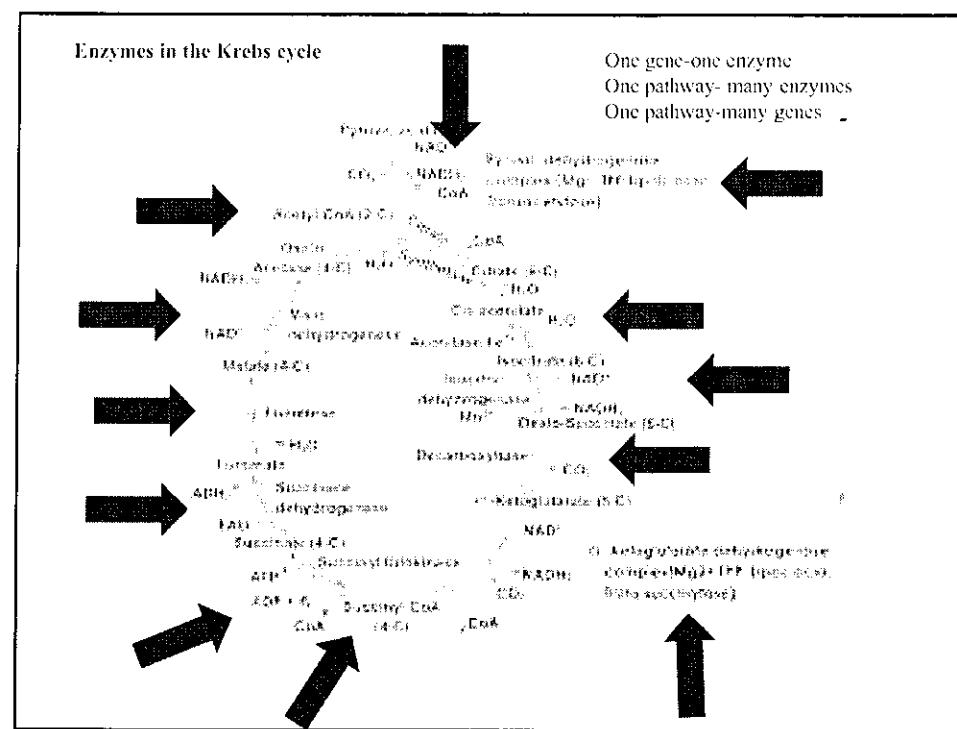
## Proposition 2

It must be true that epistasis  
is pervasive

Example: the tricarboxylic acid cycle

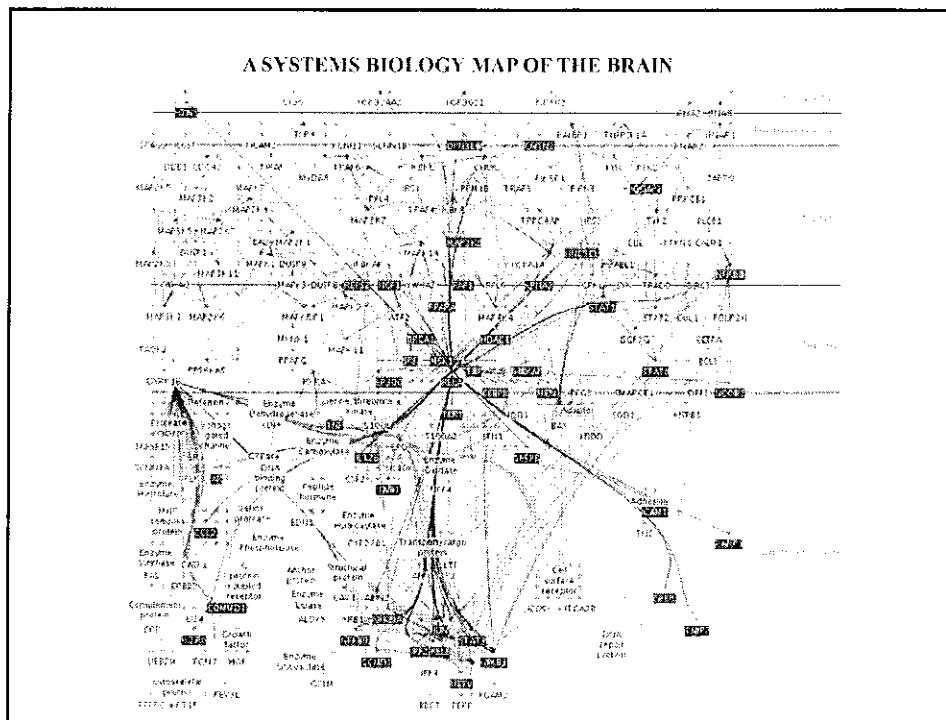


For this to work: enzymes are needed



### **Proposition 3**

A phenotype must be the result of a system involving epistasis and non-linearities of all sorts



CAN ONE WRITE A  
MECHANISTIC MODEL FOR  
SOMETHING LIKE THAT?

### Proposition 4

- It is unlikely that one could arrive to any reasonable mechanistic model satisfactory to understand, explain, learn and predict outcomes

GENOMICS (QTL)  
PROTEOMICS (P-QTL)  
METABOLOMICS (BOLO-QTL)  
EXPRESSIONOMICS (E-QTL)  
EPIGENOMICS (M-QTL)  
METAGENOMICS (META-QTL)

Need to navigate in an extraordinarily highly dimensional space  
to understand "genetic architecture"!!!!

# Welcome to the world of abstractions! Coping with complexity

First assumption: there is a genetic signal and an environmental signal  
Second assumption: the joint effect translates into a phenotype  $y$

$$Y = f(G, E) \quad \text{For some UNKNOWN function } f$$

Choices?

$$\left\{ \begin{array}{l} Y = G^E? \\ Y = E^G? \\ Y = G + E + GE? \xrightarrow{\text{Is an assumption}} \\ Y = (G + E)^{GE}? \\ Y = G + E? \xrightarrow{\text{Is an even a stronger assumption}} \end{array} \right.$$

Further,  $G$  is unknown, so has to be inferred from phenotypes and some input set:

- Pedigrees
- DNA data
- RNA data
- Pedigrees, DNA, RNA

## **THE BIGGEST SHOW ON EARTH:**

A prevailing view (Hill et al., 2008; Crow, 2010; Hill, 2010)

- Fisher's theorem of natural selection
- Interactions are second-order effects; likely tiny and hard to detect
- Detectable epistasis probably arises with genes of large effects, unlikely to be observed in outbred populations
- Epistatic systems generate additive variance and “release” it, so why worry?

## **THE BIGGEST SHOW ON EARTH: POINT-COUNTERPOINT**

### **• Fisher's theorem of natural selection (Kempthorne, 1978)**

mean”; again a basic epistemological error. On the matter of the role of variance, to say that additive genetic variance is important “since Fisher's fundamental theorem of natural selection predicts . . . is wide of the mark, and again exemplifies an error commonly made in population genetics. Fisher's theorem, if it is correct, deals with fitness, whatever that is and

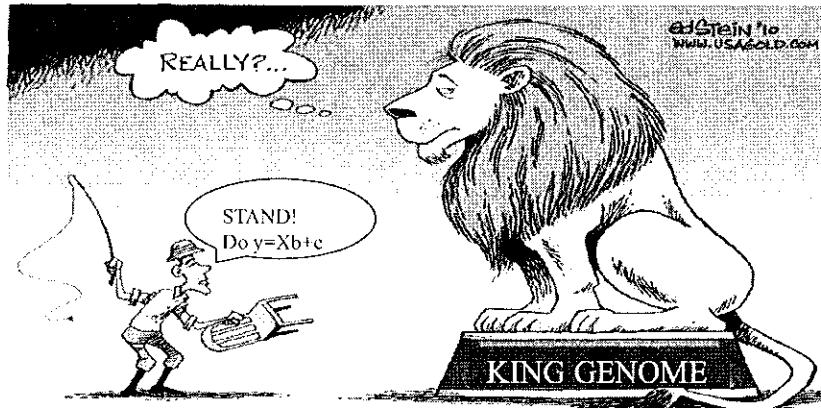
- Interactions are second-order effects; likely tiny and hard to detect  
.....perhaps, but there may be many
- Detectable epistasis probably arises with genes of large effects, unlikely to be observed in outbred populations  
....may be the instruments are not adequate?
- Epistatic systems generate additive variance and “release” it, so why worry?

.... if all we get are straight lines (even though the world is round) how can we learn about “genetic architecture” with such lines, if the world is truly round?

## THE BIGGEST SHOW ON EARTH

(The additive genetic model)

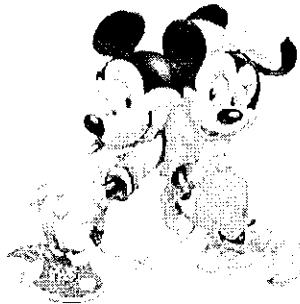
Can "Genome" the lion be tamed?



GENETIC ARCHITECTURE DETECTOR

### Another show: "Les Idiots Savants" (much less popular)

- If phenotypic prediction is crucial (medicine, precision mating) can exploitation of interaction have added value?
- Ideally, search for machine that
  - captures additivity (breeding), interaction (medicine)
  - has reasonably good predictive ability
  - general and flexible with respect to input data
  - does not fail if system is linear and non-interacting

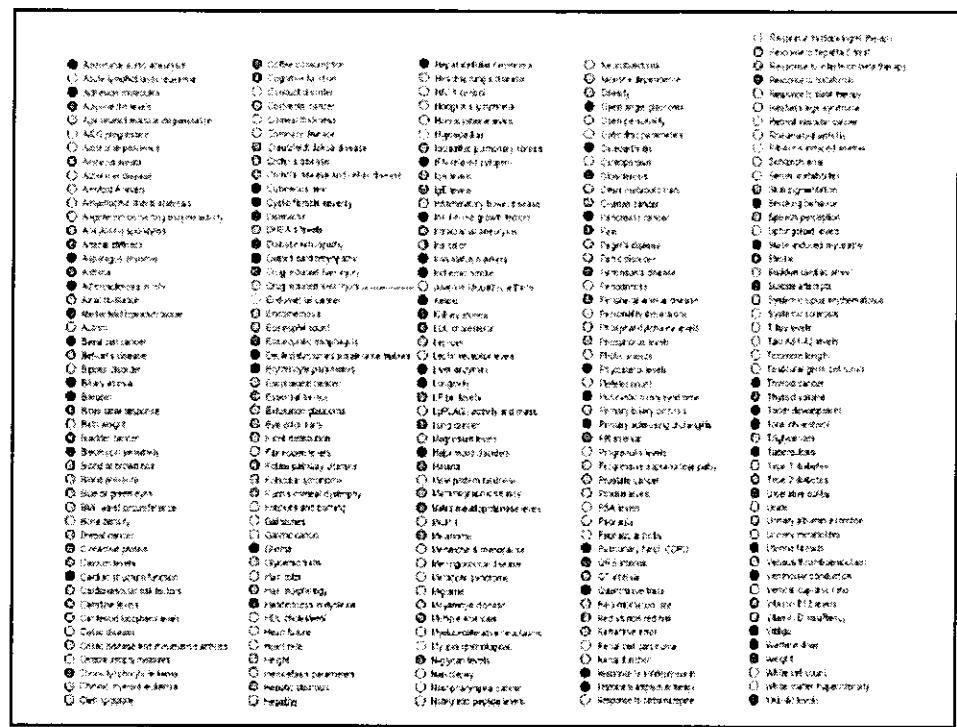
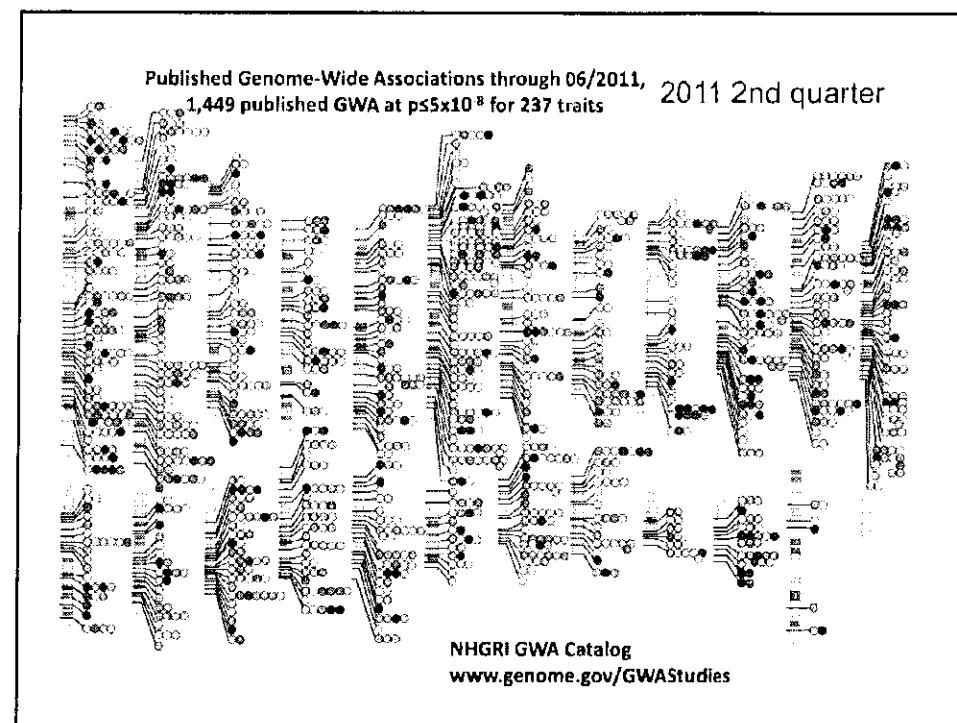


## THE AGE OF INNOCENCE

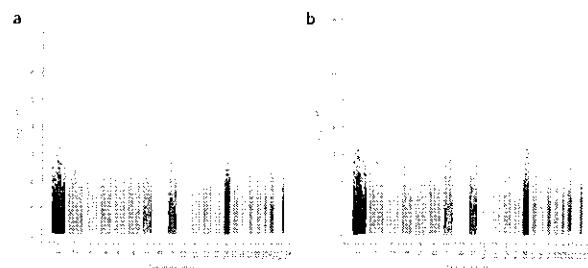
Unraveling “genetic architecture”  
with statistical models

## ARCHITECTURAL PARADIGM 1

**GWAS:** *search for association  
between some marker or genomic  
region and a phenotype*



## EXAMPLES

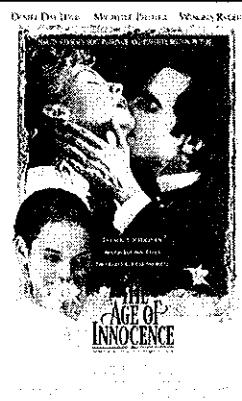
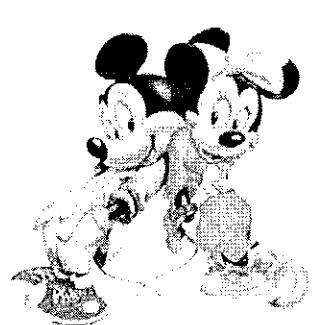


GWAS FOR PANCREATIC CANCER...  
(Nature Genetics)

OR

Kerns SL, Ostrer H, Stock R et al.

Genome-Wide Association Study to Identify Single Nucleotide Polymorphisms (SNPs) Associated With the Development of Erectile Dysfunction in African-American Men After Radiotherapy for Prostate Cancer.  
*International journal of radiation oncology, biology, physics* 2010



## THE AGE OF INNOCENCE

(issue)

Unraveling “genetic architecture”  
with statistical models

**SINGLE MARKER REGRESSION  
WITH ORDINARY LEAST-SQUARES**  
n (#number of observations << p (# markers))

"Full model"



$$y = X\beta + e \\ = (X_1\beta_1 + X_2\beta_2) + e$$

"marked phenotype"

"OLS" is biased If full model holds and one fits "smaller" model (e.g., single marker Regressions)



$$y = X_1\beta_1 + e \\ E(\tilde{\beta}_1|X_1) = (X'_1 X_1)^{-1} E(y) \\ = (X'_1 X_1)^{-1} [X_1\beta_1 + X_2\beta_2] \\ = \beta_1 + (X'_1 X_1)^{-1} X'_1 X_2 \beta_2$$

EXTRAORDINARILY NAÏVE, YET....

**SINGLE MARKER REGRESSION  
WITH ORDINARY LEAST-SQUARES**  
n (#number of observations << p (# markers))

"Full model"



$$y = X\beta + e \\ = (X_1\beta_1 + X_2\beta_2) + e$$

"marked phenotype"

"OLS" is biased If full model holds and one fits "smaller" model (e.g., single marker Regressions)



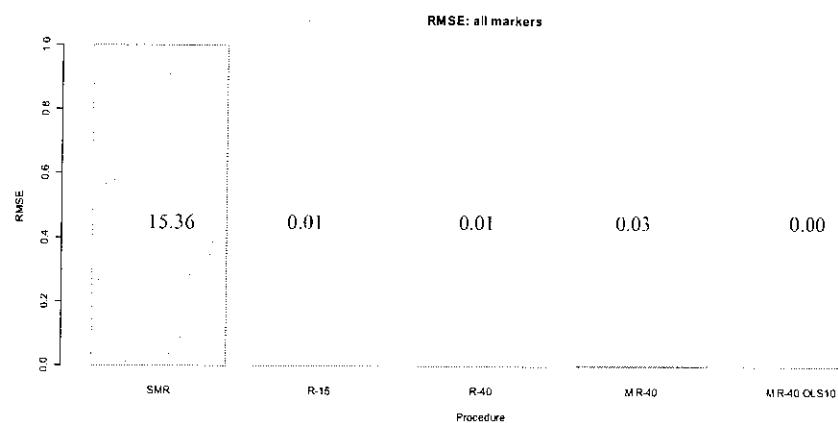
$$y = X_1\beta_1 + e \\ E(\tilde{\beta}_1|X_1) = (X'_1 X_1)^{-1} E(y) \\ = (X'_1 X_1)^{-1} [X_1\beta_1 + X_2\beta_2] \\ = \beta_1 + (X'_1 X_1)^{-1} X'_1 X_2 \beta_2$$

EXTRAORDINARILY NAÏVE, YET....

### SINGLE MARKER REGRESSION: A DISASTER

N=100, 1000 binary markers, 5 first are signal, LD~1/3

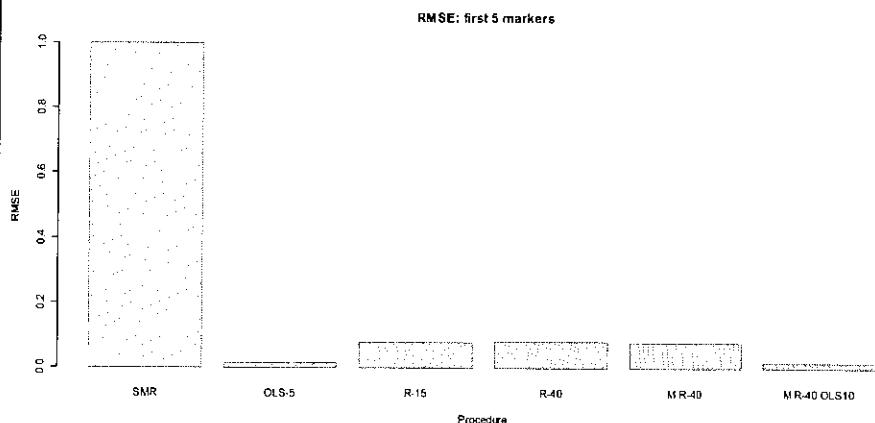
RELATIVE MEAN-SQUARED ERROR (ALL MARKERS)

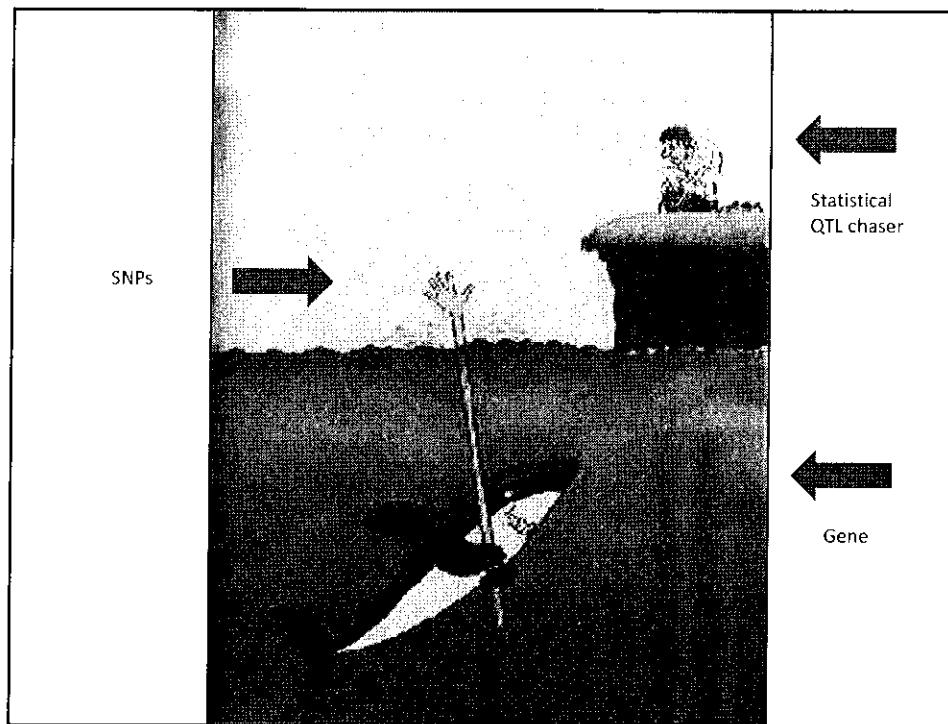


### SINGLE MARKER REGRESSION: A DISASTER

N=100, 1000 binary markers, 5 first are signal, LD~1/3

RELATIVE MEAN-SQUARED ERROR (FIRST FIVE MARKERS)



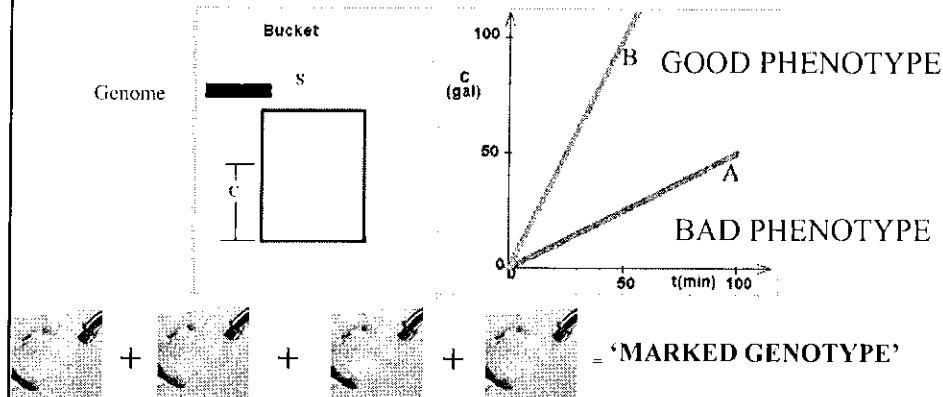


### **PEOPLE DO GWAS: THERE MUST BE ADVANTAGES...**

- Can make nice colored graphs
- Publish in high profile-journals
- Produce rapid tests
- Patent tests and sell drugs
- Probably die before lawsuits catch with you
- Make stories about “missing heritability”
- Ask for money for measuring more stuff
- Generate employment for statisticians

## ARCHITECTURAL PARADIGM 2

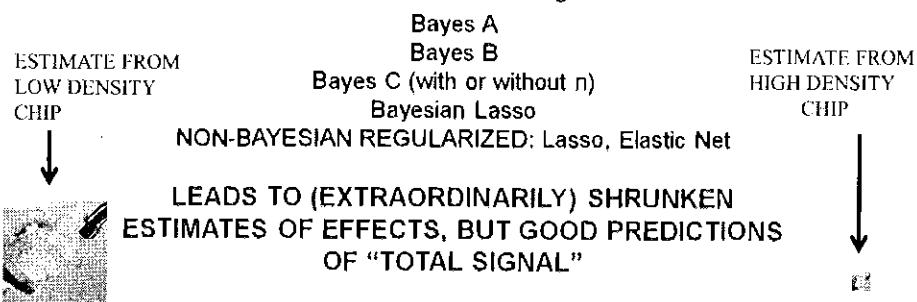
**WGP:** search for association between a linear function of (many) marker covariates and a phenotype



A (slightly) less naïve form of approximating G is the whole-genome linear model:

$$G = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \dots + w_p x_p$$

Where the x's are either pedigree relationships, or marker genotype codes or whatever the latest fad in genomic data is



Data on 4,898 progeny tested Holstein bulls were provided by the USDA-ARS Animal Improvement Programs Laboratory (Beltsville, MD) and comprised 36,778 SNP markers (minor allele frequency, MAF > 0.025) along the entire genome genotyped with the Illumina BovineSNP50 Bead Chip (Illumina Inc., San Diego, CA), as well as Predicted Transmitting Abilities (PTA) for milk protein yield.

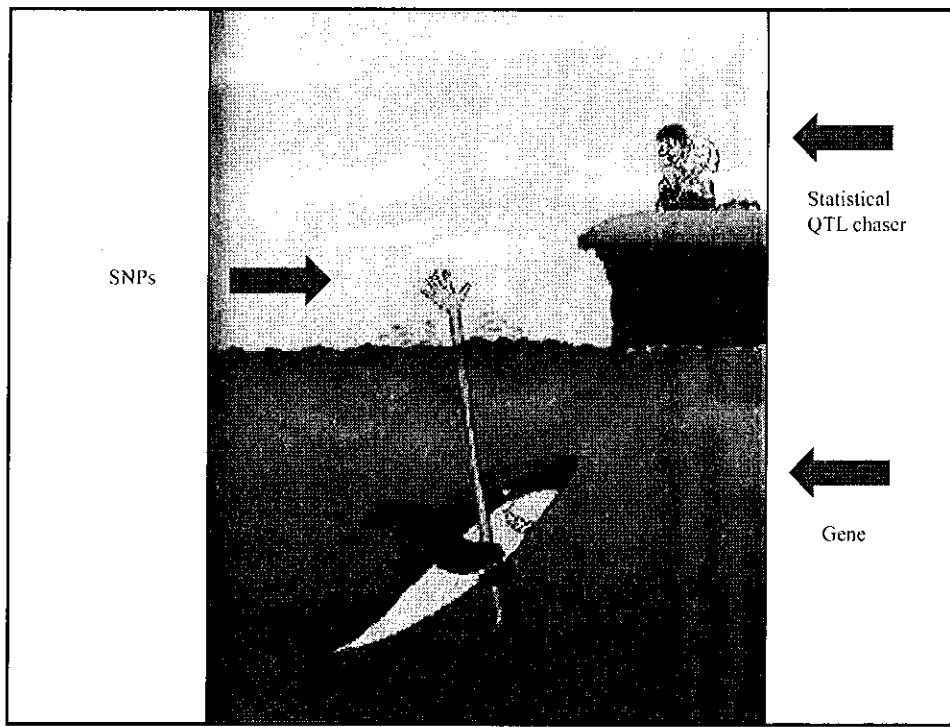
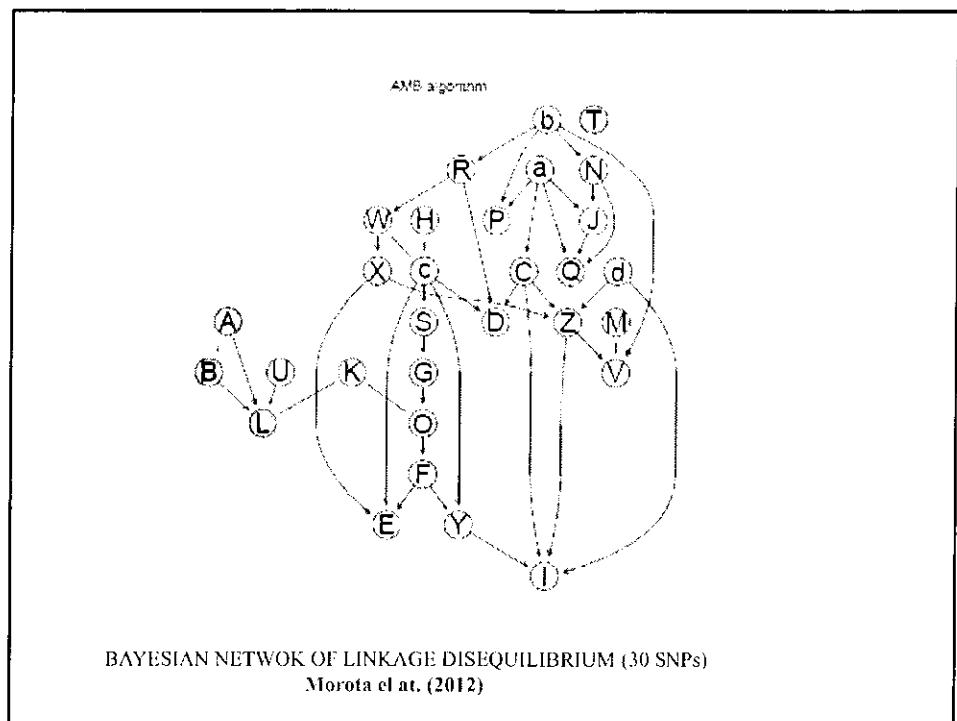
**-BAYESIAN LASSO MODEL WITH N= 4898 p=36778  
 -SNPs RANKED ACCORDING TO ABSOLUTE VALUES OF:  
 POSTERIOR MEANS.  
 STANDARDIZED POSTERIOR MEANS (USING POSTERIOR SD)  
 CONTRIBUTION TO ADDITIVE GENETIC VARIANCE**

**Morota et al. (2012)**

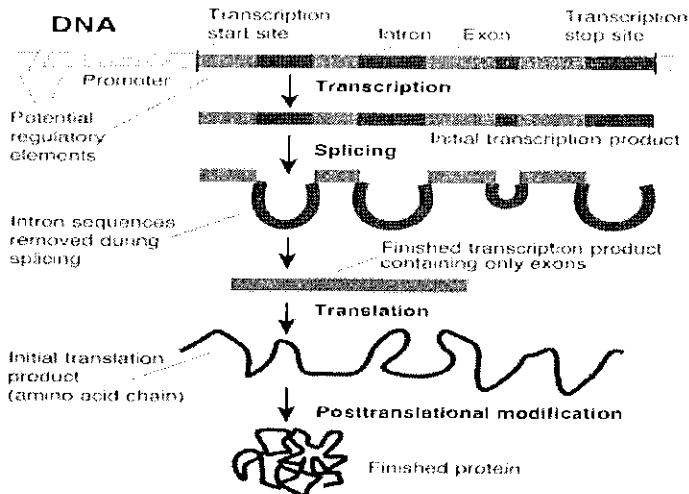
Table 1: The 30 SNPs with the largest absolute posterior means of genetic effect: SNP identification, chromosome locations, and labels

Rank	ID	Chromosome	Label
1	ARS-BFGL-NGS-55820	13	A
2	ARS-BFGL-NGS-107379	14	B
3	Hapmap58253-rs20024365	5	C
4	BFGL-NGS-115886	18	D
5	ARS-BFGL-NGS-35455	10	E
6	ARS-BFGL-NGS-57418	27	F
7	Hapmap10334-BTA-83296	9	G
8	BFGL-NGS-110691	9	H
9	Hapmap60587-rs29022097	18	I
10	ARS-BFGL-NGS-36865	13	J
11	Hapmap38418-BTA-57213	23	K
12	ARS-BFGL-NGS-91706	13	L
13	ARS-USMARC-Parent-ET026985-rs29021607	21	M
14	BFGL-NGS-110460	29	N
15	ARS-BFGL-NGS-11333	9	O
16	Hapmap51010-BTA-65142	29	P
17	Hapmap59281-rs29027629	21	Q
18	Hapmap60560-rs29011138	26	R
19	Hapmap47527-BTA-121897	9	S
20	Hapmap52113-BTA-106820	13	T
21	ARS-BFGL-NGS-11311	15	U
22	ARS-BFGL-NGS-38778	17	V
23	Hapmap31362-DBS11C60614425_1305	10	W
24	ARS-BFGL-NGS-6000	10	X
25	ARS-BFGL-NGS-55314	22	Y
26	BTA-42967-rs843	18	Z
27	ARS-BFGL-NGS-105727	5	a
28	BFGL-NGS-119107	26	b
29	ARS-BFGL-NGS-91238	10	c
30	ARS-BFGL-DAC-36979	18	d

Even if one looks  
 At just 30 SNPs with  
 largest effects, where  
 is the region?



### Reality strikes back: gene structure



**Some genes do not have introns**  
**Some genes are located within introns of other genes**

Arguably, one could do better  
than with linear Bayesian  
(regularized) linear models!

