Seattle Summer Institute 2007
Advanced QTL

Brian S. Yandell
University of Wisconsin-Madison

- Overview: Multiple QTL Approaches
- Bayesian QTL mapping & model selection
- data examples in detail
- software demo & automated strategy
- multiple phenotypes & microarrays

Real knowledge is to know the extent of one's ignorance.
Confucius (on a bench in Seattle)

contact information & resources

- email: byandell@wisc.edu
- web: www.stat.wisc.edu/~yandell/statgen
  - QTL & microarray resources
  - references, software, people
- thanks:
  - students: Jaya Satagopan, Pat Gaffney, Fei Zou, Amy Jin, W. Whipple Neely, Jee Young Moon
  - faculty/staff: Alan Attie, Michael Newton, Nengjun Yi, Gary Churchill, Hong Lan, Christina Kendziorski, Tom Osborn, Jason Fine, Tapan Mehta, Hao Wu, Samprit Banerjee, Daniel Shriner

Overview of Multiple QTL

1. What is the goal of multiple QTL study?
2. Gene action and epistasis
3. Bayesian vs. classical QTL
4. QTL model selection
5. QTL software options

1. what is the goal of QTL study?

- uncover underlying biochemistry
  - identify how networks function, break down
  - find useful candidates for (medical) intervention
  - epistasis may play key role
  - statistical goal: maximize number of correctly identified QTL
- basic science/evolution
  - how is the genome organized?
  - identify units of natural selection
  - additive effects may be most important (Wright/Fisher debate)
  - statistical goal: maximize number of correctly identified QTL
- select “elite” individuals
  - predict phenotype (breeding value) using suite of characteristics (phenotypes) translated into a few QTL
  - statistical goal: mimimize prediction error

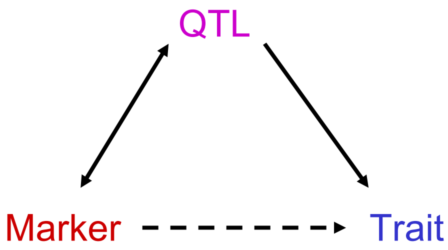
cross two inbred lines

→ linkage disequilibrium

→ associations

→ linked segregating QTL

(after Gary Churchill)



QTL

Marker

Trait

problems of single QTL approach

- wrong model: biased view
  - fool yourself: bad guess at locations, effects
  - detect ghost QTL between linked loci
  - miss epistasis completely
- low power
- bad science
  - use best tools for the job
  - maximize scarce research resources
  - leverage already big investment in experiment

advantages of multiple QTL approach

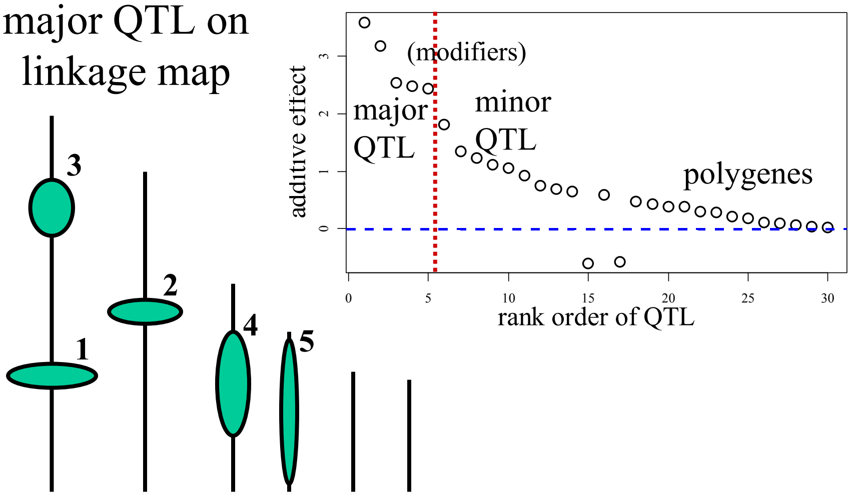
- improve statistical power, precision
  - increase number of QTL detected
  - better estimates of loci: less bias, smaller intervals
- improve inference of complex genetic architecture
  - patterns and individual elements of epistasis
  - appropriate estimates of means, variances, covariances
    - asymptotically unbiased, efficient
  - assess relative contributions of different QTL
- improve estimates of genotypic values
  - less bias (more accurate) and smaller variance (more precise)
  - mean squared error = MSE = (bias)^2 + variance

advantages of multiple QTL approach

- improve statistical power, precision
  - increase number of QTL detected
  - better estimates of loci: less bias, smaller intervals
- improve inference of complex genetic architecture
  - patterns and individual elements of epistasis
  - appropriate estimates of means, variances, covariances
    - asymptotically unbiased, efficient
  - assess relative contributions of different QTL
- improve estimates of genotypic values
  - less bias (more accurate) and smaller variance (more precise)
  - mean squared error = MSE = (bias)^2 + variance

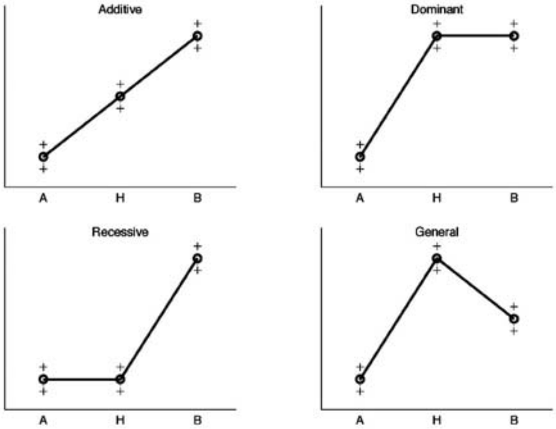
Pareto diagram of QTL effects

major QTL on linkage map



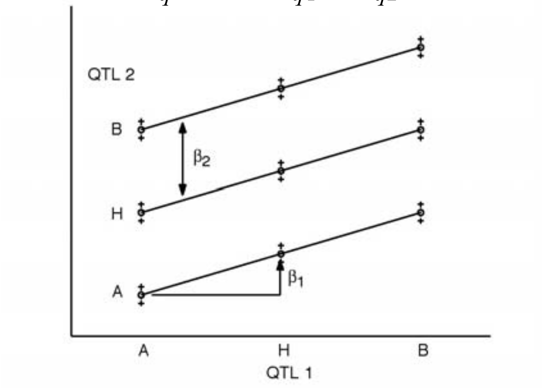
2. Gene Action and Epistasis

additive, dominant, recessive, general effects of a single QTL (Gary Churchill)



additive effects of two QTL (Gary Churchill)

$$\mu_q = \mu + \beta_{q1} + \beta_{q2}$$



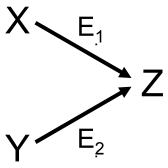
Epistasis (Gary Churchill)

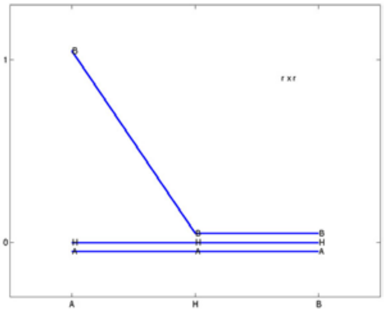
The allelic state at one locus can mask or uncover the effects of allelic variation at another.

- W. Bateson, 1907.

epistasis in parallel pathways (GAC)

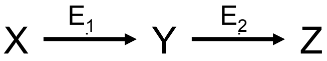
- Z keeps trait value low
- neither E_1 nor E_2 is rate limiting
- loss of function alleles are segregating from parent A at E_1 and from parent B at E_2

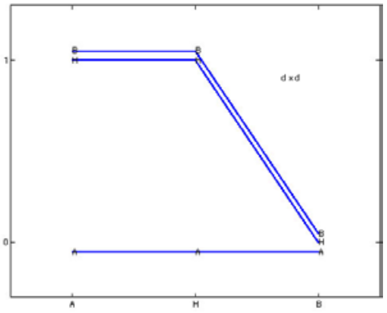




epistasis in a serial pathway (GAC)

- Z keeps trait value high
- neither E_1 nor E_2 is rate limiting
- loss of function alleles are segregating from parent B at E_1 and from parent A at E_2





epistatic interactions

- model space issues
  - 2-QTL interactions only?
    - or general interactions among multiple QTL?
  - partition of effects
    - Fisher-Cockerham or tree-structured or ?
- model search issues
  - epistasis between significant QTL
    - check all possible pairs when QTL included?
    - allow higher order epistasis?
  - epistasis with non-significant QTL
    - whole genome paired with each significant QTL?
    - pairs of non-significant QTL?
- see papers of Nengjun Yi (2000-7) in *Genetics*

limits of epistatic inference

- power to detect effects
  - epistatic model sizes grow quickly
    - $|A| = 3^{n.qtl}$ for general interactions
  - power tradeoff
    - depends sample size *vs.* model size
    - want $n / |A|$ to be fairly large (say > 5)
    - 3 QTL, $n = 100$ F2: $n / |A| \approx 4$
- rare genotypes may not be observed
  - *aa/BB* & *AA/bb* rare for linked loci
  - empty cells mess up balance
    - adjusted tests (type III) are wrong
  - confounds main effects & interactions

2 linked QTL
empty cell
with $n = 100$

	bb	*bB*	*BB*
aa	6	15	0
aA	15	25	15
AA	3	15	6

limits of multiple QTL?

- limits of statistical inference
  - power depends on sample size, heritability, environmental variation
  - “best” model balances fit to data and complexity (model size)
  - genetic linkage = correlated estimates of gene effects
- limits of biological utility
  - sampling: only see some patterns with many QTL
  - marker assisted selection (Bernardo 2001 *Crop Sci*)
    - 10 QTL ok, 50 QTL are too many
    - phenotype better predictor than genotype when too many QTL
    - increasing sample size may not give multiple QTL any advantage
  - hard to select many QTL simultaneously
    - 3^m possible genotypes to choose from

QTL below detection level?

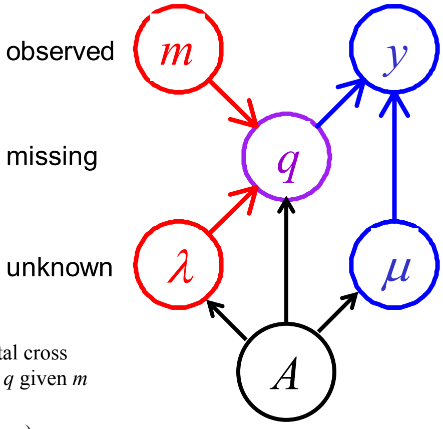
- problem of selection bias
  - QTL of modest effect only detected sometimes
  - effects overestimated when detected
  - repeat studies may fail to detect these QTL
- think of probability of detecting QTL
  - avoids sharp in/out dichotomy
  - avoid pitfalls of one “best” model
  - examine “better” models with more probable QTL
- rethink formal approach for QTL
  - directly allow uncertainty in genetic architecture
  - QTL model selection over genetic architecture

3. Bayesian vs. classical QTL study

- classical study
  - ***maximize*** over unknown effects
  - ***test*** for detection of QTL at loci
  - model selection in stepwise fashion
- Bayesian study
  - ***average*** over unknown effects
  - ***estimate*** chance of detecting QTL
  - sample all possible models
- both approaches
  - average over missing QTL genotypes
  - scan over possible loci

QTL model selection: key players

- observed measurements
  - y = phenotypic trait
  - m = markers & linkage map
  - i = individual index $(1,\ldots,n)$
- missing data
  - missing marker data
  - q = QT genotypes
    - alleles QQ, Qq, or qq at locus
- unknown quantities
  - λ = QT locus (or loci)
  - μ = phenotype model parameters
  - A = QTL model/genetic architecture
- pr($q|m,\lambda,A$) genotype model
  - grounded by linkage map, experimental cross
  - recombination yields multinomial for q given m
- pr($y|q,\mu,A$) phenotype model
  - distribution shape (assumed normal here)
  - unknown parameters μ (could be non-parametric)



after
Sen Churchill (2001)

Bayes posterior vs. maximum likelihood

- LOD: classical Log ODds
  - maximize likelihood over effects μ
  - `R/qtl scanone/scantwo: method = "em"`
- *LPD*: Bayesian *L*og *P*osterior *D*ensity
  - average posterior over effects μ
  - `R/qtl scanone/scantwo: method = "imp"`

$$\text{LOD}(\lambda) = \log_{10}\{\max_{\mu} \text{pr}(y \mid m, \mu, \lambda)\} + c$$

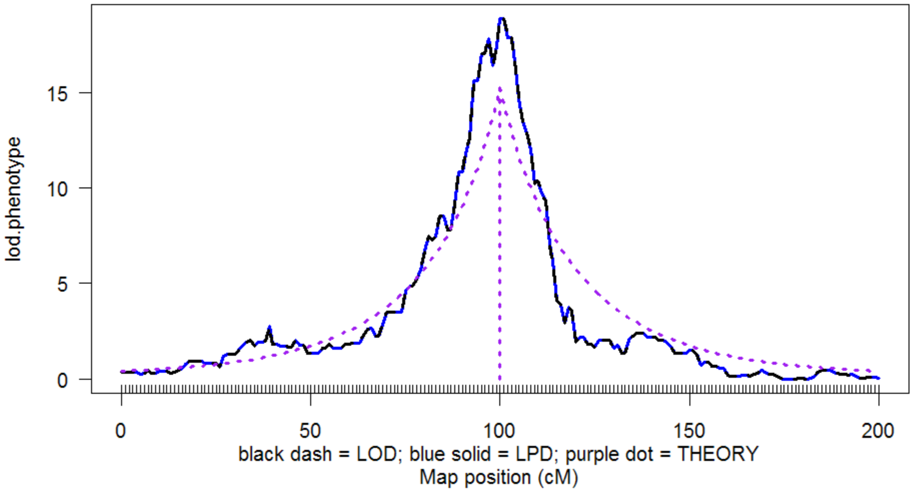
$$\text{LPD}(\lambda) = \log_{10}\{\text{pr}(\lambda \mid m) \int \text{pr}(y \mid m, \mu, \lambda)\text{pr}(\mu) d\mu\} + C$$

likelihood mixes over missing QTL genotypes:

$$\text{pr}(y \mid m, \mu, \lambda) = \sum_q \text{pr}(y \mid q, \mu)\text{pr}(q \mid m, \lambda)$$

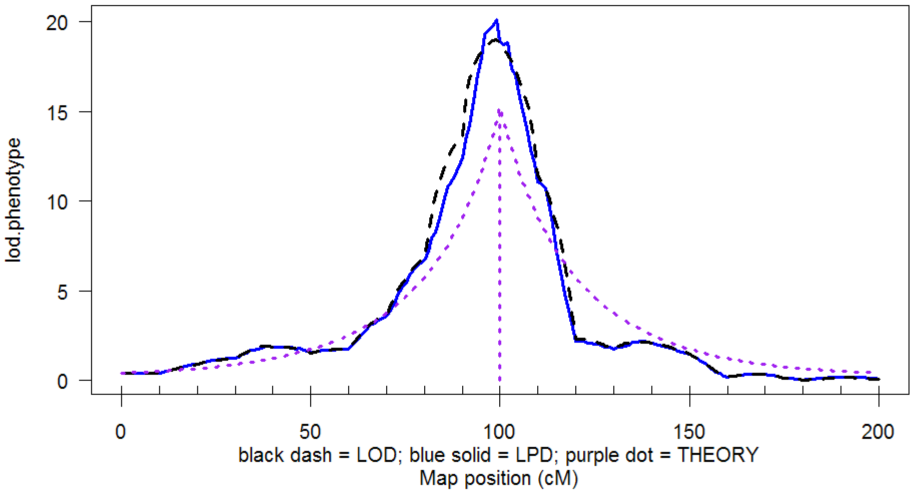
LOD & LPD: 1 QTL

n.ind = 100, 1 cM marker spacing



LOD & LPD: 1 QTL

n.ind = 100, 10 cM marker spacing



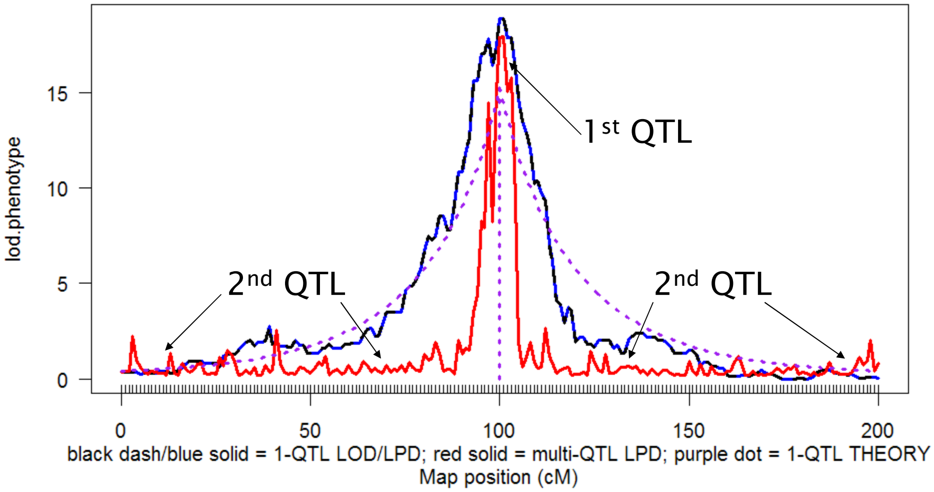
marginal LOD or LPD

- compare two architectures at each locus
  - with (A_2) or without (A_1) another QTL at separate locus λ_2
    - preserve model hierarchy (e.g. drop any epistasis with QTL at λ_2)
  - with (A_2) or without (A_1) epistasis with second locus λ_2
- allow for multiple QTL besides locus being scanned
  - allow for QTL at all other loci λ_1 in architecture A_1
- use marginal LOD, LPD or other diagnostic
  - posterior, Bayes factor, heritability

$$\mathrm{LOD}(\lambda_1, \lambda_2 \mid A_2) - \mathrm{LOD}(\lambda_1 \mid A_1)$$
$$\mathrm{LPD}(\lambda_1, \lambda_2 \mid A_2) - \mathrm{LPD}(\lambda_1 \mid A_1)$$

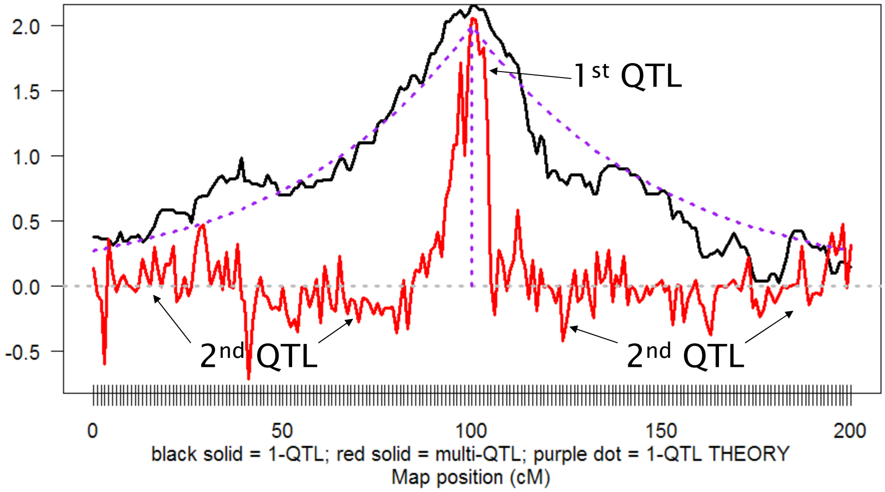
LPD: 1 QTL vs. multi-QTL

marginal contribution to LPD from QTL at λ



substitution effect: 1 QTL vs. multi-QTL

single QTL effect vs. marginal effect from QTL at λ



4. QTL model selection

- select class of models
  - see earlier slides above
- decide how to compare models
  - coming below
- search model space
  - see Bayesian QTL mapping & model selection talk
- assess performance of procedure
  - some below
  - see Kao (2000), Broman and Speed (2002)
  - be wary of HK regression assessments

pragmatics of multiple QTL

- evaluate some objective for model given data
  - classical likelihood
  - Bayesian posterior
- search over possible genetic architectures (models)
  - number and positions of loci
  - gene action: additive, dominance, epistasis
- estimate “features” of model
  - means, variances & covariances, confidence regions
  - marginal or conditional distributions
- art of model selection
  - how select “best” or “better” model(s)?
  - how to search over useful subset of possible models?