# A VIEW OF LINEAR MODELS (as employed in q. genetics)

Mathematically, can be viewed as a “local” approximation of a complex process

$$f(x) \approx f(a) + f'(a)(x - a) + \underbrace{\frac{f''(a)}{2!}(x - a)^2 + \frac{f'''(a)}{3!}(x - a)^3 + \dots}_{\text{Linear approximation}} + \underbrace{\frac{f^{(n)}(a)}{n!}(x - a)^n + \dots}_{\text{Quadratic approximation}}$$

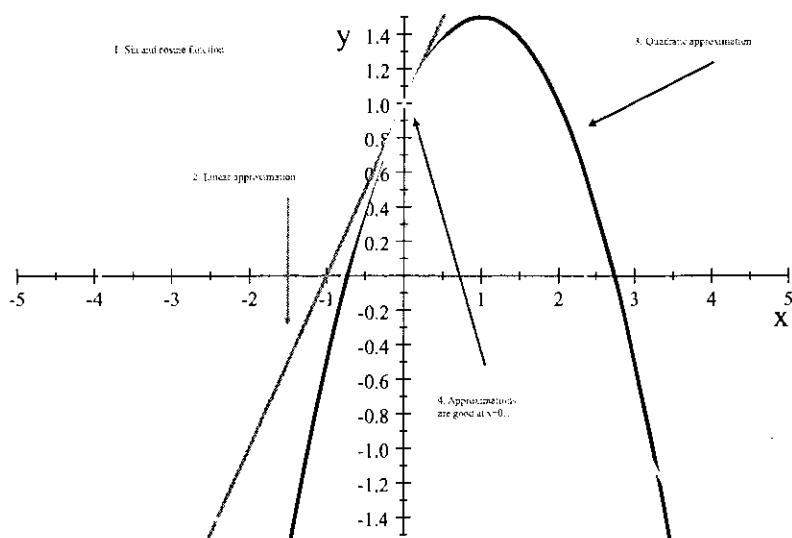
$\vdots$

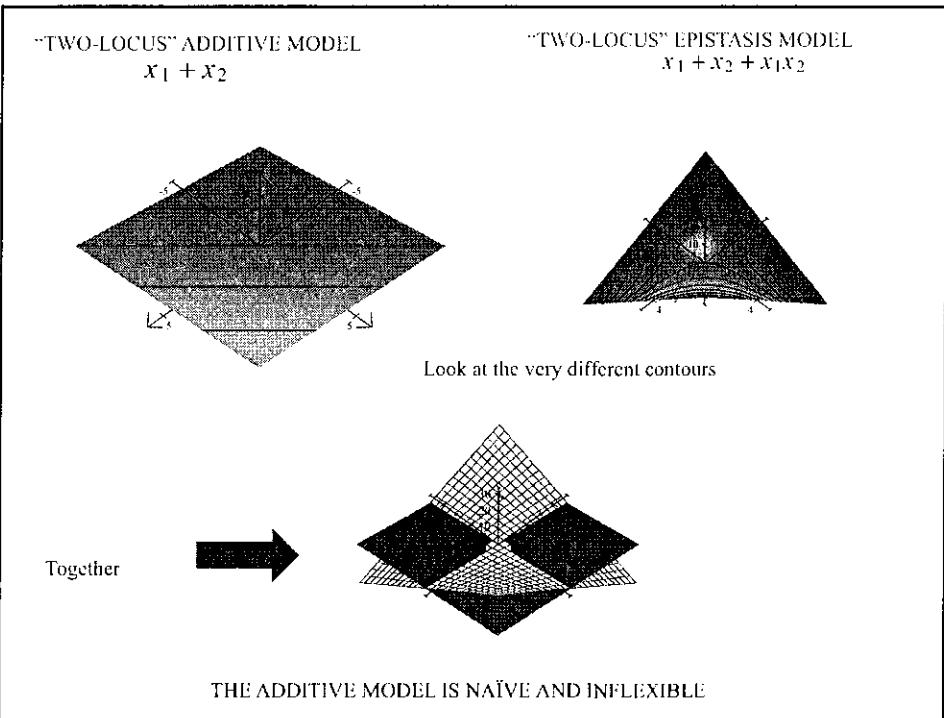
$$\underbrace{\hspace{10em}}_{\text{th order approximation}}$$

FELDMAN and LEWONTIN (1975)  
CHEVALLÉT (1994)

How good are linear and quadratic approximations? A Taylor series provides a local approximation only...

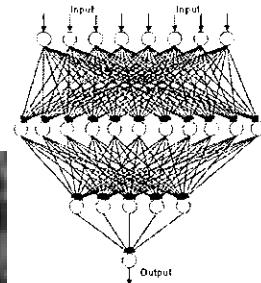
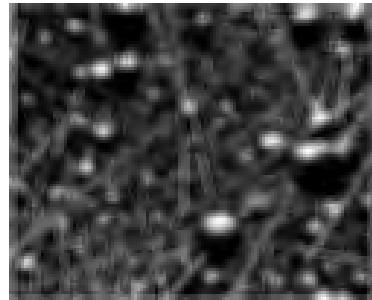
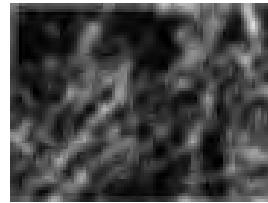
$$y = g(x) + e \quad g(x) = \sin(x) + \cos(x)$$



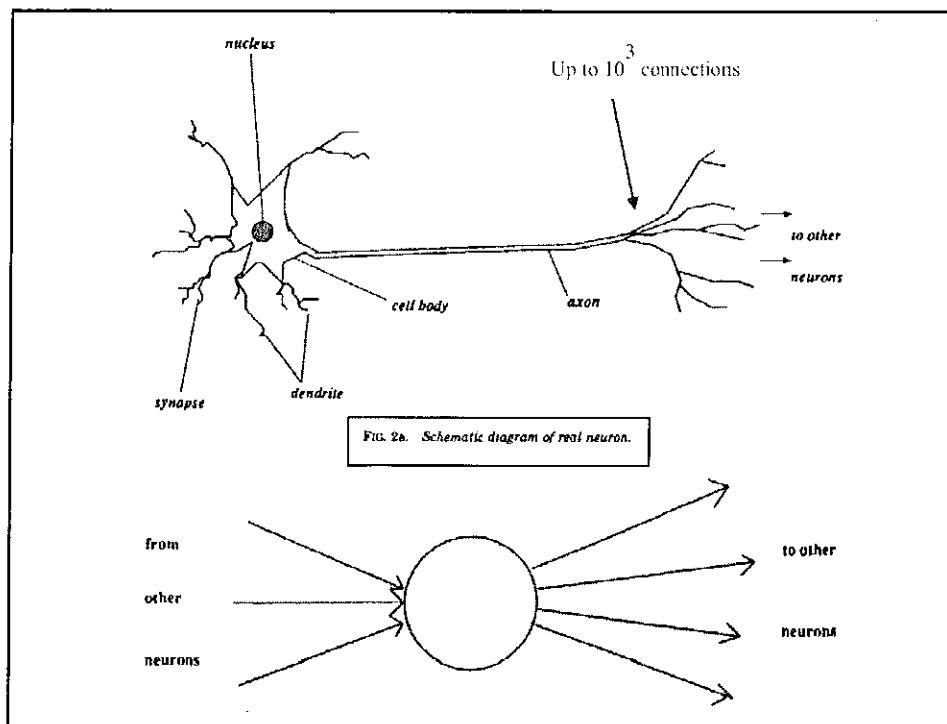


Arguably, one can do better than this

**A perhaps more universal learning machine:  
Regularized Neural Networks**



Why and how neural networks  
entered as approximators of complex  
functions...  
(a non-mathematical argument)



**McCULLOCH, W. S. and PITTS, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5 115–133.**

- Brain superior to von Neumann machines in cognitive tasks
- Microchips: nanoseconds, Brain: milliseconds
- ???

→ Brain recognizes familiar objects from unfamiliar angles  
 → Key: not speed but organization of processing

## Why?

- Tasks distributed over  $10^{12}$  neurons
- Interconnected and activated
- Massively parallel
- Neurons adapt and self-organize
- Interconnectivity: up to  $10^3$  synaptic connections

Can we attempt to emulate the  
brain, mathematically?

## Kolmogorov's Theorem

For any continuous function  $g(x_1, x_2, \dots, x_p)$  of  $p$  variables there exists continuous functions  $h_j$  in  $[0, 1]$  a continuous function  $g$  in  $[0, 1]$  such that

$$g_i(x_{i1}, x_{i2}, \dots, x_{ip}) = \sum_{q=1}^{2p+1} f \left[ \sum_{j=1}^p w_j h_j(x_{i1}, ix_2, \dots, ix_p) \right]$$

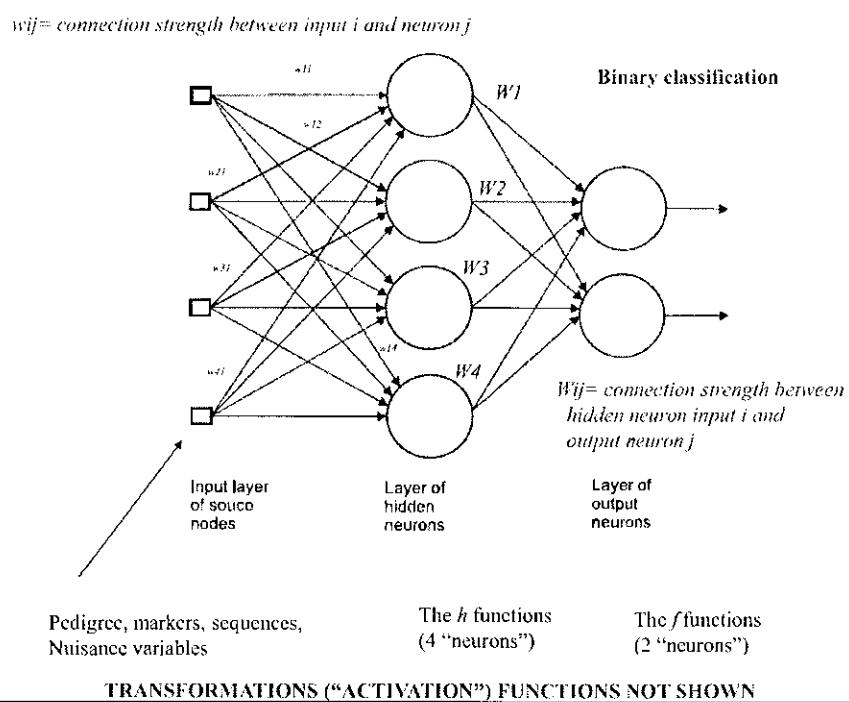
↑  
weights  
Linear or nonlinear transformation  
↑  
Linear or on-linear transformation of inputs

The subscript indicates an evaluation on a given configuration of the input

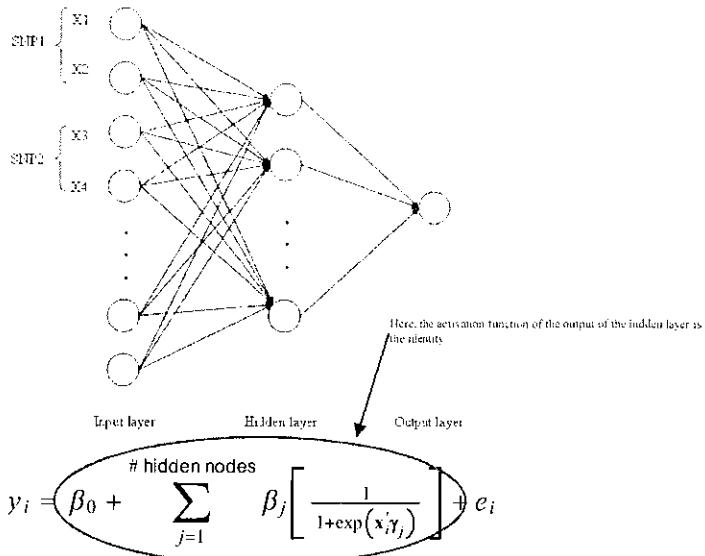
## Comments

- The theorem states that a set of functions exists
- The set includes the possibility of all possible JOINT effects (interactions) among inputs on outputs
- It does not guide on the choice of the functions or on the weights
- With noisy data the idea is to estimate the function from inputs and outputs

# KOLMOGOROV'S THEOREM CAN BE REPRESENTED AS AN ARTIFICIALNEURAL NETWORK



### Continuous output: relationship to non-parametric regression



If # nodes is known (k), the number of parameters is:

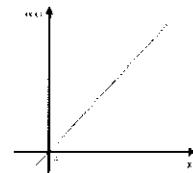
$$1+k+k(1+\# \text{x's})= 1+ k (\# \text{x's} + 2) \quad \rightarrow \quad \text{Can overfit if too many hidden nodes}$$

### Types of transformation (“activation”) functions

Linear



$$g(x) = \alpha x + \beta$$

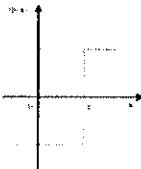


Step



$$y = \pi(x) \Leftrightarrow \begin{cases} 1 & x \geq b \\ 0 & x < b \end{cases}$$

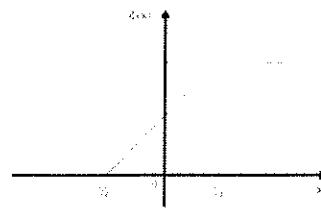
$$\pi(x)$$



b) Step or threshold function

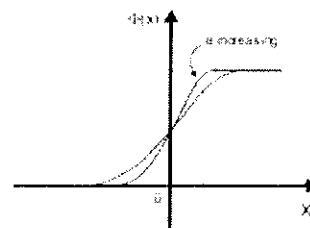
Piece-wise linear

$$y = \text{ReLU}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$



Sigmoid (logistic)

$$y = \sigma(x) = \frac{1}{1 + e^{-ax}}$$



Hyperbolic tangent

$$(e^x - e^{-x}) / (e^x + e^{-x})$$

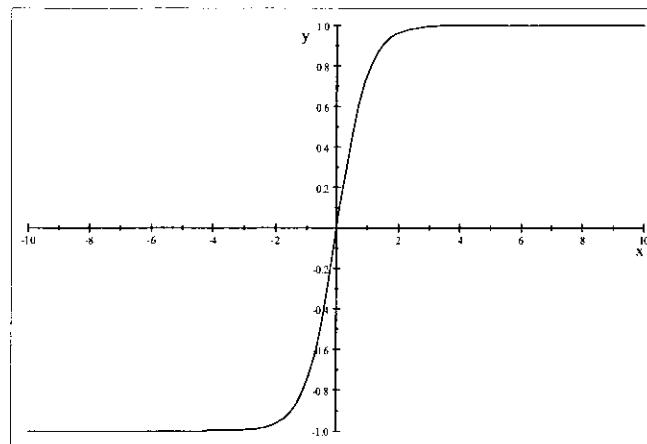


Illustration of a single-neuron model for classification with logistic activation function

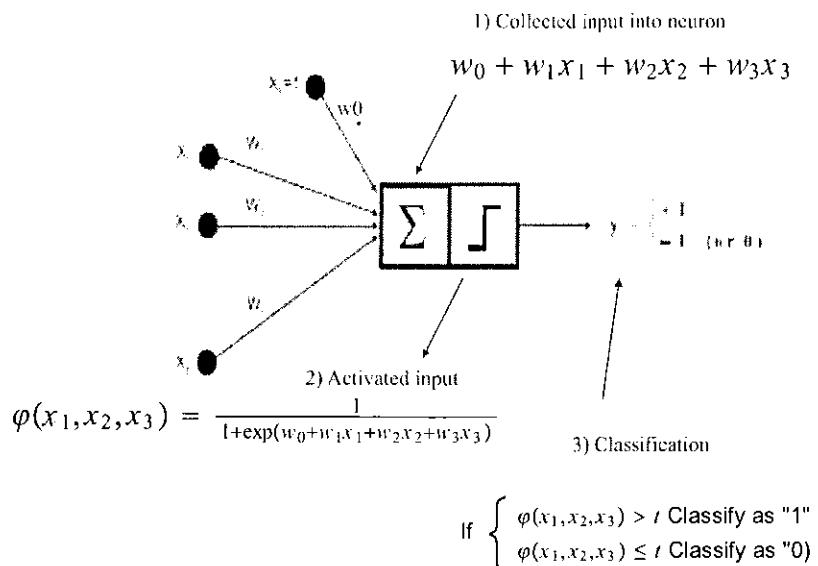
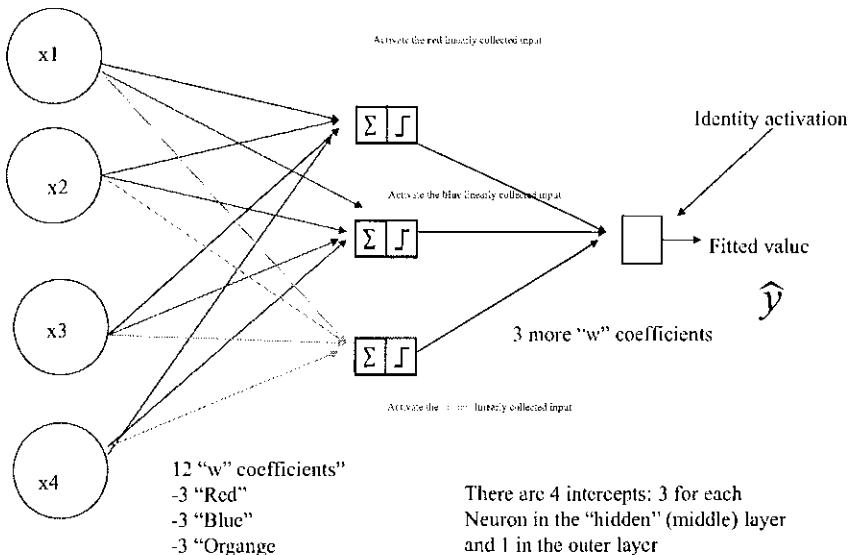


Illustration of a multi-layer model for regression with logistic activation function before emission to the output layer



Algebraically, the model looks like

$$y = \beta_0 + \beta_1 \frac{1}{1 + \exp(w_0^{[1]} + w_1^{[1]}x_1 + w_2^{[1]}x_2 + w_3^{[1]}x_3 + w_4^{[1]}x_4)} \quad \text{RED}$$

$$+ \beta_2 \frac{1}{1 + \exp(w_0^{[2]} + w_1^{[2]}x_1 + w_2^{[2]}x_2 + w_3^{[2]}x_3 + w_4^{[2]}x_4)} \quad \text{BLUE}$$

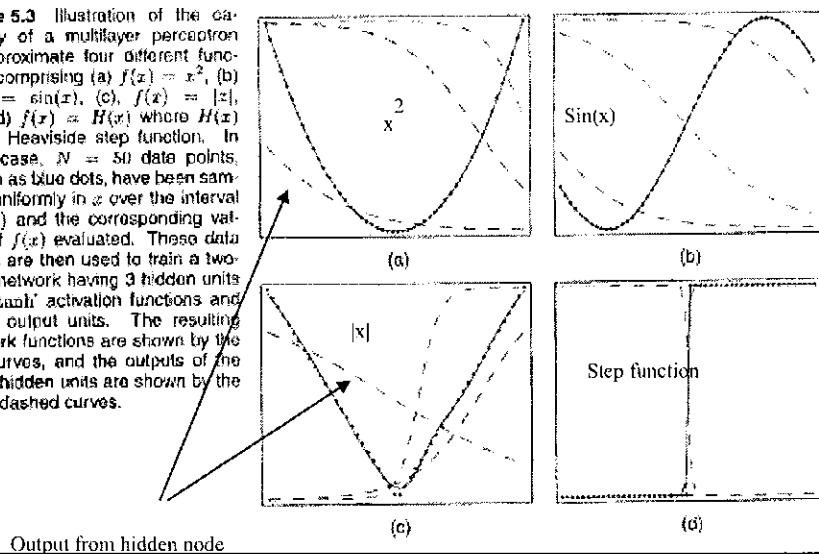
$$+ \beta_3 \frac{1}{1 + \exp(w_0^{[3]} + w_1^{[3]}x_1 + w_2^{[3]}x_2 + w_3^{[3]}x_3 + w_4^{[3]}x_4)} + e$$

4 BETAS + 15 w's = 19 regressions to estimate

### NEURAL NETWORKS ARE UNIVERSAL APPROXIMATORS (Follows from Kolmogorov's Theorem)

50 x values sampled from  $U[-1,1]$  and then evaluate  $f(x)$ . Fit a two-layer NN with 3 hidden nodes and  $\tanh$  activation functions and linear output

**Figure 5.3** Illustration of the capability of a multilayer perceptron to approximate four different functions comprising (a)  $f(x) = x^2$ , (b)  $f(x) = \sin(x)$ , (c),  $f(x) = |x|$ , and (d)  $f(x) = H(x)$  where  $H(x)$  is the Heaviside step function. In each case,  $N = 50$  data points, shown as blue dots, have been sampled uniformly in  $x$  over the interval  $(-1, 1)$  and the corresponding values of  $f(x)$  evaluated. These data points are then used to train a two-layer network having 3 hidden units with 'tanh' activation functions and linear output units. The resulting network functions are shown by the red curves, and the outputs of the three hidden units are shown by the three dashed curves.



THE INFINITESIMAL MODEL AS A REGRESSION  
ON RELATIONSHIPS

$$\begin{aligned}\mathbf{y} &= \mathbf{u} + \mathbf{e} \\ \mathbf{u} &\sim (\mathbf{0}, \mathbf{A}\sigma_a^2) \\ \mathbf{y} &= \mathbf{A}\mathbf{A}^{-1}\mathbf{u} + \mathbf{e} \\ &= \mathbf{A}\mathbf{u}^* + \mathbf{e}\end{aligned}$$

$$y_i = \sum_{j=1}^N a_{ij} u_j^* + e_i$$



Use elements of  
 $\mathbf{A}$  (or  $\mathbf{G}$ ) as inputs  
(covariates) in a regression  
Model with random effects

**Recall**  
 $\mathbf{A} = \mathbf{C}\mathbf{C}'$  (Cholesky)

The infinitesimal model as a regression on a pedigree

$$1) \quad \mathbf{t} = \mathbf{C}\mathbf{z}\sigma_u + \mathbf{e} = \mathbf{C}\mathbf{u}^* + \mathbf{e} \quad \mathbf{u}^* = \mathbf{z}\sigma_u \sim (\mathbf{0}, \mathbf{I}\sigma_u^2)$$

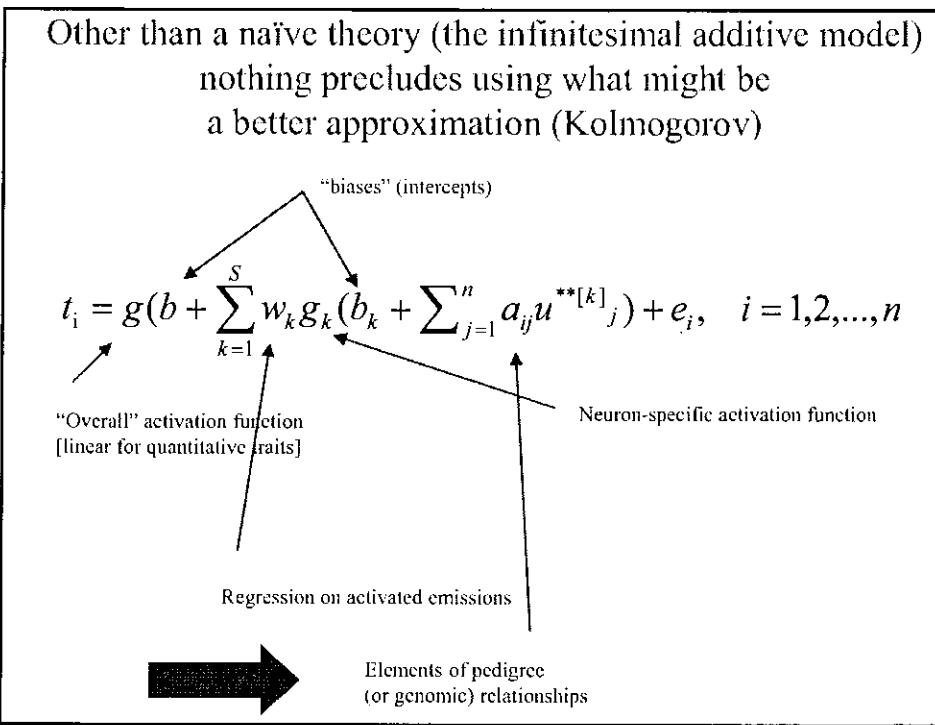
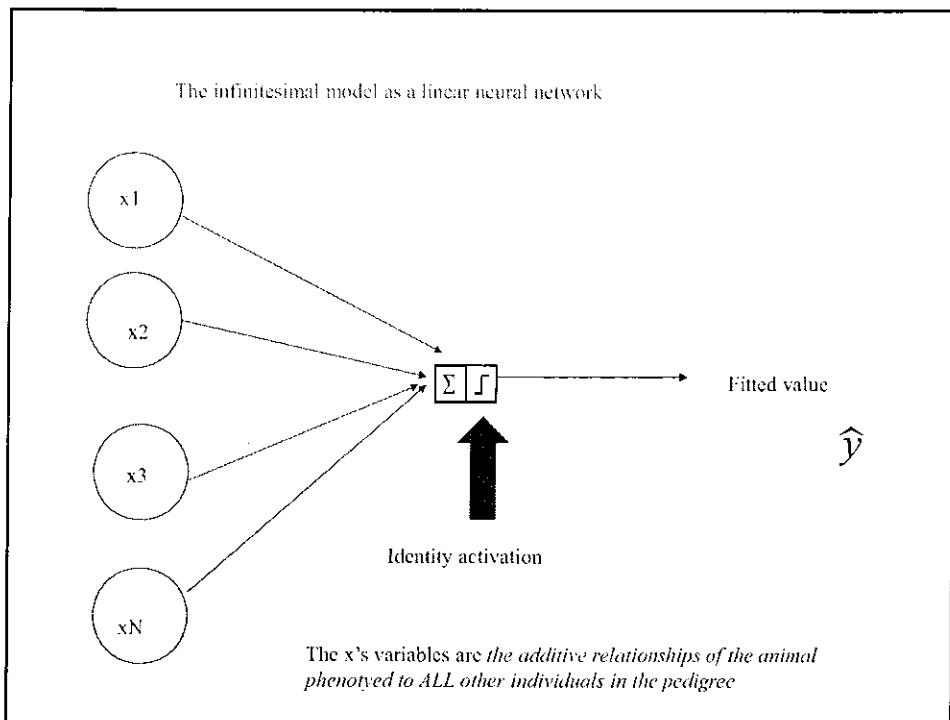
$$t_i = g(\sum_{j=1}^n c_{ij} u_j^*) + e_i, \quad \text{Identity activation}$$

$$2) \quad \mathbf{t} = \mathbf{A}\mathbf{A}^{-1}\mathbf{u} + \mathbf{e} = \mathbf{A}\mathbf{u}^{**} + \mathbf{e}, \quad \mathbf{u}^{**} = \mathbf{A}^{-1}\mathbf{u} \sim (\mathbf{0}, \mathbf{A}^{-1}\sigma_u^2)$$

$$t_i = g(\sum_{j=1}^n a_{ij} u_j^{**}) + e_i, \quad \text{Identity activation}$$

$$3) \quad \mathbf{t} = \mathbf{A}^{-1}\mathbf{A}\mathbf{u} + \mathbf{e} = \mathbf{A}^{-1}\mathbf{u}^{***} + \mathbf{e}, \quad \mathbf{u}^{***} = \mathbf{A}\mathbf{u} \sim (\mathbf{0}, \mathbf{A}^3\sigma_u^2)$$

$$t_i = g(\sum_{j=1}^n a^{ij} u_j^{***}) + e_i, \quad \text{Identity activation}$$



## Bayesian regularization (need to cope with p>>n)

$$p(D | b, \mathbf{w}, \sigma^2, M) = \prod_{i=1}^n N(t_i | b, \mathbf{w}, \sigma^2, M)$$

Likelihood

A network  
Architecture  
(number of neurons  
and activation functions)

$$p(\mathbf{w} | \sigma_w^2) = N(\mathbf{0}, \mathbf{I} \sigma_w^2)$$

(This assumes that all w coefficients are shrunken to the same extent. This is probably not a good assumption, but convenient.)

Conditional posterior

$$P(\mathbf{w} | D, \sigma^2, \sigma_w^2, M) = \frac{P(D | \mathbf{w}, \sigma^2, M) P(\mathbf{w} | \sigma_w^2, M)}{P(D | \sigma^2, \sigma_w^2, M)}$$

Marginal density of the data (used to assess variance components)

$$P(D | \sigma^2, \sigma_w^2, M) = \int P(D | \mathbf{w}, \sigma^2, M) P(\mathbf{w} | \sigma_w^2, M) d\mathbf{w}$$

$$p(D | \sigma^2, \sigma_w^2, M) = \left( \frac{1}{2\pi\sigma^2} \right)^{\frac{n}{2}} \left( \frac{1}{2\pi\sigma_w^2} \right)^{\frac{m}{2}} \times \int \exp \left[ -\frac{1}{2\sigma^2} \sum_{i=1}^n \left( t_i - b - \sum_{k=1}^s w_k g_k(b_k + \sum_{j=1}^n a_{ij} u^{*(k)}_j) \right)^2 - \frac{1}{2\sigma_w^2} \mathbf{w}^T \mathbf{w} \right] d\mathbf{w}$$

Integral not in closed form  
in non-linear networks

$$F(\alpha, \beta) = \beta \sum_{i=1}^n \left( t_i - b - \sum_{k=1}^s w_k g_k(b_k + \sum_{j=1}^n a_{ij} u^{*(k)}_j) \right)^2 + \alpha \mathbf{w}^T \mathbf{w} = \beta E_D + \alpha E_w$$

"penalized" sum of squares

$\frac{1}{2}\sigma^2$        $\frac{1}{2}\sigma_w^2$

## Laplacian approximation yields

Remember Smith and Graser (1986); Graser et al. (1987); Tempelman and Gianola (1993)

$$\log[p(D | \alpha, \beta, M)] \approx K + \frac{n}{2} \log(\beta) + \frac{m}{2} \log(\alpha) - |\beta E_D + \alpha E_w|_{w^{MAP}(\alpha, \beta)} - \frac{1}{2} \log \|H\|_{w^{MAP}(\alpha, \beta)}$$

$$\alpha_{new} = \frac{m}{2(w^{MAP}, w^{MAP} + \text{tr}H_{MAP}^{-1})}$$

Hessian of F

$$\beta_{new} = \frac{n-m+2\alpha_{MAP} \text{tr}H_{MAP}^{-1}}{2 \sum_{i=1}^n \left( t_i - b - \sum_{k=1}^S w_k g_k(b_k + \sum_{j=1}^n a_{ij} u^{**[k]}) + e_i \right)_{MAP}^2}$$

Effective number of parameters

$$\gamma = m - 2\alpha_{MAP} \text{tr}H_{MAP}^{-1}$$

## Data

(297 Jersey cows)

- Target :** Fat Yield Deviation  
Milk Yield Deviation  
Protein Yield Deviation
- Inputs :** Elements of Relationship Matrix  
(Pedigree or Genomic, or both)
- Rationale (again)**



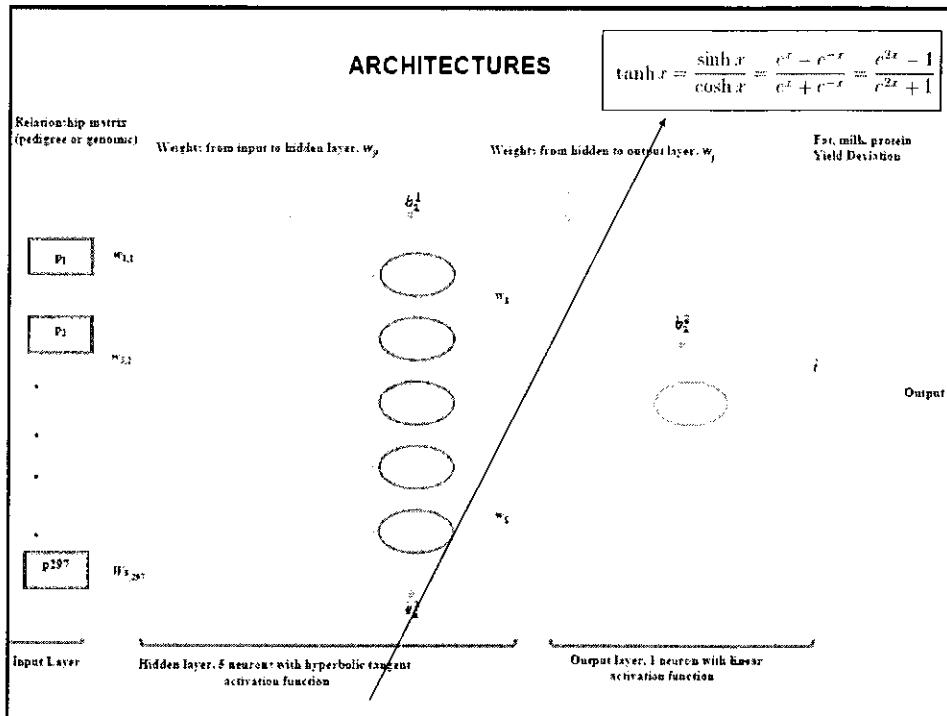
$$\begin{aligned} \mathbf{y} &= \mathbf{u} + \mathbf{e} \\ \mathbf{u} &\sim (\mathbf{0}, \mathbf{A}\sigma_u^2) \\ \mathbf{y} &= \mathbf{A}\mathbf{A}^{-1}\mathbf{u} + \mathbf{e} \\ &= \mathbf{A}\mathbf{u}^* + \mathbf{e} \\ y_i &= \sum_{j=1}^N a_{ij} u_j^* + e_i \end{aligned}$$

Use elements of  
 $\mathbf{A}$  (or  $\mathbf{G}$ ) as inputs in NN

35,798 SNPs used to build  $\mathbf{G}$   
as in Van Raden (2008)

## DATA Descriptive Statistics

Variable	N	Mean	Std Dev	(CV)	Min	Max
Yield_devMilk	297	1513	1821	(120)	-3669	7544
Yield_devFat	297		73	103 (142)	-187	1209
Yield_devProt	297		59	59. (60)	-117	267