comparing models

- balance model fit against model complexity
  - want to fit data well (maximum likelihood)
  - without getting too complicated a model

	smaller model	bigger model
fit model	miss key features	fits better
estimate phenotype	may be biased	no bias
predict new data	may be biased	no bias
interpret model	easier	more complicated
estimate effects	low variance	high variance

information criteria
to balance fit against complexity

- classical information criteria
  - penalize likelihood L by model size $|A|$
  - IC = $-2 \log L(A \mid y) + \text{penalty}(A)$
  - maximize over unknowns
- Bayes factors
  - marginal posteriors pr($y \mid A$)
  - average over unknowns

classical information criteria

- start with likelihood $L(A \mid y, m)$
  - measures fit of architecture (A) to phenotype (y)
    - given marker data (m)
  - architecture (A) depends on parameters
    - have to estimate loci (μ) and effects (λ)
- complexity related to number of parameters
  - $p = |A|$ = size of genetic architecture
  - with *n.qtl* = 4 QTL and all 2-QTL epistasis terms
    - BC: $p = 1 + n.qtl + n.qtl(n.qtl - 1) = 1 + 4 + 12 = 17$
    - F2: $p = 1 + 2n.qtl + 4\, n.qtl(n.qtl - 1) = 1 + 8 + 48 = 57$

classical information criteria

- construct information criteria
  - balance fit to complexity
  - Akaike AIC $= -2 \log(L) + 2\,p$
  - Bayes/Schwartz BIC $= -2 \log(L) + p \log(n)$
  - Broman $\mathrm{BIC}_\delta = -2 \log(L) + \delta p \log(n)$
  - general form: IC $= -2 \log(L) + p\, D(n)$
- compare models
  - hypothesis testing: designed for one comparison
    - $2 \log[LR(p_1,p_2)] = L(y|m,A_2) - L(y|m,A_1)$
  - model selection: penalize complexity
    - $\mathrm{IC}(p_1,p_2) = 2 \log[LR(p_1,p_2)] + (p_2 - p_1)\, D(n)$

Bayes factors

- ratio of model likelihoods
  - ratio of posterior to prior odds for architectures
  - averaged over unknowns

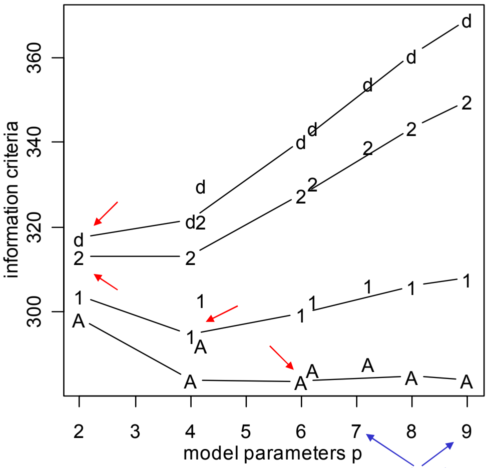
$$B_{12} = \frac{\mathrm{pr}(A_1 \mid y,m)/\mathrm{pr}(A_2 \mid y,m)}{\mathrm{pr}(A_1)/\mathrm{pr}(A_2)} = \frac{\mathrm{pr}(y \mid m,A_1)}{\mathrm{pr}(y \mid m,A_2)}$$

- roughly equivalent to BIC
  - BIC maximizes over unknowns
  - BF averages over unknowns

$$-2\log(B_{12}) = -2\log(LR) - (p_2 - p_1)\log(n)$$

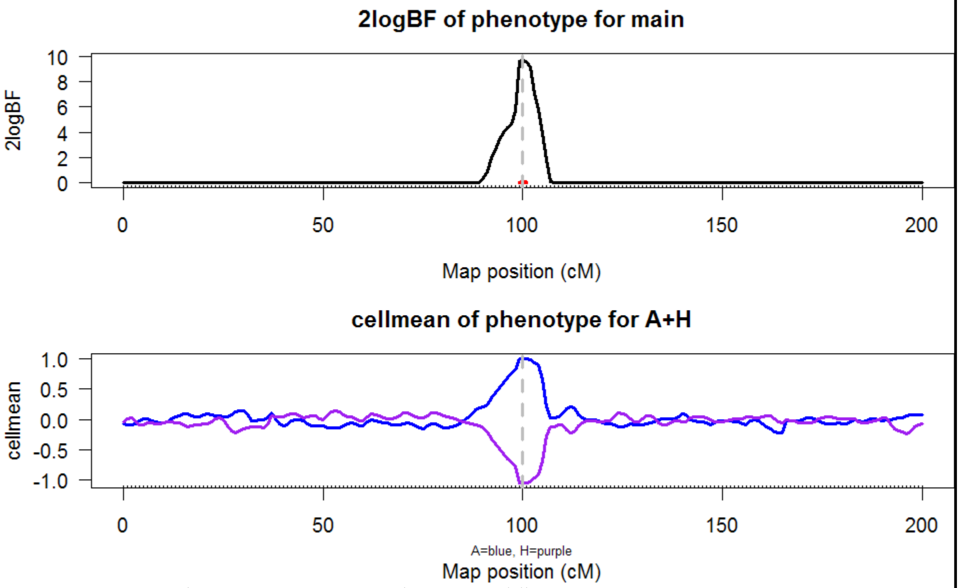
information criteria vs. model size

- WinQTL 2.0
- SCD data on F2
- A=AIC
- 1=BIC(1)
- 2=BIC(2)
- d=BIC(δ)
- models
  - 1,2,3,4 QTL
    - 2+5+9+2
  - epistasis
    - 2:2 AD



epistasis

scan of marginal Bayes factor & effect



5. QTL software options

- methods
  - approximate QTL by markers
  - exact multiple QTL interval mapping
- software platforms
  - MapMaker/QTL (obsolete)
  - QTLCart (statgen.ncsu.edu/qtlcart**)**
  - R/qtl (www.rqtl.org)
  - R/qtlbim (www.qtlbim.org)
  - Yandell, Bradbury (2007) book chapter

approximate QTL methods

- marker regression
  - locus & effect confounded
  - lose power with missing data
- Haley-Knott (least squares) regression
  - correct mean, wrong variance
  - biased by pattern of missing data (Kao 2000)
- extended HK regression
  - correct mean and variance
  - minimizes bias issue (R/qtl “ehk” method)
- composite interval mapping (QTLCart)
  - use markers to approximate other QTL
  - properties depend on marker spacing, missing data

exact QTL methods

- interval mapping (Lander, Botstein 1989)
  - scan whole genome for single QTL
  - bias for linked QTL, low power
- multiple interval mapping (Kao, Zeng, Teasdale 1999)
  - sequential scan of all QTL
  - stepwise model selection
- multiple imputation (Sen, Churchill 2001)
  - fill in (impute) missing genotypes along genome
  - average over multiple imputations
- Bayesian interval mapping (Yi et al. 2005)
  - sample most likely models
  - marginal scans conditional on other QTL

QTL software platforms

- QTLCart (statgen.ncsu.edu/qtlcart)
  - includes features of original MapMaker/QTL
    - not designed for building a linkage map
  - easy to use Windows version WinQTLCart
  - based on Lander-Botstein maximum likelihood LOD
    - extended to marker cofactors (CIM) and multiple QTL (MIM)
    - epistasis, some covariates (GxE)
    - stepwise model selection using information criteria
  - some multiple trait options
  - OK graphics
- R/qtl (www.rqtl.org)
  - includes functionality of classical interval mapping
  - many useful tools to check genotype data, build linkage maps
  - excellent graphics
  - several methods for 1-QTL and 2-QTL mapping
    - epistasis, covariates (GxE)
  - tools available for multiple QTL model selection

Bayesian QTL software options

- Bayesian Haley-Knott approximation: no epistasis
  - Berry C (1998)
    - R/bqtl (www.r-project.org contributed package)
- multiple imputation: epistasis, mostly 1-2 QTL but some multi-QTL
  - Sen and Churchill (2000)
    - matlab/pseudomarker (www.jax.org/staff/churchill/labsite/software)
  - Broman et al. (2003)
    - R/qtl (www.rqtl.org)
- Bayesian interval mapping via MCMC: no epistasis
  - Satagopan et al. (1996); Satagopan, Yandell (1996) Gaffney (2001)
    - R/bim (www.r-project.org contributed package)
    - WinQTLCart/bmapqtl (statgen.ncsu.edu/qtlcart)
  - Stephens & Fisch (1998): no code release
  - Sillanpää Arjas (1998)
    - multimapper (www.rni.helsinki.fi/~mjs)
- Bayesian interval mapping via MCMC: epistasis
  - Yandell et al. (2007)
    - R/qtlbim (www.qtlbim.org)
- Bayesian shrinkage: no epistasis
  - Wang et al. Xu (2005): no code release

R/qtlbim: www.qtlbim.org

- Properties
  - cross-compatible with R/qtl
  - new MCMC algorithms
    - Gibbs with loci indicators; no reversible jump
  - epistasis, fixed & random covariates, GxE
  - extensive graphics
- Software history
  - initially designed (Satagopan Yandell 1996)
  - major revision and extension (Gaffney 2001)
  - R/bim to CRAN (Wu, Gaffney, Jin, Yandell 2003)
  - R/qtlbim to CRAN (Yi, Yandell et al. 2006)
- Publications
  - Yi et al. (2005); Yandell et al. (2007); …

many thanks

U AL Birmingham
- Nengjun Yi
- Tapan Mehta
- Samprit Banerjee
- Daniel Shriner
- Ram Venkataraman
- David Allison

Jackson Labs
- Gary Churchill
- Hao Wu
- Hyuna Yang
- Randy von Smith

Alan Attie
- Jonathan Stoehr
- Hong Lan
- Susie Clee
- Jessica Byers
- Mark Gray-Keller

Tom Osborn
- David Butruille
- Marcio Ferrera
- Josh Udahl
- Pablo Quijada

UW-Madison Stats
- Yandell lab
  - Jaya Satagopan
  - Fei Zou
  - Patrick Gaffney
  - Chunfang Jin
  - Elias Chaibub
  - W Whipple Neely
  - Jee Young Moon
- Michael Newton
- Christina Kendziorski
- Daniel Gianola
  - Liang Li
  - Daniel Sorensen

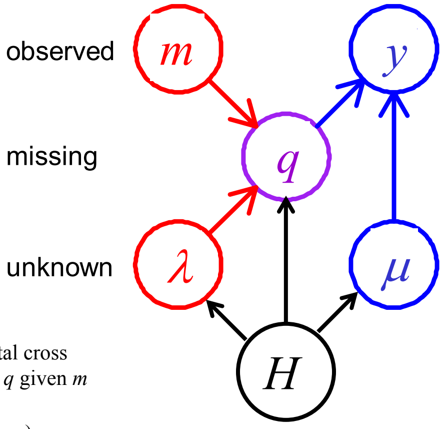
USDA Hatch, NIH/NIDDK (Attie), NIH/R01s (Yi, Broman)

Bayesian Interval Mapping

1. Bayesian strategy 3-17
2. Markov chain sampling 18-25
3. sampling genetic architectures 26-33
4. Bayesian QTL model selection 34-44

QTL model selection: key players

- observed measurements
  - y = phenotypic trait
  - m = markers & linkage map
  - i = individual index $(1,\ldots,n)$
- missing data
  - missing marker data
  - q = QT genotypes
    - alleles QQ, Qq, or qq at locus
- unknown quantities
  - λ = QT locus (or loci)
  - μ = phenotype model parameters
  - H = QTL model/genetic architecture
- pr($q|m,\lambda,H$) genotype model
  - grounded by linkage map, experimental cross
  - recombination yields multinomial for q given m
- pr($y|q,\mu,H$) phenotype model
  - distribution shape (assumed normal here)
  - unknown parameters μ (could be non-parametric)



after
Sen Churchill (2001)

1. Bayesian strategy for QTL study

- augment data (*y,m*) with missing genotypes *q*
- study unknowns (μ,λ,A) given augmented data (*y,m,q*)
  - find better genetic architectures *A*
  - find most likely genomic regions = QTL = λ
  - estimate phenotype parameters = genotype means = μ
- sample from posterior in some clever way
  - multiple imputation (Sen Churchill 2002)
  - Markov chain Monte Carlo (MCMC)
    - (Satagopan et al. 1996; Yi et al. 2005)

$$\text{posterior} = \frac{\text{likelihood} * \text{prior}}{\text{constant}}$$

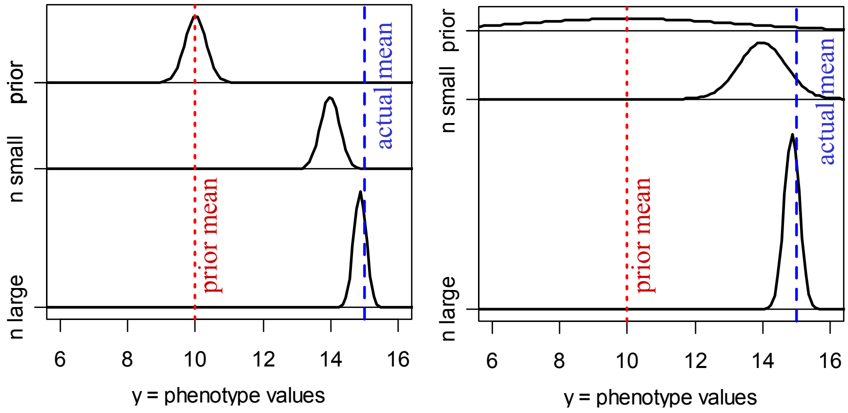
$$\text{posterior for } q,\mu,\lambda,A = \frac{\text{phenotype likelihood} * [\text{prior for } q,\mu,\lambda,A]}{\text{constant}}$$

$$\text{pr}(q,\mu,\lambda,A \mid y,m) = \frac{\text{pr}(y \mid q,\mu,A) * [\text{pr}(q \mid m,\lambda,A)\text{pr}(\mu \mid A)\text{pr}(\lambda \mid m,A)\text{pr}(A)]}{\text{pr}(y \mid m)}$$

Bayesian idea

- Reverend Thomas Bayes (1702-1761)
  - part-time mathematician
  - buried in Bunhill Cemetary, Moongate, London
  - famous paper in 1763 *Phil Trans Roy Soc London*
  - was Bayes the first with this idea? (Laplace?)
- basic idea (from Bayes' original example)
  - two billiard balls tossed at random (uniform) on table
  - where is first ball if the second is to its left?
    - prior: anywhere on the table
    - posterior: more likely toward right end of table

Bayes posterior for normal data



small prior variance

large prior variance

Bayes posterior for normal data

model $y_i = \mu + e_i$

environment e ~ $N(0, \sigma^2)$, σ^2 known

likelihood $y \sim N(\mu, \sigma^2)$

prior $\mu \sim N(\mu_0, \kappa\sigma^2)$, κ known

posterior: mean tends to sample mean

single individual $\mu \sim N(\mu_0 + b_1(y_1 - \mu_0), b_1\sigma^2)$

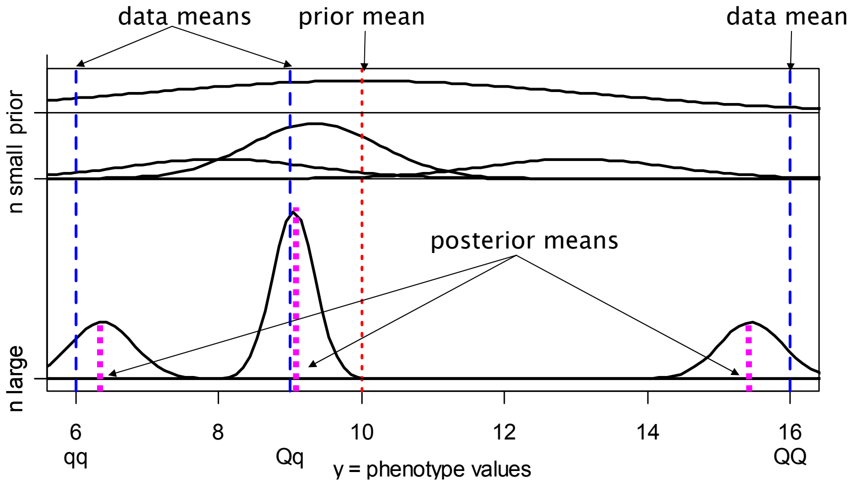
sample of n individuals $\mu \sim N\left(b_n \bar{y}_\bullet + (1-b_n)\mu_0, b_n\sigma^2 / n\right)$

with $\bar{y}_\bullet = \underset{\{i=1,\ldots,n\}}{\text{sum}} y_i / n$

fudge factor
(shrinks to 1) $b_n = \dfrac{\kappa n}{\kappa n + 1} \to 1$

what values are the genotypic means?

(phenotype mean for genotype q is μ_q)



Bayes posterior QTL means

posterior centered on sample genotypic mean
but shrunken slightly toward overall mean

prior: $\mu_q \sim N\left(\bar{y}_{\bullet}, \kappa\sigma^2\right)$

posterior: $\mu_q \sim N\left(b_q \bar{y}_q + (1-b_q)\bar{y}_{\bullet}, b_q \sigma^2 / n_q\right)$

$$n_q = \text{count}\{q_i = q\}, \bar{y}_q = \underset{\{q_i=q\}}{\text{sum}}\, y_i / n_q$$

fudge factor: $b_q = \dfrac{\kappa n_q}{\kappa n_q + 1} \to 1$

QTL with epistasis

- same phenotype model overview

$$Y = \mu_q + e, \text{var}(e) = \sigma^2$$

- partition of genotypic value with epistasis

$$\mu_q = \mu + \beta_{q1} + \beta_{q2} + \beta_{q12}$$

- partition of genetic variance & heritability

$$\text{var}(\mu_q) = \sigma_q^2 = \sigma_1^2 + \sigma_2^2 + \sigma_{12}^2$$

$$h_q^2 = \frac{\sigma_q^2}{\sigma_q^2 + \sigma^2} = h_1^2 + h_2^2 + h_{12}^2$$

partition of multiple QTL effects

- partition genotype-specific mean into QTL effects

  μ_q = mean + main effects + epistatic interactions

  $\mu_q = \mu + \beta_q = \mu + \text{sum}_{j \text{ in } A}\ \beta_{qj}$

- priors on mean and effects

μ	~ $N(\mu_0, \kappa_0\sigma^2)$	grand mean
β_q	~ $N(0, \kappa_1\sigma^2)$	model-independent genotypic effect
β_{qj}	~ $N(0, \kappa_1\sigma^2/\lvert A\rvert)$	effects down-weighted by size of *A*

- determine hyper-parameters via empirical Bayes

$$\mu_0 \approx \overline{Y}_{\bullet} \text{ and } \kappa_1 \approx \frac{h_q^2}{1 - h_q^2} = \frac{\sigma_q^2}{\sigma^2}$$

posterior mean ≈ LS estimate

$$\mu_q \mid Y, m \sim N(B_q \hat{\mu}_q, B_q C_q \sigma^2)$$

$$\approx N(\hat{\mu}_q, C_q \sigma^2)$$

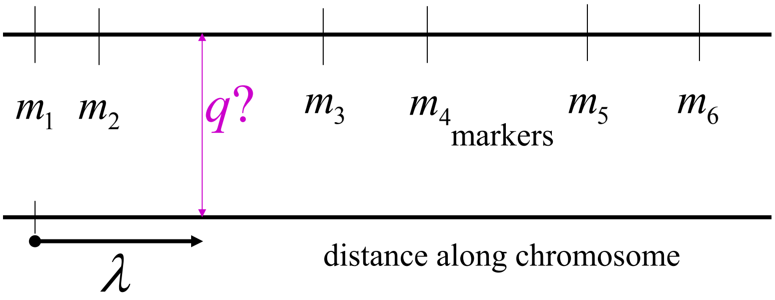
LS estimate $\hat{\mu}_q = \text{sum}_i[\text{sum}_{j \in M} \hat{\beta}_{qji}] = \text{sum}_i w_{qi} Y$

variance $V(\hat{\mu}_q) = \text{sum}_i w_{qi}^2 \sigma^2 = C_q \sigma^2$

shrinkage $B_q = \kappa / (\kappa + C_q) \to 1$

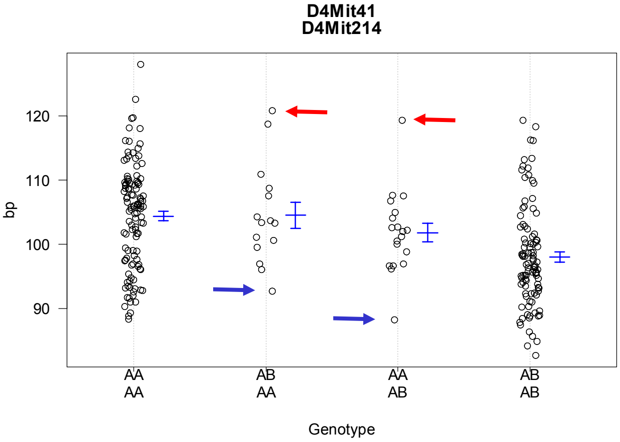
pr(q|m,λ) recombination model

pr(q|m,λ) = pr(geno | map, locus) ≈
pr(geno | flanking markers, locus)



what are likely QTL genotypes *q*?

how does phenotype *y* improve guess?



what are probabilities for genotype *q* between markers?

recombinants AA:AB

all 1:1 if ignore *y*
and if we use *y*?

posterior on QTL genotypes *q*

- full conditional of *q* given data, parameters
  - proportional to prior pr(*q* | *m, λ*)
    - weight toward *q* that agrees with flanking markers
  - proportional to likelihood pr(*y*|*q*,*μ*)
    - weight toward *q* with similar phenotype values
  - posterior recombination model balances these two
- this *is* the E-step of EM computations

$$\mathrm{pr}(q \mid y, m, \mu, \lambda) = \frac{\mathrm{pr}(y \mid q, \mu) * \mathrm{pr}(q \mid m, \lambda)}{\mathrm{pr}(y \mid m, \mu, \lambda)}$$

Where are the loci λ on the genome?