Training

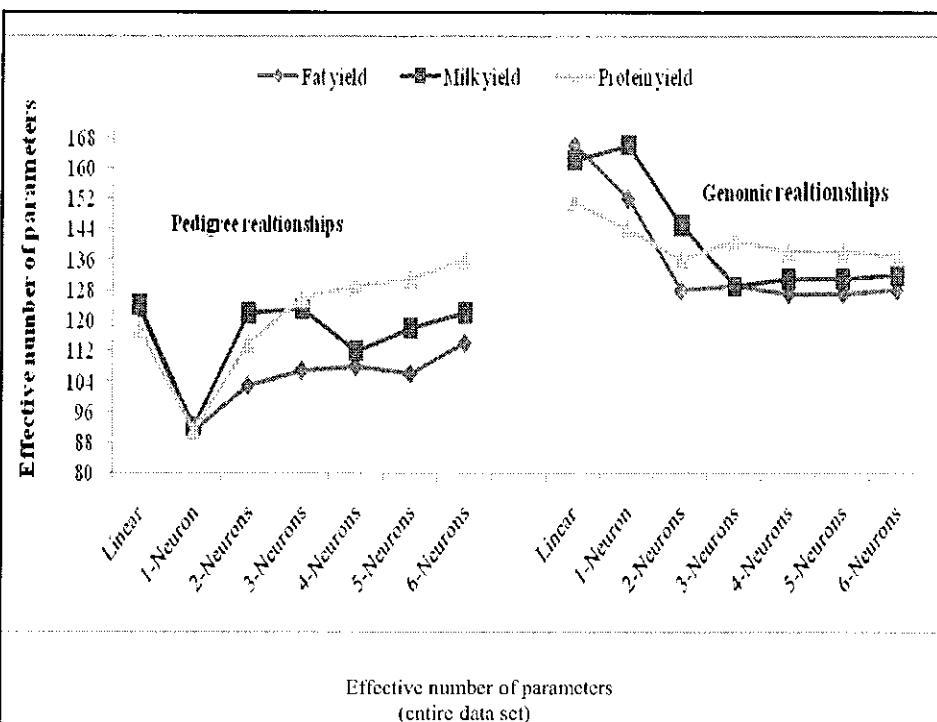
Training

Testing

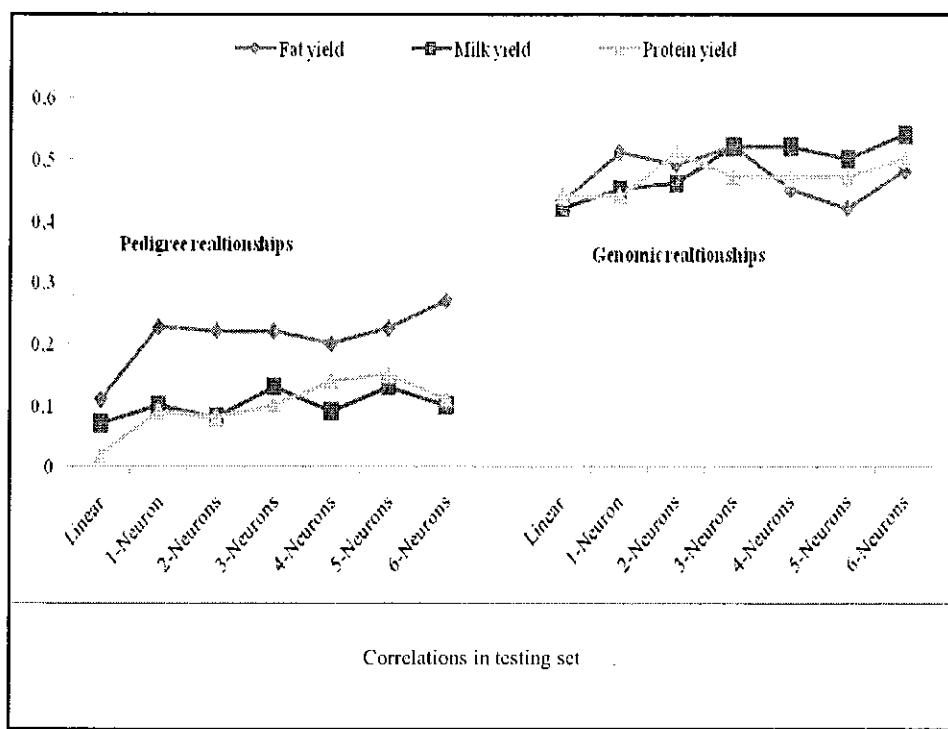
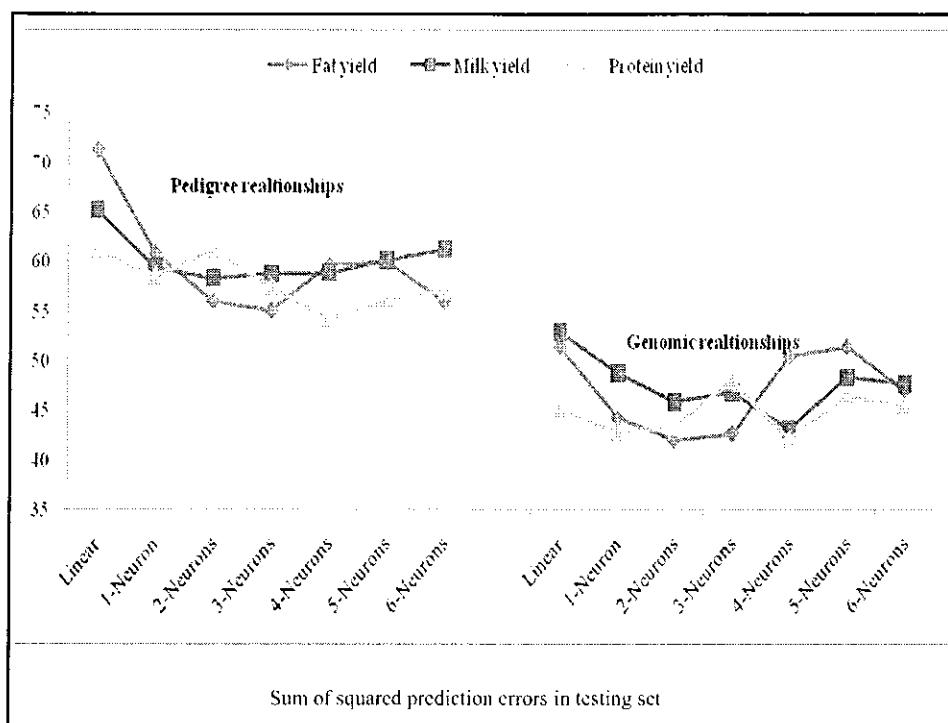
## Fitting the networks (MATLAB)

- TRAINING (60%), and TESTING (20%) sets
- Non-linear regression with Gaussian prior assigned to the weights and Gaussian likelihood
- Given variances, find mode of weights using non-linear optimization method in TRAINING set
- Examine performance in the set
- Predictive performance assessed in TESTING set
- NN with 1 Neuron and linear activation function is “animal model” with unknown variances

Run 25 times (to get more stable results) with random partitions

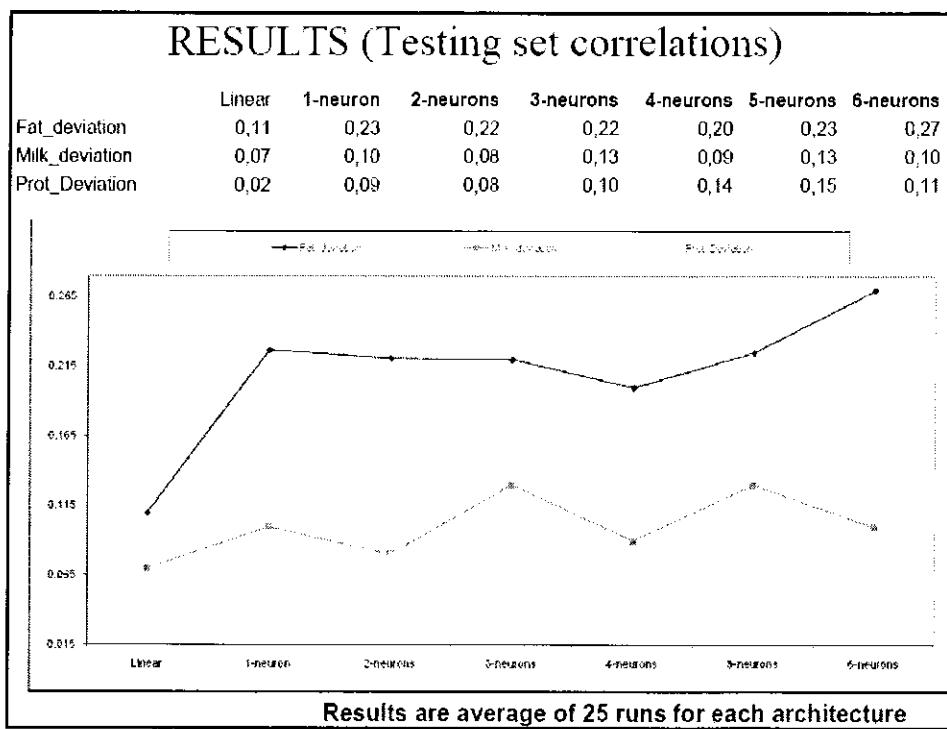


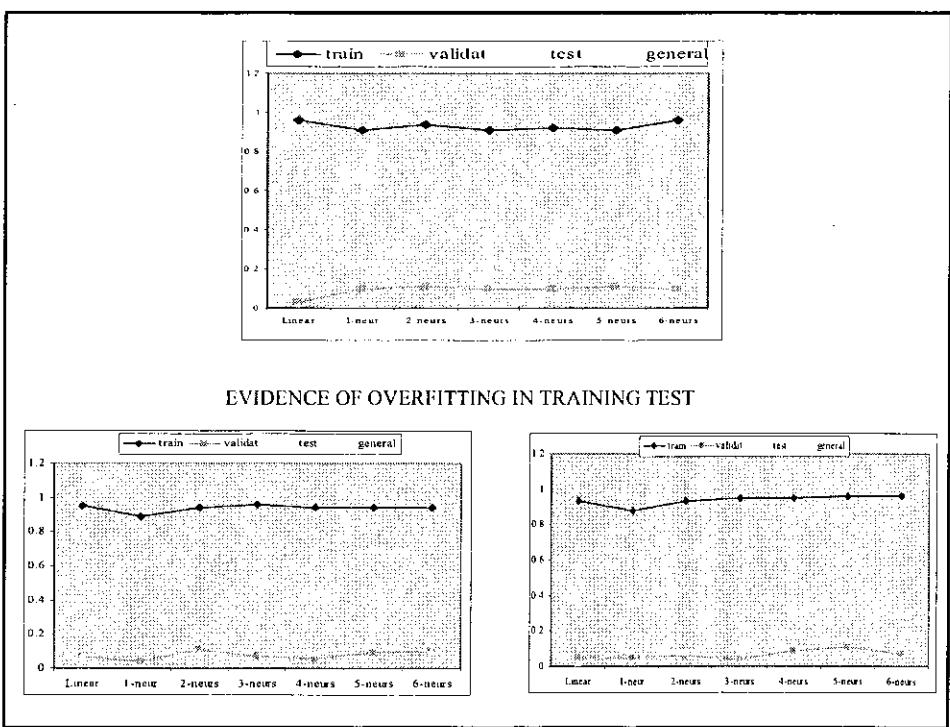
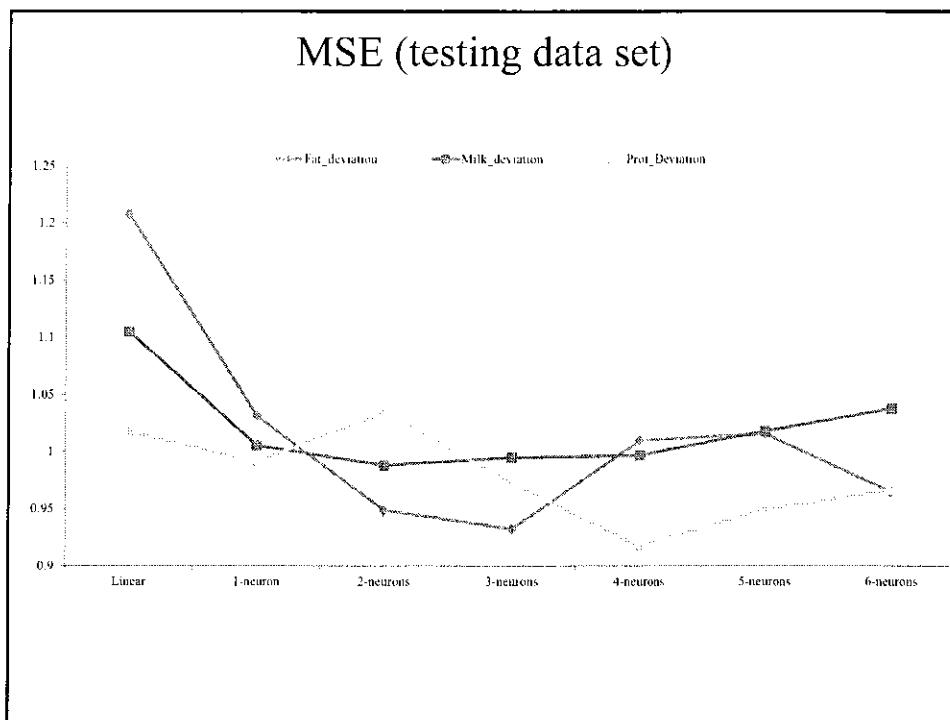
Model complexity is higher ~~not~~ in a than a while they have the same number (297)

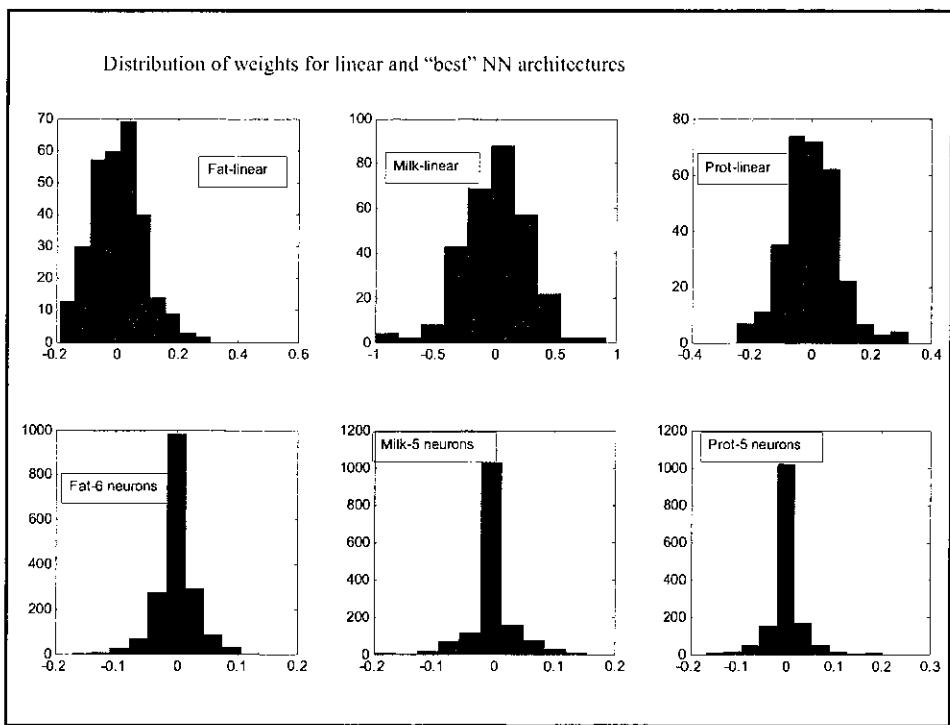
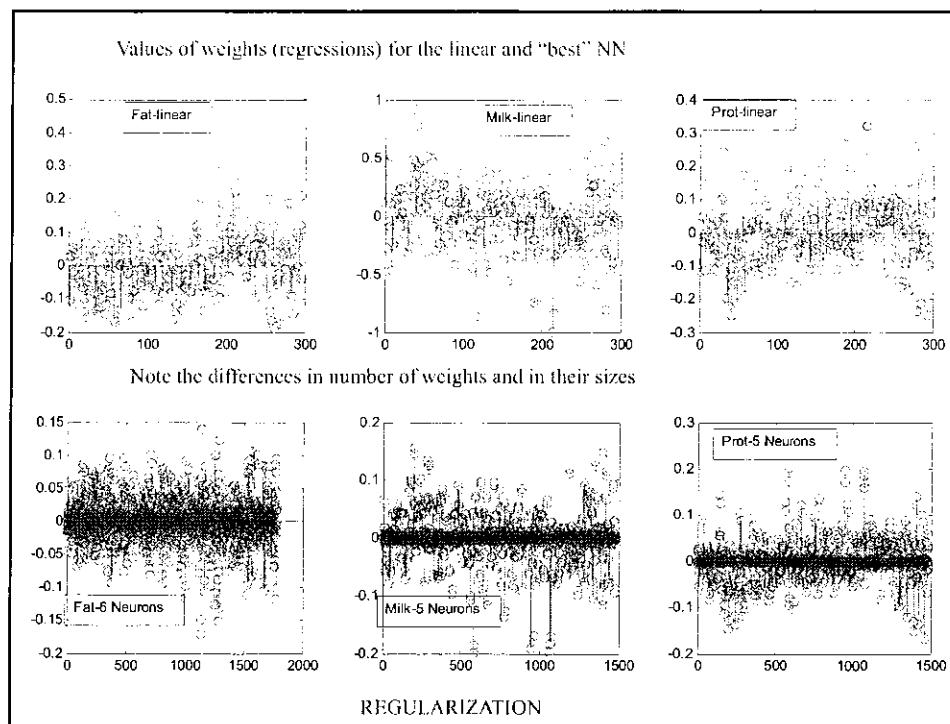


## Illustration of more results

- Using pedigree additive relationships only







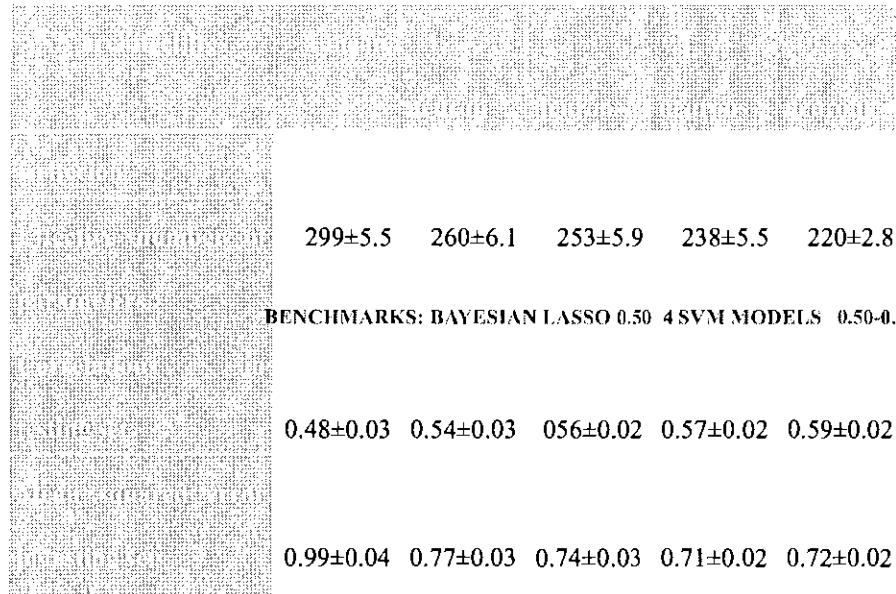
"Total" influence of inputs in neural network

<i>Value of fat yield deviation</i>	<i>Anim_Id</i>	<i>Anim_Id</i>	<i>Value of fat yield deviation</i>
212	264	168	252
212	281	194	241
234	261	278	245
241	194	208	265
245	278	214	187
251	255	257	191
252	168	296	256
256	296	211	304
265	208	255	251
304	211	281	212
308	215	215	308
316	190	190	316

$\nabla \quad I = \frac{\sum_{j=1}^s ABS(w_{ji})}{\sum_{i=1}^R \sum_{j=1}^s ABS(w_{ji})}$

Importance of one element  
in network

WHEAT DATA SET: 599 lines (480 training-119 testing, 50 random repeats)  
1279 binary markers



Long et al. Applied Genetics

ANALYSIS IN PROGRESS BY CROSSA ET AL. (CIMMYT)

Maize corn-flowering	Data used in Crossa et al. (2010)		
Trait-environment	M-BL	M-RKHS	M-RBFNN
SS-ASI	0.5425	<b>0.5926</b>	0.5821
SS-FLF	0.7417	0.6132	<b>0.7460</b>
SS-FLM	0.7404	0.6453	<b>0.7678</b>
WW-ASI	0.5153	<b>0.5580</b>	0.5365
WW-FLF	0.7268	0.5372	<b>0.7869</b>
WW-FLM	0.7428	0.5743	<b>0.7981</b>
SS-GY	0.4743	<b>0.5318</b>	0.5174
WW-GY	<b>0.5634</b>	0.5459	0.5586

Maize  
disease -  
- GLS --  
high  
density  
55k

Sites	M-		
	M-BL	M-RKHS	RBFNN
1	0.2188	0.2099	<b>0.2604</b>
2	0.4174	0.4131	<b>0.4308</b>
3	<b>0.5899</b>	0.5691	0.5823
4	<b>0.5215</b>	0.5044	0.5058
5	0.3419	0.3064	<b>0.3442</b>
6	<b>0.2842</b>	0.2535	0.2775

**Maize under 2 level of drought**  
**-- high density 55k**

Environment	M-BL	M-RKHS	M-RBFNN
GY-Moderate			
drought	0.6333	0.5591	<b>0.6531</b>
GY-Severe			
drought	<b>0.4104</b>	0.3652	0.3910

**Wheat trait 1**

Sites	M-BL	M-RKHS	M-RBFNN
1	0.5969	<b>0.6630</b>	0.6581
2	0.6861	<b>0.7278</b>	0.7069
3	0.6224	<b>0.6943</b>	0.6866
4	0.0673	0.1419	<b>0.1840</b>
5	0.6481	<b>0.6824</b>	0.6744
6	0.3798	<b>0.4659</b>	0.4586
7	0.5984	0.6235	<b>0.6284</b>
8	0.5493	0.6054	<b>0.6100</b>
9	0.5374	0.5821	<b>0.5827</b>
10	0.4775	<b>0.5024</b>	0.4274
11	0.7721	0.7422	<b>0.8039</b>

**Wheat trait2**

<b>Site</b>	<b>M-BL</b>	<b>M-RKHS</b>	<b>M-RBFNN</b>
1	0.4830	<b>0.5216</b>	0.5149
2	0.6928	0.6753	<b>0.7085</b>
3	0.2285	<b>0.3889</b>	0.3827
4	0.4610	0.5508	<b>0.5557</b>
5	0.7509	0.7147	<b>0.7880</b>
6	0.8101	0.8031	<b>0.8399</b>
7	0.4695	<b>0.5374</b>	0.5285
8	0.8345	0.8261	<b>0.8657</b>

**PUNCH LINE:**  
over 35 trials, the winner is...

<b>M-BL</b>	<b>M-RKHS</b>	<b>M-RBFNN</b>
14%	34%	52%
5	12	18

Any concerns about the predictive ability of non-parametric methods,  
relative to those that “help to understand genetic architecture”?

Table 1 Mean correlation of three models: Bayesian LASSO (BL), reproducing kernel Hilbert space (RKHS) regression, and radial basis function neural network (RBNN). The number of times one model has a higher correlation than the other (RKHS vs BL, RBNN vs BL, and RKHS vs RBNN). BL for 10 random partitions for each of 21 individual data sets (site-environment combinations) and across 11 maize datasets.

Trans-environment	Mean correlation			Number of times a model is better than the other		
	BL	RKHS	RBNN	RKHS vs BL	RBNN vs BL	RKHS vs RBNN
Maize data set						
FFL-WW	0.514	0.536	0.534	37	32	34
FFL-SS	0.554	0.763	0.757	30	32	22
MFL-WW	0.817	0.841	0.812	37	32	36
MFL-SS	0.776	0.782	0.780	31	36	27
ASI-WW	0.582	0.586	0.594	37	32	23
ASI-SS	0.612	0.621	0.605	34	23	31
GY-SS	0.326	0.330	0.288	28	13	36
GY-WW	0.557	0.548	0.529	16	13	32
GY-HI	0.633	0.663	0.623	37	37	24
GY-LOW	0.410	0.402	0.393	37	31	30
GLS 1	0.229	0.259	0.260	12	29	21
GLS 2	0.429	0.439	0.431	36	17	35
GLS 3	0.590	0.579	0.552	23	25	22
GLS 4	0.522	0.544	0.506	30	24	20
GLS 5	0.346	0.352	0.344	39	28	23
GLS 6	0.284	0.263	0.275	9	25	18
GLS 7	0.477	0.502	0.508	36	16	35
GLS 8	0.556	0.584	0.592	42	29	31
GLS 9	0.521	0.541	0.506	24	21	26
NCLB 1	0.644	0.709	0.691	49	45	40
NCLB 2	0.475	0.491	0.525	34	36	15
Combined 21 maize trans-environment						
	0.542	0.553	0.547	688	627	616

- FFL: female flowering; MFL: male flowering; ASI: FFL interval; GY: grain yield; SS: severe drought stress; WW: well-watered environment; HI: optimum environment; LOW: poor environment; GLS: *Glucosphaera zeae-maydis*; NCLB: *Excoecaria agri-*

## WHAT ABOUT THE BREEDING VALUE?

1. By network design
2. By math

a) *Infinitesimal model*

$$y_i = \mathbf{z}'_i \mathbf{u} \implies u_i = \mathbf{z}'_i \frac{\partial}{\partial \mathbf{z}_i} (\mathbf{z}'_i \mathbf{u}).$$

b) *Markers model*

$$y_i = \sum_{j=1}^p x_{ij} \beta_j + e_i = \mathbf{x}'_i \boldsymbol{\beta} + e_i.$$

Marked breeding value =

$$\mathbf{x}'_i \frac{\partial}{\partial x_i} (\mathbf{x}'_i \boldsymbol{\beta}) = \mathbf{x}'_i \boldsymbol{\beta}.$$

c) Neural network with hyperbolic tangent activation function throughout

$$t_i = b + c g \left[ \sum_{k=1}^s w_k g_k (b_k - \sum_{j=1}^n p_j u_j^{(k)}) \right] + e_i.$$

$$BV_i = p'_i \frac{\partial}{\partial p_i} t_i = c g' \left[ \sum_{k=1}^s w_k g_k (b_k - \sum_{j=1}^n p_j u_j^{(k)}) \right] p_i \left[ \sum_{k=1}^s w_k g_k' (b_k - \sum_{j=1}^n p_j u_j^{(k)}) \right] u^{(k)}$$

$$g' \left[ \sum_{k=1}^s w_k g_k (b_k - \sum_{j=1}^n p_j u_j^{(k)}) \right] = 4P(1-P),$$

$$P = \frac{\exp \left[ -2 \sum_{k=1}^s w_k g_k (b_k - \sum_{j=1}^n p_j u_j^{(k)}) \right]}{1 + \exp \left[ -2 \sum_{k=1}^s w_k g_k (b_k - \sum_{j=1}^n p_j u_j^{(k)}) \right]},$$

$$\mathbf{u}^{(k)} = \{u_j^{(k)}\},$$

and

$$g_k' (b_k - \sum_{j=1}^n p_j u_j^{(k)}) \mathbf{u}^{(k)} = 4P(1-P).$$

$$P = \frac{\exp \left[ -2(b_k + \sum_{j=1}^n p_j u_j^{(k)}) \right]}{1 + \exp \left[ -2(b_k + \sum_{j=1}^n p_j u_j^{(k)}) \right]}.$$

## WHAT ABOUT THE IMPORTANCE OF A GIVEN SNP?

Joseph, H., Huang, W. L. & Dickman, M. (2003). Neural network modelling of coastal algal blooms. *Ecology Modelling* **159**, 179–201.



*Genet. Res. Camb.* (2011) **93**, pp. 189–201. © Cambridge University Press 2011  
This is a work of the U.S. Government and is not subject to copyright protection in the United States.  
doi:10.1017/S0016672310000662

189

Prediction of body mass index in mice using dense molecular markers and a regularized neural network

---

HAYRETTIN OKUT<sup>1,2\*</sup>, DANIEL GIANOLA<sup>2,3,4</sup>, GUILHERME J. M. ROSA<sup>2,4</sup>  
AND KENT A. WEIGEL<sup>2</sup>

$$I_{SNP_k} = \frac{\sum_{j=1}^S |w_{kj}^{(1,1)}|}{\sum_{j=1}^S \sum_{k=1}^R |w_{kj}^{(1,1)}|} \cdot 100,$$

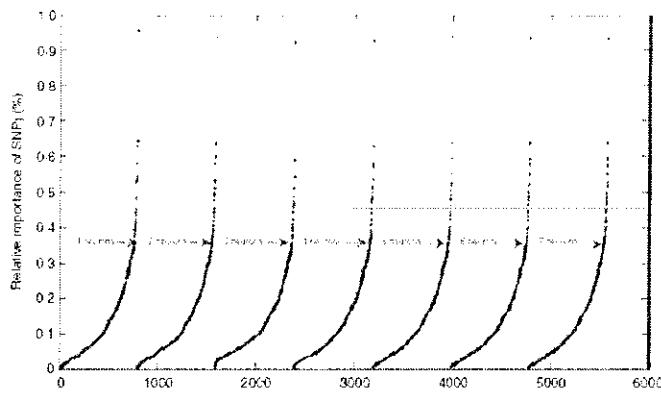
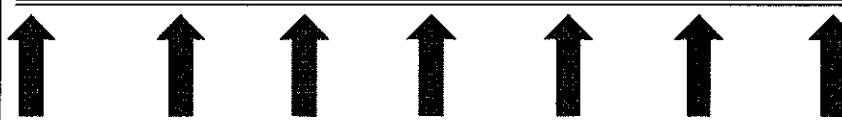


Fig. 6. Plots for the index values of 798 SNPs as prediction of BMI. The solid line gives the cutoff point separating SNPs with index values larger than 0.45%.

Table 2. Relative importance of SNPs with  $I_{SNP}$  values larger than 0.45% for the each of the non-linear networks

7 neurons		6 neurons		5 neurons		4 neurons		3 neurons		2 neurons		1 neuron	
SNP ID	$I_{SNP}$ (%)	SNP ID	$I_{SNP}$ (%)										
420	0.45	7985	0.46	420	0.45	1513	0.45	5010	0.46	4319	0.46	1513	0.45
7985	0.47	5012	0.48	7985	0.46	7985	0.45	4319	0.46	8590	0.48	7955	0.45
5012	0.47	8590	0.48	8590	0.48	348	0.48	10136	0.46	348	0.49	348	0.48
8590	0.48	4319	0.48	5012	0.48	8590	0.48	348	0.47	5012	0.50	8590	0.49
384	0.48	384	0.48	4319	0.48	3891	0.53	10141	0.47	384	0.50	3891	0.53
4319	0.48	5010	0.49	384	0.48	5012	0.53	472	0.48	5010	0.51	5012	0.53
5010	0.49	3891	0.49	5010	0.49	2487	0.53	3891	0.49	3891	0.51	2487	0.53
3891	0.49	472	0.50	3891	0.49	384	0.54	2487	0.51	472	0.53	384	0.54
472	0.50	10136	0.52	472	0.50	10136	0.54	2770	0.54	10136	0.53	10136	0.54
10136	0.52	10141	0.52	10136	0.52	5010	0.54	10961	0.55	2487	0.53	5010	0.54
10141	0.52	348	0.52	348	0.52	472	0.54	12132	0.59	10141	0.53	472	0.54
348	0.53	2487	0.55	10141	0.53	10141	0.55	3978	0.92	10961	0.58	10141	0.55
2487	0.55	2770	0.58	2487	0.55	10961	0.58			2770	0.60	10961	0.58
2770	0.58	10961	0.59	2770	0.58	2770	0.63			12132	0.64	2770	0.63
10961	0.59	12132	0.64	10961	0.59	12132	0.64			3978	0.94	12132	0.64
12132	0.64	3978	0.93	12132	0.64	3978	0.96					3978	0.96
3978	0.93			3978	0.94								



## SUMMARY

- Neural networks: universal approximators
- Need to arrive at suitable architecture (number of layers, number of neurons, choice of activation functions)
- Neural network must be assessed in predictive ability
- Important variables in a network can be detected
- Coefficients do not have obvious interpretation (except in linear networks)
- The infinitesimal model is a naïve network (*Single neuron*)
- The mechanistic value of the additive model is dubious in the face of complexity of biological systems

*The Art of War*

simplified Chinese: 孙子兵法;

traditional Chinese: 孫子兵法;

pinyin: Sunzi Bing Fa

Sun Tzu 孙武  
(722–481 BC)?

"It is said that if you know your enemies and know yourself, you will not be imperiled in a hundred battles.

If you do not know your enemies but do know yourself,  
You will win one and lose one.

If you do not know your enemies nor yourself, you will  
be imperiled in every single battle."

## SOME POSTERIOR THOUGHTS

- Cannot understand complexity ("genetic architecture") with parametric methods
- Prediction is a different ball game from inference
- For prediction, non-parametric methods are almost as good as parametric ones even when assumptions hold
- Do not spend a lot of time inventing priors, or fancy models. A simple additive model may just do well...
- Spend more time in cross-validation and less in simulation. Now there is data!!

## MORE POSTERIOR THOUGHTS

- Markers (and most types of molecular data) have ascertainment problem (Chikhi, 2008): simulations give distorted picture
- SNP assisted genetic evaluation is holding well, and has outperformed (in cross-validation) pedigree BLUP
- There is no universal prediction machine and model performance varies with species, trait and environment

### Comparison among methods in plants (Heslot et al., 2012)

Table 2. Accuracy for each trait and model, average non-cross-validated correlation for each model, and average MSE for each model.

Dataset <sup>†</sup>	Trait <sup>‡</sup>	RR-BLUP <sup>§</sup>	BL	Elastic net	wBSR	BayesCn	E-Bayes	RKHS	SVM	RF	NNET
Barley 1	Yield	0.63	0.65	0.62	0.63	0.63	0.63	0.6	0.43	0.66	0.61
Barley CAP	Betaglucan	0.57	0.57	0.57	0.57	0.57	0.57	0.6	0.35	0.55	0.64
Bay x Sha (Bay-0 x FLOSD)	DM10	0.92	0.82	0.83	0.89	0.82	0.82	0.83	0.8	0.95	0.92
Shaharai	DM10	0.63	0.63	0.63	0.64	0.63	0.63	0.64	0.56	0.57	0.56
	DM3	0.4	0.39	0.40	0.4	0.39	0.4	0.41	0.33	0.32	0.35
Panel maze	Moisture	0.75	0.75	0.75	0.76	0.75	0.73	0.79	0.45	0.73	0.73
	Yield	0.63	0.63	0.61	0.62	0.63	0.60	0.64	0.32	0.6	0.60
Digital maze	Moisture	0.74	0.74	0.72	0.72	0.74	0.70	0.76	0.56	0.61	0.72
	Yield	0.52	0.52	0.49	0.51	0.62	0.51	0.5	0.29	0.43	0.48
Wheat CIMMYT	YLD1	0.61	0.5	0.46	0.48	0.51	0.49	0.59	0.36	0.52	0.54
	YLD2	0.5	0.49	0.46	0.5	0.5	0.46	0.52	0.36	0.43	0.51
	YLD4	0.38	0.37	0.36	0.36	0.38	0.36	0.43	0.39	0.38	0.43
	YLD6	0.44	0.47	0.42	0.47	0.44	0.39	0.52	0.27	0.46	0.44
Wheat Cornell	Yield	0.36	0.36	0.37	0.37	0.34	0.28	0.29	0.22	0.39	0.30
	Height	0.45	0.44	0.41	0.44	0.44	0.41	0.35	0.37	0.46	0.45
Wheat durum	Height	0.64	0.66	0.68	0.67	0.66	0.67	0.78	0.54	0.62	0.67
	TKW	0.6	0.57	0.60	0.6	0.59	0.59	0.68	0.41	0.64	0.65
	Yield	0.63	0.62	0.61	0.62	0.63	0.51	0.59	0.39	0.52	0.57
Average accuracy (cross-validated)		0.56	0.56	0.54	0.56	0.56	0.54	0.59	0.41	0.54	0.55
Average non-cross-validated correlation		0.77	0.79	0.76	0.77	0.77	0.93	0.93	0.89	0.76	0.85
Average MSE		0.67	0.67	0.63	0.68	0.68	0.76	0.64	1.56	0.72	1.64

<sup>†</sup>Barley 1: Limagrain Europe, Rives, France; Barley CAP (Barley Coordinated Agricultural Project, 2012); Bay Sha (Leudel et al., 2002); Panel maze: Limagrain Europe; Wheat CIMMYT (Crossa et al., 2010); Wheat Cornell (Jaffee et al., 2011); Wheat durum: Limagrain Europe.  
<sup>‡</sup>Betaglucan: betaglucan content; FLOSD: flowering time in short days; DM10: dry matter in non-limiting N conditions; D65: dry matter in limiting N conditions; YLD1 to YLD6 refers to the yield traits reported in Crossa et al. (2010). TKW: thousand kernel weight.

"It ain't what you don't know that gets you into trouble.  
It's what you know for sure that just ain't so."

(Mark Twain)

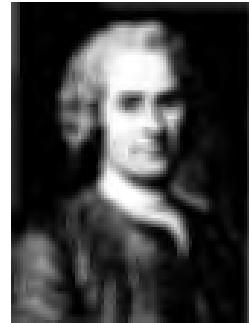


Takezawa (2005):

Model-free procedures can have better predictive performance even if the "true" model is used to generate and then fitted to the data.

## ROUSSEAU ON THE ADDITIVE GENETIC MODEL

*"...de nier ce que est, et d'expliquer ce qui n'est pas..."*  
*Rousseau "Nouvelle Héloïse"*



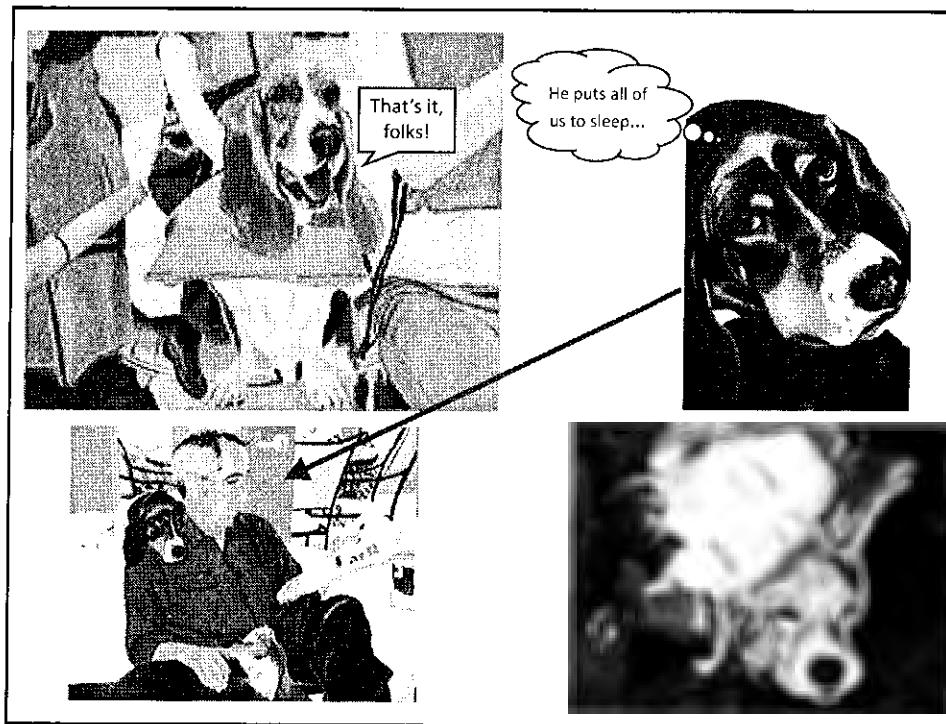
Geneve 1712- Ermenonville 1778

"Would you refuse your dinner because you do not understand the digestive system?"

quote by British mathematician in  
"The emperor of the maladies: a biography of cancer", 2010, by  
Siddhartha Mukherjee

## Conclusions

- Challenges to parametric methods posed by genomic and post-genomic data
- Future: Shift in paradigm. Semi-parametric and "machine learning" type techniques?



# A Quick Introduction to R

Gustavo de los Campos & Christine Duarte<sup>1</sup>

## Contents

1. Introduction .....	2
2. Installing R.....	2
3. The R console .....	3
4. Variable type .....	6
5. Data frames.....	7
6. Libraries.....	8
7. Listing and removing objects from the environment .....	9
8. Reading and Writing ASCII Data.....	10
9. Univariate descriptive statistics .....	10
10. Bi-variate descriptive statistics .....	11
11. Ordinary least squares regression: linear model .....	13
12. Generalized Linear Models .....	14
13. Loops and conditional statements.....	14
14. Monte Carlo Methods.....	16
15. Functions.....	19

---

<sup>1</sup> Comments to an earlier version of the handout by Kamil Suliveres are gratefully acknowledged.

## 1. Introduction

The following text (from the R-website [www.r-project.org](http://www.r-project.org) ) briefly describes R:

*R is a language and environment for statistical computing and graphics.*

*R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible.*

*R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.*

The R-package, its libraries and manuals can be downloaded from: <http://www.r-project.org/> .

## 2. Installing R

The R-package can be downloaded following these steps:

- go to [www.r-project.org](http://www.r-project.org)
- in the left side follow link 'CRAN'
- choose a repository
- In the box 'Download and Installing R' choose your operating system  
(Here I follow Windows, to illustrate)
- Follow link 'base'
- Download R from link [Download R 2.11.1 for Windows](#)
- Run the executable file, it will guide you through installation

### 3. The R console

**Assignments and simple operations.** The following box provides simple examples which illustrate how to create numeric variables in R and how to perform simple operations with these variables. The symbol "`<-`" is an assignment operator, it assigns whatever is at the tail of the arrow to the variable whose name is provided at the end of the arrow. The symbol "`#`" is used for commenting lines.

#### ## Example 3.1 ##

```
## ASSINGMENTS
x<-2
y<-3

## SIMPLE OPERATIONS
z<-y+x # try - * / for subtraction, product and division
z
```

**Numeric vectors.** The following example illustrate how to create a numeric vector using the "`c()`" function which concatenates elements into a vector. Once you created a vector, you can modify or access any entry of it by indicating the position you want to access or modify in between square brackets. Note, NA is used to denote missing values in R.

#### ## Example 3.2 ##

```
# Creates a vector
x<-c(10,20,30,40,50) # equivalently, try x<-1:5
str(x)
# Accessing elements of a vector
x[2]
x[c(1,4)]
x[-c(1,4)]

# Modifying elements of a vector
x[c(1,3)]<-NA
x

# Creating a sequence using seq
x<-seq(from=1,to=10,by=2)
y<-seq(from=20,to=1,length=5)
x
y

# Operations with vectors
x*2
x+y
x*y
```

**Matrices.** The following example shows how to create a matrix by binding columns, `cbind()`, or rows, `rbind()`. The functions `dim()`, `nrow()` and `ncol()` give you the dimensions, number of rows and number of columns of a matrix.

### ## Example 3.3 ##

```
# Creates a matrix by binding columns
x1<-1:10
x2<-11:20
x3<-21:30
X<-cbind(x1,x2,x3)
dim(X)
nrow(X)
ncol(X)

# Creates a matrix by binding rows
Z<-rbind(x1,x2,x3)
dim(Z)
nrow(Z)
ncol(Z)
```

Matrices can also be created using the `matrix()` function. To create a matrix we need to provide to `matrix()` the number of rows, number of columns and a vector containing the data that will be used to form the matrix. By default, the `matrix()` function assumes that data is sorted by column. To obtain more information about this or any other R-function type `help(functionname)`, e.g., `help(matrix)`.

### ## Example 3.4 ##

```
# Creates a matrix
X<-matrix(nrow=3,ncol=10,data=1:30)
X

Y<-matrix(nrow=3,ncol=10,data=1:30,byrow=TRUE)
Y
```

***Indexing with matrices.*** We can modify, or extract elements of a matrix using indexing. The following examples illustrate how to extract/modify single elements, columns, rows and blocks of a matrix.

## ## Example 3.5 ##

```
# Creates a matrix
x1<-1:10
x2<-11:20
x3<-21:30
X<-cbind(x1,x2,x3)

# extracting elements using indexing
X[1,1] # single element
X[1,] # entire row
X[,1] # entire column
X[c(1,2),c(2,3)] # block

# modifying elements of a matrix
X[c(1,2),c(2,3)]<-NA
X
```

***Operations with matrices.*** We can use R to perform matrix operations. Cell by cell operations can be performed using the standard symbols. To perform matrix operations we need to ‘enclose’ the symbols between “%”. The following example illustrates this.

## ## Example 3.6 ##

```
# Creates two matrices
X<-matrix(nrow=4,ncol=2,data=1:8)
Y<-matrix(nrow=4,ncol=2,data=1:8)
# Element by element operations
X+Y
X*Y
# Transpose
Z<-t(Y)
# Matrix product
X%*%Z
```

```

## Example 3.7 ##

# Creates an identity matrix
D<-diag(3)

# Adds 0.5 to the off-diagonal
D[1,c(2,3)]<-0.5
D[c(2,3),1]<-0.5
D[2,3]<-D[3,2]<-0.5
D

# Computes the inverse of D
DInv<-solve(D)

# Checks properties of the inverse

D%*%DInv
DInv%*%D

```

## 4. Variable type

So far we have used real variables only. R has 5 basic types of variables: characters, these are simply labels; integers; numeric (real); logical (TRUE/FALSE); and factors, these are variables that can take on a given set of values (the levels of the factor) which may be ordered (e.g., 'low', 'medium', 'high') or not (e.g., 'blue', 'green', 'red'). The following example illustrates how to create variables of each of the above-mentioned types. You could also create matrices of each of these types. The function `str()` gives you the structure of an R-object, in this case the variable type and dimensions of the array.

### ## Example 4.1 ##

```

# integers
x<-1:10
str(x)

# numeric
y<-x/1.15
str(y)

# logical

z<-x>5
str(z)
z

```

## ## Example 4.2 ##

```
# character
w<-c("hello", "myName", "red")
str(w)

# un-ordered factor
w<-factor(x=c("red", "blue", "red", "green"))
str(w)

# ordered factor
w<-factor(x=c("low", "medium", "high", "high"),
           ordered=TRUE, levels=c("low", "medium", "high"))
str(w)
levels(w)
is.ordered(w)

# as.factor() as.numeric(), as.integer() ,.. etc.
# can be used to coerce variables into a different type.
```

## 5. Data frames

One limitation of matrices is that all columns and rows must be of the same type. Commonly in our dataset we may have variables of different types, e.g., some factors (e.g. sex), some integers (e.g., 0/1/2 to code SNP genotypes), some characters (e.g., name). Data-frames allow you to have variables of different type within one matrix-type array. The following example creates a data frame.

```
## Example 5.1  ##
x<-as.factor(c("low", "high", "medium", "low"))
y<-c(1.1, 2.4, 3.1, 0.5)

myData<-data.frame( treatment=x, outcome=y)
str(myData)
myData
```

Indexing can be used in data frames in the same way as in matrices. Additionally, you can access individual columns using the dollar sign and the variable name. The following example illustrates this.

```
## Example 5.2 ##  
myData[c(1,2),]  
myData[,1]  
myData$treatment  
myData$outcome
```

## 6. Libraries

Specialized algorithms are provided in R through libraries. Some libraries are included in the basic installation package, but others need to be downloaded from CRAN. You can load these libraries into an R-session by using the `library()` function. Once the library is loaded, all the functions included on it become available in the environment.

To illustrate, let's consider the library MASS. This function has a function, `ginv()`, which computes generalized inverses of matrices. If you type `help(ginv)` in R without loading MASS you will get the following error message:

### ## Example 6.1 ##

```
help(ginv)  
No documentation for 'ginv' in specified packages and libraries:  
you could try '??ginv'
```

Now try the following:

### ## Example 6.2 ##

```
library(MASS)  
help(ginv)
```

To install libraries available through CRAN go to the main menu and choose: Packages/Set Cran Mirror and choose one repository (e.g., USA IA, which is a repository from Iowa State University).

Now go to the option Packages/Install packages and choose one package (e.g., `accuracy`). You should get the following message:

```
package 'accuracy' successfully unpacked and MD5 sums checked
```

## 7. Listing and removing objects from the environment

You can list the objects available in the working environment using the functions `ls()` or `objects()`. The function `rm()` can be used to remove an object (or a list of objects) from the environment. You can quit R by closing the console or by typing `quit()`. Use `quit(save='yes')`, `quit(save='no')`, to quit saving or without saving the environment, respectively. The following example illustrate these functions.

```
## Example 7.1 ##

ls() #list the objects in the environment
rm(list=ls()) # cleans the environment
ls()

# Now let's create an object
x<-1:10
ls()

quit(save='yes')

# Now open R and type ls()
```

## 8. Reading and Writing ASCII Data

The functions `read.table()` and `write.table()` can be used to read and write data in table-format. In the following example we first load a data frame (`oats`) available in the `MASS` package and then write it to the hard drive as an ASCII file and read the data again back into the R session.

### ## Example 8.1 ##

```
rm(list=ls())
library(MASS)
data(oats)
str(oats)      # shows the structure of an object
fix(oats)      # displays data (can also modify/edit/create variables)

# writing data to hard drive
write.table(x=oats,file='oats.txt', sep='') # space-delimited
write.table(x=oats,file='oats.csv', sep=',') # comma-delimited

# reading data
myData2<-read.table('oats.txt', sep='',header=TRUE)
myData3<-read.table('oats.csv', sep=',',header=TRUE)

head(myData2)
head(myData3)
```

## 9. Univariate descriptive statistics

We will look at descriptive statistics for variables in the `oats` data set to illustrate some useful functions in R. For discrete variables, the `table` function is useful (\$ operator references a certain variable in a data frame). The `summary` function will also produce descriptive statistics. The output produced by `summary()` depends on the nature of the object , below you have examples of `summary()` for vectors and data frames.

### ## Example 9.1 ##

```
table(oats$B) # look at counts for block variable
table(oats$V) # look at counts for variety variable
summary(oats$V) # can also use summary function
summary(oats)
```

For continuous variables, we will use the mean, standard deviation (`sd`), variance (`var`), quantile, and histogram functions.

```

## Example 9.2 ##
mean(oats$Y) # mean of yield
sd(oats$Y) # sd of yield
var(oats$Y) # var of yield
quantile(oats$Y) # quantiles of yield
quantile(oats$Y,probs=c(0,0.05,0.1,0.9,0.95,1)) # specify probs
help(quantile) # use help statement to get function parameters
summary(oats$Y) # summary statement for a continuous variable
hist(oats$Y) # histogram
print(hist(oats$Y)) # use print to display numeric features of hist

```

## 10. Bi-variate descriptive statistics

In the previous example we described features of the marginal distribution of a random variable (RV). Now we turn into description of the bi-variate distribution of two RVs. First we look at an example of two discrete RVs from the oats dataset.

```

## Example 10.1 ##
# Contingency tables for two discrete RV #####
table(oats$V,oats$B) # table for bivariate discrete stats
xtabs(~oats$V + oats$B) # or using xtabs and specifying a formula

highYield <- oats$Y>median(oats$Y) # binary high yield
table(oats$V,highYield) # counts of high-yielding plots by variety

```

Now we describe the association between two continuous RVs (Gas=gas consumption, and Temp=Temperature, of the whiteside dataset, also available with the MASS package) using the covariance, correlation and plot functions.

```

## Example 10.2 ##
# Two continuous RVs #####
library(MASS)
data(whiteside)
head(whiteside)

# variance-covariance matrix
var(whiteside[,2:3])

# correlation matrix
cor(whiteside[,2:3])

# or scatter plot for visualization
plot( Gas~Temp,data=whiteside)

```

Now, let's look at an example of one continuous (Gas) and one discrete RV (Ins=Insulation) also from the whiteside dataset. In this case we describe the joint distribution by first calculating the conditional mean gas consumption given insulation, and using a box-plot, which provides quantiles of a continuous RV by level of the discrete RV.

```

## Example 10.3 ##
# One continuous versus one discrete RV #####
# Conditional mean
tapply(FUN=mean, X=whiteside$Gas, INDEX=whiteside$Insul) # try FUN=sd

# boxplot, which displays several quantiles
boxplot(Gas~Insul,col='red',data=whiteside)

```

Finally, let's use graphical methods to describe features of the joint distribution of two continuous RV given a third discrete RV. The following code (taken from the documentation available with the whiteside dataset) generates a scatter plot of gas consumption versus temperature by insulation. To this end we use the xyplot() function of the R-package lattice.

```

## Example 10.4 ##
# Scatter of gas consumption versus temperature by insulation
library(lattice) # these commands make a nice plot of the data
xyplot(Gas ~ Temp | Insul, data=whiteside)

```

## 11. Ordinary least squares regression: linear model

We will use the whiteside data set to illustrate ordinary least squares using the `lm()` function of R. We begin by regressing gas consumption on temperature and insulation, using an additive model.

```
## Example 11.1 ##  
gasH0<-lm(Gas~Temp+Insul, data=whiteside)  
summary(gash0)
```

The scatter plot of gas versus temperature by insulation suggested that the effect of temperature on gas consumption depended on insulation. This suggests that we should expand the additive model (`Gas0`) with inclusion of an interaction between temperature and insulation. This is done in the next example.

```
## Example 11.2 ##  
gasHA<-lm(Gas~ Temp+Insul+Temp*Insul-1, data=whiteside)  
summary(gashA)
```

We can now compare the above models using the `anova()` function, which will make an F test between nested models. The test has only 1-df (the interaction term) and the p-value is small suggesting we should reject the null hypothesis (the additive model, in this case) in favor of the alternative.

```
## Example 11.3 ##  
anova(gash0,gashA)
```

The following code give examples of how to extract elements of the fitted model and how to obtain some diagnostics.

```
## Example 11.4 ##  
names(gashA) # these are attributes available in the lm object  
gashA$coef # print out the model coefficients  
coef(gashA) # same thing using extractor function coef  
  
## diagnostic plots  
plot(gashA)
```

## 12. Generalized Linear Models

The lm function can be used for regression with continuous response variables. Let's now look at how to fit a logistic regression to a discrete response using the glm() function. This function fits a generalized linear model using least squares.

```
## Example 12.1 ##

# Here we discretize gas consumption and append it to whiteside
threshold<- median(whiteside$Gas)
whiteside$GasHi <- ifelse(whiteside$Gas < threshold, 0, 1)

xtabs(~GasHi+Insul, data=whiteside)

gasHA_logReg<- glm(GasHi~Temp+Insul+Temp*Insul-1, data=whiteside,
                     family='binomial')
summary(gasHA_logReg)
```

## 13. Loops and conditional statements

Loops are used to repeat tasks. For example, the for loop in R can be used to run a task over a pre-defined index set. Here we have two simple examples.

```
## Example 13.1 ##

for(i in 1:10 ){
  print(i)
}

# note that the index set does not need to be a sequence
tmp<- c('aaa', 'z', 'bye')

for( x in tmp){
  print(x)
}
```

Conditional statement can be used to execute an operation if some variable is equal to TRUE. Let's look at an example.

```
## Example 13.2 ##

condition1<-c(TRUE, FALSE, TRUE, FALSE)
condition2<-c(FALSE, TRUE, FALSE, TRUE)

# AND
condition1&condition2

# OR
condition1|condition2

# IF
for(i in 1:4){
  if(i<2){
    print(i)
  }else{
    print( -i)
  }
}

# Ex. 1 ifelse(condition, action if true, action if false)

ifelse(condition1, 'a', 'b')
```

```
## Example 13.3 ##

## Here a more elaborated one.

startTime <- 1
endTime <- 5
curTime <- 2

if ((curTime>startTime) && (curTime<endTime)) {
  for (time in curTime:endTime) {
    print(paste("Time is ",time," o'clock",sep=""))
  }
  print("Time to go home!")
} else {
  print("Not work time")
}
```

## 14. Monte Carlo Methods

Monte Carlo (MC) simulations are commonly used in statistics to estimate features of distributions that may not have closed form. For instance, it can be used to estimate the power of a test statistics whose distribution over repeated sampling is unknown.

The base package of R offers functions that can be used to obtain “random” draws from several distributions. For each distribution there are usually three functions: one, whose name usually starts with `r` for “random”, which can be used to obtain random draws, one, whose name usually starts with `d` for “density”, that evaluates the density function for a give value of the random variable, one, whose name usually starts with `p` for “probability”, that will give the cumulative distribution function (CDF) at a given quantile and one, whose name usually starts with `q` for “quantile” that gives the quantile corresponding to a value of the CDF. Below we have examples of these functions for the normal density.

```
## Example 14.1 ##  
# Random draws  
x<-rnorm(n=1000, sd=1, mean=0)  
plot(density(x))  
  
# Density  
dnorm(x=0, mean=10, sd=4)  
  
# Quantile  
qnorm(p=.975, sd=1, mean=0)  
  
# CDF  
pnorm(q=1.96, sd=1, mean=0)
```

There are many other functions that can be used to draw numbers from other distribution with continuous (e.g., `rgamma`, `rexp`, `runif`) or discrete support (e.g., `rbinom`, `rpoiss`).

The function `sample()` can be used to draw numbers from a bag of labels with or without replacement.

```
## Example 14.2 ##  
# Random draws  
sample(x=c("a", "b", "c", "d"), size=10, replace=TRUE)  
sample(x=c("a", "b", "c", "d"), size=2, replace=FALSE)
```

We use computers to mimic random processes. Although the numbers generated by functions such as `sample()` or `rnorm()` look like random they are indeed deterministic. You can see this by controlling the seed of random number generator. The seed is an integer that controls the sequence of number of generated by the random generator. The following example illustrates this.

```
## Example 14.3 ##  
# Controlling the seed  
set.seed(1295490)  
runif(3)  
  
set.seed(1295490)  
runif(3)  
  
set.seed(12954)  
runif(3)
```

**Monte Carlo Estimates.** Here we have an example of a MC estimate of the mean, standard deviation and .95 quantile of a normal density using numbers randomly generated from a normal density with mean zero and variance equal to one.

```
## Example 14.3 ##  
N<-10000  
z<-rnorm(N, sd=1, mean=0)  
  
# estimating the mean  
mean(z)  
  
# estimating the sd (should be close to 1)  
sd(z)  
  
# estimating the probability of z < 1.96, should be close to .975  
mean(z<1.96)
```

In practice, we do not use MC methods to estimate the mean, standard deviation and quantiles of the standard normal density. These methods are used mostly when the distribution of the random variable is unknown. To illustrate, suppose  $X$  is a RV which is the product of  $Z_1$  and  $Z_2$ , where  $Z_1$  is a standard normal random variable and  $Z_2$  is a random variable having an exponential density with rate parameter equal to 1. The density function of  $X$  does not have a closed form. However, we can estimate features of the distribution of  $X$  using MC methods. An example is provided below.

```
## Example 14.4 ##
N<-100000
z1<-rnorm(n=N, sd=1, mean=0)
z2<-rexp(n=N, rate=1)
x<-z1*z2

# estimating the mean
mean(x)

# estimating the variance
var(x)

# estimating the probability of z > 1
mean(z>1)
```

Here we have a more elaborated example. The code below estimates the power of a t-test under different scenarios of effect size.

```
## Example 14.5 ##
nsim <- 1000 # number of Monte Carlo Replicates
eff<- c(0.2,0.5,1,1.5,2,3)
result<-matrix(nrow=length(eff) , ncol=nsim,NA)
SD<-1
power<-numeric()
n <- 10
for (i in 1:length(eff)) { # loop over effect size
  for (j in 1:nsim) { # loop over MC replicates
    group1 = rnorm(n=n,mean=0,sd=SD)
    group2 = rnorm(n=n,mean=eff[i],sd=SD)
    model = t.test(group1,group2,"two.sided")
    result[i,j] = as.numeric(model$p.value<0.05) # did we reject H0?
  }
  power[i] = mean(result[i,])
}
plot(eff,power,xlab="Effect Size",ylab="Power",main="T Test power vs.
Effect Size for n=10",type="o",col="red")
```

## 15. Functions

Most of objects in R are functions. A function takes some arguments as input, perform some internal computations and, usually, returns an object. For instance, the function `mean()` takes as argument a numeric vector and returns an integer. You can easily create your own R-functions. This allows you to automate blocks of code that can be later on used as a black-box. The following example illustrates how to create a very simple function.

```
## Example 15.1 ##  
getPower<-function( x ,power){  
    out<-x^power  
    return(out)  
}  
  
getPower (x=3,power=2)  
getPower (x=c(1,2,3),power=0)
```

Here are two simple functions for finding an item in a list.

```
## Example 15.2 ##  
IsPresent <- function(myList,item){  
    if (length(myList)==0) return( FALSE )  
    for (i in 1:length(myList)){  
        if (item==myList[i]) { return(TRUE) }  
    }  
    return(FALSE)  
}  
  
GetIndex <- function(myList,item) {  
    for (i in 1:length(myList)) {  
        if (item==myList[i]) { return(i) }  
    }  
    return(0)  
}  
  
myList = c("a","b","c","d")  
IsPresent(myList,"e")  
IsPresent(myList,"c")  
GetIndex(myList,"c")  
  
# OR, using built-in functions  
"e">%in%myList  
"c">%in%myList  
which(myList%in%"c")
```

# **Statistical Methods for Genome-Enabled Prediction**

By:

Daniel Gianola & Gustavo de los Campos  
Iowa State University, May-2012.

## **Computer Lab Handouts**

### **LAB 1: Linear Models**

1.1. Linear models and ordinary least squares (15 min).....	2
1.2. The ‘Curse’ of Dimensionality (30 min).....	2
1.3. Confronting the challenges posed by highly dimensional predictors (45 min) .....	4
Subset selection .....	4
Shrinkage estimation .....	5
References .....	7
Appendix .....	7
A1. Deriving ordinary least-squares (OLS) estimate using lm().....	7
A2. Deriving ordinary least-squares (OLS) estimate using matrix operations.....	8
A3. Deriving ordinary least-squares (OLS) estimate using iterative procedures.....	9

## **Lab 2: Shrinkage Estimation**

2.1. Penalized Estimates .....	2
2.2. Computing RR estimates.....	5
2.3. Effect of regularization on estimates, goodness of fit and model DF.....	5
2.4. The Hat Matrix of large-p with small-n genomic regressions as a local smoother.....	7
2.5. Bayesian View of Ridge Regression.....	8
2.6. G-BLUP .....	11
References .....	14

## **Lab 3: The Bayesian Alphabet**

3.1. The Bayesian Alphabet.....	2
3.2. Ridge Regression Vs Bayesian Ridge Regression.....	9
3.3. Bayesian Lasso: fixed versus random lambda.....	11
3.4. Regression using markers and pedigree .....	13
References .....	14

## **Lab 4: Semi-parametric Genomic Regression Using Reproducing Kernel Hilbert Spaces Methods**

4.1. Semi-parametric genome-enabled regression .....	2
4.2. Reproducing Kernel Hilbert Spaces (RKHS) regressions.....	3
4.3. Scatter plot smoothing with a Gaussian kernel .....	5
4.4. Inspecting the Hat Matrix .....	6
4.5. Bayesian view of RKHS .....	7
4.6. Genomic-Enabled Prediction Using RKHS .....	9
4.7. Kernel Averaging .....	12
4.8. Pedigree + Marker Models.....	15
References .....	17

## **LAB 5: Penalized Neural Networks**

5.1. Introduction .....	2
5.2. Scatterplot smoothing using a penalized NN.....	5
5.3. Penalized Neural Network Using Pre-selected Markers .....	7
5.4. Penalized Neural Networks Using Marker-derived Basis Functions as Inputs.....	8
References .....	9

## **LAB 6: Validation Methods**

6.1. Introduction .....	2
6.2. Alternative Validation Schemes .....	2
6.3. Between sub-population prediction .....	6
6.4. Across environment prediction using single-trait models .....	7
References .....	8

# *Statistical Methods for Genome-Enabled Prediction,*

## LAB 1:

### Linear Models<sup>1</sup>

( [gcampos@uab.edu](mailto:gcampos@uab.edu) )

library (BLR)  
data (~~wheat~~, wheat)  
fit (X)  
show + test  
+ summary

### Contents

1.1. Linear models and ordinary least squares (15 min).....	2
1.2. The 'Curse' of Dimensionality (30 min).....	2
1.3. Confronting the challenges posed by highly dimensional predictors (45 min) .....	4
Subset selection .....	4
Shrinkage estimation .....	5
References .....	7
Appendix .....	7
A1. Deriving ordinary least-squares (OLS) estimate using lm() .....	7
A2. Deriving ordinary least-squares (OLS) estimate using matrix operations.....	8
A3. Deriving ordinary least-squares (OLS) estimate using iterative procedures.....	9

---

<sup>1</sup> Suggestions made by Daniel Gianola are gratefully acknowledged.

## 1.1. Linear models and ordinary least squares (15 min)

Consider the following model:

$$y_i = \mu + \sum_{j=1}^p x_{ij} \beta_j + \varepsilon_i \quad i = (1, \dots, n)$$

where:  $y_i$  is the phenotype of the  $i^{th}$  individual,  $\mu$  is an effect common to all individuals (an "intercept"),  $x_{ij}$  are covariates (e.g., marker genotypes),  $\beta_j$  is the effect of the  $j^{th}$  covariate and  $\varepsilon_i$  is a model residual. In matrix notation the model is expressed as:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad [1]$$

where:  $\mathbf{y} = \{y_i\}$  is a vector of phenotypes,  $\mathbf{X} = \{\mathbf{1}, \mathbf{x}_1, \dots, \mathbf{x}_p\}$  is an incidence matrix for the vector of regression coefficients,  $\boldsymbol{\beta} = (\mu, \beta_1, \dots, \beta_p)'$  and  $\boldsymbol{\varepsilon} = \{\varepsilon_i\}$  is a vector of model residuals.

The ordinary least squares estimate of  $\boldsymbol{\beta}$  is the solution to the following optimization problem:

$$\hat{\boldsymbol{\beta}}_{OLS} = \underset{\arg \min}{\sum_i} \left( y_i - \sum_j x_{ij} \beta_j \right)^2$$

where  $\sum_i \left( y_i - \sum_j x_{ij} \beta_j \right)^2$  is a residual sum of squares. The first order conditions of [2] are satisfied by  $\hat{\boldsymbol{\beta}}_{OLS} = [\mathbf{X}'\mathbf{X}]^{-1} \mathbf{X}'\mathbf{y}$ .

The appendix provide alternative ways of deriving OLS estimates in R, including use of the lm() function, solution using matrix operations and iterative procedures.

$$MSE = [\mathbf{X}'\mathbf{X}]^{-1} \mathbf{e}'\mathbf{e}$$

## 1.2. The 'Curse' of Dimensionality (30 min)

The mean-squared error (MSE) of an estimator is:  $MSE(\hat{\theta}) = E[(\theta - \hat{\theta})^2]$  where  $\theta$  is the true value of the parameter and  $\hat{\theta}$  is the estimator, which is a function of the data ( $\mathbf{X}$  and  $\mathbf{y}$  in the regression example discussed above). The expectation in the MSE formula is taken with respect to all possible samples of data. Commonly  $\mathbf{X}$  is treated as fixed and the expectation is taken only with respect to possible realizations of  $\mathbf{y}$  given  $\mathbf{X}$ .

The MSE can be decomposed in two components:  $MSE(\hat{\theta}) = [\theta - E(\hat{\theta})]^2 + Var(\hat{\theta})$ , where  $[\theta - E(\hat{\theta})]$  and  $Var(\hat{\theta})$  are the bias and variance of the estimator.

The expectation of the OLS estimate of regression coefficients in [1] is:

$$\begin{aligned} E[\hat{\beta}_{OLS} | \mathbf{X}] &= [\mathbf{X}'\mathbf{X}]^{-1} \mathbf{X}'E[\mathbf{y}] \\ &= [\mathbf{X}'\mathbf{X}]^{-1} \mathbf{X}'\mathbf{X}\beta + [\mathbf{X}'\mathbf{X}]^{-1} \mathbf{X}'E[\epsilon] \\ &= \beta + [\mathbf{X}'\mathbf{X}]^{-1} \mathbf{X}'E[\epsilon] \end{aligned}$$

When model [1] holds,  $E[\epsilon] = \mathbf{0}$ , therefore:  $E[\hat{\beta}_{OLS} | \mathbf{X}] = \beta$ . In words, if the linear model holds, OLS gives unbiased estimates of regression coefficients. The second term of the MSE formula,  $Var(\hat{\theta})$ , is a frequentist measure of uncertainty and reflects variability of the estimator over repeated sampling. The asymptotic (co)variance matrix of OLS estimates of regression coefficients, given  $\mathbf{X}$ , is,  $Var(\hat{\beta}) = [\mathbf{X}'\mathbf{X}]^{-1} \sigma^2$ , where  $\sigma^2$  is the variance of model residuals. This is also the finite-sample covariance matrix of estimates under normality. Therefore, the MSE of the estimate of the  $j$ th regression coefficient is  $C^{jj} \sigma^2$  where  $C^{jj}$  is the  $j$ th diagonal entry of the inverse of the matrix of coefficients, that is  $\mathbf{C}^{-1} = [\mathbf{X}'\mathbf{X}]^{-1}$ . This element decreases with sample size. In the following example we study how MSE of estimates of regression coefficients changes with  $n$  and  $p$ .