- prior over genome for QTL positions
  - flat prior = no prior idea of loci
  - or use prior studies to give more weight to some regions
- posterior depends on QTL genotypes q

  $\text{pr}(\lambda \mid m,q) = \text{pr}(\lambda)\ \text{pr}(q \mid m,\lambda)$ / constant
  - constant determined by averaging
    - over all possible genotypes q
    - over all possible loci λ on entire map
- no easy way to write down posterior

what is the genetic architecture A?

- which positions correspond to QTLs?
  - priors on loci (previous slide)
- which QTL have main effects?
  - priors for presence/absence of main effects
    - same prior for all QTL
    - can put prior on each d.f. (1 for BC, 2 for F2)
- which pairs of QTL have epistatic interactions?
  - prior for presence/absence of epistatic pairs
    - depends on whether 0,1,2 QTL have main effects
    - epistatic effects less probable than main effects

Bayesian priors & posteriors

- augmenting with missing genotypes q
  - prior is recombination model
  - posterior is (formally) E step of EM algorithm
- sampling phenotype model parameters μ
  - prior is “flat” normal at grand mean (no information)
  - posterior shrinks genotypic means toward grand mean
  - (details for unexplained variance omitted here)
- sampling QTL loci λ
  - prior is flat across genome (all loci equally likely)
- sampling QTL model A
  - number of QTL
    - prior is Poisson with mean from previous IM study
  - genetic architecture of main effects and epistatic interactions
    - priors on epistasis depend on presence/absence of main effects

2. Markov chain sampling

- construct Markov chain around posterior
  - want posterior as stable distribution of Markov chain
  - in practice, the chain tends toward stable distribution
    - initial values may have low posterior probability
    - burn-in period to get chain mixing well
- sample QTL model components from full conditionals
  - sample locus λ given q,A (using Metropolis-Hastings step)
  - sample genotypes q given λ,μ,y,A (using Gibbs sampler)
  - sample effects μ given q,y,A (using Gibbs sampler)
  - sample QTL model A given λ,μ,y,q (using Gibbs or M-H)

$$(\lambda, q, \mu, A) \sim \mathrm{pr}(\lambda, q, \mu, A \mid y, m)$$

$$(\lambda, q, \mu, A)_1 \rightarrow (\lambda, q, \mu, A)_2 \rightarrow \cdots \rightarrow (\lambda, q, \mu, A)_N$$

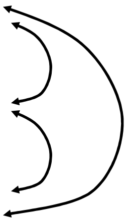
MCMC sampling of (λ,q,μ)

- Gibbs sampler
  - genotypes q
  - effects μ
  - *not* loci λ

$$q \sim \text{pr}(q \mid y_i, m_i, \mu, \lambda)$$

$$\mu \sim \frac{\text{pr}(y \mid q, \mu)\text{pr}(\mu)}{\text{pr}(y \mid q)}$$

$$\lambda \sim \frac{\text{pr}(q \mid m, \lambda)\text{pr}(\lambda \mid m)}{\text{pr}(q \mid m)}$$



- Metropolis-Hastings sampler
  - extension of Gibbs sampler
  - does not require normalization
    - pr(q | m) = sum$_\lambda$ pr(q | m, λ) pr(λ)

Gibbs sampler
for two genotypic means

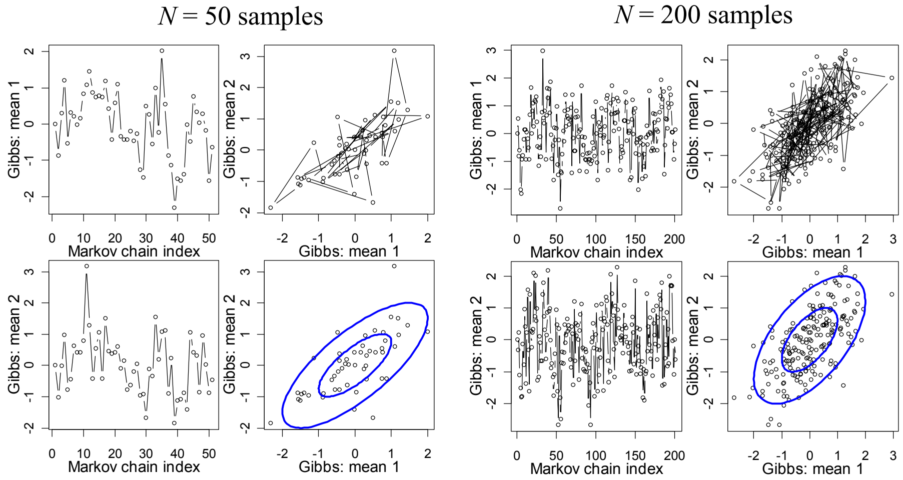
- want to study two correlated effects
  - could sample directly from their bivariate distribution
  - assume correlation ρ is known
- instead use Gibbs sampler:
  - sample each effect from its full conditional given the other
  - pick order of sampling at random
  - repeat many times

$$\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$$

$$\mu_1 \sim N\left(\rho\mu_2, 1 - \rho^2\right)$$

$$\mu_2 \sim N\left(\rho\mu_1, 1 - \rho^2\right)$$

Gibbs sampler samples: $\rho = 0.6$



$N = 50$ samples

$N = 200$ samples

full conditional for locus

- cannot easily sample from locus full conditional

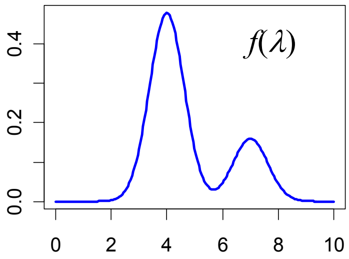
  $\text{pr}(\lambda\,|y,m,\mu,q) \quad = \text{pr}(\ \lambda \mid m,q)$

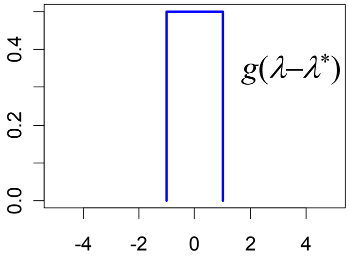
  $\qquad\qquad\qquad\qquad = \text{pr}(\ q \mid m,\ \lambda\ )\ \text{pr}(\lambda\ )\ /\ \text{constant}$
- constant is very difficult to compute explicitly
  - must average over all possible loci λ over genome
  - must do this for every possible genotype q
- Gibbs sampler will not work in general
  - but can use method based on ratios of probabilities
  - Metropolis-Hastings is extension of Gibbs sampler

Metropolis-Hastings idea

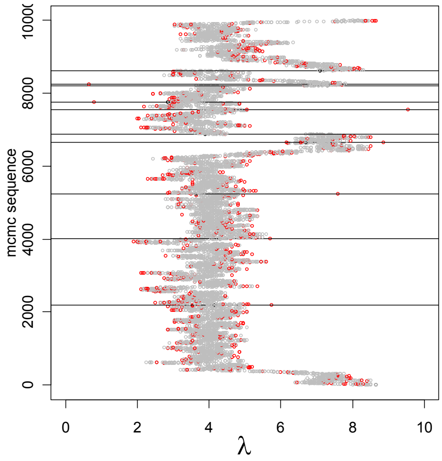
- want to study distribution $f(\lambda)$
  - take Monte Carlo samples
    - unless too complicated
  - take samples using ratios of f
- Metropolis-Hastings samples:
  - propose new value λ^*
    - near (?) current value λ
    - from some distribution g
  - accept new value with prob a
    - Gibbs sampler: $a = 1$ always

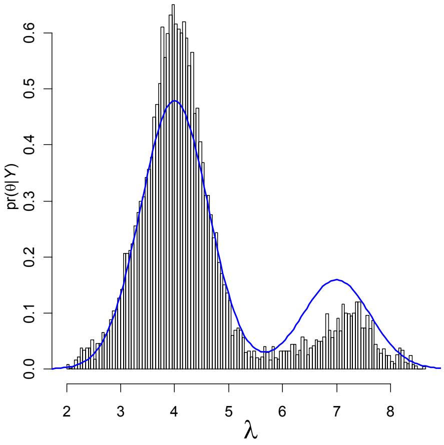
$$a = \min\left(1, \frac{f(\lambda^*)g(\lambda^* - \lambda)}{f(\lambda)g(\lambda - \lambda^*)}\right)$$





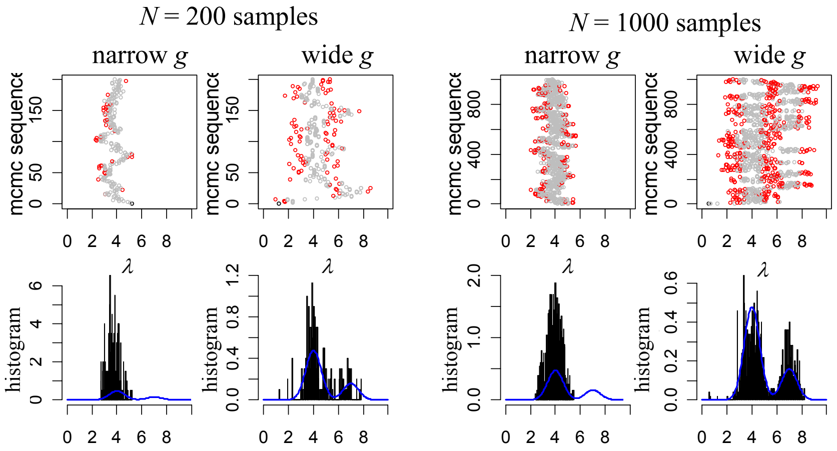
Metropolis-Hastings for locus λ





added twist: occasionally propose from entire genome

Metropolis-Hastings samples



3. sampling genetic architectures

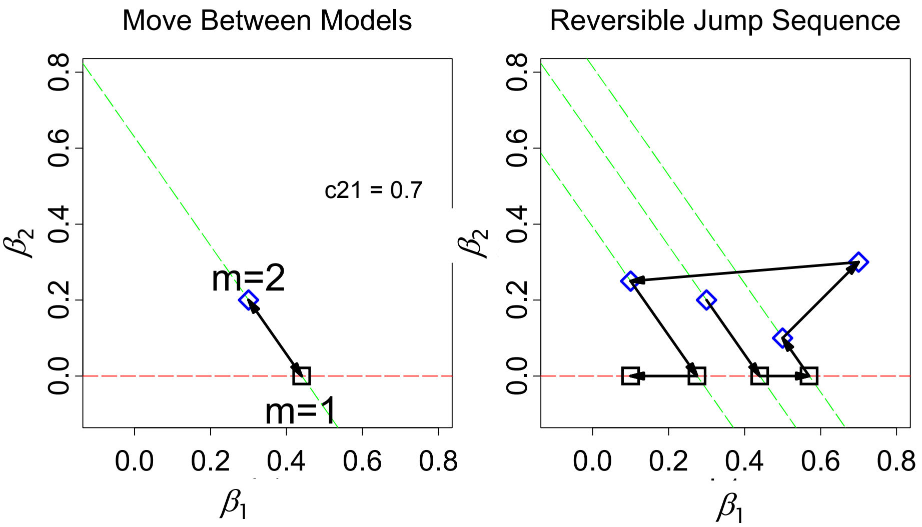
- search across genetic architectures A of various sizes
  - allow change in number of QTL
  - allow change in types of epistatic interactions
- methods for search
  - reversible jump MCMC
  - Gibbs sampler with loci indicators
- complexity of epistasis
  - Fisher-Cockerham effects model
  - general multi-QTL interaction & limits of inference

reversible jump MCMC

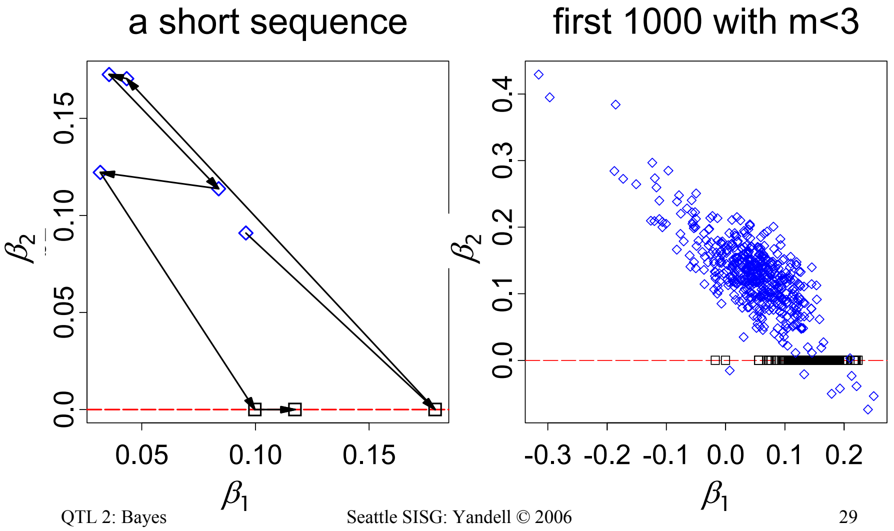
- consider known genotypes q at 2 known loci λ
  - models with 1 or 2 QTL
- M-H step between 1-QTL and 2-QTL models
  - model changes dimension (via careful bookkeeping)
  - consider mixture over QTL models H

$$
\begin{aligned}
n.qtl &= 1 : Y = \beta_0 + \beta_{q1} + e \\
n.qtl &= 2 : Y = \beta_0 + \beta_{q1} + \beta_{q2} + e
\end{aligned}
$$

geometry of reversible jump



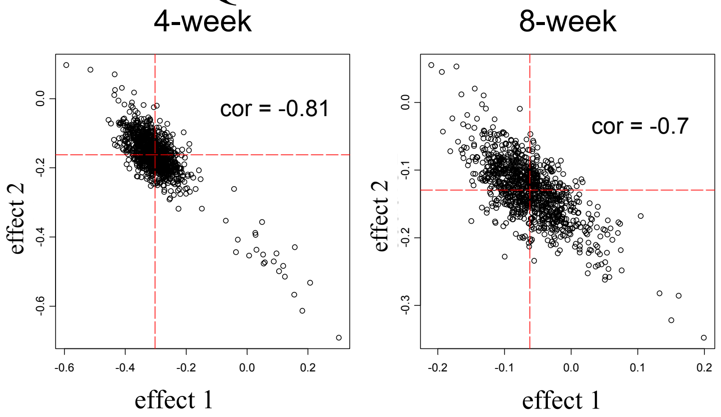
geometry allowing q and λ to change



a short sequence

first 1000 with m<3

collinear QTL = correlated effects

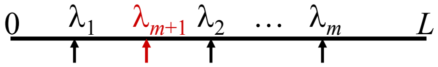


4-week

8-week

- linked QTL = collinear genotypes
  - correlated estimates of effects (negative if in coupling phase)
  - sum of linked effects usually fairly constant

sampling across QTL models *A*



action steps: draw one of three choices

- update QTL model *A* with probability 1-*b*(*A*)-*d*(*A*)
  - update current model using full conditionals
  - sample QTL loci, effects, and genotypes
- add a locus with probability *b*(*A*)
  - propose a new locus along genome
  - innovate new genotypes at locus and phenotype effect
  - decide whether to accept the "birth" of new locus
- drop a locus with probability *d*(*A*)
  - propose dropping one of existing loci
  - decide whether to accept the "death" of locus

Gibbs sampler with loci indicators

- consider only QTL at pseudomarkers
  - every 1-2 cM
  - modest approximation with little bias
- use loci indicators in each pseudomarker
  - $\delta = 1$ if QTL present
  - $\delta = 0$ if no QTL present
- Gibbs sampler on loci indicators δ
  - relatively easy to incorporate epistasis
  - Yi, Yandell, Churchill, Allison, Eisen, Pomp (2005 *Genetics*)
    - (see earlier work of Nengjun Yi and Ina Hoeschele)

$$\mu_q = \mu + \delta_1 \beta_{q1} + \delta_2 \beta_{q2}$$

Bayesian shrinkage estimation

- soft loci indicators
  - strength of evidence for λ_j depends on variance of β_j
  - similar to $\gamma > 0$ on grey scale
- include all possible loci in model
  - pseudo-markers at 1cM intervals
- Wang et al. (2005 *Genetics*)
  - Shizhong Xu group at U CA Riverside

$$Y = \beta_0 + \beta_1(q_1) + \beta_2(q_1) + \ldots + e$$

$$\beta_j(q_j) \sim N(0, \sigma_j^2), \sigma_j^2 \sim \text{inverse - chisquare}$$

4. Bayesian QTL model selection

- Bayes factor details
- Bayesian model averaging
- false discovery rate (FDR)

Bayes factors

- ratio of model likelihoods
  - ratio of posterior to prior odds for architectures
  - averaged over unknowns

$$B_{12} = \frac{\mathrm{pr}(A_1 \mid y,m)/\mathrm{pr}(A_2 \mid y,m)}{\mathrm{pr}(A_1)/\mathrm{pr}(A_2)} = \frac{\mathrm{pr}(y \mid m, A_1)}{\mathrm{pr}(y \mid m, A_2)}$$

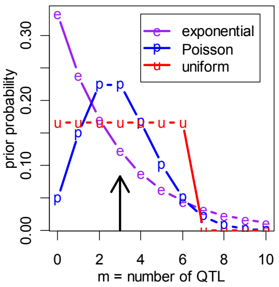
- roughly equivalent to BIC
  - BIC maximizes over unknowns
  - BF averages over unknowns

$$-2\log(B_{12}) = -2\log(LR) - (p_2 - p_1)\log(n)$$

issues in computing Bayes factors

- *BF* insensitive to shape of prior on *A*
  - geometric, Poisson, uniform
  - precision improves when prior mimics posterior
- *BF* sensitivity to prior variance on effects θ
  - prior variance should reflect data variability
  - resolved by using hyper-priors
    - automatic algorithm; no need for user tuning
- easy to compute Bayes factors from samples
  - sample posterior using MCMC
  - posterior pr(*A* | *y, m*) is marginal histogram

Bayes factors and genetic model *A*

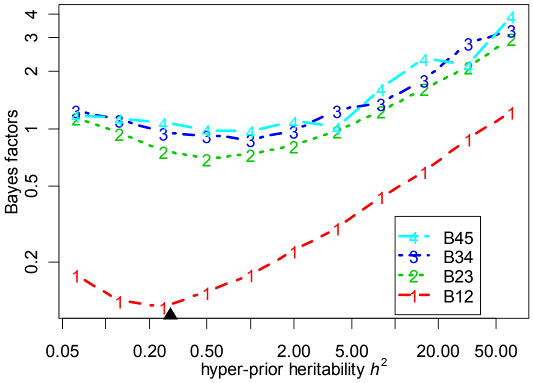


- |*A*| = number of QTL
  - prior pr(*A*) chosen by user
  - posterior pr(*A*|*y,m*)
    - sampled marginal histogram
    - shape affected by prior pr(*A*)

$$BF_{A,A+1} = \frac{\mathrm{pr}(A|y,m)/\mathrm{pr}(A)}{\mathrm{pr}(A+1|y,m)/\mathrm{pr}(A+1)}$$

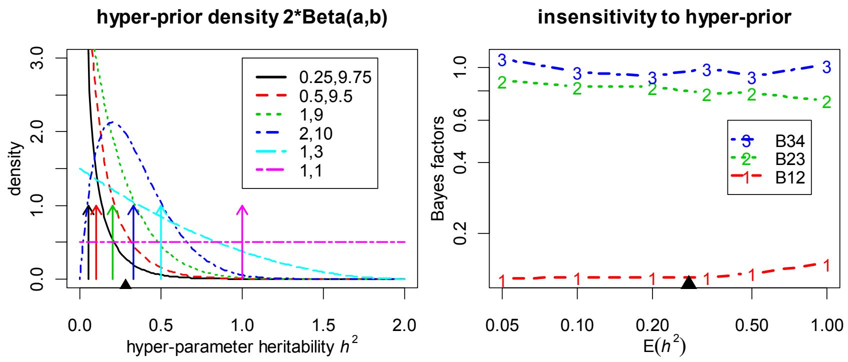
- pattern of QTL across genome
- gene action and epistasis

BF sensitivity to fixed prior for effects



$$\beta_{qj} \sim \mathrm{N}\left(0, \sigma_G^2 / m\right), \sigma_G^2 = h^2 \sigma_{\text{total}}^2, h^2 \text{ fixed}$$