#### Example 1. Effects of $n$ and $p$ on Mean-Squared Error of OLS estimates

```
rm(list=ls())
n<-seq(from=100,to=300,by=10) # vector defining sample size
p<-seq(from=5,to=80,by=4) # vector defining number of predictors
x<-rbinom(prob=.5,n=max(p)*max(n),size=1) # sample predictors
X<-matrix(nrow=max(n),ncol=max(p),data=x)
varE<-1
VAR<-matrix(nrow=length(n),ncol=length(p),NA)
colnames(VAR)<-p
rownames(VAR)<-n
for(i in 1:length(n)){ # loop over sample size
  for(j in 1:length(p)){ # loop over number of predictors
    tmpX<-X[1:n[i],1:p[j]]
    C<-crossprod(tmpX) # X' * X
    CInv<-chol2inv(chol(C)) # should be symmetric positive definite
    VAR[i,j]<-mean(diag(CInv))*varE # average variance of estimates
  }
}
## plot Variance (equal to MSE in this case) Vs. n and p
persp(z=VAR,x=n,y=p,xlab="Sample Size",
      ylab="Number of Predictors",zlab="MSE(bj)",col=2)
```

*library (rgl) for move around 3 dimensions*

NOTE. When  $p > n$ , the OLS estimate is not unique because  $\mathbf{X}'\mathbf{X}$  is singular. Nevertheless, predictions,  $\hat{\mathbf{y}} = \mathbf{X}[\mathbf{X}'\mathbf{X}]^{-1}\mathbf{X}'\mathbf{y}$ , are unique; here  $[\mathbf{X}'\mathbf{X}]^{-1}$  is a generalized inverse of  $\mathbf{X}'\mathbf{X}$ . The function `ginv()` of `library(MASS)` can be used to compute a Moore-Penrose generalized inverse. The function `svd()` can be used to compute the singular value decomposition of  $\mathbf{X}$  from where  $\hat{\mathbf{y}}$  can also be computed.

In genomic models  $p > n$ , because of this, estimation methods other than OLS are required. In the following sections we consider alternative methods.

### 1.3. Confronting the challenges posed by highly dimensional predictors (45 min)

In this section we discuss two different approaches designed to confront the challenges posed by ‘large  $p$  with small  $n$ ’ regressions. In the first one (**subset selection**) we design an algorithm to select  $k$  out of  $p$  ( $k \leq p$ ) predictors; our final model will include only these  $k$  predictors. Subset selection is a commonly used practice, and it is based on the idea that ‘highly dimensional predictors are dangerous’; therefore, the approach seeks to reduce the number of predictors. The second approach (**shrinkage estimation**) uses all available predictors and confronts the challenges posed by regressions with  $p > n$  by using shrinkage estimation methods. We illustrate this approach using ridge regression. In both examples we use a genomic dataset made available with R-package BLR ('wheat'). This dataset contains 4 phenotypes evaluated in 599 wheat lines that were genotyped for 1,279 markers. In the examples we use 450 lines for training and evaluate the prediction accuracy of each of the methods on the remaining 149 lines (testing).

**Subset selection.** The problem of selecting  $k$  out of  $p$  ( $k < p$ ) predictors can be viewed as a model comparison problem. Ideally, we would fit all possible models and select the one that is best according to some model comparison criterion (e.g., AIC, Akaike Information Criterion, Akaike 1973). In practice, when  $p$  is large fitting all possible models is not feasible. Instead model search algorithms are used. A very simple search algorithm consists of regressing the response in each of the predictors one at a time ('single marker regression'). Each of these regressions yields a measure of association between markers and phenotypes (e.g., a p-value). Then, we can form our final model by using the first  $k$  predictors ranked according to the association measure. This approach is commonly used in Genome Wide Association Studies (GWAS). The following example fits models with  $k$  predictors ( $k=1,\dots,300$ ) chosen based on the marginal association between markers and phenotypes. The examples use the ‘wheat dataset’ of the BLR package of R (G. de los Campos and Pérez 2010; Paulino Pérez et al. 2010).

### Example 2. Subset selection using p-values derived from single-marker regressions

```

rm(list=ls())
##### DATA #####
library(BLR)
data(wheat)
objects()
N<-nrow(X) ; p<-ncol(X)
y<-Y[,2]
set.seed(1235)
tst<-sample(1:N, size=150, replace=FALSE)
XTRN<-X[-tst,] ; yTRN<-y[-tst]
XTST<-X[tst,] ; yTST<-y[tst]
##### SINGLE MARKER REGRESSIONS #####
pValues<-numeric()
for(i in 1:p){
  fm<-lm(yTRN~XTRN[,i])
  pValues[i]<-summary(fm)$coef[2,4]
  print(paste('Fitting Marker ',i,'.',sep=''))
}
plot(-log(pValues),base=10,cex=.5,col=2)
##### VARIABLE SELECTION #####
myRanking<-order(pValues)
sqCor<-numeric()
for(i in 1:300){
  tmpIndex<- myRanking[1:i]
  fm<-lm(yTRN~XTRN[,tmpIndex])
  bHat<-coef(fm)[-1] ; bHat<-ifelse(is.na(bHat),0,bHat)
  yHat<-as.matrix(XTST[,tmpIndex])%*%bHat
  sqCor[i]<-cor(yTST,yHat)^2
  print(paste('Fitting Model with ',i,' markers!',sep=''))
}
plot(sqCor,type='o',col=2,ylab='Squared Correlation',
     xlab='Number of markers',ylim=c(0,.28))

```

**Shrinkage estimation.** We have seen that when  $n$  is small and  $p$  is large OLS estimates have high variance, and therefore high MSE. In addition, when  $p$  is large relative to  $n$ , over-fitting may occur, yielding poor predictive ability. Penalized estimates of regression coefficients are designed to confront these problems. The main idea is to reduce MSE by reducing the variance of the estimator, even at the expense of introducing bias. We will cover penalized estimation procedures in more detail in Lab 2; here we briefly illustrate their performance using Ridge Regression (Hoerl and Kennard 1970). Recall that in the linear model of eq. 1

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \quad [1]$$

the OLS estimates of regression coefficients are the solution to the following systems of equations

$$[\mathbf{X}'\mathbf{X}]\hat{\boldsymbol{\beta}}_{OLS} = \mathbf{X}'\mathbf{y} \quad [2]$$

The RR estimates has a very similar form, we simply add a constant to the diagonal of the matrix of coefficients, that is:

$$[\mathbf{X}'\mathbf{X} + \lambda \mathbf{D}] \hat{\boldsymbol{\beta}}_{RR} = \mathbf{X}'\mathbf{y} \quad [5]$$

where  $\lambda$  is a constant and  $\mathbf{D}$  is a diagonal matrix with zero in its first diagonal entry (this, to avoid shrinking the estimate of the intercept) and ones in the remaining diagonal entries and zeroes everywhere else. When either  $\lambda$  equals zero, the solution to the above problem is OLS. Adding a constant to the diagonal entries of the coefficient matrix makes it non-singular and shrinks the estimates of regression coefficients other than the intercept towards zero. This induces bias but reduces the variance of the estimates; in large-p with small-n problems this may reduce MSE of estimates and may yield more accurate predictions. The following R-code computes RR estimates.

### Example 3. Ridge Regression

```
MSx<-0
for(i in 1:ncol(XTRN)) { MSx<-MSx+mean((XTRN[,i]-mean(XTRN[,i]))^2) }
h2<-0.5
lambda<-round(MSx*(1-h2)/h2)

TMP<-cbind(1,XTRN)
C<-crossprod(TMP)
rhs<-crossprod(TMP,yTRN)
for(i in 2:ncol(C)){ C[i,i]<-C[i,i]+lambda } #adds a constant to diag
CInv<-chol2inv(chol(C))
bHatRR<-crossprod(CInv,rhs)
yHatRR<-cbind(1,XTST) %*%bHatRR
tmp<-cor(yHatRR,yTST)^2
lines(x=c(0,30),y=rep(tmp,2),col=4,lwd=2)
lines(x=c(150,300),y=rep(tmp,2),col=4,lwd=2)
text(x=90,y=tmp,label=expression(paste('RR (lambda=',lambda, ')))),col=4 )
```

## References

- Akaike, H. 1973. Information theory and an extension of the maximum likelihood principle. In *Second international symposium on information theory*, 1:267–281.
- de los Campos, G., and P. Pérez. 2010. *BLR: Bayesian linear regression. R package version 1.2.* <http://cran.r-project.org/web/packages/BLR/index.html>.
- Hoerl, A. E, and R. W Kennard. 1970. “Ridge regression: Biased estimation for nonorthogonal problems.” *Technometrics* 12 (1): 55–67.
- Pérez, Paulino, Gustavo de los Campos, José Crossa, and Daniel Gianola. 2010. “Genomic-Enabled Prediction Based on Molecular Markers and Pedigree Using the Bayesian Linear Regression Package in R.” *The Plant Genome Journal* 3 (2): 106-116. doi:10.3835/plantgenome2010.04.0005.

## Appendix

### A1. Deriving ordinary least-squares (OLS) estimate using lm()

The OLS estimate of  $\beta$  can be obtained using the function `lm()`, which fits a linear model by OLS. Alternatively, we can compute the solution using matrix operations. The code below simulates data for regression [1], and fits the linear model using `lm()`.

#### Example A1. Deriving Ordinary Least Squares estimates using lm()

```
rm(list=ls())
## SIMULATES DATA FOR A LINEAR MODEL
set.seed(12345)
n<-100
p<-6
set.seed(12345)
X<-matrix(nrow=n,ncol=p,
           data=rbinom(n=n*p,p=.5,size=1))
beta<-rnorm(p,mean=0,sd=2)
ERROR<-rnorm(n=n, sd=1,mean=0)
y<-124 +X%*%beta+ERROR # note %*% computes matrix product

## FITS THE MODEL USING lm() #####
fm<-lm(y~X)
summary(fm)
bHat1<-fm$coeff
#(continues below)
```

## A2. Deriving ordinary least-squares (OLS) estimate using matrix operations

In the system of equations

$$[\mathbf{X}'\mathbf{X}]\hat{\boldsymbol{\beta}}_{OLS} = \mathbf{X}'\mathbf{y} \quad [2]$$

we will refer to  $\mathbf{C} = [\mathbf{X}'\mathbf{X}]$  as the matrix of coefficients and to  $\mathbf{rhs} = \mathbf{X}'\mathbf{y}$  as the right-hand side of the system. The matrix of coefficients can be computed using `C<-t(X) %*% X`, or, equivalently, `C<-crossprod(X)`. Similarly, the right-hand-side can be computed using `rhs<-t(X) %*% y`, or, equivalently, `rhs<-crossprod(X, y)`. `crossprod()` is usually faster. The system can be solved using the function `solve()`, as illustrated below.

### Example A2. Deriving Ordinary Least Squares Using Matrix Operations

```
# (continued from Example 1)
## FITS LINEAR MODEL USING MATRIX OPERATIONS #####
X2<-cbind(1,X) ## note a vector of 1s is added type head(X)
C<-crossprod(X2)
rhs<-crossprod(X2, y)
bHat2<-solve(C, rhs)
# (continues in Example 3)
```

The matrix of coefficients is symmetric and positive definite. The cholesky decomposition of this matrix (**C**) is an upper-triangular matrix satisfying  $\mathbf{C}=\mathbf{U}'\mathbf{U}$ . **U** can then be used to invert **C** using `chol2inv()` function (see below). This is usually faster than using function `solve()`. Other factorizations of **C**, such as the eigen-value decomposition, `eigen()`, or the QR decompositions, `qr()`, can also be used to invert **C** as well. An example using the cholesky decomposition of **C** is given below.

### Example A3. Inversion of positive definite matrices using the Cholesky factorization

```
# (continued from Ex. 1 and 2)
X2<-cbind(1,X) # note a vector of 1s is added type head(X)
C<-crossprod(X2)
rhs<-crossprod(X2, y)
U<-chol(C)      # computes the Cholesky decomposition
CInv<-chol2inv(U) # obtains the inverse from a Cholesky decomp.
bHat3<-CInv%*%rhs
# compare bHat1, bHat2, bHat3
round(cbind(bHat1,bHat2,bHat3), 4)
# (continues in example 4)
```

### A3. Deriving ordinary least-squares (OLS) estimate using iterative procedures

In practice, when  $p$  is large, the system of equation is solved using some type of iterative methods. Here is one possible algorithm. Suppose that we know all but the  $j^{th}$  regression coefficient, then, from the data-equation we can write:

$$\begin{aligned} y_i &= \sum_{k=1}^p x_{ik} \beta_k + \varepsilon_i \\ y_i &= \sum_{k \neq j}^p x_{ik} \beta_k + x_{ij} \beta_j + \varepsilon_i \\ y_i - \sum_{k \neq j}^p x_{ik} \beta_k &= x_{ij} \beta_j + \varepsilon_i \\ \tilde{y}_{i(-j)} &= x_{ij} \beta_j + \varepsilon_i \end{aligned} \quad [3]$$

where:  $\tilde{y}_{i(-j)} = y_i - \sum_{k \neq j}^p x_{ik} \beta_k$  is an off-set formed by subtracting from the original phenotypes the contribution to the conditional expectation of all but the  $j^{th}$  predictor, that is  $\sum_{k \neq j}^p x_{ik} \beta_k$ . The OLS estimate of  $\beta_j$  in [3] is simply

$$\hat{\beta}_j = \frac{\sum_i x_{ij} \tilde{y}_{i(-j)}}{\sum_i x_{ij}^2}. \quad [4]$$

A back-fitting algorithm can then be formed by iterating over regression coefficients using [4]. This is implemented in the following R-code.

- Run the code. How do estimates computed using the above-described algorithm compare with the exact solution?
- Change nIter (the number of iterations) from 2 to 30 and compare.

#### Example A4. Deriving Ordinary Least Squares Using Iterative Procedures

```
# Computes OLS using a back-fitting algorithm
SSx<-colSums(X2^2)                      # the diagonal elements of X'X
nIter<-2                                    # number of iterations of the algorithm
bHat4<-rep(0,ncol(X2))                     # initialvalues bj=zero
bHat4[1]<-mean(y)                         # initial values mu=mean(y)
e<-y-mean(y)                               # initial model residuals

for(i in 1:nIter){                         # loop for iterations of the algorithm
  for(j in 1:ncol(X2)){                   # loop over predictors
    yStar<-e+X2[,j]*bHat4[j]            # forming off-sets
    bHat4[j]<- sum(X2[,j]*yStar)/SSx[j]  # eq. [4]
    e<-yStar-X2[,j]*bHat4[j]           # updates residuals
  }
}

# compare bHat1, bHat2, bHat3, bHat4
round(cbind(bHat1,bHat2,bHat3,bHat4),4)
```

# *Statistical Methods for Genome-Enabled Prediction,*

## LAB 2:

### Shrinkage Estimation<sup>1</sup>

( [gcampos@uab.edu](mailto:gcampos@uab.edu) )

## Contents

2.1. Penalized Estimates .....	2
2.2. Computing RR estimates.....	5
2.3. Effect of regularization on estimates, goodness of fit and model DF.....	5
2.4. The Hat Matrix of large-p with small-n genomic regressions as a local smoother.....	7
2.5. Bayesian View of Ridge Regression.....	8
2.6. G-BLUP .....	11
References .....	14

**NOTE:** In many examples in this lab we use Bayesian methods. In those examples we make inferences based on a relatively small number of samples and this is done due to time constraints. In practice, accurate inferences require much more samples.

---

<sup>1</sup> Suggestions made by Daniel Gianola are gratefully acknowledged.

## 2.1. Penalized Estimates

Ordinary least squares (OLS) and Maximum likelihood (ML) are examples of estimation methods in which estimates are derived by maximizing the fitness (as measured by the residual sum of squares or likelihood function) of the model to the training data. When the number of predictors ( $p$ ) is large relative to sample size ( $n$ ) this is not a good strategy: estimates can have high mean-squared error (MSE) and over-fitting may occur. Penalized estimates are obtained as the solution to an optimization problem that balances two components: how well the model fits the data and how-complex the model is. The general form of the optimization problem is:

$$\hat{\beta} = \underset{\beta}{\arg \min} \{ L(\mathbf{y}, \beta) + \lambda J(\beta) \} \quad [1]$$

where,  $L(\mathbf{y}, \beta)$  is a loss function that measure lack of fit of the model to the data,  $J(\beta)$  is a measure of model complexity and  $\lambda \geq 0$  is a regularization parameter controlling the trade-offs between fitness and model complexity.

**Ridge Regression** (Hoerl and Kennard 1970) is a particular case of [1] and is obtained by setting  $L(\mathbf{y}, \beta)$  to be a residual sum of squares  $L(\mathbf{y}, \beta) = \sum_i \left( y_i - \sum_j x_{ij} \beta_j \right)^2$  and  $J(\beta)$  to be the sum of square of the regression coefficients; typically, some of the regression coefficients (e.g., the intercept) are not penalized; therefore,  $J(\beta) = \sum_{j \in S} \beta_j^2$  where  $S$  define the set of coefficients to be penalized.

$$\hat{\beta} = \underset{\beta}{\arg \min} \left\{ \sum_i \left( y_i - \sum_j x_{ij} \beta_j \right)^2 + \lambda \sum_{j \in S} \beta_j^2 \right\}$$

subject to  $\beta$  for  
nonzero (e.g.)  
constraint (e.g.)  
are start const  
Simplik p.m.

$\lambda < 0 \rightarrow \text{solution} \approx \text{OLS}$   
 $\lambda \rightarrow 0 \Rightarrow \beta \rightarrow 0$

[2]

When  $\lambda \rightarrow \infty$  the solution is  $\hat{\beta}_{RR} = \mathbf{0}$ . On the other extreme, as  $\lambda = 0$  the solution is the OLS estimates of  $\beta$ . In matrix notation problem [2] can be represented as:

$$\hat{\beta}_{RR} = \underset{\beta}{\arg \min} \left\{ (\mathbf{y} - \mathbf{X}\beta)' (\mathbf{y} - \mathbf{X}\beta) + \lambda \beta' \mathbf{D} \beta \right\}$$

No  
penalize  
for linear

where:  $(\mathbf{y} - \mathbf{X}\beta)' (\mathbf{y} - \mathbf{X}\beta) = \sum_i \left( y_i - \sum_j x_{ij} \beta_j \right)^2$  is a RSS and  $\beta' \mathbf{D} \beta = \sum_{j \in S} \beta_j^2$  is a sum of squares of the regression coefficients. Here,  $\mathbf{D}$  is a diagonal matrix whose entries are 1 for  $j \in S$  and zero otherwise. The first order conditions of the above optimization problem are satisfied by the following system of linear equations:

$$[\mathbf{X}' \mathbf{X} + \lambda \mathbf{D}] \hat{\beta}_{RR} = \mathbf{X}' \mathbf{y} \quad [3]$$

Relative to OLS, RR adds a constant ( $\lambda$ ) to the diagonal entry corresponding to regression coefficients that are included in S (i.e., those whose effects are penalized). When either  $D$  or  $\lambda$  equals zero, the solution to the above problem is OLS. Adding a constant to the diagonal of the matrix of coefficients shrink estimates towards zero. This induces bias but reduces the variance of the estimates. And in large-p small-n regressions this may smaller MSE than those of OLS estimates and better predictions.

**A simplified example.** Let us consider a simple example where each subject was assigned to one of two possible treatments (treatments 1 and 2). The treatment-means parameterization of this model is:  $y_i = x_{1i}\beta_1 + x_{2i}\beta_2 + \varepsilon_i$  where  $y_i$  is the response,  $x_{1i}$  is a dummy variable indicator of treatment 1,  $x_{2i} = (1 - x_{1i})$  is a dummy variable indicator of treatment 2,  $\beta_1$  and  $\beta_2$  the means of treatments 1 and 2, respectively, and  $\varepsilon_i$  is a model residual. The OLS estimates of regression coefficients in this model are:

$$\begin{bmatrix} \sum_i x_{1i}^2 & \sum_i x_{1i}x_{2i} \\ \sum_i x_{1i}x_{2i} & \sum_i x_{2i}^2 \end{bmatrix} \begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{pmatrix} = \begin{pmatrix} \sum_i x_{1i}y_i \\ \sum_i x_{2i}y_i \end{pmatrix}$$

Moreover,  $\sum_i x_{1i}^2$  and  $\sum_i x_{2i}^2$  equal the number of individuals in treatment 1 and 2 (denoted as  $n_1$  and  $n_2$  respectively), since  $x_{1i}$  and  $x_{2i}$  are orthogonal  $\sum_i x_{1i}x_{2i} = 0$ , and, finally,  $\sum_i x_{1i}y_i$  and  $\sum_i x_{2i}y_i$  are the sum of the response variable for subjects assigned to treatments 1 and 2, respectively. Therefore,

$$\begin{bmatrix} n_1 & 0 \\ 0 & n_2 \end{bmatrix} \begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{pmatrix} = \begin{pmatrix} \sum_{i:x_{1i}=1} y_i \\ \sum_{i:x_{2i}=1} y_i \end{pmatrix}$$

, from where we conclude that the OLS estimate of the treatment mean are simply the average of the phenotypes observed in each treatment, that is  $\hat{\beta}_1 = \frac{\sum_{i:x_{1i}=1} y_i}{n_1}$  and  $\hat{\beta}_2 = \frac{\sum_{i:x_{2i}=1} y_i}{n_2}$ . Now, considering the RR estimates, according to [3] these will be

$$\begin{bmatrix} n_1 + \lambda & 0 \\ 0 & n_2 + \lambda \end{bmatrix} \begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{pmatrix} = \begin{pmatrix} \sum_{i:x_{1i}=1} y_i \\ \sum_{i:x_{2i}=1} y_i \end{pmatrix}$$

; therefore the RR estimates are  $\hat{\beta}_1 = \frac{\sum_{i:x_{1i}=1} y_i}{n_1 + \lambda}$  and  $\hat{\beta}_2 = \frac{\sum_{i:x_{2i}=1} y_i}{n_2 + \lambda}$ . Therefore, adding  $\lambda$  to the diagonal entries of the matrix of coefficients will shrink estimates towards zero. By how much? This will depend on the relationship between  $\lambda$  and sample size. From here we can also see that with fix  $\lambda$ , the amount

of shrinkage will decrease as sample size increases. Asymptotically, if we fix  $\lambda$  and let the number of individuals in each treatment approach infinity, RR estimates converge to OLS estimates.

**Other penalized estimators.** Several alternative penalized estimation procedures have been proposed, and they differ on the choice of penalty function,  $J(\beta)$ . As we discussed above, in RR, the penalty is proportional to the sum of squares of the regression coefficients or L2 norm,  $J(\beta) = \sum_{j=1}^p \beta_j^2$ . A more general formulation, known as **Bridge regression** (Frank and Friedman 1993), uses  $J(\beta) = \sum_{j=1}^p \|\beta_j\|^\gamma$  with  $\gamma > 0$ . RR is a particular case with  $\gamma = 2$  yielding RR. **Subset selection** occurs as a limiting case with  $\gamma \rightarrow 0$ , this penalizes the number of non-zero effects regardless of their magnitude,  $J(\beta) = \sum_{j=1}^p I(\beta_j \neq 0)$ . Another special case, known as **LASSO** (Least Absolute Angle and Selection Operator, (Tibshirani 1996) occurs with  $\gamma = 1$ , yielding the L1 penalty:  $J(\beta) = \sum_{j=1}^p \|\beta_j\|$ . Using this penalty induces a solution that may involve zeroing-out some regression coefficients and shrinkage estimates of the remaining effects; therefore combining in features of subset selection with shrinkage estimation. LASSO has become very popular in several fields of applications. However LASSO and subset selection approaches have two important limitations. First, by construction, in these methods the solution admits at most  $n$  non-zero estimates of regression coefficients. In GS and with complex traits, there is no reason to restrict the number of markers with non-zero effect to be limited by  $n$  (the number of observations). Second, when predictors are correlated, something which occurs in GS, methods performing variable selection such as the LASSO are usually outperformed by RR (Hastie, Tibshirani, and Friedman 2009). Therefore, in an attempt to combine the good features of RR and of Lasso in a single estimation framework (Zou and Hastie 2005) proposed to use as penalty a weighted average of the L1 and L2 norm, that is, for  $0 \leq \alpha \leq 1$ ,  $J(\beta) = \alpha \sum_{j=1}^p \|\beta_j\| + (1-\alpha) \sum_{j=1}^p \beta_j^2$  and termed the method the **Elastic Net** (EN), this model involves then two tuning parameters which need to be specified, the regularization parameter ( $\lambda$ ) and  $\alpha$ .

## 2.2. Computing RR estimates

In the following example we present two ways of computing ridge regression estimates. The first one implements [3] using matrix operations; the second one uses an iterative procedure. Run this last algorithm with 10 and 500 iterations.

### Example 1. Alternative ways of deriving Ridge-Regression Estimates

```

rm(list=ls())
## Using Cholesky factor #####
library(BLR)
data(wheat)
X2<-cbind(1,X)
y<-Y[,2]
C<-crossprod(X2)
rhs<-crossprod(X2,y)
MSx<-0 ; for(i in 1:ncol(X)) { MSx<-MSx+var(X[,i])}
h2<-0.5
lambda<-MSx*(1-h2)/h2
for(i in 2:ncol(C)) { C[i,i]<-C[i,i]+lambda }
CInv<-chol2inv(chol(C))
bHatRR_1<-crossprod(CInv, rhs)

## Using an iterative procedure #####
diagC<-numeric()
for(i in 1:ncol(X2)) {diagC[i]<-sum(X2[,i]^2)+ifelse(i==1, 0, lambda) }
bHatRR_2<-rep(0, ncol(X2))
bHatRR_2[1]<-mean(y)
e<-y-mean(y)
nIter<-10
for(i in 1:nIter){
  for(j in 1:ncol(X2)){
    tmpY<-e+X2[,j]*bHatRR_2[j]
    rhs<-sum(X2[,j]*tmpY)
    bHatRR_2[j]<-rhs/diagC[j]
    e<-tmpY-X2[,j]*bHatRR_2[j]
  }
  print(i)
}
tmp<-range(c(bHatRR_1[-1],bHatRR_2[-1]))
plot(bHatRR_1[-1],bHatRR_2[-1], ylim=tmp, xlim=tmp, col=2, main="")
## Change nIter, set it equal to 500 and then equal to 1000

```

Convergence is faster if there is diagonal dominant ( $\lambda \downarrow$ )  
 (because more eigen values are positive)

## 2.3. Effect of regularization on estimates, goodness of fit and model DF

In penalized estimation, the regularization parameter ( $\lambda$ ) controls the trade-offs between model goodness of fit and model complexity. This affects parameter estimates (their value, and the statistical properties of the estimator) model goodness of fit to the training dataset and the ability of the model to predict un-observed phenotypes.

More iterations are needed for less dominant.

**Model complexity.** The complexity of a linear model can be measured by the degree of freedom of the model. In RR, predictions are computed as  $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}_{RR} = \mathbf{X}[\mathbf{X}'\mathbf{X} + \lambda\mathbf{D}]^{-1}\mathbf{X}'\mathbf{y} = \mathbf{H}_{RR}\mathbf{y}$  where  $\mathbf{H}_{RR} = \mathbf{X}[\mathbf{X}'\mathbf{X} + \lambda\mathbf{D}]^{-1}\mathbf{X}'$  is the Hat matrix. If we set  $\lambda = 0$  we obtain the Hat matrix of OLS:  $\mathbf{H}_{OLS} = \mathbf{X}[\mathbf{X}'\mathbf{X}]^{-1}\mathbf{X}'$ . In linear models degree of freedom are equal to the sum of the diagonal entries of  $\mathbf{H}$ . In OLS this just equals the number of predictors (provided that  $\mathbf{X}$  is full rank). In RR  $\lambda$  also affects DF. The following R-code fits RR over a grid of values of  $\lambda$  and evaluates the impact that  $\lambda$  has on goodness of fit to the training data, prediction accuracy, and model degree of freedom.

### Example 2. Effects of regularization on goodness of fit and model DF

```

rm(list=ls())
##### DATA #####
library(BLR)
data(wheat)
objects()
N<-nrow(X) ; p<-ncol(X)
y<-Y[,2]
set.seed(12345)
tst<-sample(1:N, size=150, replace=FALSE)
XTRN<-X[-tst,]
yTRN<-y[-tst]
XTST<-X[tst,]
yTST<-y[tst]

## FITTING MODEL OVER A GRID OF VALUES OF lambda
lambda<-c(5,10,50,100,200,500,700,1000, 2000, 5000,20000)
ZTRN<-cbind(1,XTRN) ; ZTST<-cbind(1,XTST)
sqCorTRN<-numeric(); sqCorTST<-numeric(); DF<-numeric()
BHat<-matrix(nrow=ncol(XTRN), ncol=length(lambda), NA)

C0<-crossprod(ZTRN)
rhs<-crossprod(ZTRN, yTRN)

for(i in 1:length(lambda)){ #loop over values of lambda
  C<-C0
  # adds lambda to the diagonal of C (starts at 2)
  for(j in 2:ncol(C)){ C[j,j]<-C[j,j]+lambda[i] }
  CInv<-chol2inv(chol(C))
  sol<-crossprod(CInv, rhs)
  BHat[,i]<-sol[-1]
  yHatTRN<-ZTRN%*%sol
  sqCorTRN[i]<-cor(yTRN, yHatTRN)^2
  yHatTST<-ZTST%*%sol
  sqCorTST[i]<- cor(yTST, yHatTST)^2
  H<-ZTRN%*%CInv%*%t(ZTRN)
  DF[i]<-sum(diag(H))
  print(i)
}
write(sqCorTST,file="sqCorTST.txt")
write(lambda,file="lambda.txt")
# (Plots in next page)

```

### Example 2. (from previous page)

```

## PLOT 1: Model Degree of freedom
plot(DF~log(lambda), type="o", col=2,
      xlab= expression(paste(log(lambda))), ylab="DF", ylim=c(0, max(DF))); abline(h=1, lty=2)

## PLOT 2: Estimates (shrinkage by marker)
marker<-1 # (choose a number between 1 and 1279)
plot(BHAT[marker,], type="o", col=2,
      xlab=expression(paste(log(lambda))), ylab="Estimate")
abline(h=0)
tmp<-range(BHAT[,c(1,5)])
## PLOT 3: Estimates (shrinkage all markers)
plot(BHAT[,5]~BHAT[,1], xlim=tmp, ylim=tmp,
      xlab='Lambda=5', ylab='Lambda=200', col=2, cex=.5);
lines(x=c(-10,10), y=c(-10,10))

## PLOT 4: Goodness of fit to TRN dataset
plot(sqCorTRN~log(lambda), type="o", col=2, main="Training data",
      xlab=expression(paste(log(lambda))), ylab="Squared Corr.")

## PLOT 5 Prediction Accuracy
plot(sqCorTST~log(lambda), type="o", col=2, main="Testing data",
      xlab=expression(paste(log(lambda))), ylab="Squared Corr.")

```

## 2.4. The Hat Matrix of large-p with small-n genomic regressions as a local smoother

Above we introduce the hat matrix as applied to the training dataset,

$\hat{\mathbf{y}}_{TRN} = \mathbf{X}_{TRN} \hat{\boldsymbol{\beta}}_{RR} = \mathbf{X}_{TRN} [\mathbf{X}'_{TRN} \mathbf{X}_{TRN} + \lambda \mathbf{D}]^{-1} \mathbf{X}'_{TRN} \mathbf{y}_{TRN} = \mathbf{H}_{TRN} \mathbf{y}_{TRN}$ . Similarly, we can define a hat matrix for the testing dataset,  $\hat{\mathbf{y}}_{TST} = \mathbf{X}_{TST} \hat{\boldsymbol{\beta}}_{RR} = \mathbf{X}_{TST} [\mathbf{X}'_{TST} \mathbf{X}_{TST} + \lambda \mathbf{D}]^{-1} \mathbf{X}'_{TST} \mathbf{y}_{TST} = \mathbf{H}_{TST} \mathbf{y}_{TST}$ . In both cases, predictions are simply weighted sums of phenotypes of the training dataset,

$$\hat{y}_{TRN,i} = \sum_{j \in TRN} h_{TRN,ij} y_j \text{ and } \hat{y}_{TST,i} = \sum_{j \in TST} h_{TST,ij} y_j, \text{ where } h_{ij} \text{ is the } (i,j)^{th} \text{ entry of either } \mathbf{H}_{TRN} \text{ or } \mathbf{H}_{TST}.$$

The relative absolute value of each entry,  $|h_{ij}|$ , indicates, according to the model, how informative the  $j^{th}$  phenotype of the training dataset is for estimating the conditional expectation at the  $i^{th}$  point of either the training or testing dataset. The following code computes the hat matrix a training and testing dataset and plots the one of the rows of  $\mathbf{H}_{TRN}$  and of  $\mathbf{H}_{TST}$ .

### Example 3. The Hat Matrix of Ridge Regression

```

rm(list=ls())
##### DATA #####
library(BLR)
data(wheat)
objects()
N<-nrow(X) ; p<-ncol(X)
y<-Y[,2]
set.seed(1235)
tst<-sample(1:N,size=150,replace=FALSE)
XTRN<-X[-tst,]
yTRN<-y[-tst]
XTST<-X[tst,]
yTST<-y[tst]

## FITTING THE MODEL
lambda<-200
ZTRN<-cbind(1,XTRN)
ZTST<-cbind(1,XTST)

C<-crossprod(ZTRN)
for(j in 2:ncol(C)){ C[j,j]<-C[j,j]+lambda}
CInv<-chol2inv(chol(C))
TMP<-tcrossprod(CInv,ZTRN)

HTRN<-ZTRN%*%TMP
HTST<-ZTST%*%TMP
yHatTRN<-HTRN%*%yTRN
yHatTST<-HTST%*%yTST

## Plot of row 100 of HTRN
plot(abs(HTRN[100,]),xlab=' j (TRN)',
     ylab='h(100 , j)',col=2,main='Training dataset');abline(v=100)

## Plot of row 30 of HTST
plot(abs(HTST[30,]),xlab=' j (TRN)',
     ylab='h(30 , j)',col=2,main='Testing dataset')

```

## 2.5. Bayesian View of Ridge Regression

Most penalized can be viewed as posterior modes in certain class of Bayesian models. For instance, RR estimates are equivalent to the posterior mode of the vector of regression coefficients in a Bayesian model with a Gaussian likelihood and a Gaussian prior for the vector of regression coefficients. To see this, recall that that estimates in RR are obtained as the solution to the following optimization problem:

$$\hat{\beta}_{RR} = \underset{\arg\min}{\left\{ (\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta'\mathbf{D}\beta \right\}}$$

Multiplying the objective function by -1/2 and switching from minimization to maximization do not affect the solution; therefore,

$$\hat{\beta}_{RR} = \underset{\arg \max}{\left\{ -\frac{1}{2} (\mathbf{y} - \mathbf{X}\beta)' (\mathbf{y} - \mathbf{X}\beta) - \lambda \frac{1}{2} \beta' \mathbf{D} \beta \right\}}$$

Let  $\lambda = \frac{\sigma_e^2}{\sigma_\beta^2}$  where,  $\sigma_e^2$  and  $\sigma_\beta^2$  are non-negative constants. Replacing above and dividing the objective function by  $\sigma_e^2$  maintains the solution unchanged, with this we get:

$$\hat{\beta}_{RR} = \underset{\arg \max}{\left\{ -\frac{1}{2\sigma_e^2} (\mathbf{y} - \mathbf{X}\beta)' (\mathbf{y} - \mathbf{X}\beta) - \frac{1}{2\sigma_\beta^2} \beta' \mathbf{D} \beta \right\}}$$

Finally, applying the exponential function to the objective function maintains the solution unchanged, therefore:

$$\begin{aligned} \hat{\beta}_{RR} &= \underset{\arg \max}{\left\{ \exp \left[ -\frac{1}{2\sigma_e^2} (\mathbf{y} - \mathbf{X}\beta)' (\mathbf{y} - \mathbf{X}\beta) - \frac{1}{2\sigma_\beta^2} \beta' \mathbf{D} \beta \right] \right\}} \\ &= \underset{\arg \max}{\left\{ \exp \left[ -\frac{1}{2\sigma_e^2} (\mathbf{y} - \mathbf{X}\beta)' (\mathbf{y} - \mathbf{X}\beta) \right] \exp \left[ -\frac{1}{2\sigma_\beta^2} \beta' \mathbf{D} \beta \right] \right\}} \end{aligned}$$

The first component of the objective function,  $\exp \left[ -\frac{1}{2\sigma_e^2} (\mathbf{y} - \mathbf{X}\beta)' (\mathbf{y} - \mathbf{X}\beta) \right]$ , is proportional to a

Gaussian likelihood, centered at  $\mathbf{X}\beta$  and with (co)variance matrix  $\mathbf{I}\sigma_e^2$ . The second component,

$\exp \left[ -\frac{1}{2\sigma_\beta^2} \beta' \mathbf{D} \beta \right]$ , is proportional a Gaussian prior for the regression coefficients, centered at zero

and with (co)variance matrix  $\mathbf{D}^{-1}\sigma_\beta^2$ . Therefore, estimates obtained with RR are equivalent to the posterior mode of regression coefficients in the following Bayesian model.

$$\begin{cases} \text{Likelihood: } [\mathbf{y} | \beta, \sigma_e^2] \sim N(\mathbf{X}\beta, \mathbf{I}\sigma_e^2) \\ \text{Prior: } [\beta | \sigma_\beta^2] \sim N(\mathbf{0}, \mathbf{D}^{-1}\sigma_\beta^2) \end{cases} \quad [4]$$

The posterior distribution of  $\beta$  is multivariate normal with a mean (co-variance matrix) equal to the solution (inverse of the coefficient matrix) of the following system:  $[\mathbf{X}'\mathbf{X} + \lambda\mathbf{D}] \hat{\beta} = \mathbf{X}'\mathbf{y}$ ; this is just the RR equations. This is also the Best Linear Unbiased Predictor (BLUP) of  $\beta$  given  $\mathbf{y}$ .

Recall that the ratio  $\frac{\sigma_e^2}{\sigma_\beta^2}$  is equivalent to  $\lambda$  in RR. In fully-Bayesian models we assign priors to

each of these variance parameters, this allows inferring these unknowns from the same training data that is used to estimate marker effects. The following example fits a Bayesian RR using the R-package BLR ('Bayesian Linear Regression'), after you run the model:

- The BLR package returns a list with posterior means and other information, type `str(fm)` and inspect what BLR returns
- Check the posterior mean of  $\sigma_e^2$  and  $\sigma_\beta^2$  (`fm$varE` and `fm$varBR`, respectively), remember the ratio of these variances is interpretable as  $\lambda$  in RR.
- Examine trace plots
- Compare prediction accuracy of the fully-Bayesian method versus RR.

#### Example 4. Bayesian Ridge Regression Using BLR

```

rm(list=ls())
##### DATA (same as Example 2) #####
library(BLR)
data(wheat)
objects()
N<-nrow(X) ; p<-ncol(X)
y<-Y[,2]
set.seed(12345)
tst<-sample(1:N,size=150,replace=FALSE)
XTRN<-X[-tst,]
yTRN<-y[-tst]
XTST<-X[tst,]
yTST<-y[tst]

## Fits the model
prior<-list(varE=list(df=4,S=1),varBR=list(df=5,S=.01))
fm<-BLR(y=yTRN,XR=XTRN,nIter=12000,burnIn=2000,prior=prior)

## Prediction Accuracy: Bayesian vs grid search
x<-scan(file="lambda.txt")
y<-scan(file="sqCorTST.txt")

plot(y~log(x),type="o",col=2,
      xlab=expression(paste(log(lambda))),ylab="Squared Corr.",
      ylim=c(0.1,.3))

abline(v= log(fm$varE/fm$varBR),col=4)
abline(h=cor(yTST,XTST%*%fm$bR)^2,col=4)

## trace plots
plot(scan("varE.dat"),type="o",col=2)
abline(h=fm$varE,col=4)
abline(v=200,col=4)

```

## 2.6. G-BLUP

Here we show the equivalence between estimates (posterior modes) derived from model [4] and the so-called G-BLUP ('Genomic Best Linear Unbiased Predictor', e.g., VanRaden, 2008). We show this, using [4] and properties of the multivariate-normal density that are outlined below.

### **Properties of Multivariate Normal Density**

Let  $\boldsymbol{\theta} = (\boldsymbol{\theta}'_1, \boldsymbol{\theta}'_2)'$  be a multivariate normal random vector with expectation  $E\begin{bmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}$  and (co)variance matrix  $Cov\begin{bmatrix} \boldsymbol{\theta}_1 \\ \boldsymbol{\theta}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix}$ .

(1) All **marginal densities are also normal densities**, specifically:

$$\boldsymbol{\theta}_1 \sim MVN(\boldsymbol{\theta}_1, \boldsymbol{\Sigma}_{11}) \text{ and } \boldsymbol{\theta}_2 \sim MVN(\boldsymbol{\theta}_2, \boldsymbol{\Sigma}_{22}).$$

The **conditional densities are also normal densities**, with mean and (co)variance matrices given by the following:

$$E[\boldsymbol{\theta}_1 | \boldsymbol{\theta}_2] = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1} (\boldsymbol{\theta}_2 - \boldsymbol{\mu}_2) \text{ and } E[\boldsymbol{\theta}_2 | \boldsymbol{\theta}_1] = \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21} \boldsymbol{\Sigma}_{11}^{-1} (\boldsymbol{\theta}_1 - \boldsymbol{\mu}_1). \quad [5]$$

$$Cov[\boldsymbol{\theta}_1 | \boldsymbol{\theta}_2] = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1} \boldsymbol{\Sigma}_{21} \text{ and } Cov[\boldsymbol{\theta}_2 | \boldsymbol{\theta}_1] = \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21} \boldsymbol{\Sigma}_{11}^{-1} \boldsymbol{\Sigma}_{12}. \quad [6]$$

Above,  $\mathbf{B}_{21} = \boldsymbol{\Sigma}_{21} \boldsymbol{\Sigma}_{11}^{-1} = \{b_{ij}\}$  and  $\mathbf{B}_{12} = \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1} = \{b_{ij}\}$  are matrix of regression coefficients of the  $i$ th on the  $j$ th random variable of  $\boldsymbol{\theta}$ .

The multivariate normal density is closed under linear operations in the sense that **linear combinations of MVN random variables** of the form  $\boldsymbol{\delta} = \boldsymbol{\alpha} + \mathbf{T}\boldsymbol{\theta}$  are **multivariate normal** random variables, with mean vector and (co)variance matrices given by the following:

$$E[\boldsymbol{\delta}] = \boldsymbol{\alpha} + \mathbf{T}E[\boldsymbol{\theta}] = \boldsymbol{\alpha} + \mathbf{T}\boldsymbol{\mu}, \quad [7]$$

and (co)variance matrix

$$Cov[\boldsymbol{\delta}] = \mathbf{T}Cov[\boldsymbol{\theta}]\mathbf{T}' = \mathbf{T}\boldsymbol{\Sigma}\mathbf{T}', \quad [8]$$

### **Best Linear Unbiased Predictor (BLUP)**

We are now ready to derive the conditional expectation of marker effects and of genomic values. The **conditional expectation is the best predictor in the mean-squared error sense**. Also, as we show here, in the context of model [4] the conditional expectations of marker effects and of genomic values are linear functions of data and are un-biased. Therefore, the conditional expectations of genomic values and of marker effects from model [4] are BLUP ('Best Linear Unbiased Predictor').

For ease of notation we omit the intercept and therefore in [4] we set  $\mathbf{D}$  equal to an identity matrix. The model is then described by:

$$\begin{cases} \text{Likelihood: } [\mathbf{y} | \boldsymbol{\beta}, \sigma_e^2] \sim N(\mathbf{X}\boldsymbol{\beta}, \mathbf{I}\sigma_e^2) \\ \text{Prior: } [\boldsymbol{\beta} | \sigma_\beta^2] \sim N(\mathbf{0}, \mathbf{I}\sigma_\beta^2) \end{cases} \quad [4b]$$

From [4b] and using [7] and [8], we obtain that the joint density of  $\mathbf{y}$  and  $\boldsymbol{\beta}$ :

$$[\mathbf{y} \ \boldsymbol{\beta}] \sim MVN\left[\mathbf{0}, \begin{bmatrix} \mathbf{X}\mathbf{X}'\sigma_\beta^2 + \mathbf{I}\sigma_e^2 & \mathbf{X}\sigma_\beta^2 \\ \mathbf{X}'\sigma_\beta^2 & \mathbf{I}\sigma_\beta^2 \end{bmatrix}\right] \quad [9]$$

Using [5] we get the BLUP of marker effects:

$$E[\boldsymbol{\beta} | \mathbf{y}, \sigma_e^2] = \mathbf{X}'\sigma_\beta^2 \underbrace{[\mathbf{X}\mathbf{X}'\sigma_\beta^2 + \mathbf{I}\sigma_e^2]^{-1}}_{\mathbf{Q}} \mathbf{y} = \mathbf{X}'[\mathbf{X}\mathbf{X}' + \lambda\mathbf{I}]^{-1} \mathbf{y} \quad [10]$$

which is the posterior mean of  $\boldsymbol{\beta}$ . Here,  $\lambda = \sigma_e^2 \sigma_\beta^{-2}$ . Because of the equivalence between the posterior mode of  $\boldsymbol{\beta}$  and the RR estimate, the solution given by [10] is also equivalent to the RR estimate given by [3]. Importantly, note that computing the solution using [3] requires inverting a  $p \times p$  matrix. On the other hand, we can obtain the same solution using [10] with inversion of  $n \times n$  matrix. Expression [10] is **linear** on data and it is **unbiased** with respect to the prior mean,  $E(\boldsymbol{\beta}) = \mathbf{0}$ . To see this we take expectations in [10] with respect to  $\mathbf{y}$  to get  $E\{E[\boldsymbol{\beta} | \mathbf{y}, \sigma_e^2]\} = \mathbf{X}'[\mathbf{X}\mathbf{X}' + \lambda\mathbf{I}]^{-1} E[\mathbf{y}]$ . From [9],  $E[\mathbf{y}] = \mathbf{0}$ ; therefore:  $E\{E[\boldsymbol{\beta} | \mathbf{y}, \sigma_e^2]\} = \mathbf{0}$ . Therefore, [10] gives the BLUP of marker effects.

We now derive the conditional expectation of genomic values given the data.

$$\begin{aligned} E[\mathbf{X}\boldsymbol{\beta} | \mathbf{y}, \sigma_e^2] &= \mathbf{X}E[\boldsymbol{\beta} | \mathbf{y}, \sigma_e^2] \\ &= \mathbf{X}\mathbf{X}'[\mathbf{X}\mathbf{X}' + \lambda\mathbf{I}]^{-1} \mathbf{y} \\ &= [\mathbf{I} + \lambda\mathbf{G}^{-1}]^{-1} \mathbf{y} \end{aligned} \quad [11]$$

Where  $\mathbf{G} = \mathbf{XX}'$ . This is the so-called G-BLUP of genomic values. Expression [11] is the best predictor of genomic value and it is linearly on data. Also, taking expectation with respect to phenotypes  $E \left\{ \left[ \mathbf{I} + \lambda \mathbf{G}^{-1} \right]^{-1} \mathbf{y} \right\} = \left[ \mathbf{I} + \lambda \mathbf{G}^{-1} \right]^{-1} E \left\{ \mathbf{y} \right\} = \mathbf{0}$ ; therefore [11] is the BLUP of genomic values.

The following example computes G-BLUP for the wheat dataset, and illustrate the equivalence with predictions from the RR.

### Example 5. Ridge Regression and G-BLUP

```

rm(list=ls())
#### DATA #####
library(BLR)
data(wheat)
for(i in 1:ncol(X)) {X[,i]<- (X[,i]-mean(X[,i]))}
y<-Y[,1]
h2<-0.5
lambda<-ncol(X)
#### Computing RR estimates and prediction using eq. [3] #####
C<-crossprod(X)
diag(C)<-diag(C)+lambda
CInv<-chol2inv(chol(C))
rhs<-crossprod(X,y)
sol<-crossprod(CInv, rhs)
yHat_1<-X%*%sol

#### GBLUP
G<-tcrossprod(X)
C<-chol2inv(chol(G))*lambda
diag(C)<-diag(C)+1
CInv<-chol2inv(chol(C))
yHat_2<-crossprod(CInv, y)

#### Comparison
plot(yHat_2~yHat_1,col=2,xlab='Predictions from RR equations',
      ylab='Predictions from GBLUP equations')

```

## References

- Frank, I.E., and J.H. Friedman. 1993. "A Statistical View of Some Chemometrics Regression Tools." *Technometrics*: 109–135.
- Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. 2nd ed. 2009. Corr. 3rd printing 5th Printing. Springer.
- Hoerl, A. E., and R. W. Kennard. 1970. "Ridge Regression: Biased Estimation for Nonorthogonal Problems." *Technometrics* 12 (1): 55–67.
- Tibshirani, R. 1996. "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society. Series B (Methodological)* 58 (1): 267–288.
- Zou, H., and T. Hastie. 2005. "Regularization and Variable Selection via the Elastic Net." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67 (2): 301–320.

# ***Statistical Methods for Genome-Enabled Prediction,***

## **Lab 3:**

### **The Bayesian Alphabet<sup>1</sup>**

( [gcampos@uab.edu](mailto:gcampos@uab.edu) )

## **Contents**

3.1. The Bayesian Alphabet .....	2
3.2. Ridge Regression Vs Bayesian Ridge Regression.....	9
3.3. Bayesian Lasso: fixed versus random lambda .....	11
3.4. Regression using markers and pedigree.....	13
References.....	14

**NOTE:** In many examples in this lab we use Bayesian methods. In those examples we make inferences based on a relatively small number of samples and this is done due to time constraints.  
In practice, accurate inferences require much more samples.

---

<sup>1</sup> Suggestions made by Daniel Gianola are gratefully acknowledged.

### 3.1. The Bayesian Alphabet

In standard parametric models for genomic selection (GS) phenotypes,  $y_i$ , are regressed on marker covariates,  $\{x_i\}$ , using a linear model of the form  $y_i = \mu + \sum_{j=1}^p x_{ij}\beta_j + \varepsilon_i$ , where  $\mu$  is an effect common to all subjects (i.e., an ‘intercept’),  $\{x_y\}$  are marker genotypes (usually coded as 0,1,2),  $\{\beta_j\}$  are marker effects and  $\varepsilon_i$  is a model residuals. A standard practice for continuous traits is to assume that model residuals are IID normal, this yields the following likelihood function:

$$\text{Likelihood: } p(\mathbf{y}|\mu, \boldsymbol{\beta}, \sigma^2) = \prod_{i=1}^n N(y_i | \mu + \sum_{j=1}^p x_{ij}\beta_j, \sigma^2), \quad [1]$$

where,  $N(y_i | \mu + \sum_{j=1}^p x_{ij}\beta_j, \sigma^2)$  is a normal density for the random variable  $y_i$  centered at  $\mu + \sum_{j=1}^p x_{ij}\beta_j$  and with variance  $\sigma^2$ .

With dense panels, the number of markers ( $p$ ) vastly exceeds the number of data points ( $n$ ) and because of this penalized or Bayesian shrinkage estimation methods are commonly used. In a Bayesian setting, shrinkage of estimates of effects is controlled by the choice of prior density assigned to marker effects. The joint prior density of the unknowns is commonly structured as follows:

**Prior:**

$$p(\mu, \boldsymbol{\beta}, \sigma^2 | df, S, \omega) \propto \left\{ \prod_{j=1}^p p(\beta_j | \boldsymbol{\theta}_{\beta_j}, \sigma^2) p(\boldsymbol{\theta}_{\beta_j} | \omega) \right\} \chi^{-2}(\sigma^2 | df, S) \quad [2]$$

Above, a flat prior was assigned to the intercept,  $\chi^{-2}(\sigma^2 | df, S)$  is a scaled-inverse Chi-squared density assigned to the residual variance and with  $df$  degree of freedom and scale equal to  $S$ ,  $p(\beta_j | \theta_{\beta_j}, \sigma^2)$  denotes the prior density of the  $j$ th marker effect,  $\theta_{\beta_j}$  is a vector of parameters indexing the prior density assigned to marker effects,  $p(\theta_{\beta_j} | \omega)$  is the prior density assigned to  $\theta_{\beta_j}$  and  $\omega$  are parameters indexing this density. The marginal prior density of marker effects is obtained by integrating  $\theta_{\beta_j}$  out,

$$p(\beta_j | \sigma^2, \omega) = \int p(\beta_j | \theta_{\beta_j}, \sigma^2) p(\theta_{\beta_j} | \omega) d\theta_{\beta_j}.$$

Note that, a-priori, all marker effects are assigned the same marginal prior density; therefore, contrary what it is sometimes said, in all members of the Bayesian alphabet, the prior variances of marker effects are the same for all markers.

Using Bayes rule, the posterior density of model unknowns given the data is proportional to the product of the likelihood, given in eq. [1], and the prior density, eq. [2], that is:

### **Posterior density:**

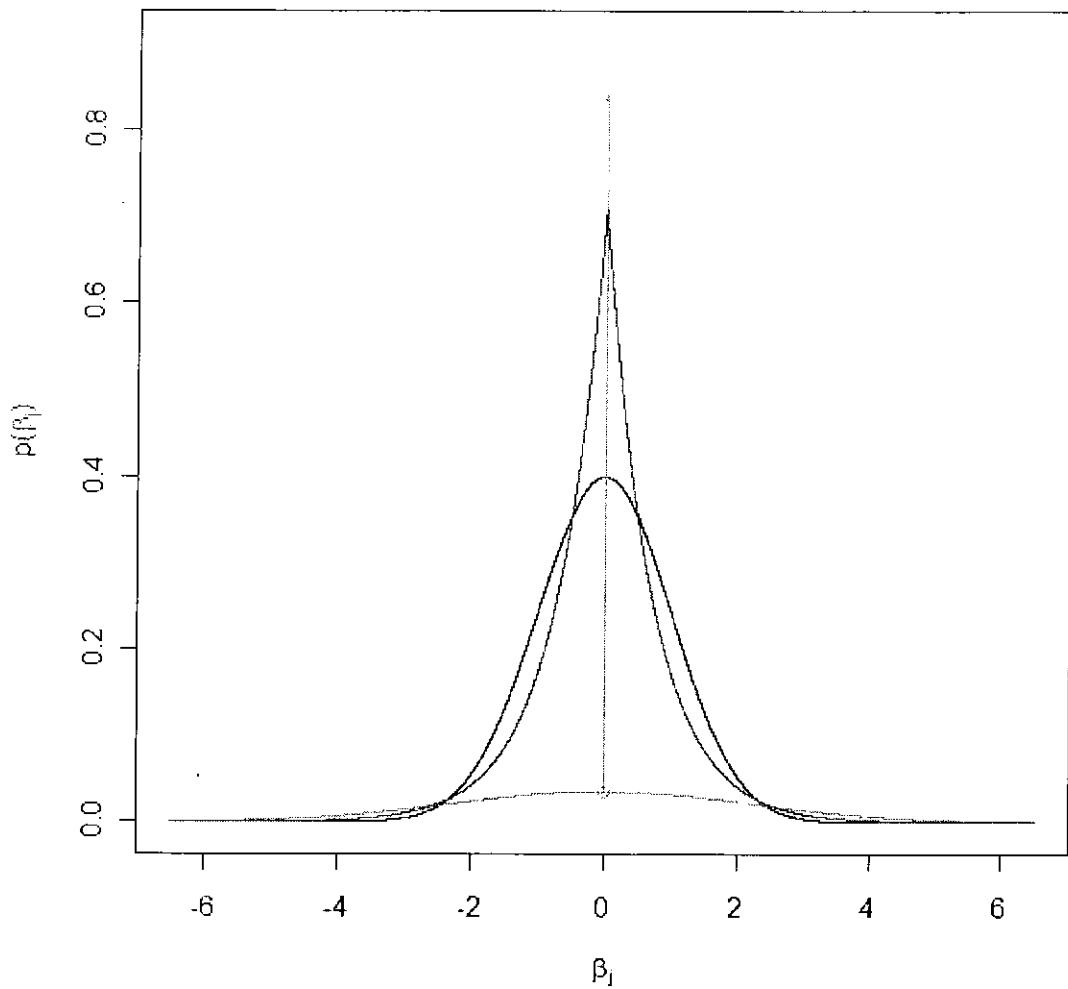
$$p(\mu, \beta, \sigma^2 | \mathbf{y}, df, S, \omega) \propto \prod_{i=1}^n N(y_i | \mu + \sum_{j=1}^p x_{ij} \beta_j, \sigma^2) \times \left\{ \prod_{j=1}^p p(\beta_j | \theta_{\beta_j}, \sigma^2) p(\theta_{\beta_j} | \omega) \right\} \chi^{-2}(\sigma^2 | df, S), \quad [3]$$

**The Bayesian Alphabet.** Following the seminal contribution of Meuwissen, Hayes, and Goddard (2001) several linear Bayesian regression methods have been proposed and used for simulation and real data analysis. They differed in the choice of prior density

assigned to marker effects. In a **Bayesian Ridge** regression (BRR), the conditional prior assigned of marker effects are IID normal,  $p(\beta_j | \theta_{\beta_j}, \sigma^2) = N(\beta_j | 0, \sigma_{\beta_j}^2)$  and  $p(\theta_{\beta_j} | \omega) = \chi^{-2}(\sigma_{\beta_j}^2 | df_{\beta_j}, S_{\beta_j})$ .

A second group of models, which includes **Bayes A** (Meuwissen, Hayes, and Goddard 2001) and the **Bayesian LASSO** (BL, Park and Casella 2008) use thick tail prior densities (t in Bayes A and Double Exponential in the BL). These priors induce a different type of shrinkage than that induced by the BRR.

A third group of models, which include Bayes B (Meuwissen, Hayes, and Goddard 2001) and the spike-slab models (Ishwaran and Rao 2005) use priors that are mixtures of a peak (or a spike) of mass at (in the vicinity of) zero and of a continuous density (e.g., t, or normal). Figure 1 shows the densities of a Gaussian and Double Exponential densities and that of a mixture model with a peak of mass at zero and a Gaussian slab. The three densities have mean equal to zero and variance equal to one.



**Figure 1.** Density of a standard normal random variable (black), of a double-exponential random variable (blue) and of a random variable following a mixture density with a mass point at zero (with probability 0.8) and a Gaussian process with probability 0.2. All variables with zero mean and variance equal to one.

Many of the thick tail distributions, such as the t or the double-exponential densities can be represented as infinite mixtures of scaled normal densities. For instance, the t-prior density assigned to marker effects in **Bayes A** (Meuwissen, Hayes, and

Goddard 2001) can be represented as  $t(\beta_j | df_{\beta_j}, S_{\beta_j}) = \int N(\beta_j | 0, \sigma_{\beta_j}^2) \chi^{-2}(\sigma_{\beta_j}^2 | df_{\beta_j}, S_{\beta_j}) d\sigma_{\beta_j}^2$

where  $df_{\beta_j}$  and  $S_{\beta_j}$  are prior degree of freedom and scale parameters and  $\chi^{-2}(\sigma_{\beta_j}^2 | df_{\beta_j}, S_{\beta_j})$  is a scaled-inverse Chi-squared density.

In the **Bayesian LASSO** (Park and Casella 2008) the Double-exponential prior density is represented as:  $DE(\beta_j | \lambda^2, \sigma_{\epsilon}^2) = \int N(\beta_j | 0, \sigma_{\epsilon}^2 \tau_j^2) Exp\left(\tau_j^2 | \frac{\lambda^2}{2}\right) d\sigma_{\beta_j}^2$ . In the fully-Bayesian LASSO,  $\lambda^2$  is treated as unknown and is assigned a Gamma prior. This prior is indexed by two parameters (rate and shape, see `help(rgamma)`) which are assumed to be known. Alternative priors for the regularization parameter are discussed in de los Campos et al. (2009).

In **Bayes B** (Meuwissen, Hayes, and Goddard 2001) marker effects are assumed to be equal to zero with probability  $\pi$  and with probability  $(1-\pi)$  the effect is assumed to be a draw from a t-distribution such as the one described in Bayes A. Model **Bayes C** (Habier et al. 2011) is similar to Bayes B but uses a Gaussian slab instead of the t-density used in Bayes B.

For infinitesimal traits, zeroing-out marker effects, such as in Bayes B or C, may harm predictive ability. Therefore, an alternative is to replace the peak of mass at zero used in Bayes B or C with a continuous density with small variance. This strategy is commonly used in what it is referred as to **Spike-Slab models** (Ishwaran and Rao 2005); for instance one can mix two Gaussian densities, one with very small variance and one with larger variance.

**Choosing hyper-parameters.** In the above mentioned models, the parameters indexing the prior density of marker effects play a central role in controlling the extent of

shrinkage of estimates of markers effect (similar to that of  $\lambda$  of the ridge regression. These parameters can be chosen in several ways, one of which is to select their values based on heritability-based rules.

**Choosing Hyper parameters using heritability based rules.** In linear models for genomic selection, genetic values are represented as regressions on marker covariates, that is  $g_i = \sum_j x_{ij} \beta_j$ . In these models, marker genotypes are fixed and marker effects are random variables drawn from an IID process; therefore:

$$Var(g_i) = \sum_j x_{ij}^2 Var(\beta_j) = \sigma_\beta^2 \sum_j x_{ij}^2$$

where  $\sigma_\beta^2$  is the prior variance of marker effects. Summing over individuals and dividing by  $n$  yields

$$\frac{h^2}{1-h^2} = \frac{\sigma_\beta^2}{\sigma_\epsilon^2} n^{-1} \sum_i \sum_j x_{ij}^2 = \frac{\sigma_\beta^2}{\sigma_\epsilon^2} K \quad [4]$$

where  $K = n^{-1} \sum_i \sum_j x_{ij}^2$  is the average sum of square of marker genotypes in the dataset,

and  $h^2$  is the heritability of the trait. Commonly, the model uses an intercept and we measure variance at the genomic values as deviations from the center of the sample. Therefor, a common practice is to compute  $K$  after centering genotypes, that is:

$K = n^{-1} \sum_i \sum_j (x_{ij} - 2\theta_j)$  where  $\theta_j$  is the frequency of the allele coded as one at the  $j$ th

marker. Moreover, if markers are centered and standardized to a unit variance, that is if

$\tilde{x}_{ij} = \frac{x_{ij} - 2\theta_j}{\sqrt{2\theta_j(1-\theta_j)}}$  are used as marker codes in the regression, then  $K$  equals the number

of markers ( $p$ ).

We can now use [4] to solve for the values of the parameters controlling regularization as a function of  $K$ ,  $h^2$  and of the phenotypic variance ( $\sigma_p^2$ ).

**Ridge Regression.** Recall from the Bayesian standpoint the regularization parameter of a ridge regression  $\lambda$  equals the ratio of the residual variance to the prior variance of marker effects,  $\sigma_\varepsilon^2 \sigma_\beta^{-2}$ . Replacing this in [4] and solving for  $\lambda$  we get

$$\frac{h^2}{1-h^2} = \frac{K}{\lambda} \Rightarrow \lambda = \frac{1-h^2}{h^2} K \quad [5]$$

Therefore, according to [5] the larger the noise-signal ratio, the strongest shrinkage of estimates should be. Also,  $K$  increases as the number of marker does; therefore, according to [5]  $\lambda$  should be increased as the number of markers does.

**Bayesian Ridge Regression.** In the Bayesian Ridge regression, instead of choosing  $\lambda$  we need to assign a prior to  $\sigma_\beta^2$  and to  $\sigma_\varepsilon^2$ . If these priors are scaled-inverse chi square, the prior expectations are:  $E(\sigma_\cdot^2 | df, S) = \frac{S_\cdot}{df_\cdot - 2}$  where  $(\cdot)$  equals  $\beta$  or  $\varepsilon$ . Typically we choose  $df$  to be a small value, usually greater than 4 to guarantee finite prior variance. Then, we can solve for  $S$  as a function of  $df$ ,  $K$ ,  $\sigma_p^2$  and  $h^2$ , so that the prior expectation of each of the variance components matches the value we expect according to  $\sigma_\beta^2$ ,  $h^2$  and [4],

specifically, equating  $\sigma_p^2(1-h^2)$  to  $E(\sigma_\varepsilon^2 | df, S)$  we get,

$\sigma_p^2(1-h^2) = E(\sigma_\varepsilon^2 | df, S) = \frac{S_\varepsilon}{df_\varepsilon - 2}$  and equating  $\sigma_p^2 h^2$  to  $K \times E(\sigma_\beta^2 | df_\beta, S_\beta)$  we get

$$S_\varepsilon = (1-h^2) \sigma_p^2 (df_\varepsilon - 2) \\ S_\beta = \frac{h^2 \sigma_p^2}{K} (df_\beta - 2) \quad [6]$$

**Bayes A.** The above formulas can also be used to define the scale parameters in Bayes B.

**Bayesian Lasso.** In this model, as originally formulated by (Park and Casella 2008), marker effects are assigned IID double-exponential priors with rate parameter,

$\frac{\lambda^2}{\sigma_e^2}$  (note,  $\lambda$  here is a different parameter than that of the ridge regression). The prior

variance of marker effects is:  $Var(\beta_j | \lambda^2, \sigma_e^2) = \sigma_\beta^2 = 2 \frac{\sigma_s^2}{\lambda^2}$ ; therefore,  $\frac{\sigma_\beta^2}{\sigma_e^2} = \frac{2}{\lambda^2}$ . Using

this in [4] we get:  $\frac{h^2}{1-h^2} = \frac{2}{\lambda^2} K$  or

$$\lambda = \sqrt{2 \frac{1-h^2}{h^2} K} \quad [7]$$

For the scale parameter of the residual variance we can use formula [6].

Note. The regularization parameter of the Bayesian Lasso is a function of the noise-signal ratio, and also of the number of markers. Specifically we expect  $K$  at a rate proportional to the square-root of the number of markers. The same occurs in RR (see [5]).

**Bayes B and C.** Here, the prior variance of marker effects are  $\sigma_\beta^2 = \frac{\sigma_A^2}{1-\pi}$  where

$\pi$  is the proportion of marker effects coming from the zero-state of the mixture and  $\sigma_A^2$  is the variance of the ‘slab’ (a Gaussian density in Bayes C and a t in Bayes B); therefore we can use the following formulas to chose the scale parameters as functions of  $df$ ,  $K$ ,  $\sigma_p^2$ ,  $h^2$  and  $\pi$ ,

$$S_e = \frac{(1-h^2)\sigma_p^2}{df_e - 2}, \quad S_\beta = \frac{h^2\sigma_p^2}{K(df_\beta - 2)(1-\pi)} \quad [8]$$

### 3.2. Ridge Regression Vs Bayesian Ridge Regression

In this section we compare estimates of marker effects derived from a ridge regression using lambda from eq. [5] with those obtained with a Bayesian Ridge Regression using

hyper-parameters chosen according to [6]. For the BRR we use the BLR package. Here, the prior is provided as a list. There is one component in the list for each of the variance parameters. In each component you need to provide prior degree of freedom and scale. For more details refer to `help(BLR)` or see ( Pérez et al. 2010).

### Example 1. Ridge regression Vs Bayesian Ridge Regression

```

rm(list=ls())
library(BLR)
data(wheat)
y<-Y[,2]
h2<-.2
df0<-5
for(i in 1:ncol(X)){ X[,i]<-(X[,i]-mean(X[,i]))/sd(X[,i]) }

K<-ncol(X) # after standardization, K=# of markers
lambda<-K*(1-h2)/h2
Se<- (1-h2)*var(y)*(df0-2)
Sb<-h2*var(y)*(df0-2)/K
round(Se/Sb,5)==lambda

## Ridge Regression
X2<-cbind(1,X)
C<-crossprod(X2)
for(i in 2:ncol(C)){ C[i,i]<- C[i,i]+lambda }
CInv<-chol2inv(chol(C))
rhs<-crossprod(X2,y)
bHat_RR<-crossprod(CInv,rhs)
yHat_RR<-X2%*%bHat_RR

## Bayesian Ridge Regression
library(BLR)
prior<-list(varE=list(df=df0,S=Se) , varBR=list(df=df0,S=Sb))
fmBRR<-BLR(y=y,XR=X,prior,
            nIter=13000,burnIn=3000, saveAt='BRR_')

fmBRR$varE/fmBRR$varBR
lambda

tmp<-range(c(bHat_RR[-1],fmBRR$bR))
plot(fmBRR$bR ~bHat_RR[-1],xlim=tmp,
     ylim=tmp, ,main='Estimates of Marker Effects',
     xlab='Ridge Regression', ylab='Bayesian Ridge Regression')
lines(x=c(-1,1),y=c(-1,1),col=2)

tmp<-range(c(yHat_RR,fmBRR$yHat))
plot(fmBRR$yHat~yHat_RR,xlim=tmp,ylim=tmp,main='Predictions',
     xlab='Ridge Regression', ylab='Bayesian Ridge Regression')
lines(x=c(-10,10),y=c(-10,10),col=2,lwd=2)
## Change the prior scale (e.g., double it) and evaluate the
## inferences

```

### 3.3. Bayesian Lasso: fixed versus random lambda

In this example we fit the Bayesian LASSO using BLR. The prior for parameter lambda of the BL has four arguments: `type`, `value`, `rate` and `shape`. If `type='fixed'` lambda is set equal to `value` and kept fixed. If `type='random'` lambda is treated as unknown; in this case a gamma prior is assigned to  $\lambda^2$  as described in Park and Casella (2008). For more details type `help(BLR)` in R or see Pérez et al. (2010). We chose values of the rate and shape parameters of the gamma prior so that the prior is flat in the neighborhood of the value of lambda we derive from eq. [4]. The following code displays the prior, run it and evaluates sensitivity with respect to rate and shape.