BF insensitivity to random effects prior

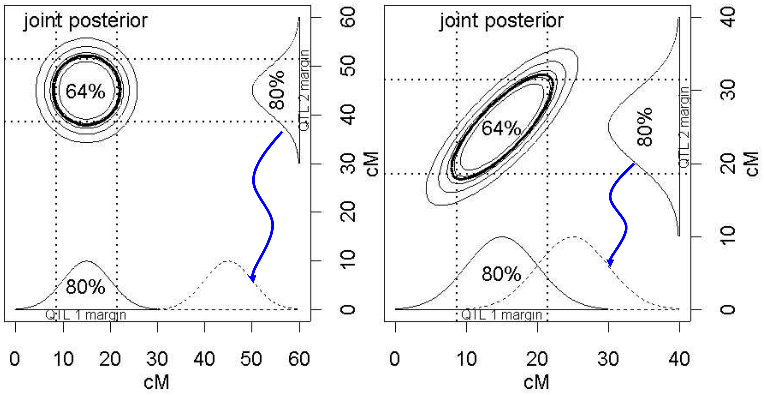


$$\beta_{qj} \sim \mathrm{N}\left(0, \sigma_G^2 / m\right), \sigma_G^2 = h^2 \sigma_{\text{total}}^2, \tfrac{1}{2} h^2 \sim \mathrm{Beta}(a, b)$$

Bayesian model averaging

- average summaries over multiple architectures
- avoid selection of “best” model
- focus on “better” models
- examples in data talk later

1-D and 2-D marginals pr(QTL at λ | *Y,X, m*)



unlinked loci

linked loci

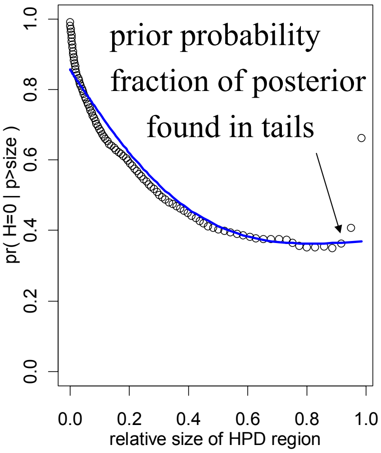
false detection rates and thresholds

- multiple comparisons: test QTL across genome
  - size = pr(LOD(λ) > threshold | no QTL at λ)
  - threshold guards against a single false detection
    - very conservative on genome-wide basis
  - difficult to extend to multiple QTL
- positive false discovery rate (Storey 2001)
  - pFDR = pr(no QTL at λ | LOD(λ) > threshold)
  - Bayesian posterior HPD region based on threshold
    - $\Lambda = \{\lambda \mid \text{LOD}(\lambda) > \text{threshold}\} \approx \{\lambda \mid \text{pr}(\lambda \mid Y,X,m) \text{ large}\}$
  - extends naturally to multiple QTL

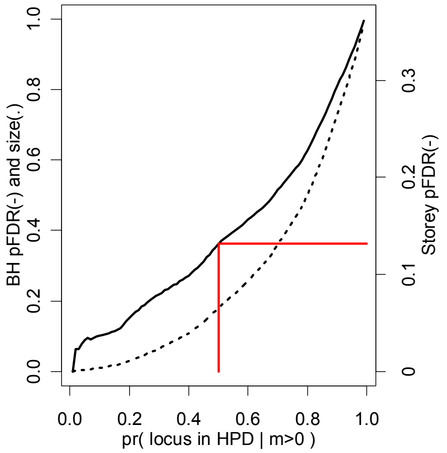
pFDR and QTL posterior

- positive false detection rate
  - pFDR = pr(no QTL at *λ* | *Y,X, λ* in *Λ*)
  - $\text{pFDR} = \dfrac{\text{pr}(H\text{=}0)\text{*size}}{\text{pr}(m\text{=}0)\text{*size+pr}(m\text{>}0)\text{*power}}$
  - power = posterior = pr(QTL in *Λ* | *Y,X, m*>0)
  - size = (length of *Λ*) / (length of genome)
- extends to other model comparisons
  - *m* = 1 vs. *m* = 2 or more QTL
  - pattern = ch1,ch2,ch3 vs. pattern > 2*ch1,ch2,ch3

pFDR for SCD1 analysis



prior probability fraction of posterior found in tails

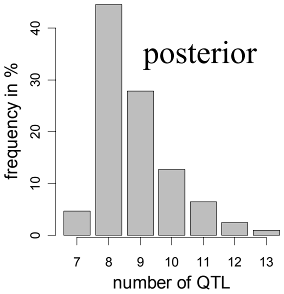


examples in detail

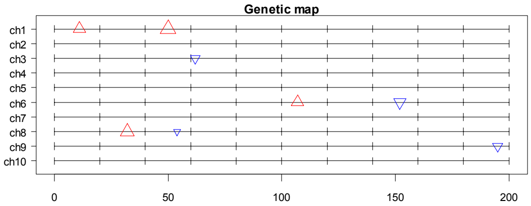
- simulation study (after Stephens & Fisch (1998) 2-3
- days to flower for *Brassica napus* (plant) (*n* = 108) 4-10
  - single chromosome with 2 linked loci
  - whole genome
- gonad shape in *Drosophila* spp. (insect) (*n* = 1000) 11-16
  - multiple traits reduced by PC
  - many QTL and epistasis
- expression phenotype (SCD1) in mice (*n* = 108) 17-26
  - multiple QTL and epistasis
- obesity in mice (*n* = 421) 27-35
  - epistatic QTLs with no main effects
- hypertension in mice set (*n* = 250) 36-48
  - classic R/qtl example (used in R demo)

simulation with 8 QTL

- •simulated F2 intercross, 8 QTL
  - (Stephens, Fisch 1998)
  - *n*=200, heritability = 50%
  - detected 3 QTL
- •increase to detect all 8
  - *n*=500, heritability to 97%



QTL	chr	loci	effect
1	1	11	–3
2	1	50	–5
3	3	62	+2
4	6	107	–3
5	6	152	+3
6	8	32	–4
7	8	54	+1
8	9	195	+2



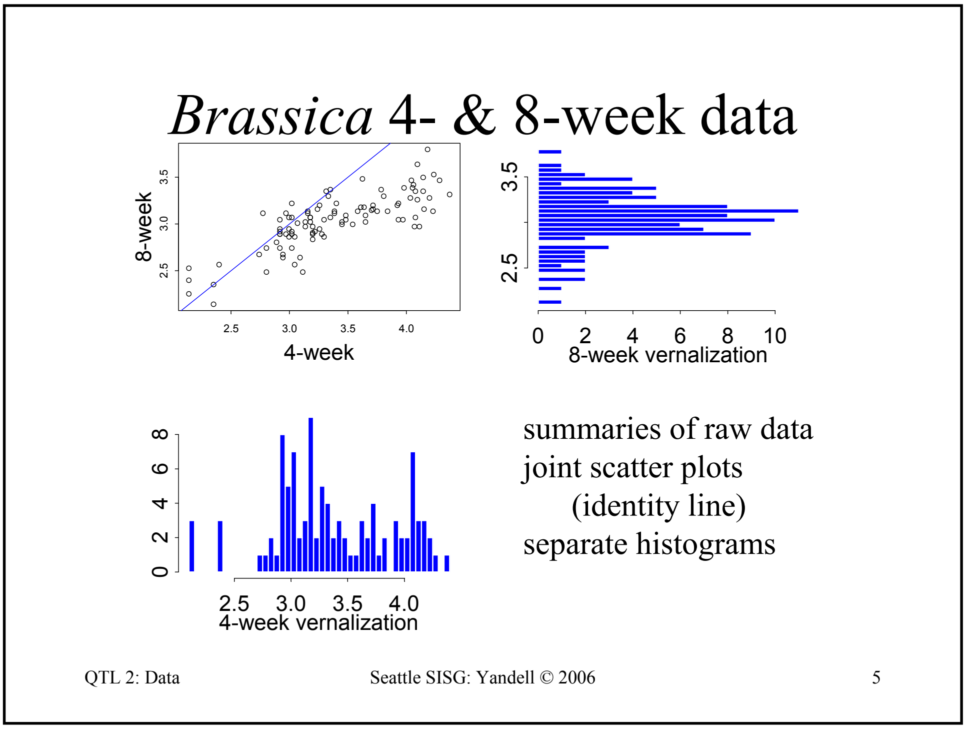
loci pattern across genome

- notice which chromosomes have persistent loci
- best pattern found 42% of the time

	Chromosome										
m	**1**	2	**3**	4	5	**6**	7	**8**	**9**	10	**Count of 8000**
8	**2**	**0**	**1**	**0**	**0**	**2**	**0**	**2**	**1**	**0**	3371
9	*3*	0	1	0	0	2	0	2	1	0	751
7	2	0	1	0	0	2	0	*1*	1	0	377
9	2	0	1	0	0	2	0	2	1	0	218
9	2	0	1	0	0	*3*	0	2	1	0	218
9	2	0	1	0	0	2	0	2	*2*	0	198

Brassica napus: 1 chromosome

- 4-week & 8-week vernalization effect
  - log(days to flower)
- genetic cross of
  - Stellar (annual canola)
  - Major (biennial rapeseed)
- 105 F1-derived double haploid (DH) lines
  - homozygous at every locus (*QQ* or *qq*)
- 10 molecular markers (RFLPs) on LG9
  - two QTLs inferred on LG9 (now chromosome N2)
  - corroborated by Butruille (1998)
  - exploiting synteny with *Arabidopsis thaliana*



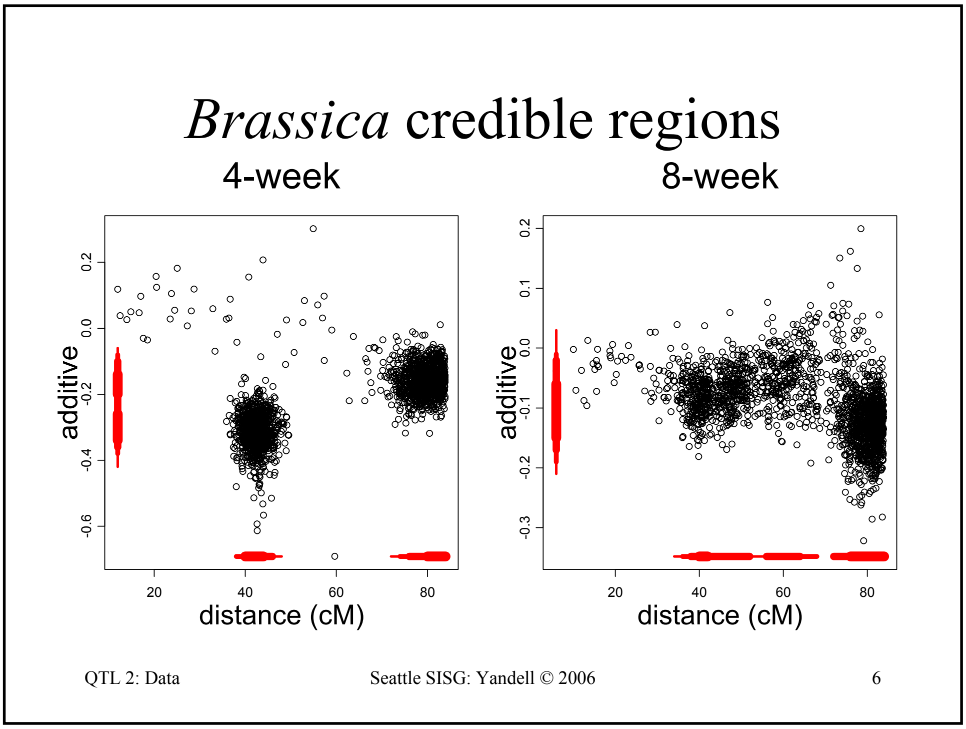
Brassica 4- & 8-week data

summaries of raw data

joint scatter plots

(identity line)

separate histograms



Brassica credible regions

B. napus 8-week vernalization whole genome study

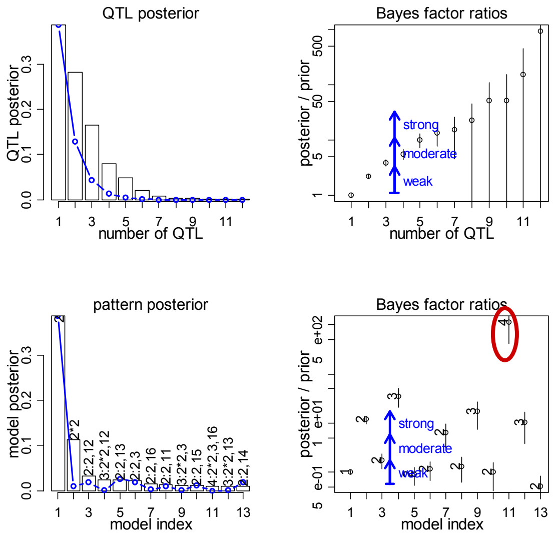
- 108 plants from double haploid
  - similar genetics to backcross: follow 1 gamete
  - parents are Major (biennial) and Stellar (annual)
- 300 markers across genome
  - 19 chromosomes
  - average 6cM between markers
    - median 3.8cM, max 34cM
  - 83% markers genotyped
- phenotype is days to flowering
  - after 8 weeks of vernalization (cooling)
  - Stellar parent requires vernalization to flower
- Ferreira et al. (1994); Kole et al. (2001); Schranz et al. (2002)

Bayesian model assessment

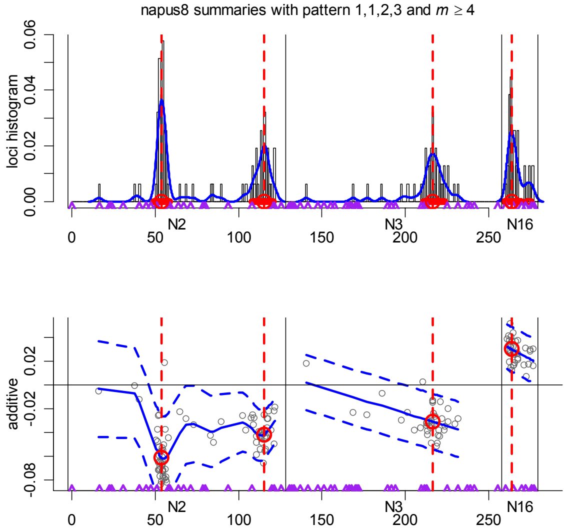
row 1: # QTL
row 2: pattern

col 1: posterior
col 2: Bayes factor
note error bars on bf

evidence suggests
4-5 QTL
N2(2-3),N3,N16



Bayesian estimates of loci & effects



histogram of loci
blue line is density
red lines at estimates

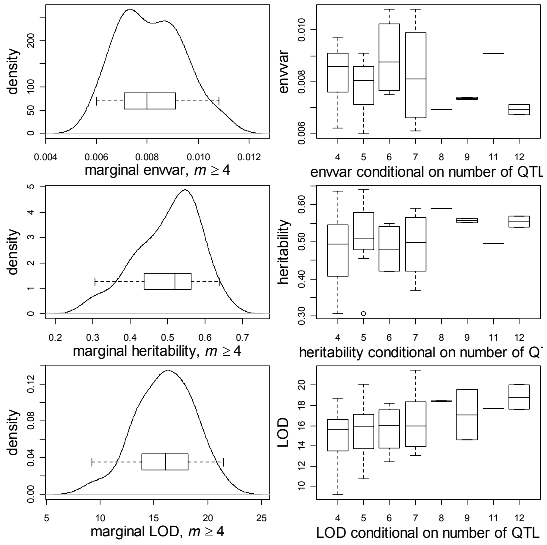
estimate additive effects
  (red circles)
grey points sampled
  from posterior
blue line is cubic spline
dashed line for 2 SD

Bayesian model diagnostics

pattern: N2(2),N3,N16
col 1: density
col 2: boxplots by *m*

environmental variance
  $\sigma^2 = .008$, $\sigma = .09$
heritability
  $h^2 = 52\%$
LOD = 16
(highly significant)

but note change with *m*



shape phenotype in BC study indexed by PC1

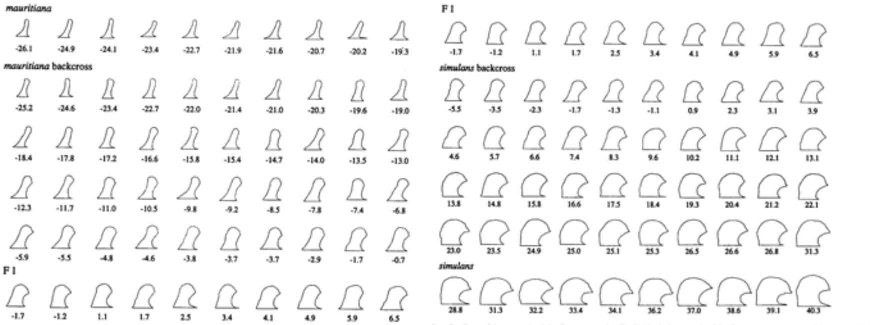
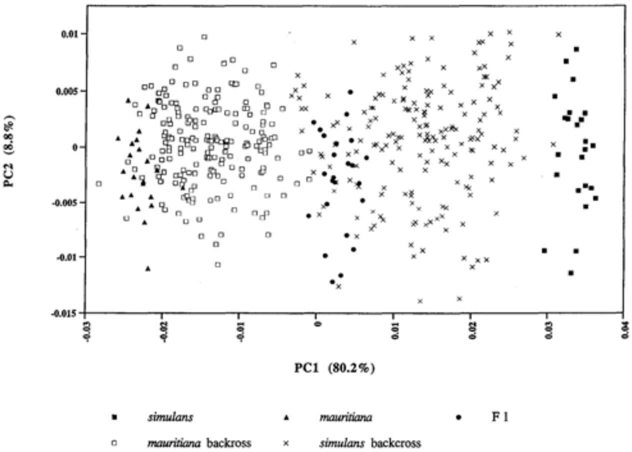


FIGURE 6.—Outlines of the posterior lobe from a sample of individuals from each of the five groups: pure *mauritiana*, *mauritiana* backcross, F_1, *simulans* backcross, and pure *simulans*. Within each group, the outlines are presented in order of their PC1 score (sampled at even intervals from the range of variation). The number below each specimen is its PC1 score. The outlines are drawn to scale with the origin at the centroid of each outline and with all baselines parallel.

Liu et al. (1996) *Genetics*

shape phenotype via PC



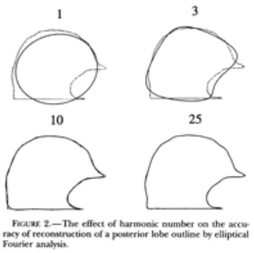
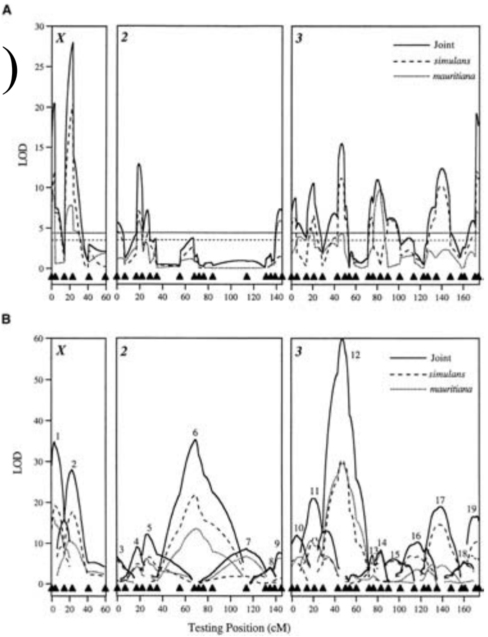


FIGURE 2.—The effect of harmonic number on the accuracy of reconstruction of a posterior lobe outline by elliptical Fourier analysis.

FIGURE 5.—A plot of the first two principal components of the Fourier coefficients from posterior lobe outlines. Many individuals from each of five genotypic classes are represented. Each point represents an average of scores from the left and right sides of an individual (with a few exceptions for which the score is from one side only). The percentage of variation in the Fourier coefficients accounted for by each principal component is given in parentheses. Liu et al. (1996) *Genetics*

Zeng et al. (2000) CIM vs. MIM

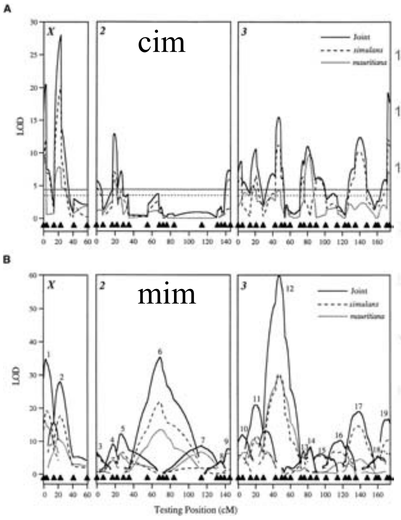


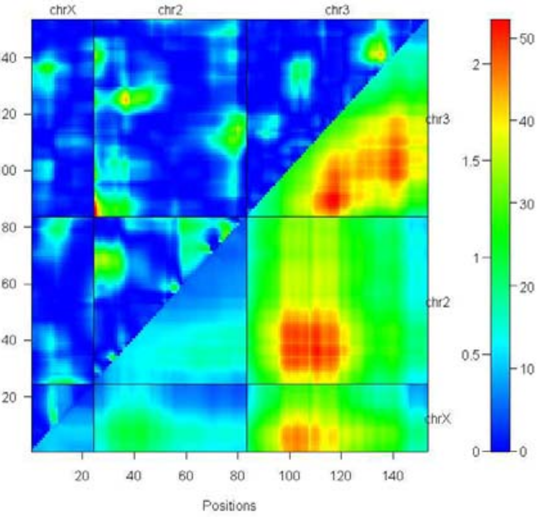
composite interval mapping
- (Liu et al. 1996)
- narrow peaks
- miss some QTL

multiple interval mapping
- (Zeng et al. 2000)
- triangular peaks

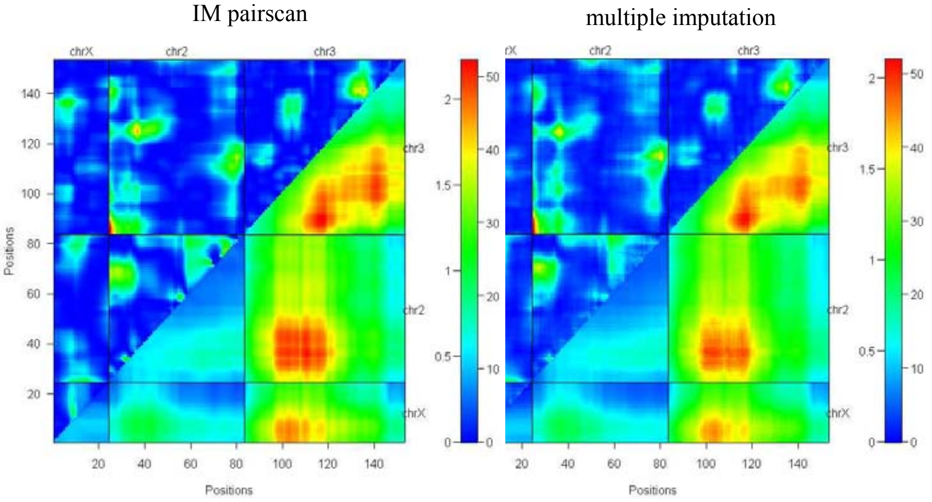
both conditional 1-D scans
- fixing all other "QTL"

CIM, MIM and IM pairscan





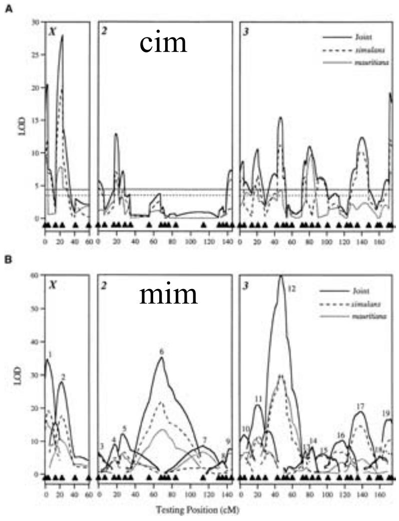
2 QTL + epistasis: IM versus multiple imputation



IM pairscan

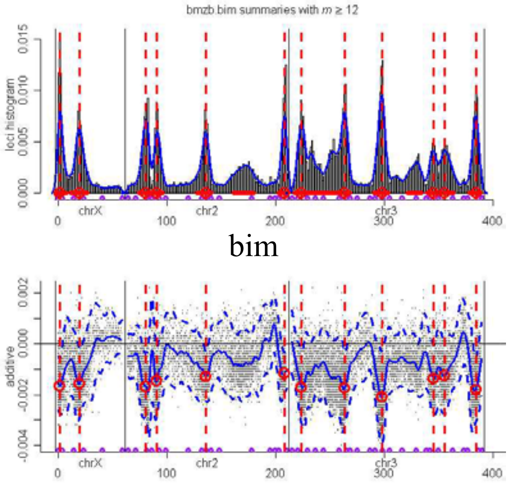
multiple imputation

multiple QTL: CIM, MIM and BIM



cim

mim

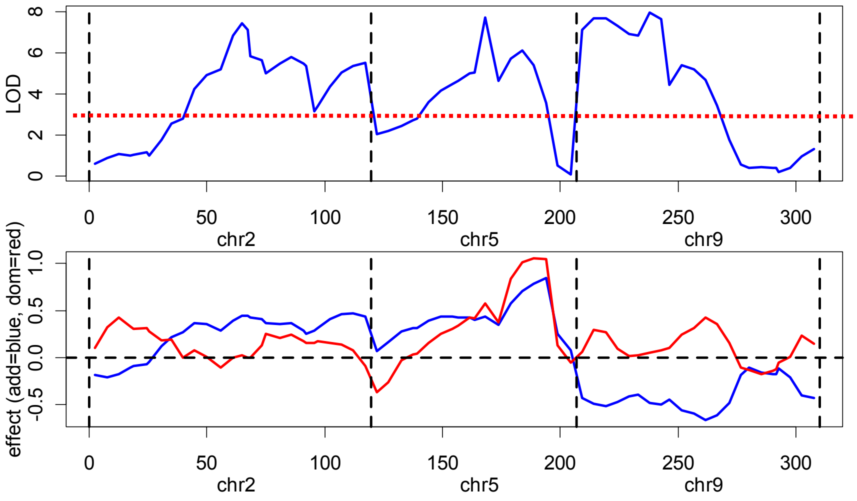


bim

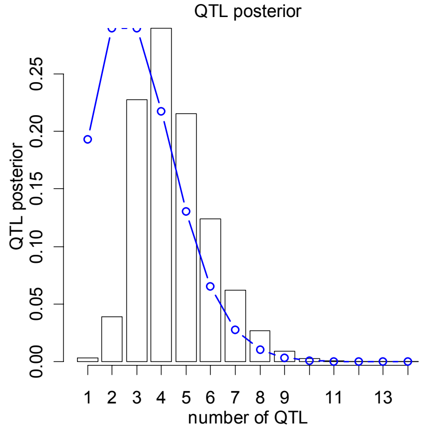
studying diabetes in an F2

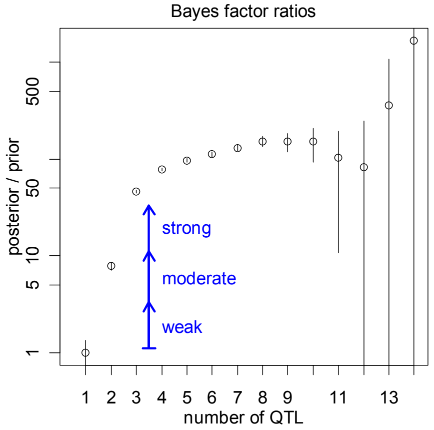
- segregating cross of inbred lines
  - B6.ob x BTBR.ob → F1 → F2
  - selected mice with ob/ob alleles at leptin gene (chr 6)
  - measured and mapped body weight, insulin, glucose at various ages (Stoehr et al. 2000 Diabetes)
  - sacrificed at 14 weeks, tissues preserved
- gene expression data
  - Affymetrix microarrays on parental strains, F1
    - key tissues: adipose, liver, muscle, β-cells
    - novel discoveries of differential expression (Nadler et al. 2000 PNAS; Lan et al. 2002 in review; Ntambi et al. 2002 PNAS)
  - RT-PCR on 108 F2 mice liver tissues
    - 15 genes, selected as important in diabetes pathways
    - SCD1, PEPCK, ACO, FAS, GPAT, PPARgamma, PPARalpha, G6Pase, PDI,…

Multiple Interval Mapping (QTLCart)
SCD1: multiple QTL plus epistasis!

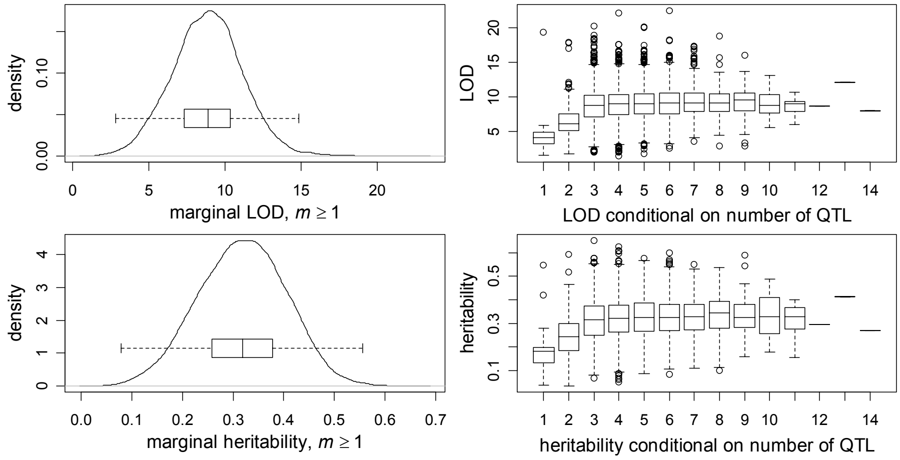


Bayesian model assessment: number of QTL for SCD1

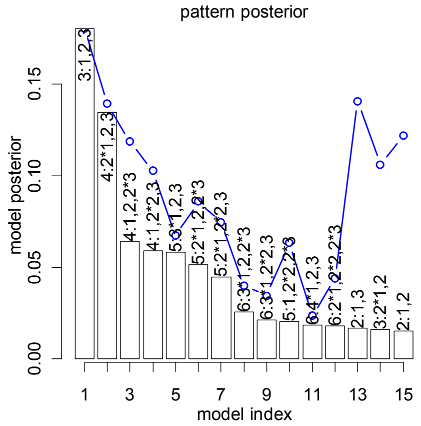


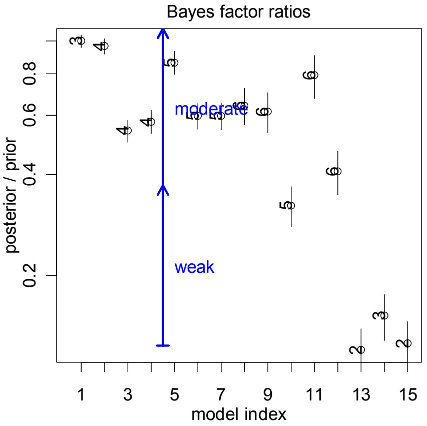


Bayesian LOD and h^2 for SCD1



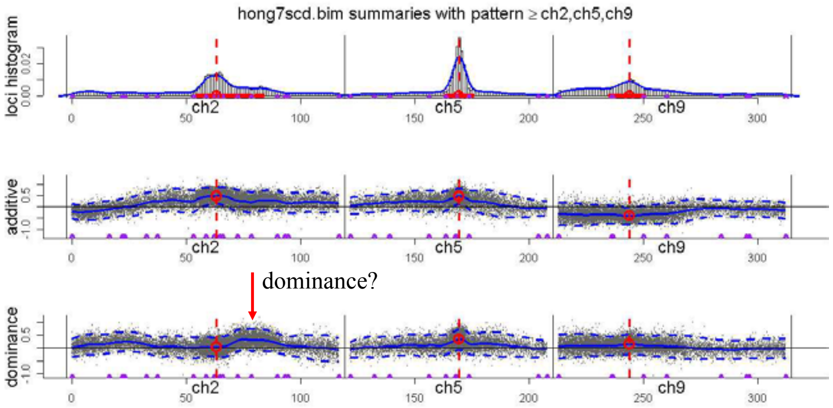
Bayesian model assessment: chromosome QTL pattern for SCD1



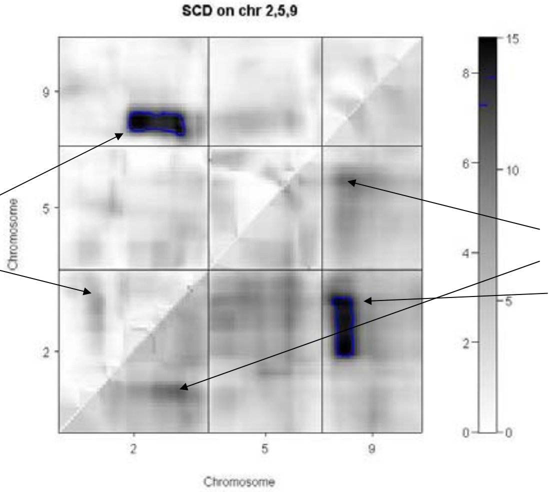


trans-acting QTL for SCD1

(no epistasis yet: see Yi, Xu, Allison 2003)



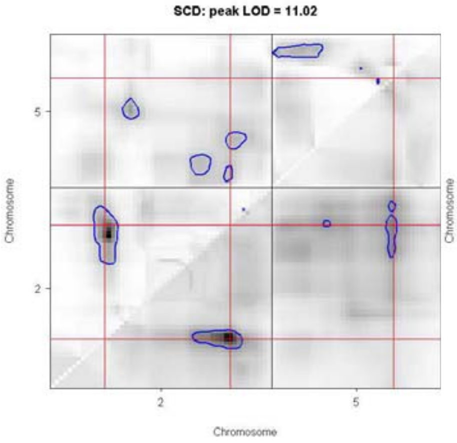
2-D scan: assumes only 2 QTL!

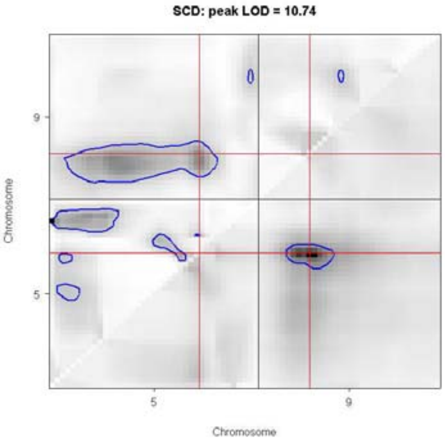


epistasis LOD peaks

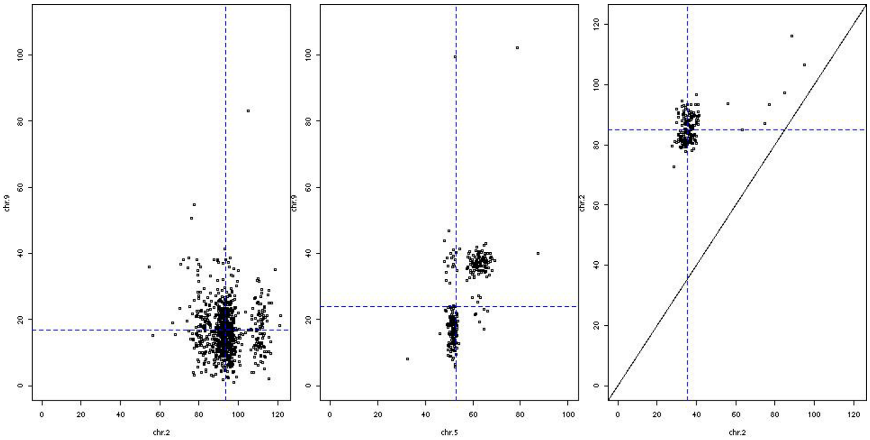
joint LOD peaks

sub-peaks can be easily overlooked!

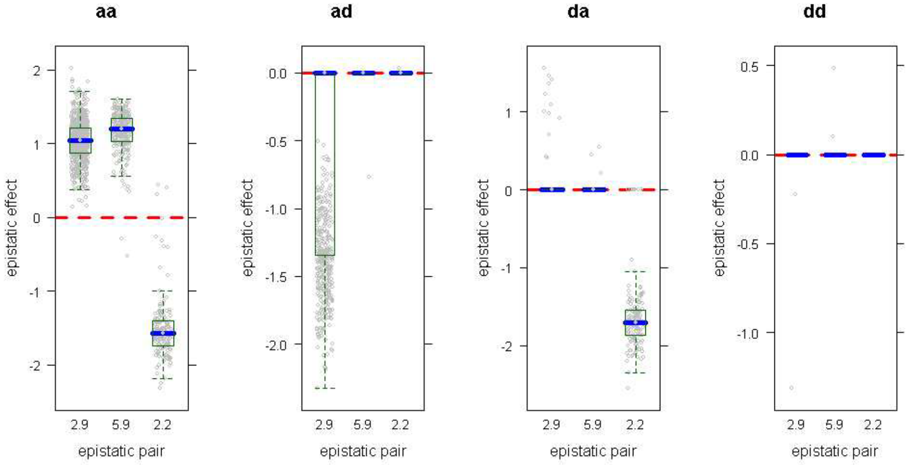




epistatic model fit



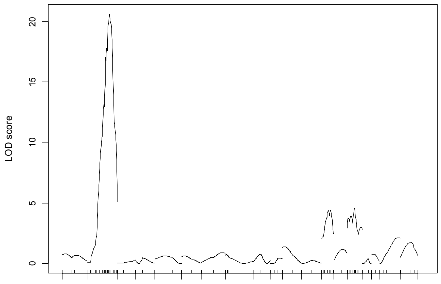
Cockerham epistatic effects



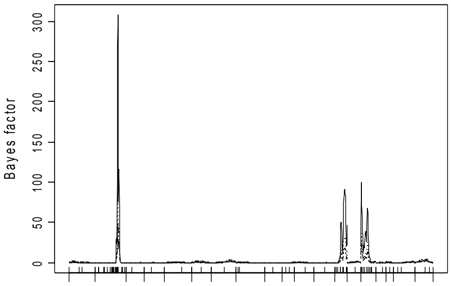
obesity in CAST/Ei BC onto M16i

- 421 mice (Daniel Pomp)
  - (213 male, 208 female)
- 92 microsatellites on 19 chromosomes
  - 1214 cM map
- subcutaneous fat pads
  - pre-adjusted for sex and dam effects
- Yi, Yandell, Churchill, Allison, Eisen, Pomp (2005) *Genetics* (in press)

non-epistatic analysis

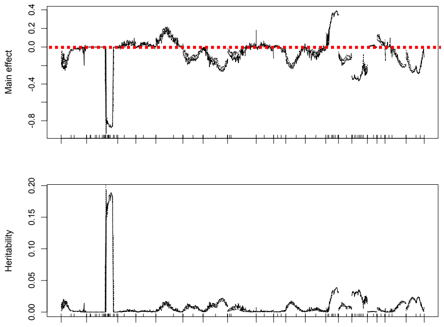


single QTL LOD profile

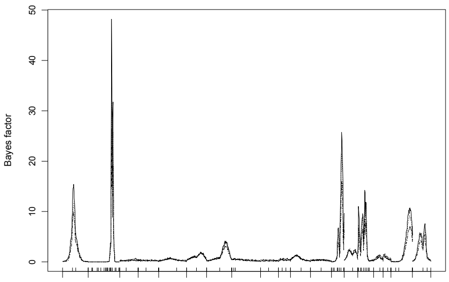


multiple QTL
Bayes factor profile

posterior profile of main effects in epistatic analysis

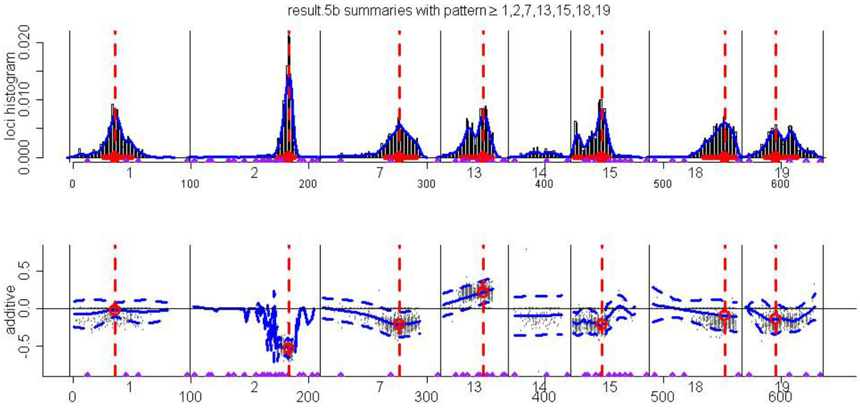


main effects & heritability profile



Bayes factor profile

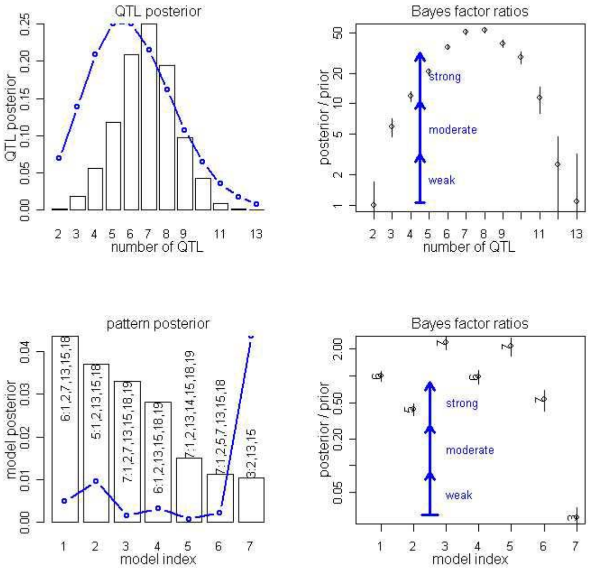
posterior profile of main effects in epistatic analysis