#### Example 2. Displaying prior of lambda of the BL

```
h2<-0.5
lambda0<-sqrt(2*K*(1-h2)/h2)
lambda<-seq(from=0,to=250,by=1)
dLambda<-2*lambda*dgamma(x=lambda^2,rate=1e-5,shape=0.53)
plot(dLambda~lambda, type='l')
abline(v=lambda0,col=2)

# change rate and shape and evaluate sensitivity of the prior
```

Bayesian Lasso is similar to Bayes A and  
differs from frequentist Lasso (via  
shrinkage)

Now we fit the BL with fix and random lambda.

### Example 3. Bayesian Lasso with fixed and random

```

rm(list=ls())
library(BLR)
data(wheat)
y<-Y[,2] ; h2<- .5
df0<-5
for(i in 1:ncol(X)){ X[,i]<- (X[,i]-mean(X[,i]))/sd(X[,i]) }

Se<- (1-h2)*var(y)*(df0-2)
lambda0<-sqrt(2*(1-h2)/h2*ncol(X))

## Bayesian Lasso fixed lambda #####
prior<-list(varE=list(df=df0,S=Se),
            lambda=list(value=lambda0,
                         type='fixed',rate=1e-5,shape=.53))

fmBL_fixed<-BLR(y=y,XL=X,prior=prior,
                  nIter=12000,burnIn=2000,saveAt='BL_fixed_')

fmBL_fixed$lambda
lambda0

tmp<-range(c(bHat_RR[-1],fmBL_fixed$bL))
plot(fmBL_fixed$bL ~ bHat_RR[-1],xlim=tmp,ylim=tmp)
lines(x=c(-1,1),y=c(-1,1),col=2)

tmp<-range(c(yHat_RR,fmBL_fixed$yHat))
plot(fmBL_fixed$yHat~yHat_RR,xlim=tmp,ylim=tmp)
lines(x=c(-10,10),y=c(-10,10),col=2,lwd=2)

## Now: change the value of lambda (e.g., 30 and 200) and
##       evaluate the impact on shrinkage of estimates

## Bayesian Lasso random lambda #####
prior$lambda$type='random'

fmBL_rand<-BLR(y=y,XL=X,prior=prior,
                  nIter=12000,burnIn=2000,saveAt='BL_rand_')

fmBL_rand$lambda
lambda0

tmp<-range(fmBL_rand$bL,fmBL_fixed$bL)
plot(fmBL_rand$bL ~ fmBL_fixed$bL,xlim=tmp,ylim=tmp)
lines(x=c(-1,1),y=c(-1,1),col=2)

tmp<-range(c(fmBL_rand$yHat,fmBL_fixed$yHat))
plot(fmBL_rand$yHat~fmBL_fixed$yHat,xlim=tmp,ylim=tmp)
lines(x=c(-10,10),y=c(-10,10),col=2,lwd=2)

```

### 3.4. Regression using markers and pedigree

So far we have regressed phenotypes on markers only. The following code gives an example of models with and without pedigree. In the wheat dataset, matrix A is an additive relationship matrix computed from the pedigree.

#### Example 4. Bayesian Lasso with & without pedigree

```
##### DATA #####
rm(list())
library(BLR)
data(wheat)
objects()
y<-Y[,2]
set.seed(1235)
tst<-sample(1:599,size=150,replace=FALSE)
yNA<-y
yNA[tst]<-NA

## Markers model
prior<-list(varE=list(df=df0,S=Se) ,
            lambda=list(value=lambda0,type='random',
                         rate=1e-5,shape=.53))

## Model with only markers
fmM<-BLR(y=yNA,XL=X,prior=prior,
           nIter=12000,burnIn=2000,saveAt='BL_M_')

prior$varU=list(df=df0,S=Se/3)
fmPM<-BLR(y=yNA,XL=X,prior=prior,GF=list(A=A,ID=1:599),
           nIter=12000,burnIn=2000,saveAt='BL_PM_')

fmPM$varE/fmM$varE
fmPM$lambda/fmM$lambda

cor(y[tst],fmM$yHat[tst])
cor(y[tst],fmPM$yHat[tst])

tmp<-range(c(fmM$bL,fmPM$bL))
plot(fmM$bL ~ fmPM$bL,xlim=tmp,ylim=tmp)
lines(x=c(-1,1),y=c(-1,1),col=2)

tmp<-range(c(fmPM$yHat,fmM$yHat))
plot(fmPM$yHat~fmM$yHat,xlim=tmp,ylim=tmp)
lines(x=c(-10,10),y=c(-10,10),col=2,lwd=2)
```

## References

- Habier, D., R. Fernando, K. Kizilkaya, and D. Garrick. 2011. "Extension of the Bayesian Alphabet for Genomic Selection." *BMC Bioinformatics* 12 (1): 186.
- Ishwaran, H., and J. S Rao. 2005. "Spike and Slab Variable Selection: Frequentist and Bayesian Strategies." *The Annals of Statistics* 33 (2): 730–773.
- Meuwissen, T H, B J Hayes, and M E Goddard. 2001. "Prediction of Total Genetic Value Using Genome-wide Dense Marker Maps." *Genetics* 157 (4) (April): 1819-1829.
- Park, T., and G. Casella. 2008. "The Bayesian Lasso." *Journal of the American Statistical Association* 103 (482): 681–686.
- Pérez, Paulino, Gustavo de los Campos, José Crossa, and Daniel Gianola. 2010. "Genomic-Enabled Prediction Based on Molecular Markers and Pedigree Using the Bayesian Linear Regression Package in R." *The Plant Genome Journal* 3 (2): 106-116. doi:10.3835/plantgenome2010.04.0005.

*Statistical Methods for Genome-Enabled Prediction,*

**Lab 4:**

**Semi-parametric Genomic Regression Using Reproducing**

**Kernel Hilbert Spaces Methods<sup>1</sup>**

( [gcampos@uab.edu](mailto:gcampos@uab.edu) )

**Contents**

4.1. Semi-parametric genome-enabled regression .....	2
4.2. Reproducing Kernel Hilbert Spaces (RKHS) regressions .....	2
4.3. Scatter plot smoothing with a Gaussian kernel.....	5
4.4. Inspecting the Hat Matrix .....	6
4.5. Bayesian view of RKHS .....	7
4.6. Genomic-Enabled Prediction Using RKHS .....	9
4.7. Kernel Averaging.....	12
4.8. Pedigree + Marker Models .....	15
References.....	17

NOTE: In many examples in this lab we use Bayesian methods. In those examples we make inferences based on a relatively small number of samples and this is done due to time constraints. In practice, accurate inferences require much more samples.

---

<sup>1</sup> Suggestions made by Daniel Gianola are gratefully acknowledged.

#### 4.1. Semi-parametric genome-enabled regression

In a standard regression model, the response,  $y_i$ , is expressed as the sum of a conditional expectation function,  $g(\mathbf{x}_i)$ , and a model residual,  $\varepsilon_i$ , that is  $y_i = g(\mathbf{x}_i) + \varepsilon_i$ . In previous labs we have focused on the case where  $g(\mathbf{x}_i)$  is a linear function of marker genotypes, that is  $g(\mathbf{x}_i) = \sum_{j=1}^p x_{ij}\beta_j$ . Departures from the linear model could theoretically be captured by extending the regression formula with addition of contrasts between marker genotypes, for instance dominance (i.e., within-loci interaction of alleles) could be modeled using dummy variables of the form  $d_{ij} = \{1 \text{ if } x_{ij} = 1; 0 \text{ otherwise}\}$ , and similar contrasts could be used to model interaction of alleles at different loci (i.e., epistasis). However, with large  $p$  the number of possible interaction terms needed to model even modest degree of interactions (e.g., 1st order epistatic interactions) is extremely large and the problem becomes intractable.

Alternatively, we could try to capture departures from the linear model using semi-parametric procedures. This was first suggested in the context of Genomic Selection (GS) by Gianola, Fernando, and Stella (2006) who propose implementing GS using various semi-parametric procedures. Since then, several existing semi parametric procedures have been evaluated in GS. In this lab we focus on Reproducing Kernel Hilbert Spaces (RKHS). Penalized Neural Networks are introduced in LAB 5.

#### 4.2. Reproducing Kernel Hilbert Spaces (RKHS) regressions

Reproducing kernel Hilbert spaces (RKHS) methods are used for semi-parametric modeling in different areas of application such as scatter-plot smoothing (e.g., smoothing spline, Wahba, 1990; spatial smoothing (e.g., Kriging, Cressie 1988); classification problems (e.g., support vector, Vapnik 1998), just to mention a few. Gianola, Fernando, and Stella (2006) suggested using this methodology for semi-parametric genomic enabled prediction. Since then, several authors have discussed and evaluated this methodology in a genomic context.

Estimates in RKHS can be motivated as solution to a penalized optimization problem in a RKHS of real-valued functions or, simply, as posterior modes in certain class of Bayesian models. Next, we provide an overview of the methodology. Detailed discussions of RKHS regressions in the context of genome-enabled prediction can be found in Gianola and van Kaam (2008), de los Campos, Gianola, and Rosa 2009) and de los Campos et al. (2010).

### ***Penalized Regression in Reproducing Kernel Hilbert Spaces***

In RKHS regressions we define the set of functions, or space, in which we perform the regression by choosing a reproducing kernel (RK). Technically, the RK can be any positive definite function<sup>2</sup> mapping from pairs of points in input space onto the real line, that is  $K(\mathbf{x}_i, \mathbf{x}_{i'}) : \{(\mathbf{x}_i, \mathbf{x}_{i'}) \rightarrow \mathbb{R}\}$ . For reasons that we will discuss later in this handout you can also think  $K(\mathbf{x}_i, \mathbf{x}_{i'})$  as a co-variance function. For example, if the input

---

<sup>2</sup>For  $K(\mathbf{x}_i, \mathbf{x}_{i'})$  to be positive semi definite it must satisfy  $\sum_i \sum_{i'} \alpha_i \alpha_{i'} K(\mathbf{x}_i, \mathbf{x}_{i'}) K(\mathbf{x}_i, \mathbf{x}_{i'}) \geq 0$  for every non-null sequence  $\{\alpha_i\}$ .

space consists of a pedigree additive relationships  $K(ID_i, ID_j) = \alpha(ID_i, ID_j)$  constitute a valid RK.

In RKHS regressions the evaluations of functions are expressed as linear combinations of the basis functions provided by the reproducing kernel, RK,  $K(\mathbf{x}_i, \mathbf{x}_{i'})$ , that is  $g(\mathbf{x}_i) = \sum_{i'} K(\mathbf{x}_i, \mathbf{x}_{i'}) \alpha_{i'}$ , and the squared of the norm of the function is given by

$$\|g\|^2 = \sum_i \sum_{i'} K(\mathbf{x}_i, \mathbf{x}_{i'}) \alpha_{i'}.$$

Stacking the evaluations of the function into a vector yields:  $\mathbf{g} = \mathbf{K}\boldsymbol{\alpha}$  and  $\|\mathbf{g}\|^2 = \boldsymbol{\alpha}' \mathbf{K} \boldsymbol{\alpha}$ , where  $\mathbf{g} = \{g_i\}$ ,  $\mathbf{K} = \{K_{ii'} = K(\mathbf{x}_i, \mathbf{x}_{i'})\}$  and  $\boldsymbol{\alpha} = \{\alpha_i\}$ .

Estimates in RKHS are usually obtained as the solution to the following penalized residual sum of squares (intercept and non-maker effects omitted for ease of notation):

$$\hat{\boldsymbol{\alpha}} = \underset{\arg \min}{\left\{ (\mathbf{y} - \mathbf{K}\boldsymbol{\alpha})' (\mathbf{y} - \mathbf{K}\boldsymbol{\alpha}) + \lambda \boldsymbol{\alpha}' \mathbf{K} \boldsymbol{\alpha} \right\}} \quad [1]$$

above,  $(\mathbf{y} - \mathbf{K}\boldsymbol{\alpha})' (\mathbf{y} - \mathbf{K}\boldsymbol{\alpha})$  is a residual sum of squares,  $\boldsymbol{\alpha}' \mathbf{K} \boldsymbol{\alpha}$  is a penalty on model complexity, which is taken to be the square of the norm of the function and  $\lambda$  is a regularization parameters.

The solution to the above optimization problem can be shown to be:

$$\hat{\boldsymbol{\alpha}} = [\mathbf{K}' \mathbf{K} + \lambda \mathbf{K}]^{-1} \mathbf{K}' \mathbf{y}. \quad [2]$$

Predictions are then obtained as follows:

$$\mathbf{K}\hat{\boldsymbol{\alpha}} = \mathbf{K}[\mathbf{K}' \mathbf{K} + \lambda \mathbf{K}]^{-1} \mathbf{K}' \mathbf{y} = [\mathbf{I} + \lambda \mathbf{K}^{-1}]^{-1} \mathbf{y}; \quad [3]$$

therefore,  $\mathbf{K}[\mathbf{K}' \mathbf{K} + \lambda \mathbf{K}]^{-1} \mathbf{K}' = [\mathbf{I} + \lambda \mathbf{K}^{-1}]^{-1}$  is the Hat matrix of RKHS.



Model specification in RKHS regression is defined by two main elements<sup>3</sup>: the choice of the reproducing kernel, this functions provide the basis functions and the inner product which define the Hilbert Space, and  $\lambda$  which, as in ridge regression, represents a shrinkage parameter.

#### 4.3. Scatter plot smoothing with a Gaussian kernel

In the following example we will use a RKHS regression to estimate a conditional expectation function non-parametrically. In the example, there is a single predictor,  $x_i \in [0, 2\pi]$  and the true conditional expectation function is  $g(x_i) = 120 + \sin(x_i)$ . Data was generated as  $y_i = 120 + \sin(x_i) + \varepsilon_i$  where  $\varepsilon_i \stackrel{IID}{\sim} N(0, 1)$ . With this setting, approximately 1/3<sup>rd</sup> of the variance of the response is explained by the conditional expectation function and 2/3<sup>rd</sup> by model residuals.

In this example we use the Gaussian kernel,

$$K(x_i, x_{i'}) = \exp\{-h \times d(x_i, x_{i'})\}$$

where:  $d(x_i, x_{i'})$  is a distance function which in this example we set to be a squared-Euclidean distance,  $d(x_i, x_{i'}) = (x_i - x_{i'})^2$ , and  $h$  is a bandwidth parameter controlling how fast the kernel decay as the two points,  $(x_i, x_{i'})$ , get further apart. In the example we evaluate the effects of  $h$  (which defines the RK) and of  $\lambda$ .

---

<sup>3</sup> A third element pertains to the choice of the function used to measure model goodness/lack of fit to the training data. Here we focus on the case where lack of fit is measured by the residual sum of squares; other common choices are the negative of the log-likelihood, this allows modeling continuous, binary and other types of outcomes. For binary outcomes another popular choice is the hinge function, the support vector machine (Vapnik 1998) is a special case of RKHS where the loss-function is chosen to be a hinge function (Wahba 1990).

- Run the code with the values of  $h$  and  $\lambda$  given in the example.
- Set  $h=1/1000$ , this makes the kernel extremely global, and run the code.
- Set  $h=50$ , this makes the kernel extremely local, and run the code.
- Now fix  $h=1$  and change lambda, evaluate  $\lambda=200$ , then  $\lambda=1/100$ , evaluate results.

### Example 1. Scatter-plot smoothing with a Gaussian kernel

```
#### SIMULATION#####
set.seed(12345)
N<-200
x<-seq(from=0,to=2*pi,length=N)
signal<-sin(x)
error<-rnorm(N)
y<-signal+error
h<-1
lambda<-10
#### DISTANCE FUNCTION AND REPRODUCING KERNEL #####
D<-as.matrix(dist(x,method="euclidean")) ^2
K<-exp(-h*D)
diag(K)<-diag(K) + .001

#### FITTING THE MODEL #####
yStar<-y-mean(y)
KInv<-chol2inv(chol(K))
C<-KInv*lambda
diag(C)<-diag(C)+1
H<-chol2inv(chol(C)) # the Hat matrix
uHat<-H%*%(y-mean(y))

plot(y~x, main=paste("lambda=",lambda," h=",h,sep=""))
lines(x=x,y=signal,col=2,lwd=2)
lines(x=x,y=uHat+mean(y),col=4,lwd=2)

## want to make the function less local? set h=1/1000,
## want to make it extremely local? set h=100
## Now fix h=1 and change lambda = 200 then lambda= 1/100
```

$h \downarrow \rightarrow$  global  $\rightarrow$  linear

$h \uparrow \rightarrow$  local

lambda  $\downarrow \rightarrow$  less shrinkage

To only problem with RKHS  
is choosing  $h$  & lambda  
(you can obtain Lambda  
from R&L models).

#### 4.4. Inspecting the Hat Matrix

From eq. [3] predictions are obtained as  $\hat{\mathbf{y}} = [\mathbf{I} + \mathbf{K}^{-1}\lambda]^{-1}\mathbf{y} = \mathbf{H}\mathbf{y}$ , where,

$\mathbf{H} = \{h_{ij}\} = [\mathbf{I} + \mathbf{K}^{-1}\lambda]^{-1}$ , therefore,  $\hat{g}_i = \sum_j h_{ij} y_j$ . The following code displays the

entries of the hat matrix of Example 1. You can evaluate the impact of the bandwidth parameter on the weights by changing (in Example 1)  $h$ .

### Example 2. Displaying the entries of the Hat matrix in RKHS

```
#### SIMULATION#####
rm(list=ls())
set.seed(12345)
N<-200
x<-seq(from=0,to=2*pi,length=N)
signal<-sin(x)
error<-rnorm(N)
y<-signal+error
h<-1
lambda<-10
## DISTANCE FUNCTION AND REPRODUCING KERNEL #####
D<-as.matrix(dist(x,method="euclidean"))^2
K<-exp(-h*D)
diag(K)<-diag(K) + .001

#### Hat Matrix #####
yStar<-y-mean(y)
KInv<-chol2inv(chol(K))
C<-KInv*lambda
diag(C)<-diag(C)+1
H<-chol2inv(chol(C)) # the Hat matrix
## Plots the ith row of H #####
row<-50
plot(H[row,]~x, main="", xlab="x(j)", type="l", ylab="h(i,j)", col=2)
abline(v=x[row], col=4); abline(h=0)
```

## 4.5. Bayesian view of RKHS

The solution to the penalized RKHS regression (see eq. [1]) can be shown to be the same than the posterior mode of the vector of regression coefficients in the following Bayesian model:

$$\left\{ \begin{array}{l} \mathbf{y} = \mathbf{K}\boldsymbol{\alpha} + \boldsymbol{\varepsilon} \\ \left( \begin{array}{c} \boldsymbol{\varepsilon} \\ \boldsymbol{\alpha} \end{array} \middle| \sigma_{\varepsilon}^2, \sigma_{\alpha}^2 \right) \sim N \left[ \begin{array}{c} \mathbf{0}, \begin{pmatrix} \mathbf{I} \sigma_{\varepsilon}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{K}^{-1} \sigma_{\alpha}^2 \end{pmatrix} \end{array} \right] \end{array} \right.$$

[4]

where  $\lambda = \sigma_{\varepsilon}^2 \sigma_{\alpha}^{-2}$ . The proof of the equivalence between the posterior mode of  $\boldsymbol{\alpha}$  in the Bayesian model described in [4] and the solution given in [2] can be obtained following the same steps used in section 2.5 of LAB 2.

Further, changing variables in [4] from  $\mathbf{K}\boldsymbol{\alpha}$  to  $\mathbf{g} = \mathbf{K}\boldsymbol{\alpha}$ , and noting from the properties of the MVN density (see section 2.6 of LAB 2) that  $\mathbf{g} \sim MVN(\mathbf{0}, \mathbf{K}\sigma_g^2)$ , where  $\sigma_{\alpha}^2 = \sigma_g^2$ , we obtain an equivalent representation of [4],

$$\left\{ \begin{array}{l} \mathbf{y} = \mathbf{g} + \boldsymbol{\varepsilon} \\ \left( \begin{array}{c} \boldsymbol{\varepsilon} \\ \mathbf{g} \end{array} \middle| \sigma_{\varepsilon}^2, \sigma_g^2 \right) \sim N \left[ \begin{array}{c} \mathbf{0}, \begin{pmatrix} \mathbf{I} \sigma_{\varepsilon}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{K} \sigma_g^2 \end{pmatrix} \end{array} \right] \end{array} \right.$$

[5]

Therefore, from the Bayesian perspective, the evaluations of functions at points in the input space,  $\mathbf{g} = \{g(\mathbf{x}_i)\}$  are viewed as realizations from Gaussian process satisfying:

$\text{Cor}[g(\mathbf{x}_i), g(\mathbf{x}_{i'})] = \frac{K(\mathbf{x}_i, \mathbf{x}_{i'})}{\sqrt{K(\mathbf{x}_i, \mathbf{x}_i)K(\mathbf{x}_{i'}, \mathbf{x}_{i'})}}$ . Here, the RK  $K(\mathbf{x}_i, \mathbf{x}_{i'})$  is viewed as a (co)variance function which defines a notion of smoothness of the function with respect to points in the input space (genotypes in our case). A high value of  $\text{Cor}[g(\mathbf{x}_i), g(\mathbf{x}_{i'})]$  implies that, a-priori, we expect the function to behave smoothly when we jump from  $\mathbf{x}_i$  to  $\mathbf{x}_{i'}$ . At the same time, this means  $y_i$  is informative about  $g(\mathbf{x}_{i'})$  and that  $y_{i'}$  informs us something about  $g(\mathbf{x}_i)$ .

**Special cases.** Certain parametric models appear as special cases of RKHS regression. For instance, if our information set consists of a pedigree and  $\mathbf{K}$  is a matrix of additive relationship matrix, the model defined by [1] is equivalent to the infinitesimal additive model, the so-called **Animal Model**. The Bayesian ridge regression and GBLUP (see section 2.6 of LAB 2) is another example of a parametric model that can be represented as a RKHS, this is obtained by setting  $\mathbf{K} = \mathbf{X}\mathbf{X}'$ . These are examples where the RK is chosen so as to represent the types of patterns expected under a parametric model. Another alternative is to choose kernels based on their performance (e.g., predictive ability). In this lab we will focus on this second approach.

*A-*  
*Veda.*

#### 4.6. Genomic-Enabled Prediction Using RKHS

In this section we use the Gaussian kernel for genomic-enabled prediction. To this end, we replace the distance function by a genomic-distance. For instance, we can set

Idea: you can weight the distance between individual marker effects in iteration matrix.

$$d(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_j (x_{ij} - x_{i'j})^2 ; \text{ the Gaussian kernel becomes: } K(x_i, x_{i'}) = \exp\{-h \times d(\mathbf{x}_i, \mathbf{x}_{i'})\}.$$

The function `dist()` of R takes two arguments: `x` which should be a numeric vector or matrix, and `method`, which should be a string indicating the method for computing distances. By default the Euclidean distance is computed. Type `help(dist)` for further details. The function returns an object, which can be converted to an  $n \times n$  matrix, containing pairwise distance between the rows of  $\mathbf{X}$ .

The example below fits the model over a grid of values of the bandwidth parameter ( $h$ ) and evaluates the effect of it on goodness of fit, model complexity and predictive ability.

- Run the code;
- Evaluate how goodness of fit and predictive ability changes with  $h$
- How does  $\lambda = \frac{\sigma_e^2}{\sigma_g^2}$  changes with  $h$ ?

As the # of man increase the distance increase and we should be careful about choosing  $h$ .

### Example 3. RKHS for Genomic Prediction

```

rm(list=ls())
setwd('~/Dropbox/Armidale/')
load("PROGRAMS/RKHS/RKHS.rda")
library(BLR)
data(wheat)

#### DISTANCE MATRIX #####
D<-as.matrix(dist(X,method="euclidean"))^2
D<-D/mean(D)
h<-c(1e-2,.1,.4,.8,1.5,3,5)

#### GENERATES TESTING SET #####
set.seed(12345)
tst<-sample(1:599,size=100,replace=FALSE)
y<-Y[,4]
yNA<-y
yNA[tst]<-NA      ← missing phenotype for validation sample.

#### FITS MODELS #####
PMSE<-numeric() ; VARE<-numeric(); VARU<-numeric() ;
pD<-numeric(); DIC<-numeric()
fmList<-list()
for(i in 1:length(h)){
  print(paste('Working with h=',h[i],sep=''))
  # COMPUTES THE KERNEL
  K<-exp(-h[i]*D)
  # FITS THE MODEL
  prefix<- paste(h[i], "_", sep="")
  fm<-RKHS(y=yNA,K=list(list(K=K,df0=5,S0=2)),
            nIter=5000,burnIn=1000,df0=5,S0=2,saveAt=prefix)
  fmList[[i]]<-fm
  PMSE[i]<-mean((y[tst]-fm$yHat[tst])^2)
  VARE[i]<-fm$varE
  VARU[i]<-fm$K[[1]]$varU
  DIC[i]<-fm$fit$DIC
  pD[i]<-fm$fit$pD
}
R2<-1-PMSE/mean((y[tst]-mean(y[-tst]))^2)

#### PLOTS #####
plot(VARE~h,xlab="Bandwidth", ylab="Residual Variance",type="o",col=4)

plot(I(VARE/VARU)~h,xlab="Bandwidth",
      ylab="variance ratio (noise/signal)",type="o",col=4)

plot(pD~h,xlab="Bandwidth", ylab="pD",type="o",col=2)

plot(DIC~h,xlab="Bandwidth", ylab="DIC",type="o",col=2)

plot(R2~h,xlab="Bandwidth", ylab="R-squared",type="o",col=2)

```

Gibbs Sampler

$h \rightarrow 0 \rightarrow$  fit only intercept on  $\mathbb{A}$  &  $D \rightarrow \mathbb{J}$   
 output of Example 3

$h \uparrow$  Residual var ↓

$h \uparrow$   $\lambda \uparrow$

$h \uparrow$  effective number of parameters ↑ (although  $\lambda \uparrow$  but not proportional)

#### 4.7. Kernel Averaging

The choice of the RK (its functional form and the values of parameters such as the bandwidth) constitutes the central element of model specification in RKHS regressions.

There are several ways of choosing a kernel. In **parametric models**, the RK is chosen to represent the type of patterns expected under a particular parametric model (e.g., additive infinitesimal,  $\mathbf{K}=\mathbf{A}$ ; linear model,  $\mathbf{K}=\mathbf{XX}'$ ). From a **non-parametric** perspective one can choose kernels based on the performance of the model, e.g., predictive ability; an illustration of this was provided in the previous example where a validation set was used to evaluate predictive ability of RKHS using a Gaussian kernel, over a grid of values of the bandwidth parameter.

A third way is by **inferring the kernel** from the data. For instance, in a Bayesian context one could assign a prior to the bandwidth parameter and infer this parameter jointly with other unknowns. While this is appealing, it is computationally demanding for at least two reasons: (a) the RK must be re-computed every time a new value of the bandwidth parameter is sampled; (b) mixing may be poor. This occurs because, usually, variance parameters and the bandwidth parameter are highly correlated at the posterior distribution. An alternative which we consider here is to offer the algorithm all candidate kernels jointly. For instance, we can make the conditional expectation to be a sum of several random effects,  $\{\mathbf{g}_1, \dots, \mathbf{g}_{N_k}\}$ , each of which has its own (co)variance function, the model becomes:

$$\begin{cases} \mathbf{y} = \mathbf{1}\mu + \sum_{k=1}^{N_k} \mathbf{g}_k + \boldsymbol{\epsilon} \\ p(\boldsymbol{\epsilon}, \mathbf{g}_1, \dots, \mathbf{g}_{N_k} | \sigma_e^2, \sigma_{g_1}^2, \dots, \sigma_{g_{N_k}}^2) = N(\boldsymbol{\epsilon} | \mathbf{0}, \mathbf{I}\sigma_e^2) \prod_{k=1}^{N_k} N(\mathbf{g}_k | \mathbf{0}, \mathbf{K}_k \sigma_{g_k}^2) \end{cases}$$

It can be shown that, conditional on variance parameters, the above model is equivalent to one with a single random effect,  $\mathbf{g}$ , whose prior distribution is  $N(\mathbf{g} | \mathbf{0}, \bar{\mathbf{K}}\sigma_g^2)$

where:  $\bar{\mathbf{K}} = \mathbf{K}_1\alpha_1 + \mathbf{K}_2\alpha_2 + \dots + \mathbf{K}_{N_k}\alpha_{N_k}$  is a weighted sum of the candidate kernels with

weight given by  $\alpha_k = \frac{\sigma_{g_k}^2}{\sigma_g^2}$  and  $\sigma_g^2 = \sum_k \sigma_{g_k}^2$ . Variance parameter here can then be seen

as weights associated to each kernel which can be inferred from the data. The larger the variance associated to a given kernel the larger the contribution of that random effect to the conditional expectation. We refer to this approach as kernel averaging (KA, de los Campos et al., 2010).

The following example illustrates the use of KA; the sequence of kernels was generated using the Gaussian kernel and the values of the bandwidth parameter used in our previous example.

- Run the code below.
- What Kernel gets higher weight?
- Is that the Kernel that gave highest predictive ability in our previous example?
- Compare the predictive ability of KA with that of models fitted in our previous example (i.e., single kernel with fixed bandwidth).

#### Example 4. Kernel Averaging

```

rm(list=ls())
setwd('~/Dropbox/Armidale/') ; load("PROGRAMS/RKHS/RKHS.rda")

library(BLR)
data(wheat)
D<-as.matrix(dist(X,method="euclidean"))^2
D<-D/mean(D)
h<-c(1e-2,.1,.4,.8,1.5,3,5)

#### GENERATES TESTING SET #####
set.seed(12345)
tst<-sample(1:599,size=100,replace=FALSE)
y<-Y[,4]
yNA<-y
yNA[tst]<-NA

#### FITS MODELS #####
PMSE<-numeric()
VARE<-numeric()
KList<-list()
for(i in 1:length(h)){
  KList[[i]]<-list(K=exp(-h[i]*D),df0=5,S0=.5)
}

## Displays entries of different kernels
plot(KList[[1]]$K[100,],ylim=c(0,1),col=2);abline(v=100)

plot(KList[[5]]$K[100,],ylim=c(0,1),col=2);abline(v=100)

fmKA<-RKHS(y=yNA,K=KList,thin=10,
             nIter=25000,burnIn=5000,df0=5,S0=1,saveAt="KA_")

VARG<-numeric()
for(i in 1:length(KList)){ VARG[i]<-fmKA$K[[i]]$varU }
weights<-round(VARG/sum(VARG),5)

PMSE<-mean((y[tst]-fmKA$yHat[tst])^2)
R2_KA<-1-PMSE/mean((y[tst]-mean(y[-tst]))^2)

# compare with results obtained in the previous example
# take a look at the trace plots of variance parameters

```

use proper prior otherwise sometimes it not  
 converge  
 if kernel get zero it confused with residual

#### 4.8. Pedigree + Marker Models

The following code compares the entries of a pedigree-based additive relationship matrix versus that of two marker-based genomic relationships. The first one ( $XX'$ , denoted as  $XXt$ ) is the co-variance structure corresponding to a linear regression on marker-covariates with IID normal marker effects (what we have called the Bayesian Ridge Regression). The second one (denoted as  $K$ ) is a Gaussian kernel.

##### Example 5. Pedigree Vs marker based relationship matrices

```
rm(list=ls())
library(BLR)
setwd('~/Dropbox/Armidale/') ; load("PROGRAMS/RKHS/RKHS.rda")
data(wheat) ; for(i in 1:ncol(X)){ X[,i]<-(X[,i]-mean(X[,i]))/sd(X[,i]) }

D<-as.matrix(X,method='euclidean')^2
D<-D/mean(D)
K<-exp(-2*D)
G<-tcrossprod(X)/ncol(X)

## plot of entries of XXt versus A
tmpX<-as.vector(A)
tmpY<-as.vector(G)
tmp<-range(c(tmpX,tmpY))
plot(tmpY~tmpX,xlab='A',ylab='G',cex=0.3,col=2,xlim=tmp,ylim=tmp)
```

### Example 6. RKHS with markers and pedigree

```

rm(list=ls())
library(BLR)
setwd('~/Dropbox/Armidale/') ; load("PROGRAMS/RKHS/RKHS.rda")
data(wheat) ; for(i in 1:ncol(X)){ X[,i]<-(X[,i]-mean(X[,i]))/sd(X[,i]) }

#### Generates Testing Sets #####
set.seed(12345)
tst<-sample(1:599,size=100,replace=FALSE)
y<-Y[,4] ; yNA<-y; yNA[tst]<-NA; KList<-list()

#### First the pedigree-model #####
KList[[1]]<-list(K=A,df0=5,S0=.2)
fmP<-RKHS(y=yNA,K=KList,thin=10,
            nIter=6000,burnIn=1000,df=5,S0=1,saveAt="P_")
PMSE<- mean((y[tst]-fmP$yHat[tst])^2)
R2_P<-1-PMSE /mean((y[tst]-mean(y[-tst]))^2)

#### Now Markers #####
G<-tcrossprod(X)/ncol(X)
KList[[1]]<-list(K=G,df0=5,S0=.2)
fmM<-RKHS(y=yNA,K=KList,thin=10,
            nIter=6000,burnIn=1000,df=5,S0=1,saveAt="M_")
PMSE<- mean((y[tst]-fmM$yHat[tst])^2)
R2_M<-1-PMSE /mean((y[tst]-mean(y[-tst]))^2)

#### Now Markers and pedigree #####
KList[[1]]<-list(K=A,df0=5,S0=.1)
KList[[2]]<-list(K=G,df0=5,S0=.1)

fmPM<-RKHS(y=yNA,K=KList,thin=10,
             nIter=6000,burnIn=1000,df=5,S0=1,saveAt="PM_")
PMSE<- mean((y[tst]-fmPM$yHat[tst])^2)
R2_PM<-1-PMSE /mean((y[tst]-mean(y[-tst]))^2)

## Now Lets add xxt#xxt #####
KList[[1]]<-list(K=A,df0=5,S0=.1)
KList[[2]]<-list(K=G,df0=5,S0=.05)
KList[[3]]<-list(K=I(G^2),df0=5,S0=.05)

fmPM2<-RKHS(y=yNA,K=KList,thin=10,
              nIter=15000,burnIn=5000,df=5,S0=1,saveAt="PM2_")
PMSE<- mean((y[tst]-fmPM2$yHat[tst])^2)
R2_PM2<-1-PMSE /mean((y[tst]-mean(y[-tst]))^2)

library(graphics)
barplot(height=c(R2_P,R2_M,R2_PM,R2_PM2),
        names.arg=c('P','M','PM','PM2'), ylab='R-sq. TRN set',col=2)
## Take a look at trace plots of variance parameters

```

## References

- de los Campos, G., D. Gianola, G. J. M. Rosa, K. A Weigel, and J. Crossa. 2010. "Semiparametric Genomic-enabled Prediction of Genetic Values Using Reproducing Kernel Hilbert Spaces Methods." *Genetics Research* 92 (04): 295–308.
- de los Campos, G., D. Gianola, and G. J.M Rosa. 2009. "Reproducing Kernel Hilbert Spaces Regression: a General Framework for Genetic Evaluation." *Journal of Animal Science* 87 (6): 1883.
- Cressie, N. 1988. "Spatial Prediction and Ordinary Kriging." *Mathematical Geology* 20 (4): 405–421.
- Gianola, D., and J. B van Kaam. 2008. "Reproducing Kernel Hilbert Spaces Regression Methods for Genomic Assisted Prediction of Quantitative Traits." *Genetics* 178 (4): 2289.
- Gianola, Daniel, Rohan L. Fernando, and Alessandra Stella. 2006. "Genomic-Assisted Prediction of Genetic Value With Semiparametric Procedures." *Genetics* 173 (3) (July 1): 1761-1776. doi:10.1534/genetics.105.049510.
- Vapnik, V. N. 1998. "Statistical Learning Theory."
- Wahba, G. 1990. "Spline Methods for Observational Data." *SIAM: Philadelphia*.