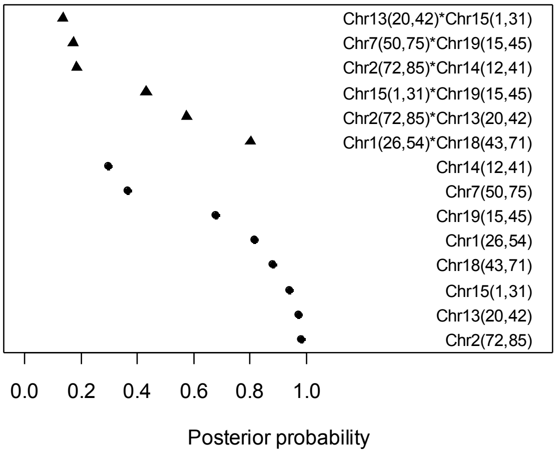
model selection
via
Bayes factors
for
epistatic model

number of QTL

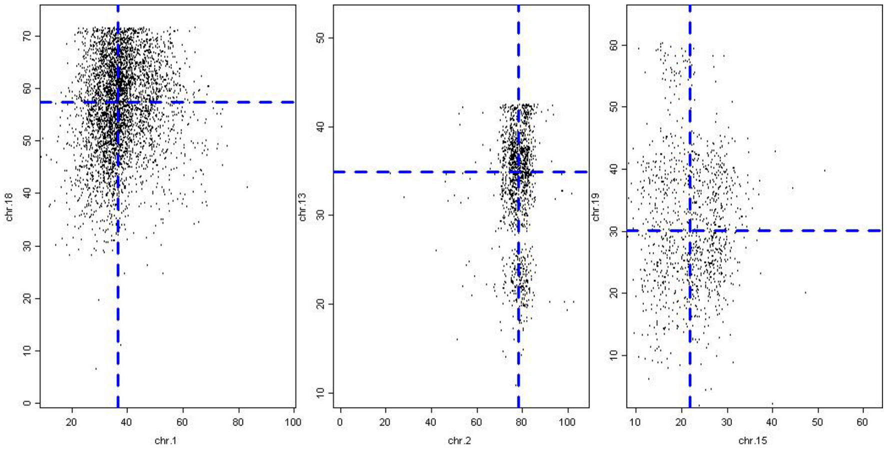
QTL pattern



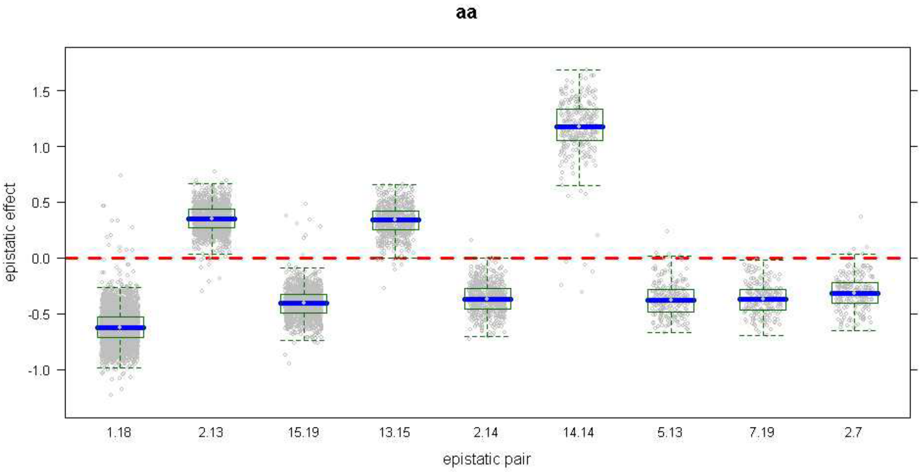
posterior probability of effects



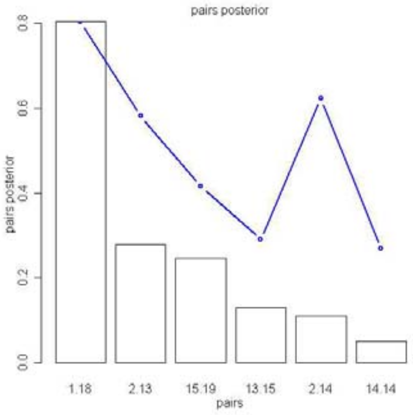
scatterplot estimates of epistatic loci

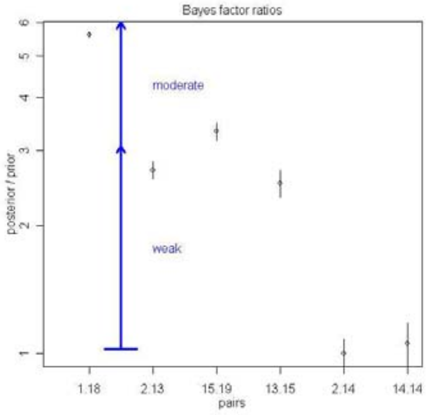


stronger epistatic effects

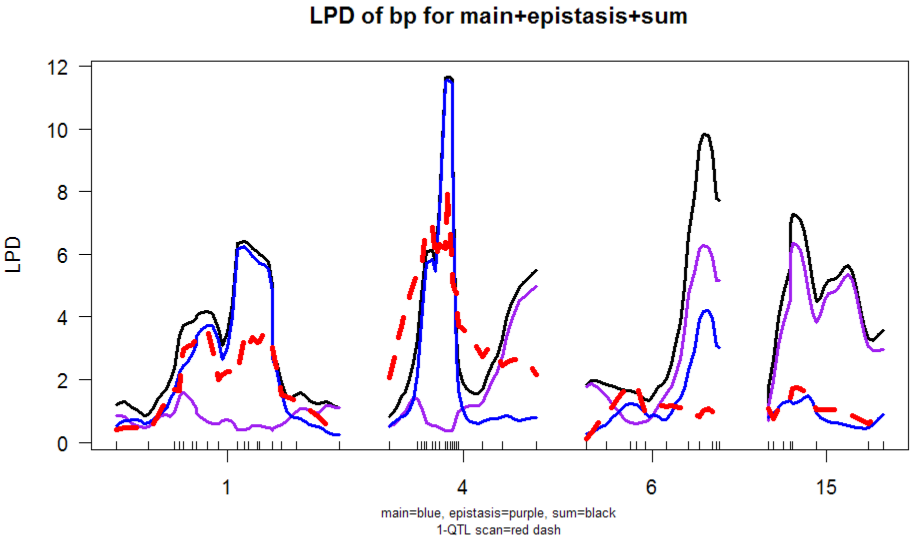


model selection for pairs

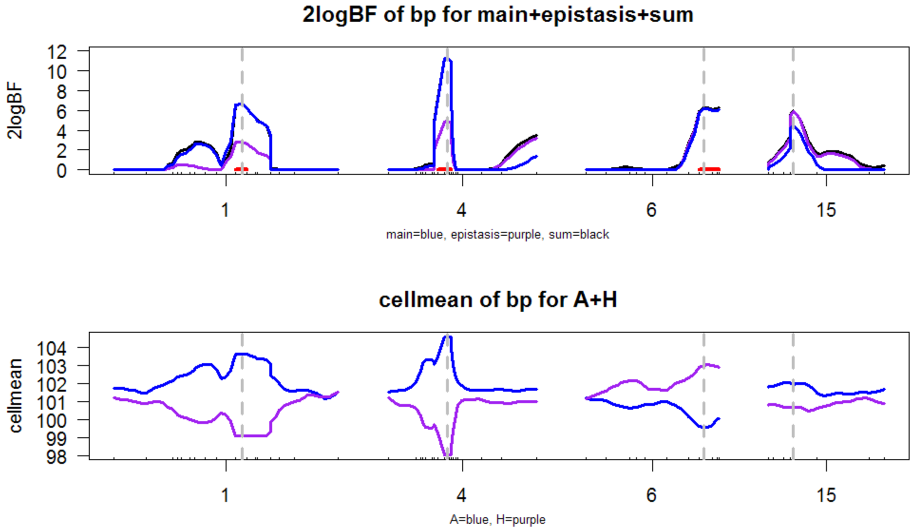




hyper data: scanone

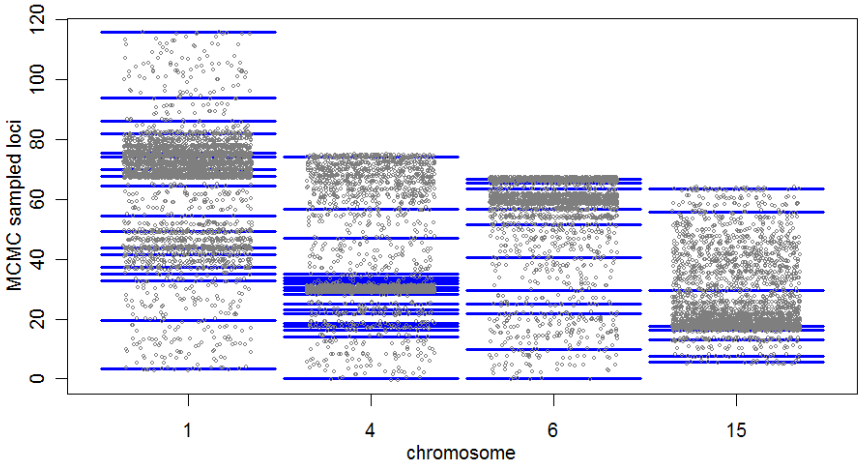


2log(BF) scan with 50% HPD region

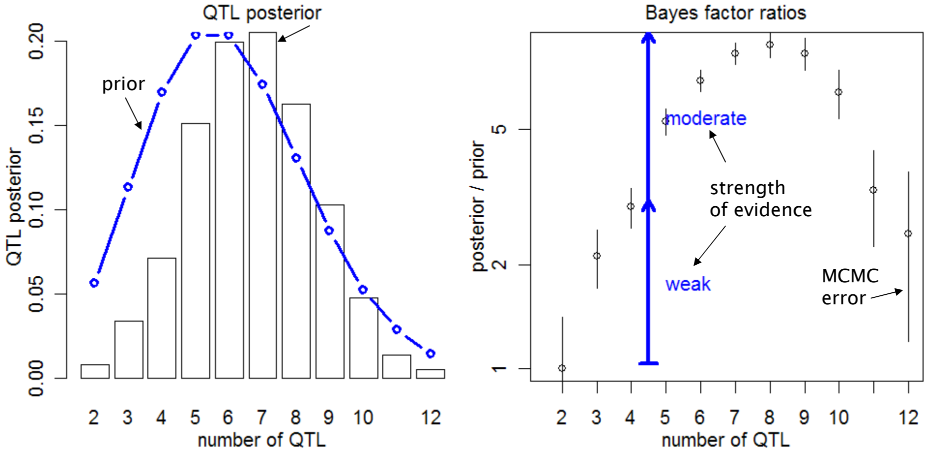


sampled QTL by chromosome

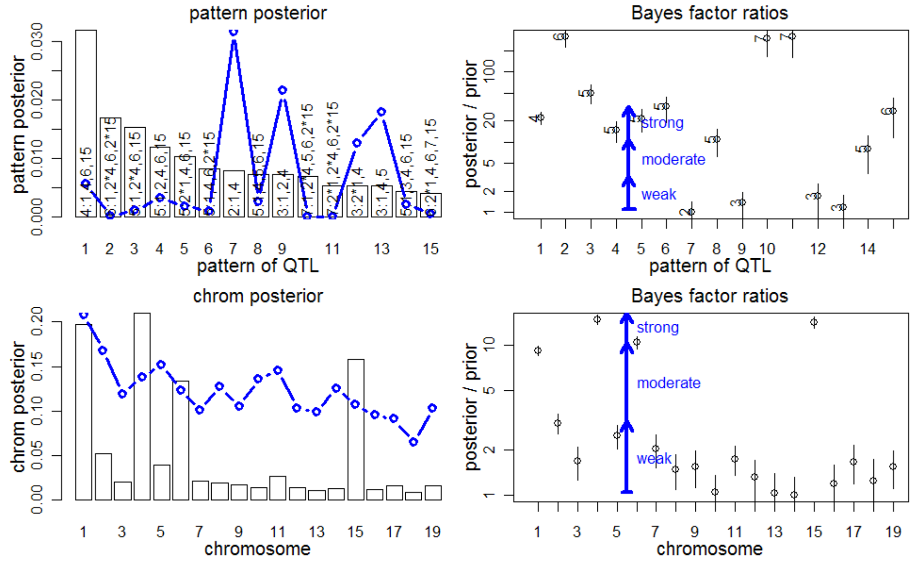
blue lines = markers



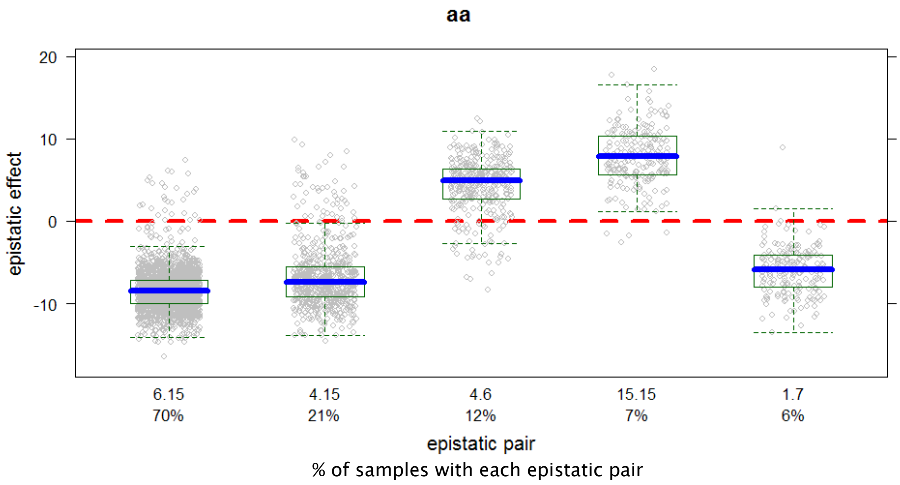
hyper: number of QTL posterior, prior, Bayes factors



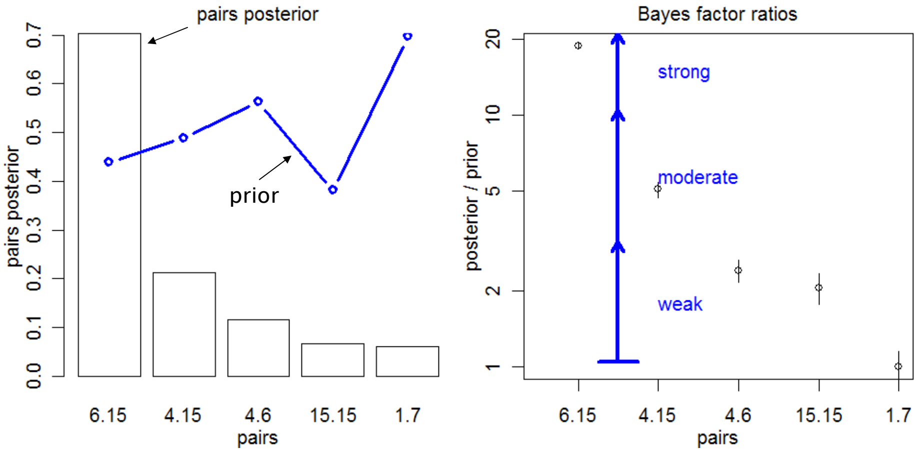
pattern of QTL on chromosomes



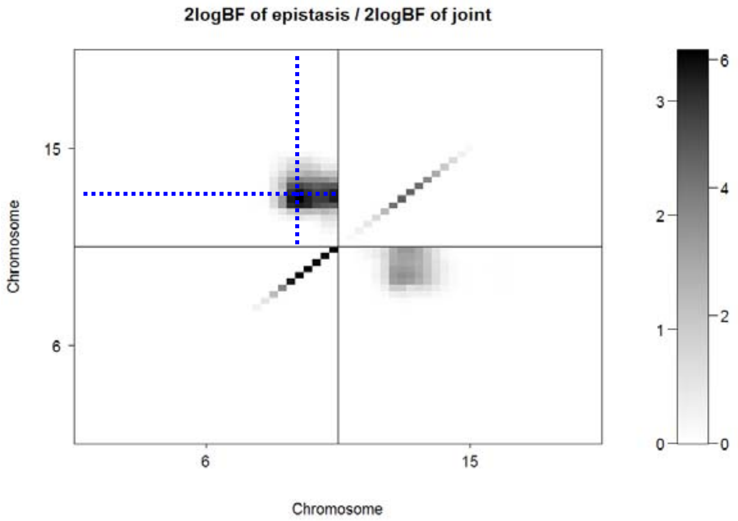
Cockerham epistatic effects



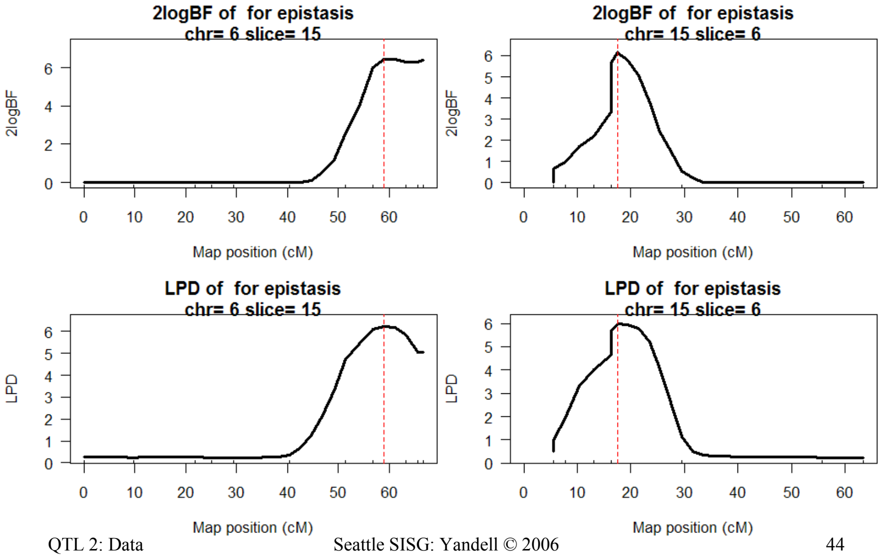
relative importance of epistasis



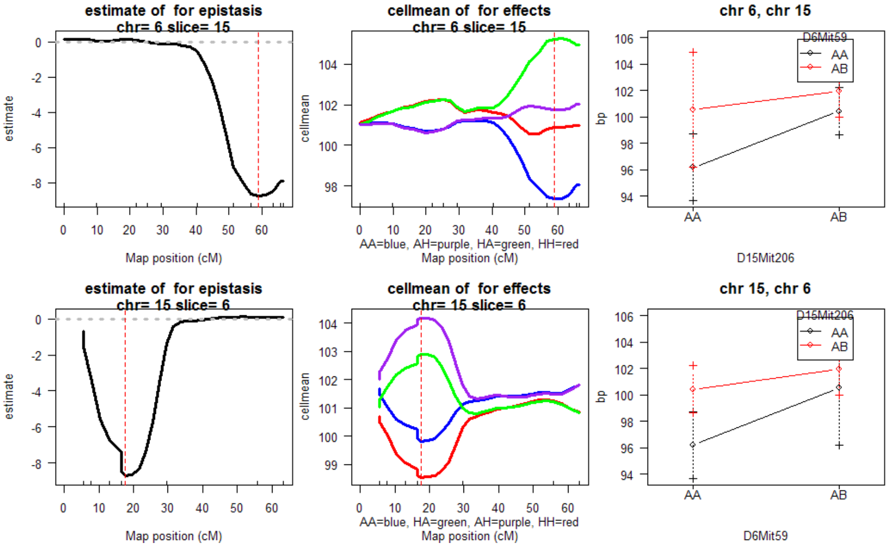
2-D plot of 2logBF: chr 6 & 15



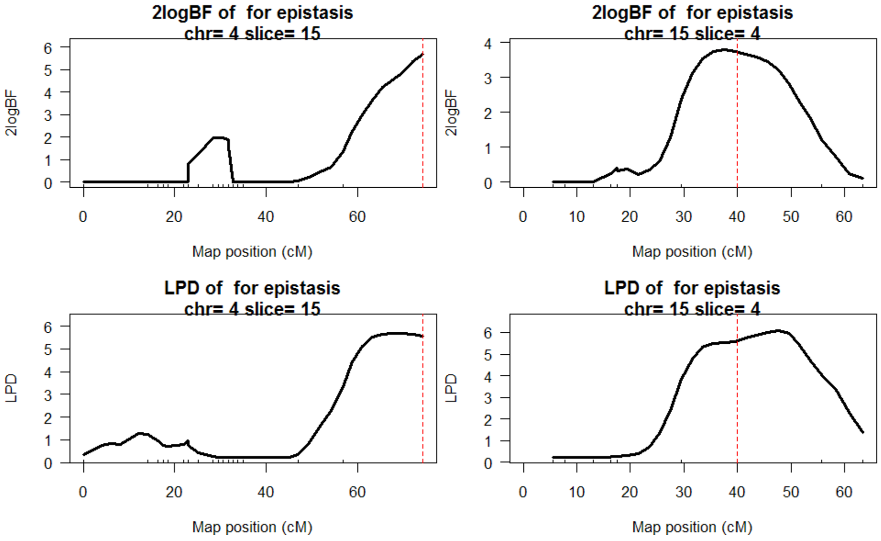
1-D Slices of 2-D scans: chr 6 & 15



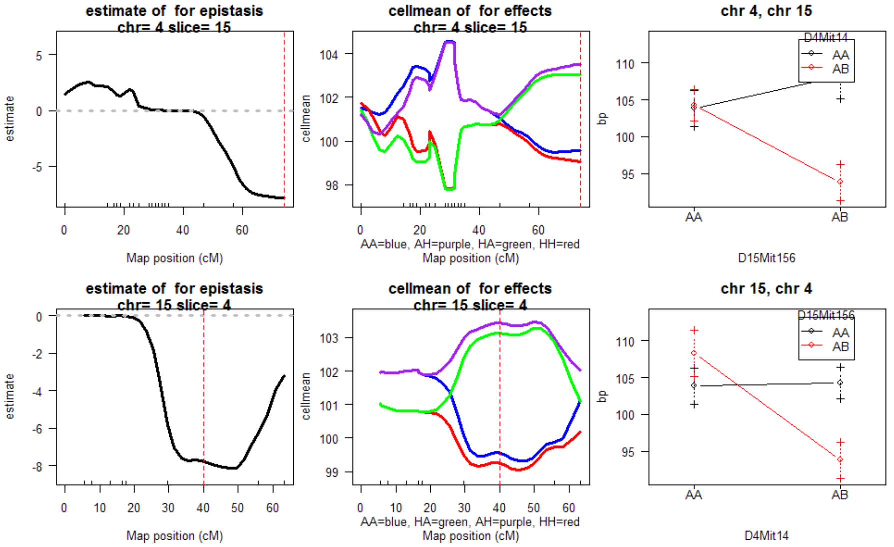
1-D Slices of 2-D scans: chr 6 & 15



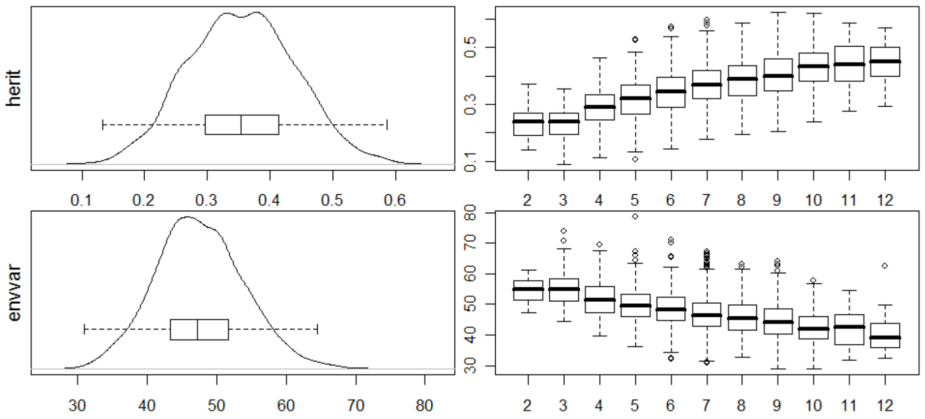
1-D Slices of 2-D scans: chr 4 & 15



1-D Slices of 2-D scans: chr 4 & 15



diagnostic summaries

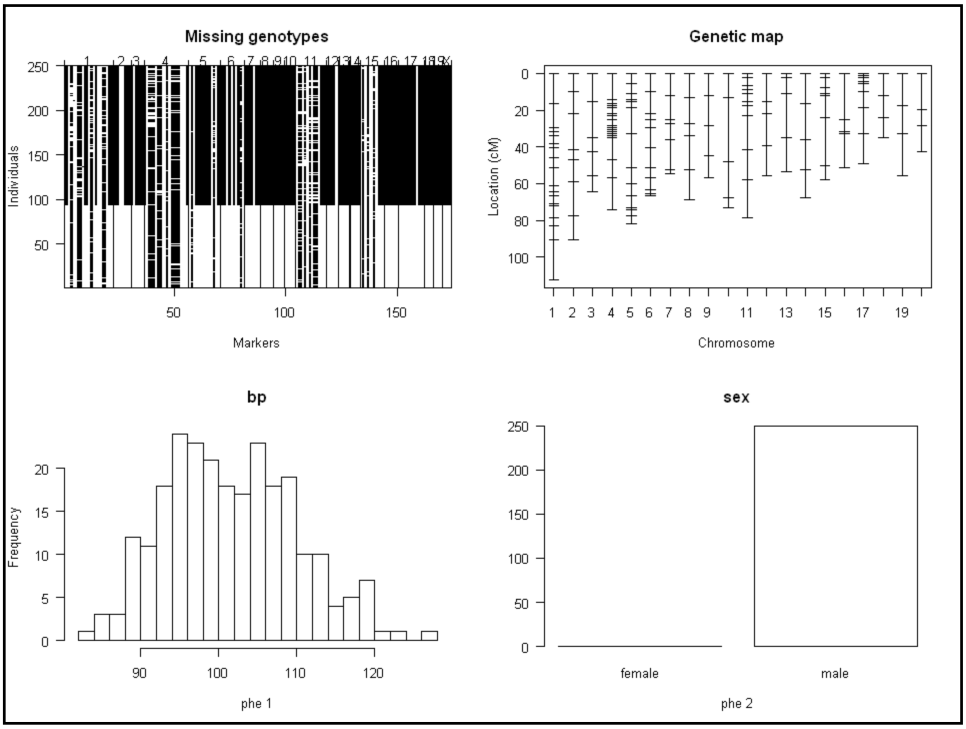


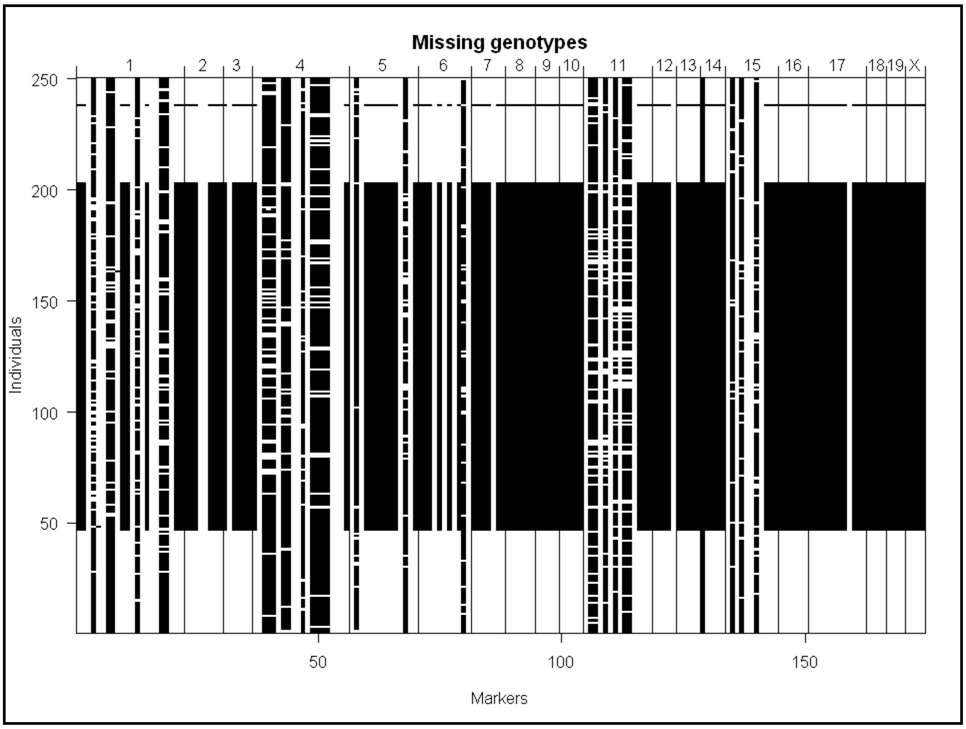
R/qtl & R/qtlbim Tutorials

- R statistical graphics & language system
- R/qtl tutorial
  - R/qtl web site: www.rqtl.org
  - Tutorial: www.rqtl.org/tutorials/rqtltour.pdf
  - R code: www.rqtl.org/tutorials/rqtltour.R
- R/qtlbim tutorial
  - R/qtlbim web site: www.qtlbim.org
  - Tutorial: www.stat.wisc.edu/~yandell/qtlbim/rqtltour.pdf
  - R code: www.stat.wisc.edu/~yandell/qtlbim/rqtltour.R

R/qtl tutorial (www.rqtl.org)

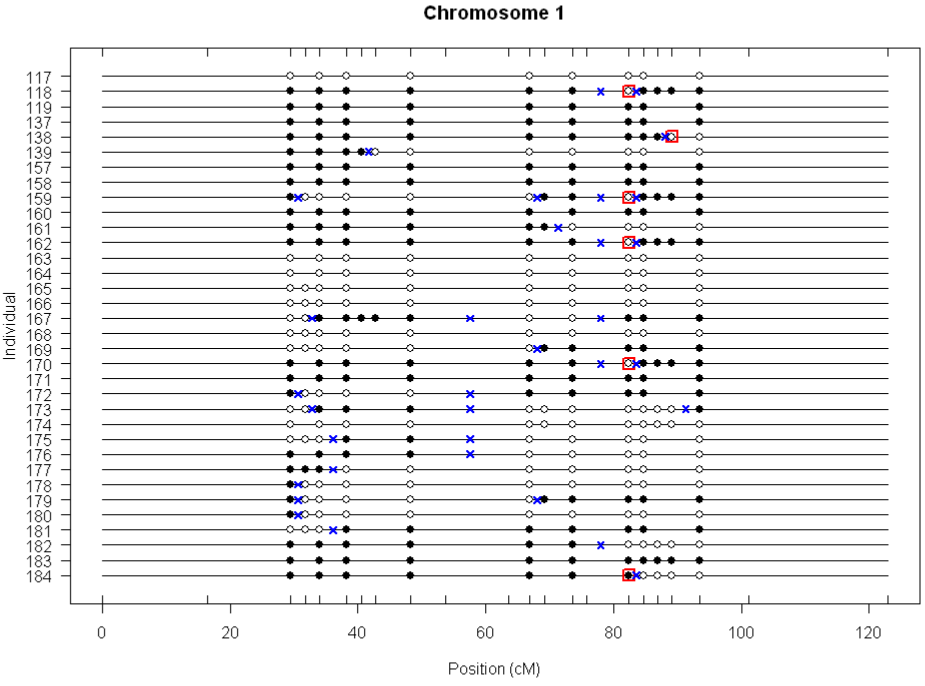
```
> library(qtl)
> data(hyper)
> summary(hyper)
    Backcross

    No. individuals:    250

    No. phenotypes:     2
    Percent phenotyped: 100 100

    No. chromosomes:    20
        Autosomes:      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
        X chr:          X

    Total markers:      174
    No. markers:        22 8 6 20 14 11 7 6 5 5 14 5 5 5 11 6 12 4 4 4
    Percent genotyped:  47.7
    Genotypes (%):      AA:50.2  AB:49.8
> plot(hyper)
> plot.missing(hyper, reorder = TRUE)
```


Missing genotypes
Individuals
250
200
150
100
50
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 X
50
100
150
Markers
Genetic map
Location (cM)
0
20
40
60
80
100
1 2 3 4 5 6 7 8 9 11 13 15 17 19
Chromosome
bp
Frequency
0
5
10
15
20
90
100
110
120
phe 1
sex
0
50
100
150
200
250
female
male
phe 2


Missing genotypes
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 X
Individuals
250
200
150
100
50
50
100
150
Markers

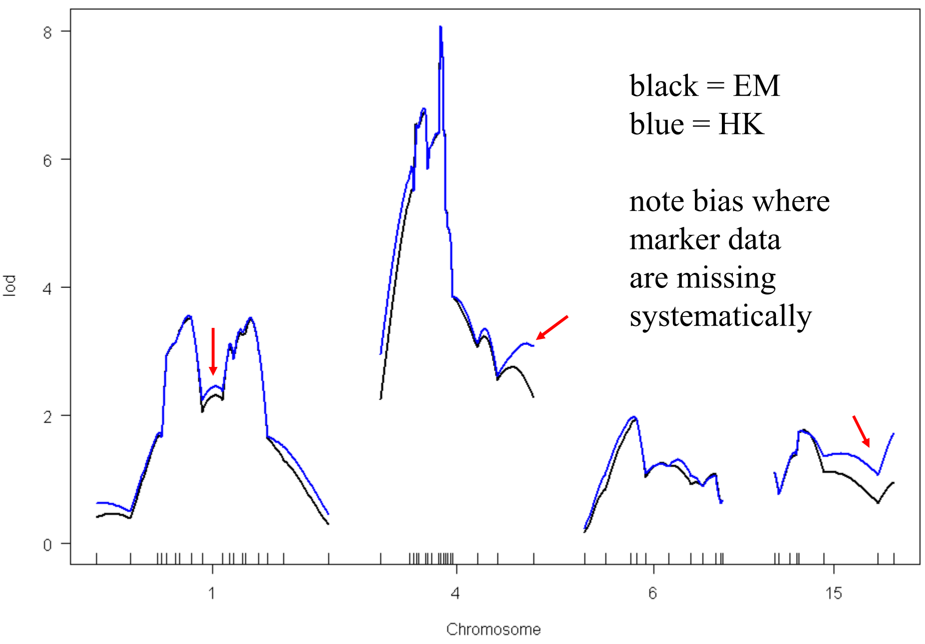
R/qtl: find genotyping errors

```
> hyper <- calc.errorlod(hyper, error.prob=0.01)
> top.errorlod(hyper)
   chr  id    marker errorlod
1    1 118   D1Mit14 8.372794
2    1 162   D1Mit14 8.372794
3    1 170   D1Mit14 8.372794
4    1 159   D1Mit14 8.350341
5    1  73   D1Mit14 6.165395
6    1  65   D1Mit14 6.165395
7    1  88   D1Mit14 6.165395
8    1 184   D1Mit14 6.151606
9    1 241   D1Mit14 6.151606
...
16   1 215  D1Mit267 5.822192
17   1 108  D1Mit267 5.822192
18   1 138  D1Mit267 5.822192
19   1 226  D1Mit267 5.822192
20   1 199  D1Mit267 5.819250
21   1  84  D1Mit267 5.808400
> plot.geno(hyper, chr=1, ind=c(117:119,137:139,157:184))
```



R/qtl: 1 QTL interval mapping

```
> hyper <- calc.genoprob(hyper, step=1,
  error.prob=0.01)
> out.em <- scanone(hyper)
> out.hk <- scanone(hyper, method="hk")
> summary(out.em, threshold=3)
         chr  pos  lod
c1.loc45   1 48.3 3.52
D4Mit164   4 29.5 8.02
> summary(out.hk, threshold=3)
         chr  pos  lod
c1.loc45   1 48.3 3.55
D4Mit164   4 29.5 8.09
```

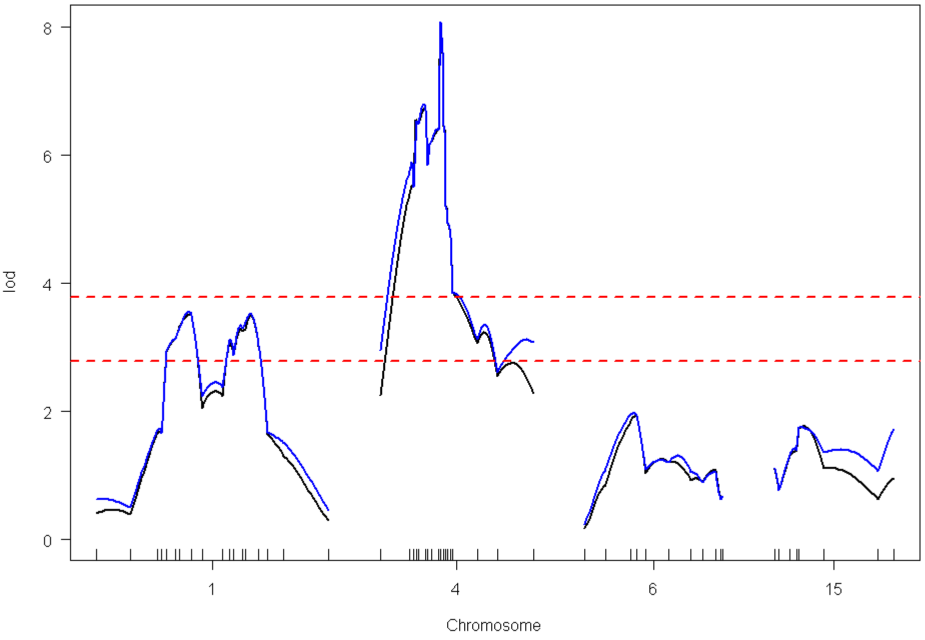


black = EM
blue = HK

note bias where marker data are missing systematically

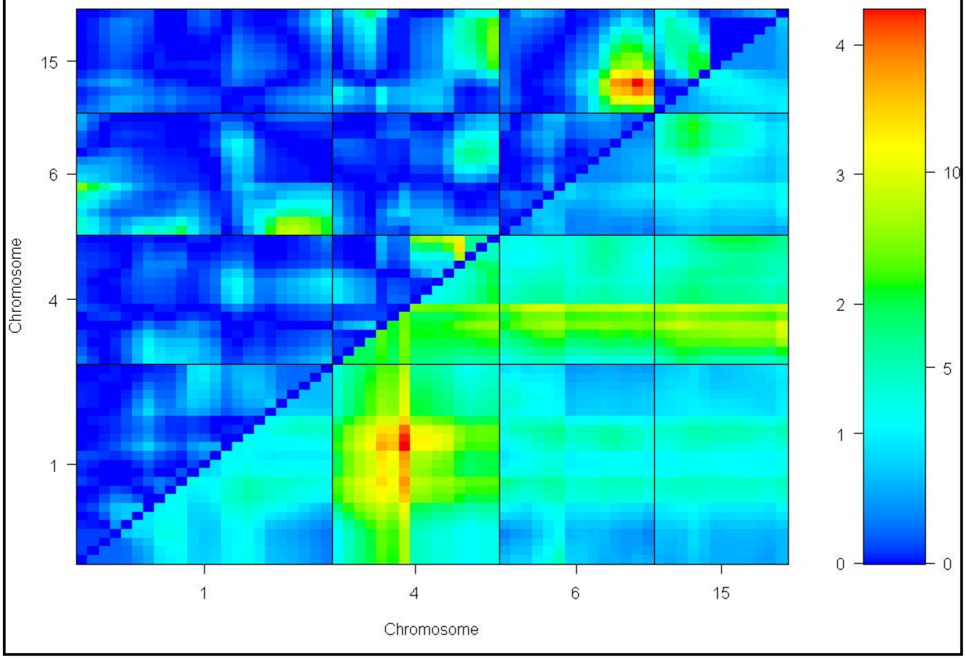
R/qtl: permutation threshold

```
> operm.hk <- scanone(hyper, method="hk",
  n.perm=1000)
Doing permutation in batch mode ...
> summary(operm.hk, alpha=c(0.01,0.05))
LOD thresholds (1000 permutations)
    lod
1% 3.79
5% 2.78
> summary(out.hk, perms=operm.hk, alpha=0.05,
  pvalues=TRUE)
  chr  pos  lod  pval
1   1 48.3 3.55 0.015
2   4 29.5 8.09 0.000
```



R/qtl: 2 QTL scan

```
> hyper <- calc.genoprob(hyper, step=5, error.prob=0.01)
>
> out2.hk <- scantwo(hyper, method="hk")
 --Running scanone
 --Running scantwo
 (1,1)
 (1,2)
...
 (19,19)
 (19,X)
 (X,X)
> summary(out2.hk, thresholds=c(6.0, 4.7, 4.4, 4.7, 2.6))
        pos1f pos2f lod.full lod.fv1 lod.int     pos1a pos2a lod.add lod.av1
c1 :c4   68.3  30.0    14.13    6.51   0.225      68.3  30.0   13.90   6.288
c2 :c19  47.7   0.0     6.71    5.01   3.458      52.7   0.0    3.25   1.552
c3 :c3   37.2  42.2     6.10    5.08   0.226      37.2  42.2    5.87   4.853
c6 :c15  60.0  20.5     7.17    5.22   3.237      25.0  20.5    3.93   1.984
c9 :c18  67.0  37.2     6.31    4.79   4.083      67.0  12.2    2.23   0.708
c12:c19   1.1  40.0     6.48    4.79   4.090       1.1   0.0    2.39   0.697
> plot(out2.hk, chr=c(1,4,6,15))
```