*Statistical Methods for Genome-Enabled Prediction,*

**LAB 5:**

**Penalized Neural Networks<sup>1</sup>**

( [gcampos@uab.edu](mailto:gcampos@uab.edu) )

**Contents**

5.1. Introduction .....	2
5.2. Scatterplot smoothing using a penalized NN.....	4
5.3. Penalized Neural Network Using Pre-selected Markers.....	7
5.4. Penalized Neural Networks Using Marker-derived Basis Functions as Inputs .....	8
References .....	9

---

<sup>1</sup> Software and suggestions provided by Dr. Paulino Pérez are gratefully acknowledged.

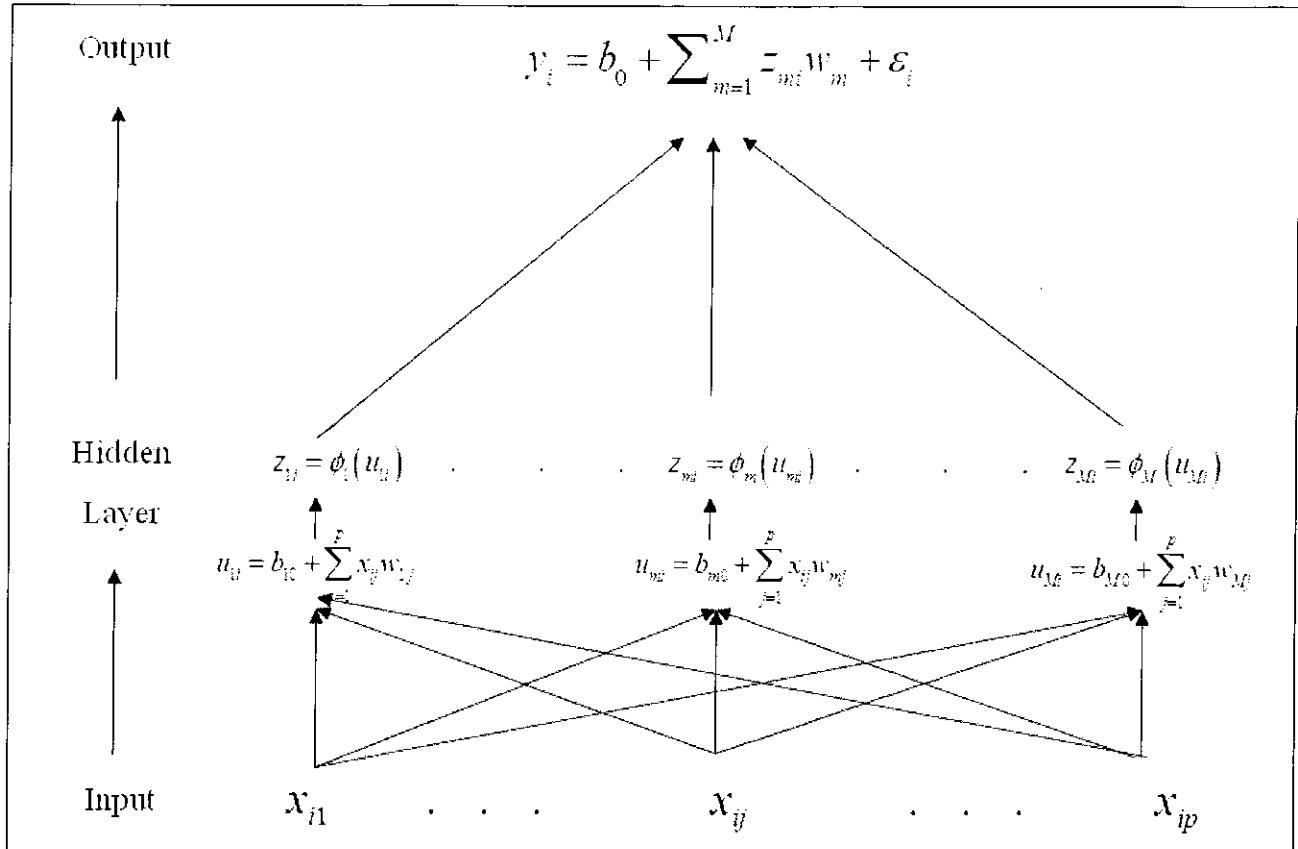
## 5.2. Introduction

In linear regression models the conditional expectation is represented as a weighted sum of input variables,  $E(y_i | \mathbf{x}_i) = \sum_{j=1}^n x_{ij} \beta_j$ . Many **non-linear patterns** can be represented linearly by appropriate choice of **basis functions**:  $E(y_i | \mathbf{x}_i) = \sum_{m=0}^M \phi_m(\mathbf{x}_i) \beta_m$  where,  $\{\phi_m(\mathbf{x}_i)\}_{m=1}^M$  are the basis functions, which map from the input variables onto the real line. An example of these are the polynomial basis functions:  $\Phi = \{\phi_m(x_i) = x_i^m\}_{m=0}^M$ . For instance, if  $M=2$  we have the 2<sup>nd</sup> degree polynomial basis functions,  $\Phi = \{1, x_i, x_i^2\}$ ; therefore,  $E(y_i | \mathbf{x}_i) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$ . Other common examples of non-linear basis functions are the power, logarithm and exponential functions. With this types of basis functions each of the regression coefficients affect the behavior of the conditional expectation in the entire input space, and this may limit the ability of a model to capture the local behavior of the conditional expectation.

**Local basis functions** can be used to model a conditional expectation within certain regions of the input space. **Splines** represent an example of this. In a spline, polynomial basis functions are used to represent the regression function within boundaries defined by a set of knots. The **Gaussian kernel** discussed in LAB4 is another example of a local basis function, here  $\phi_m(\mathbf{x}_i, \mathbf{t}_m, h) = e^{-h\|\mathbf{x}_i - \mathbf{t}_m\|^2}$  where  $\mathbf{t}_m$  is a focal point and  $h$  is a bandwidth parameter which controls how fast the basis function decay as  $\mathbf{x}_i$  gets further apart from the focal point. Model specification in this case pertains to the choice of focal points (how many and where in input space should be placed) and of the bandwidth parameter. In the RKHS regressions of LAB4, the strategy was to ‘offer’ the model a large set of basis functions (one per subject in the sample) generated by setting  $\mathbf{t}_1 = \mathbf{x}_1, \mathbf{t}_2 = \mathbf{x}_2, \dots, \mathbf{t}_n = \mathbf{x}_n$ ; therefore  $E(y_i | \mathbf{x}_i) = \sum_{i'=1}^n \alpha_{i'} \times e^{-h\|\mathbf{x}_i - \mathbf{x}_{i'}\|^2}$ . This strategy may induce over-fitting and this was confronted by using shrinkage estimation procedures. This approach is also used in smoothing spline (Craven and Wahba 1978; Wahba 1991).

Non-linear basis functions such as the ones described above offer great potential for capturing potentially complex patterns between input and output variables; however, the set of basis functions needs to be defined a-priori. In Neural Networks (NN) the basis functions used for regression are inferred (i.e., are data driven), this gives NN great potential for capturing potentially complex patterns.

One of the simplest NNs is the **single hidden layer feed-forward NN**. This NN can be thought as non-linear regressions consisting of two steps (Hastie, Tibshirani, and Friedman 2009): in the first one (or hidden layer) the basis functions are inferred, and in the second one (or output layer) the output,  $y_i$ , is regressed on the basis function inferred in the hidden layer. A graphical representation of such NN is given in Figure 1. The term feed-forward is used to highlight that in these NNs information flows from inputs (the  $\mathbf{x}_i$ 's) to output (the  $y_i$ 's), other NN allow feedbacks.



**Figure 1.** Graphical Representation of Single Hidden Layer Feed-Forward Neural Network for a Continuous Response ( $y_i$ ) and  $p$  predictor variables ( $x_{i1}, \dots, x_{ip}$ ). The network contains  $M$  neurons. At each neuron, linear combinations of the predictors ( $u_{mi} = b_{m0} + \sum_{j=1}^p x_{ij} w_{mj}$ ) are inferred and subsequently activated ( $z_{mi} = \phi_m(u_{mi})$ ). These basis functions are then used in the output layer to regress the output variable using a linear model ( $y_i = b_0 + \sum_{m=1}^M z_{mi} w_m + \varepsilon_i$ ).

As illustrated in Figure 1, in the **hidden layer**  $M$  basis functions,  $\phi_m\left(b_{m0} + \sum_{j=1}^p x_{ij} w_{mj}\right)$ , are inferred (one at each **neuron**). Each of these basis functions consist of a linear score,  $u_{mi} = b_{m0} + \sum_{j=1}^p x_{ij} w_{mj}$ , activated by a non-linear **activation function**,  $\phi_m(\cdot)$ .

In the **output layer**, the outcome,  $y_i$ , is regressed on the basis functions using an additive model. The example of Figure 1 is for a continuous response; in many applications with NN the outcome is either binary or polychotomous. In those cases an additional activation functions are added in the output layer.

Note that, if the activation function of the hidden and output layers are identity functions (i.e.,  $\varphi_m(u_{im}) = u_{im}$ ) the model of Figure 1 becomes a standard multiple linear regression model. Moreover, if we set the  $\varphi_m(\cdot)$  to be the basis functions of a reproducing kernel (see LAB4), the NN of Figure 1 becomes the RKHS regression. Therefore, we can view the NN of figure 1 as a general framework that includes the linear model and the RKHS as special cases.

The **activation functions** of the hidden layers map from the real line onto the [0,1] interval, and a common choice is to set this to be a sigmoid function. For instance we could use  $\phi_m(z_{mi}) = \frac{1}{1 + e^{-\theta \times z_{mi}}}$  for some  $\theta > 0$ .

**Architecture of a Neural Network.** The elements that define model specification in NN are: (a) the choice of input variables, (b) the type of network (e.g., feed-forward), (c) the number of layers, (d) the number of neurons per layer, and (d) the choice of activation functions. In general the term ‘architecture’ of the network is used to referred to the choices made in (b)-(d).

**Penalized Neural Networks.** The set of parameters to be estimated in the NN of Figure 1 include: all the intercepts and regression coefficients at each neurons, the parameters of the activation functions, and the intercept and regression coefficients of the output layer. With large  $p$ , and with several neurons, the total number of parameters to be estimated can be huge. This, together with the intrinsic flexibility of the NN, can easily yield over-fitting and poor predictive performance. To prevent this, a common strategy is to fit the neural network using penalized methods such as those discussed in LAB2. Therefore, in a penalized NN, parameters are estimated by minimizing an objective function consisting of a lack-of fit function (e.g., a residual sum of squares) plus a penalty on model complexity. Any of the penalties discussed in LAB 2 can be used; however, a common choice is to set the penalty to be the of regression coefficients (usually intercepts are not penalized).

In what remains of the lab we illustrate the use of penalized NN using a beta version of the R-package `trainbr`. This package was developed and kindly shared by Paulino Perez.

## 5.2. Scatterplot smoothing using a penalized NN

The following example illustrates the use of penalized NN for scatter-plot smoothing.

### Example 1: Scatter-plot smoothing Using a Neural Network

```
rm(list=ls());library(trainbr) ; library(splines)
### SIMULATION (same as the one used in Ex. 1 of LAB4) #####
set.seed(12345)
N<-200
x<-seq(from=0,to=2*pi,length=N)
signal<-sin(x)
error<-rnorm(N)
y<-signal+error

# for train-br the outcome variable needs to be standardized to [0,1]
yStd<-normalize(y) # NN assume your data is standard normal (between 0 & 1)
signalStd<-2*(signal-min(y))/(max(y)-min(y))-1
#Std

## Various parametric models
lm1<-lm(y~x)
poly3<-lm(yStd~x+I(x^2)+I(x^3))
## Natural spline with 4 knots
X<-ns(x=x,df=4)
fmNS<-lm(yStd~X)
## Neural Networks with 1,2,3 and 5 neurons
NN1<-trainbr(y=yStd,X=as.matrix(x),neurons=1)
yHatNN_1<-predictions.nn(X=as.matrix(x),theta=NN1$theta, neurons=1)

NN2<-trainbr(y=yStd,X=as.matrix(x),neurons=2)
yHatNN_2<-predictions.nn(X=as.matrix(x),theta=NN2$theta, neurons=2)

NN3<-trainbr(y=yStd,X=as.matrix(x),neurons=3)
yHatNN_3<-predictions.nn(X=as.matrix(x),theta=NN3$theta, neurons=3)

NN4<-trainbr(y=yStd,X=as.matrix(x),neurons=4)
yHatNN_4<-predictions.nn(X=as.matrix(x),theta=NN4$theta, neurons=4)

NN5<-trainbr(y=yStd,X=as.matrix(x),neurons=5)
yHatNN_5<-predictions.nn(X=as.matrix(x),theta=NN5$theta, neurons=5)

#(continues next page)
```

### Example 1: Scatter-plot smoothing Using a Neural Network

```

# (FROM PREVIOUS PAGE)
## R-Squared #####
R2_lm<-1-mean((signalStd-predict(lm1))^2)/var(signalStd)
R2_poly3<-1- mean((signalStd-predict(poly3))^2)/var(signalStd)
R2_NS<-1- mean((signalStd-predict(fmNS))^2)/var(signalStd)
R2_NN<-numeric()
R2_NN[1]<-1-mean((signalStd-yHatNN_1)^2)/var(signalStd)
R2_NN[2]<-1-mean((signalStd-yHatNN_2)^2)/var(signalStd)
R2_NN[3]<-1-mean((signalStd-yHatNN_3)^2)/var(signalStd)
R2_NN[4]<-1-mean((signalStd-yHatNN_5)^2)/var(signalStd)
R2_NN[5]<-1-mean((signalStd-yHatNN_5)^2)/var(signalStd)

## Plots #####
plot(yStd~x,col=1,cex=.5)
lines(x=x,y=signalStd,lwd=2,col=2)
lines(x=x,y=yHatNN_3,col=4,lwd=4,lty=2)

plot(R2_NN~I(1:5),
      xlab='Number of Neurons',ylab= 'R2(Pred. vs signal',type='o'
      , col=4)
abline(h=R2_NS,col=4,lty=2)

```

Example 1 illustrates the flexibility that NNs have in terms of capturing complex patterns: starting from a single predictor, the NN generated complexity by inferring multiple basis functions which were able to capture the non-linear patterns between inputs and outputs very well. The example uses a single predictor, but as illustrated in Figure 1 the method could also be applied to multiple-predictors. However, with large  $p$  and with multiple neurons, the computational requirements increase substantially.

### 5.3. Penalized Neural Network Using Pre-selected Markers

In Example 2 we first select the top p markers from single marker regressions and subsequently offer these markers to a NN with 3 neurons.

#### Example 2: Penalized Neural Network Applied to Pre-selected Markers

```
rm(list=ls())
### DATA #####
library(BLR) ; library(trainbr) ; data(wheat)
N<-nrow(X) ; p<-ncol(X)
y<-Y[,4]
y<-normalize(y)
set.seed(1235)
tst<-sample(1:N, size=150, replace=FALSE)
XTRN<-X[-tst,] ; yTRN<-y[-tst]
XTST<-X[tst,] ; yTST<-y[tst]
#### SINGLE MARKER REGRESSIONS #####
pValues<-numeric()
for(i in 1:p){
  fm<-lm(yTRN~XTRN[,i])
  pValues[i]<-summary(fm)$coef[2,4]
  print(paste('Fitting Marker ',i,'!',sep=''))
}
nMarkers<-75
selSNPs<-order(pValues)[1:nMarkers]
XTRN<-XTRN[,selSNPs]
XTST<-XTST[,selSNPs]

#### Neural Network #####
NN<-trainbr(y=yTRN, X=XTRN, neurons=4, epochs=100)
yHatNN<-predictions.nn(x=XTST, theta=NN$theta, neurons=4)
cor(yHatNN,y[tst])

## Change the number of pre-selected markers (line 22) and number of
## Neurons (lines 28 and 29) and experiment.
```

#### 5.4. Penalized Neural Networks Using Marker-derived Basis Functions as Inputs

In Example 2 we pre-selected markers, another strategy consist of first mapping the input information into some basis functions (e.g., using a reproducing kernel or using genomic relationships) and then applying the NN to these basis functions. For instance, Gianola et al. (2011) suggested using the additive relationships as basis functions, by so doing we reduce the number of input variables of the NN from  $p$  to  $n$ . In Example 3 we illustrate this approach by using as inputs to the NN marker-derived principal components.

#### Example 3: Penalized Neural Network Applied to Marker-derived Principal Components

```
rm(list=ls())
#### DATA #####
library(BLR); library(trainbr); data(wheat)
for(i in 1:ncol(X)){ X[,i]<-X[,i]-mean(X[,i])}
N<-nrow(X); p<-ncol(X)
y<-Y[,4]
y<-normalize(y)
## Pcs
SVD<-svd(X,nu=599,nv=0)
PC<-SVD$u; for(i in 1:ncol(PC)){ PC[,i]<-PC[,i]*SVD$d[i] }
plot(PC[,1:2],col=4)
set.seed(1235)
tst<-sample(1:N,size=150,replace=FALSE)
yTRN<-y[-tst]
yTST<-y[tst]
PCTrn<-PC[-tst,]

PCTst<-PC[tst,]

nPC<-300
NN<-trainbr(y=yTRN,X=PCTrn[,1:nPC],neurons=3, epochs=150)
yHatNN<-predictions.nn(X=PCTst[,1:nPC],theta=NN$theta,
                        neurons=c(length(NN$theta)-1))
cor(yHatNN,yTST)
```

## References

- Craven, P., and G. Wahba. 1978. "Smoothing Noisy Data with Spline Functions." *Numerische Mathematik* 31 (4): 377–403.
- Gianola, D., H. Okut, K. Weigel, and G. Rosa. 2011. "Predicting Complex Quantitative Traits with Bayesian Neural Networks: a Case Study with Jersey Cows and Wheat." *BMC Genetics* 12 (1): 87.
- Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. 2nd ed. 2009. Corr. 3rd printing 5th Printing. Springer.
- Wahba, G. 1991. "Spline Functions for Observational Data." *SIAM, Philadelphia, PA*.

# *Statistical Methods for Genome-Enabled Prediction,*

## LAB 6: Validation Methods<sup>1</sup>

( [gcampos@uab.edu](mailto:gcampos@uab.edu) )

### Contents

6.1. Introduction .....	2
6.2. Alternative Validation Schemes .....	3
6.3. Between sub-population prediction .....	7
6.4. Across environment prediction using single-trait models .....	8
References .....	9

**NOTE:** In many examples in this lab we use Bayesian methods. In those examples we make inferences based on a relatively small number of samples and this is done due to time constraints. In practice, accurate inferences require much more samples.

---

<sup>1</sup> Suggestions made by Daniel Gianola are gratefully acknowledged.

## 6.1. Introduction

Prediction is a central problem in plant and animal breeding and in many other domains. It is natural to compare models based on their ability to predict future outcomes. Validation methods aim at estimating the distribution (or features of it, e.g., the variance) of prediction errors.

**Prediction error.** Let  $S_{TRN} = \{y_i, \mathbf{x}_i\}$  denote the available training data,  $M$  a model (or algorithm) and  $\{y_{new}, \mathbf{x}_{new}\}$  an un-observed data point that we want to predict. The algorithm processes the training sample and derives a prediction:  $\hat{y}_{new}(\mathbf{x}_{new}, M, S_{TRN})$ . Example: using training data,  $S_{TRN}$ , and a linear model ( $M$ ) we estimate marker effects and then we use the estimated marker effects and the genotypes of candidates of selection ( $\mathbf{x}_{new}$ ) to derive predictions. The prediction error is  $\hat{\epsilon}_{new} = y_{new} - \hat{y}_{new}$ . Model performance can then be assessed using features of the distribution of prediction errors.

(Condition prediction error on training)

\* **Validation methods.** Deriving a closed form for the distribution of prediction errors requires making assumptions about the true data generating process. In practice we do not know such process and models are, at best, good approximations. However, if we are able to draw a large number of samples from the desired prediction errors  $\{\hat{\epsilon}_{new,i}\}$ , we can then estimate features of the density of prediction errors using Monte Carlo methods. For instance, given a large number of sample of prediction errors we could estimate the proportion of variance of future phenotypes accounted for by predictions

$$\text{using an R-squared type statistic: } R_{TST}^2 = 1 - \frac{\sum_i \hat{\epsilon}_{new,i}^2}{\sum_i (y_{new,i} - \bar{y}_{new})^2}.$$

In practice we have only a finite sample of data and most validation methods emulate the sampling process by sampling data points using some type of resampling method. There are different types of prediction errors, and the design of the validation scheme will determine what type of prediction errors are we describing.

**Conditional error.** Typically, we want to estimate the distribution of the prediction error given the training sample, that is,  $p(\hat{\epsilon}_{new} | S_{TRN})$ . Here, prediction errors are random variables because they are functions of yet-to-be-observed genotypes and phenotypes. Intuitively, we can obtain draws from the distribution of conditional errors by first fitting the model (only once) to the available TRN sample and subsequently evaluating the prediction accuracy of the model we derived by sampling testing samples.

**Marginal prediction errors** are obtained by averaging the density of conditional errors over all possible realizations of the training sample:  $p(\hat{\epsilon}_{new}) = E[p(\hat{\epsilon}_{new} | S_{TRN})] = \int p(\hat{\epsilon}_{new} | S_{TRN}) p(S_{TRN}) dS_{TRN}$ . Intuitively we can estimate the marginal distribution of prediction error with re-sampling of both training and testing datasets.

In most applications, our interest is to estimate the density of conditional errors; however this density is difficult to estimate and most of the methods we will see estimate  $p(\hat{\varepsilon}_{new})$  (Hastie, Tibshirani, and Friedman 2009).

## 6.2. Alternative Validation Schemes

### ***Training-Testing (TRN-TST) Validation***

If sample size is large we can simply assign some individuals for training (TRN) and some for testing (TST). We use TRN to fit the model and derive prediction errors from TST. We have done so in previous labs by partitioning at random the wheat dataset into TRN and TST. If the prediction problem of interest has certain structure (e.g., ancestors will be used for training with the goal of predicting performance of progeny) the partition of the data into TRN and TST should reflect such structure. This has been done, for instance for validation of methods for genomic selection in dairy cattle. Unfortunately we can't do this with the wheat dataset because we lack a pedigree.

### ***Cross-validation (CV)***

One disadvantage of the TRN/TST design above described is that individuals are either used for training or testing. When the total sample size is large this is not a problem; however, with small sample size one would like to use all individuals both for training and testing CV allows this. In CV individuals are randomly assigned to disjoint sets using an index, for example, in 2-fold CV each individual is assigned to either 1<sup>st</sup> or 2<sup>nd</sup> fold. Then, a TRN/TST evaluation is done for every fold. In those evaluations, individuals assigned to that fold are regarded as TST set and the remaining ones as TRN set. The following R-code implements a 5-fold CV using the wheat dataset.

### Example 1: 5-fold CV

```

#### LOADS DATA #####
rm(list=ls()); library(BLR); data(wheat)
y<-Y[,4]
for(i in 1:ncol(X)){ X[,i]<-(X[,i]-mean(X[,i]))/sd(X[,i]) }
h2<-0.5 ; lambda<-(1-h2)/h2*ncol(X)
#### ASSIGNMENT TO FOLDS (5-fold CV) #####
set.seed(124292)
sets<-sample(1:5,size=nrow(X),replace=TRUE)
yHatCV_RR<-rep(NA,length(y))
yHatCV_0<- rep(NA,length(y))
varE<-numeric()
indexH<-rep(NA,length(y))
for(fold in 1:5{
  tst<-which(sets==fold) # here we partition the data
  C<-crossprod(X[-tst,])
  for(j in 1:ncol(C)){ C[j,j]<- C[j,j]+lambda }

  CInv<-chol2inv(chol(C))
  H<-X[tst,]*%CInv%*%t(X[-tst,])
  indexH[tst]<-rowSums(abs(H)>.15) # count entries > 0.15 in H
  yHatCV_RR[tst]<- H%*%y[-tst]
  yHatCV_0[tst]<-mean(y[-tst])
  print(fold)
}

sqErrorRR<-(y-yHatCV_RR)^2
sqError0<-(y-yHatCV_0)^2

PMSE_RR<-tapply(X=sqErrorRR,FUN=mean,INDEX=sets)
PMSE_0<-tapply(X=sqError0,FUN=mean,INDEX=sets)
R2<-1-PMSE_RR/PMSE_0 # compare to cor(y,yHatCV)^2
sqrt(R2)

## Three different ways of computing R2: discuss!
cor(y,yHatCV_RR)^2
1-var(y-yHatCV_RR)/var(y)
1-sum((y-yHatCV_RR)^2)/sum((y-yHatCV_0)^2)

## Relationships between entries of hat matrix and pred. errors
tapply(FUN=mean,X=sqErrorRR,INDEX=indexH)

plot(sqErrorRR~indexH,ylab='Sq.Error',xlab='Index',col=2,cex=.5)

```

**NOTE 1.** While CV is commonly used in statistics and computer science, one needs to be aware that CV is not always an appropriate validation design. For instance, as previously mentioned, in breeding applications the prediction problem usually consists of inferring genetic values of candidates to selection. This prediction problem involves a generational order that is not considered in a standard CV with random assignment of individuals to folds. This may or may not induce biases, but one needs to be aware that CV is not the solution to any validation problem.

**NOTE 2.** The observed the variability in PMSE and R-squared across partitions of the CV reflects uncertainty associated to the sampling of TRN and TST sets. Evaluating such uncertainty is very important, especially when the number of records in the TRN and/or TST set is small. Note however, that ideally we would like to hold the training data fixed and evaluate the uncertainty associated to sampling of un-observed data (i.e., TST) only.

**NOTE 3.** We also observed that sq.-error diminishes as 'local sample size', measured, for example using the entries of the hat matrix, increases.

### ***Replicated Training-Testing***

In CV the number of folds affects the size of the training and testing datasets and the number of replicates of estimates of prediction accuracy. For instance, in a 5-fold CV the size of the TRN (TST) datasets is 80% (20%) of that of the available data and we only obtain 5 estimates of prediction accuracy (one per fold), this is a very small number if we wish to construct a confidence interval on estimates of prediction accuracy. An alternative is to replicate TRN-TST experiments a large number of times, each time re-assigning at random subjects into TRN and TST samples. The following R-code illustrates this with 30 replicates (example in next page).

### Example 3: Replicated TRN-TST partitions

```

rm(list=ls())
##### DATA #####
library(BLR)
data(wheat)
N<-nrow(X) ; p<-ncol(X)
for(i in 1:ncol(X)){ X[,i]<-(X[,i]-mean(X[,i]))/sd(X[,i]) }
y<-Y[,2]
nTst<-150
nRep<-30
set.seed(1235)
COR<-matrix(nrow=nRep, ncol=3, NA)
colnames(COR)<-c('lambda=10', 'lambda=1279', 'lambda=5000')
lambda<-c(10, 1279, 10000)

for(i in 1:nRep){
  print(paste('TRN-TST Replicate ', i, sep=''))
  tst<-sample(1:N, size=nTst, replace=FALSE)
  XTRN<-X[-tst,]
  yTRN<-y[-tst]
  XTST<-X[tst,]
  yTST<-y[tst]
  ZTRN<-cbind(1, XTRN)
  ZTST<-cbind(1, XTST)
  rhs<-crossprod(ZTRN, yTRN)
  C0<-crossprod(ZTRN)
  for(j in 1:3){
    C<-C0
    for(k in 2:ncol(C)){ C[k,k]<-C[k,k]+lambda[j] }
    CInv<-chol2inv(chol(C))
    sol<- CInv%*%rhs
    yHatTST<- ZTST%*%sol
    COR[i,j]<-cor(yTST, yHatTST)
  }
}
## Plots in next page
### PLOTS (Results from previous page)
## One way of looking at the problem (not quite correct)
x<-rep(lambda, nRep)
boxplot(as.vector(COR)~x, xlab=expression(paste(lambda)),
       ylab='Correlation')

## A better way
plot(y=COR[,2], x=COR[,1], xlim=range(COR), ylim=range(COR),
      xlab=expression(paste(lambda[10])),
      ylab=expression(paste(lambda[1279])), main='Correlation', col=2)
abline(a=0, b=1, col=4)

plot(y=COR[,3], x=COR[,2], xlim=range(COR), ylim=range(COR),
      xlab=expression(paste(lambda[1279])),
      ylab=expression(paste(lambda[10000])), main='Correlation', col=2)
abline(a=0, b=1, col=4)

```

### 6.3. Between sub-population prediction

So far we have assigned lines from training and testing completely at random. In this example we explore the impacts of training and validating in different subpopulations.

#### Example 3: Across sub-population prediction

```
rm(list=ls())
##### DATA #####
library(BLR)
data(wheat) ;
for(i in 1:ncol(X)){ X[,i]<- (X[,i]-mean(X[,i]))/sd(X[,i]) }

## Clustering based on q principal components
q<-2 # number of PCs used for clustering
for(i in 1:ncol(X)){X[,i]<-X[,i]-mean(X[,i])}
SVD<-svd(X,nu=q,nv=0)
myClusters<-kmeans(x=SVD$u%*%diag(SVD$d[1:q]),centers=2)

## Ploting principal components
tmp<-which(myClusters$cluster==1)
plot(x=SVD$u[tmp,1],y=SVD$u[tmp,2], ylim=range(SVD$u[,2]),
      xlim=range(SVD$u[,1]), col=2, xlab='1st PC', ylab='2nd PC' )
points(x=SVD$u[-tmp,1],y=SVD$u[-tmp,2],col=4)

## Fitting models
prior=list(varE=list(df=5,S=1),
           lambda=list(type='random',value=20,rate=1e-5,shape=.53))

group1<-myClusters$cluster==1
y<-Y[,4]
yNA1<-y
yNA1[which(group1)]<-NA
yNA2<-y
yNA2[which(!group1)]<-NA

## Training in sub-population 1
fm1<-BLR(y=yNA1,XL=X,nIter=7000,burnIn=2000,prior=prior,saveAt='1_')

# training in sub-population 2
fm2<-BLR(y=yNA2,XL=X,nIter=7000,burnIn=2000,prior=prior,saveAt='2_')

## Across group prediction
cor(X[which(group1),] %*% fm1$bL,y[which(group1)])
cor(X[which(!group1),] %*% fm2$bL,y[which(!group1)])

## Estimates of marker effects
plot(fm1$bL~fm2$bL,col=2)
```

## 6.4. Across environment prediction using single-trait models

In this example we address the problem of across environment (or trait prediction), this appear, for example when we want to select individuals based on expected performance in an environment in which these genotypes have not been evaluated. Most of the models we have discussed so far can be extended to accommodate multiple traits. Here, we explore the problem of prediction across correlated environments using single-trait models alone or combined using an ad-hoc procedure. A fully multi-environment evaluation of genome-enabled prediction methods for this dataset is presented in Burgueño et al. (2012).

### Example 4: Across environment prediction

```
rm(list=ls())
#####
# DATA #####
library(BLR)
data(wheat)
for(i in 1:ncol(X)){ X[,i]<- (X[,i]-mean(X[,i]))/sd(X[,i]) }
round(cor(Y), 3) #

prior=list(varE=list(df=5,S=1),
           lambda=list(type='random',value=20,rate=1e-5,shape=.53))

## Training models in environments 1-4
fm<-list()
for(i in 1:4){
  fm[[i]]<-BLR(y=Y[,i],XL=X,nIter=7000,burnIn=2000,
                 prior=prior,saveAt=paste('E_',i,sep=''))
}

## 1st strategy
COR<-matrix(nrow=4,ncol=4,NA)
colnames(COR)<-paste('TRN_',1:4,sep='')
rownames(COR)<-paste('TST_',1:4,sep='')
for(i in 1:4){
  for(j in 1:4){
    if(i!=j){ COR[i,j]<-cor(Y[,i],fm[[j]]$yHat) }
  }
}
## 2nd strategy (a bit of cheating)
covP<-cov(Y)
W<-matrix(ncol=4,nrow=4,0)
wCor<-rep(NA,4)
for(i in 1:4){
  W[i,-i]<-covP[i,-i] %*% solve(covP[-i,-i])
  TMP<-cbind(fm[[1]]$yHat,fm[[2]]$yHat,fm[[3]]$yHat,fm[[4]]$yHat)
  wCor[i]<-cor(Y[,i],TMP%*%W[i,])
}
## compare COR & wCor
```

## References

- Burgueño, J., G. de los Campos, K. Weigel, and J. Crossa. 2012. "Genomic Prediction of Breeding Values When Modeling Genotype\$\times\$ Environment Interaction Using Pedigree and Dense Molecular Markers." *Crop Science* 52 (2): 707.
- Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. 2nd ed. 2009. Corr. 3rd printing 5th Printing. Springer.