R/qtl: ANOVA imputation at QTL

```
> hyper <- sim.geno(hyper, step=2, n.draws=16, error.prob=0.01)
> qtl <- makeqtl(hyper, chr = c(1, 1, 4, 6, 15), pos = c(50, 76, 30, 70, 20))

> my.formula <- y ~ Q1 + Q2 + Q3 + Q4 + Q5 + Q4:Q5
> out.fitqtl <- fitqtl(hyper$pheno[,1], qtl, formula=my.formula)
> summary(out.fitqtl)

Full model result
----------------------------------
Model formula is:  y ~ Q1 + Q2 + Q3 + Q4 + Q5 + Q4:Q5

       df        SS        MS      LOD     %var Pvalue(Chi2) Pvalue(F)
Model   6  5789.089 964.84822 21.54994 32.76422            0         0
Error 243 11879.847  48.88826
Total 249 17668.936

Drop one QTL at a time ANOVA table:
----------------------------------
                 df Type III SS      LOD     %var F value Pvalue(F)
Chr1@50           1     297.149    1.341    1.682   6.078   0.01438 *
Chr1@76           1     520.664    2.329    2.947  10.650   0.00126 **
Chr4@30           1    2842.089   11.644   16.085  58.134  5.50e-13 ***
Chr6@70           2    1435.721    6.194    8.126  14.684  9.55e-07 ***
Chr15@20          2    1083.842    4.740    6.134  11.085  2.47e-05 ***
Chr6@70:Chr15@20  1     955.268    4.199    5.406  19.540  1.49e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

R/qtlbim (www.qtlbim.org)

- cross-compatible with R/qtl
- model selection for genetic architecture
  - epistasis, fixed & random covariates, GxE
  - samples multiple genetic architectures
  - examines summaries over nested models
- extensive graphics

R/qtlbim: tutorial

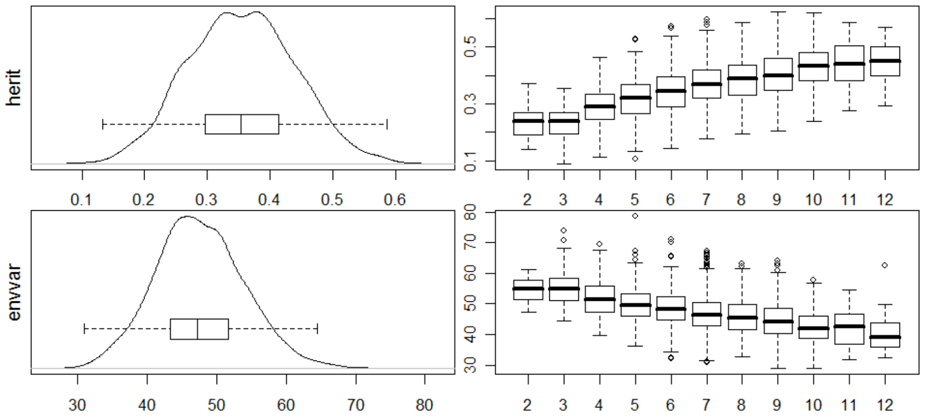
(www.stat.wisc.edu/~yandell/qtlbim)

```
> data(hyper)
## Drop X chromosome (for now).
> hyper <- subset(hyper, chr=1:19)
> hyper <- qb.genoprob(hyper, step=2)
## This is the time-consuming step:
> qbHyper <- qb.mcmc(hyper, pheno.col = 1)
## Here we get stored samples.
> qb.load(hyper, qbHyper)
> summary(qbHyper)
```

R/qtlbim: initial summaries

```
> summary(qbHyper)
Bayesian model selection QTL mapping object qbHyper on cross object hyper
had 3000 iterations recorded at each 40 steps with 1200 burn-in steps.

Diagnostic summaries:
          nqtl   mean envvar varadd  varaa    var
Min.     2.000  97.42  28.07  5.112  0.000  5.112
1st Qu.  5.000 101.00  44.33 17.010  1.639 20.180
Median   7.000 101.30  48.57 20.060  4.580 25.160
Mean     6.543 101.30  48.80 20.310  5.321 25.630
3rd Qu.  8.000 101.70  53.11 23.480  7.862 30.370
Max.    13.000 103.90  74.03 51.730 34.940 65.220

Percentages for number of QTL detected:
 2  3  4  5  6  7  8  9 10 11 12 13
 2  3  9 14 21 19 17 10  4  1  0  0

Percentages for number of epistatic pairs detected:
pairs
 1  2  3  4  5  6
29 31 23 11  5  1

Percentages for common epistatic pairs:
 6.15  4.15   4.6   1.7 15.15   1.4   1.6   4.9  1.15  1.17   1.5  5.11   1.2  7.15   1.1
   63    18    10     6     6     5     4     4     3     3     3     2     2     2     2
> plot(qb.diag(qbHyper, items = c("herit", "envvar")))
```

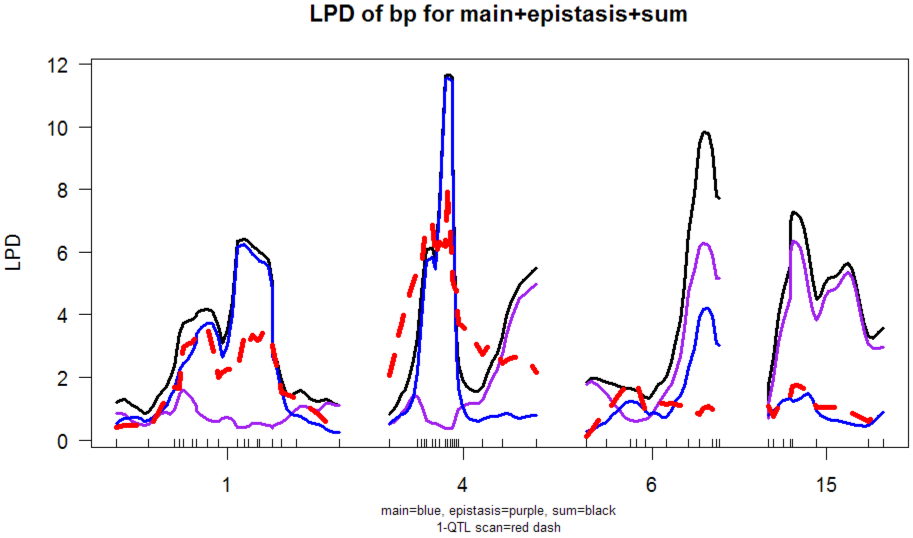
diagnostic summaries



R/qtlbim: 1-D (*not* 1-QTL!) scan

```
> one <- qb.scanone(qbHyper, chr = c(1,4,6,15),
  type = "LPD")
> summary(one)
LPD of bp for main,epistasis,sum

    n.qtl  pos m.pos e.pos  main epistasis   sum
c1  1.331 64.5  64.5  67.8  6.10     0.442  6.27
c4  1.377 29.5  29.5  29.5 11.49     0.375 11.61
c6  0.838 59.0  59.0  59.0  3.99     6.265  9.60
c15 0.961 17.5  17.5  17.5  1.30     6.325  7.28
> plot(one)
> plot(out.em, chr=c(1,4,6,15), add = TRUE, col =
  "red", lty = 2)
```

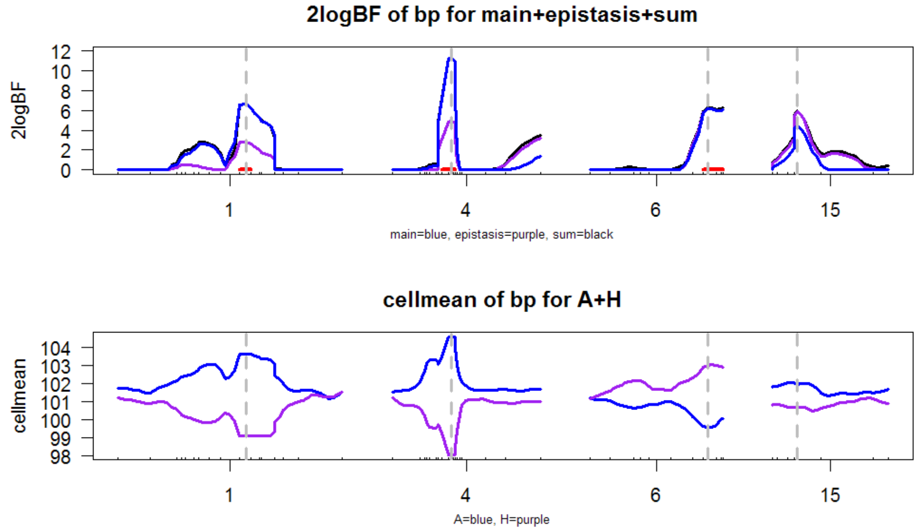
hyper data: scanone



R/qtlbim: automated QTL selection

```
> hpd <- qb.hpdone(qbHyper, profile = "2logBF")
> summary(hpd)
   chr n.qtl  pos lo.50% hi.50% 2logBF       A       H
1    1 0.829 64.5   64.5   72.1  6.692 103.611  99.090
4    4 3.228 29.5   25.1   31.7 11.169 104.584  98.020
6    6 1.033 59.0   56.8   66.7  6.054  99.637 102.965
15  15 0.159 17.5   17.5   17.5  5.837 101.972 100.702
> plot(hpd)
```

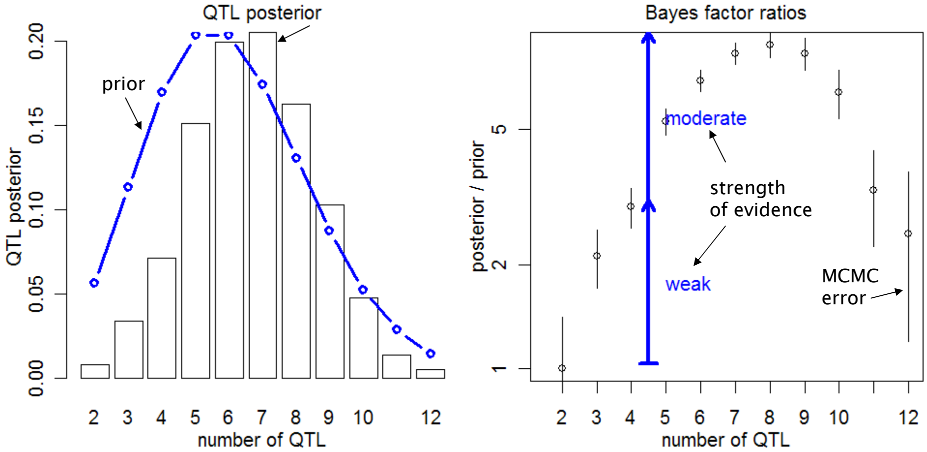
2log(BF) scan with 50% HPD region



R/qtlbim: Bayes Factor evaluations

```
> tmp <- qb.BayesFactor(qbHyper)
> summary(tmp)
$nqtl

$pattern
                 posterior    prior     bf   bfse
7:2*1,2*15,2*4,6   0.00500 3.17e-07 220.00 56.700
6:1,2*15,2*4,6     0.01400 1.02e-06 192.00 29.400
7:1,2*15,2*4,5,6   0.00600 4.49e-07 186.00 43.800
7:1,2*15,2,2*4,6   0.00433 5.39e-07 112.00 31.000
5:1,15,2*4,6       0.00867 5.81e-06  20.80  4.060
5:1,15,4,2*6       0.00733 5.22e-06  19.60  4.170
4:1,15,4,6         0.03770 2.71e-05  19.40  1.790

$chrom
   posterior  prior    bf  bfse
4     0.2100 0.0595 15.00 0.529
15    0.1470 0.0464 13.40 0.589
6     0.1280 0.0534 10.10 0.483
1     0.2030 0.0901  9.55 0.345
> plot(tmp)
```

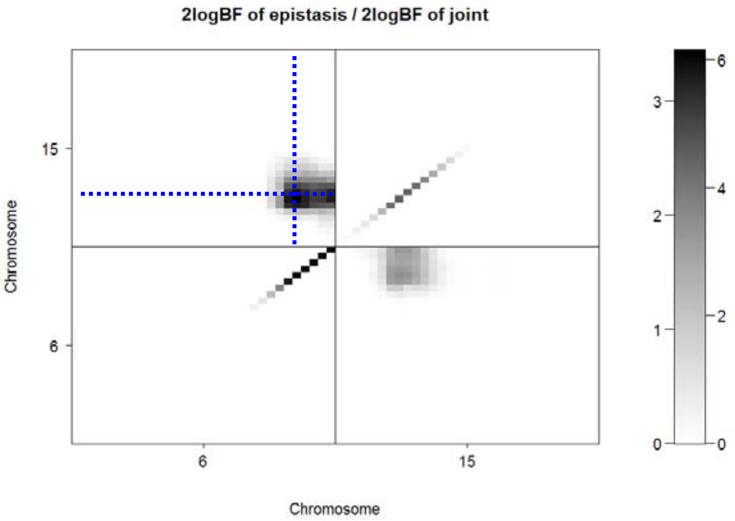
hyper: number of QTL posterior, prior, Bayes factors



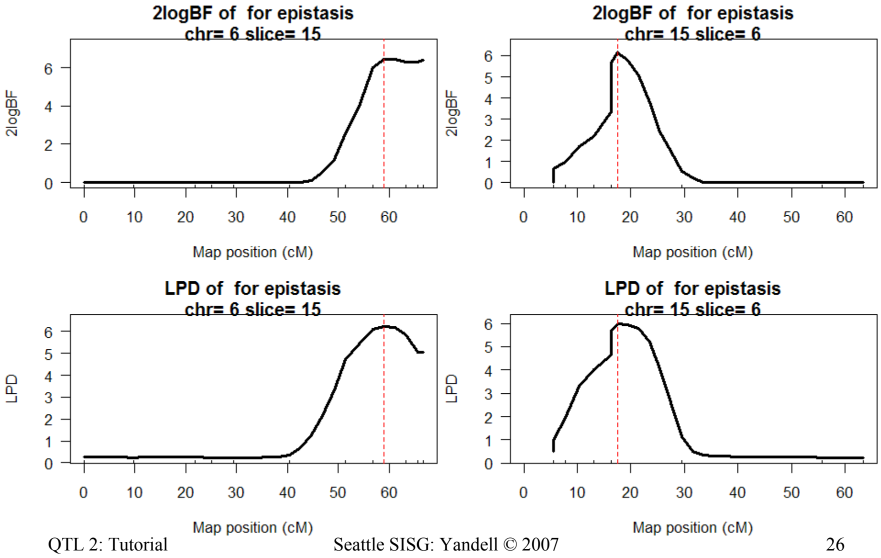
R/qtlbim: 2-D (*not* 2-QTL) scans

```
> two <- qb.scantwo(qbHyper, chr = c(6,15),
  type = "2logBF")
> plot(two)
> plot(two, chr = 6, slice = 15, show.locus =
  FALSE)
> plot(two, chr = 15, slice = 6, show.locus =
  FALSE)
> two <- qb.scantwo(qbHyper, chr = c(6,15),
  type = "LPD")
> plot(two, chr = 6, slice = 15, show.locus =
  FALSE)
> plot(two, chr = 15, slice = 6, show.locus =
  FALSE)
```

2-D plot of 2logBF: chr 6 & 15



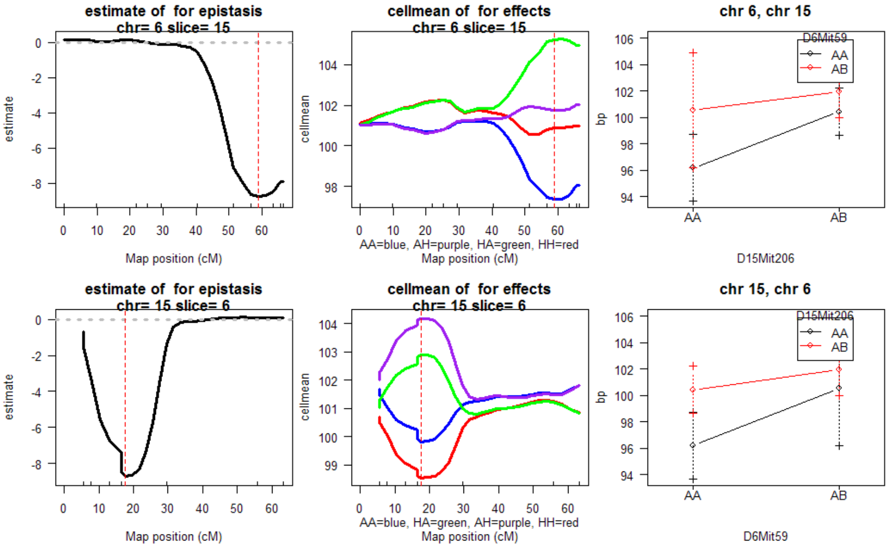
1-D Slices of 2-D scans: chr 6 & 15

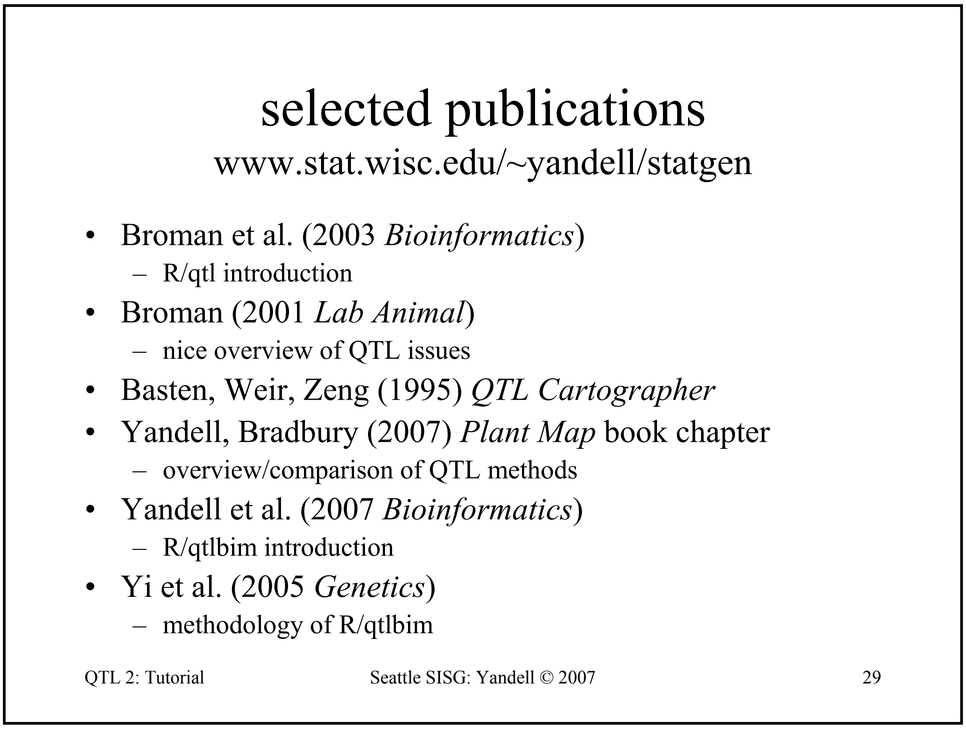


R/qtlbim: slice of epistasis

```
> slice = qb.slicetwo(qbHyper, c(6,15), c(59,19.5))
> summary(slice)
2logBF of bp for epistasis

    n.qtl  pos m.pos e.pos epistasis slice
c6  0.838 59.0  59.0  66.7      15.8  18.1
c15 0.961 17.5  17.5  17.5      15.5  60.6

cellmean of bp for AA,HA,AH,HH

    n.qtl  pos m.pos   AA  HA  AH    HH slice
c6  0.838 59.0  59.0 97.4 105 102 100.8  18.1
c15 0.961 17.5  17.5 99.8 103 104  98.5  60.6

estimate of bp for epistasis

    n.qtl  pos m.pos e.pos epistasis slice
c6  0.838 59.0  59.0  66.7     -7.86  18.1
c15 0.961 17.5  17.5  17.5     -8.72  60.6
> plot(slice, figs = c("effects", "cellmean", "effectplot"))
```

1-D Slices of 2-D scans: chr 6 & 15





selected publications

www.stat.wisc.edu/~yandell/statgen

- Broman et al. (2003 *Bioinformatics*)
  - R/qtl introduction
- Broman (2001 *Lab Animal*)
  - nice overview of QTL issues
- Basten, Weir, Zeng (1995) *QTL Cartographer*
- Yandell, Bradbury (2007) *Plant Map* book chapter
  - overview/comparison of QTL methods
- Yandell et al. (2007 *Bioinformatics*)
  - R/qtlbim introduction
- Yi et al. (2005 *Genetics*)
  - methodology of R/qtlbim

A brief tour of R/qtl

Karl W Broman

Department of Biostatistics, Johns Hopkins University

http://www.rqtl.org

16 January 2007

Overview of R/qtl

R/qtl is an extensible, interactive environment for mapping quantitative trait loci (QTL) in experimental crosses. It is implemented as an add-on package for the freely available and widely used statistical language/software R (see www.r-project.org). The development of this software as an add-on to R allows us to take advantage of the basic mathematical and statistical functions, and powerful graphics capabilities, that are provided with R. Further, the user will benefit by the seamless integration of the QTL mapping software into a general statistical analysis program. Our goal is to make complex QTL mapping methods widely accessible and allow users to focus on modeling rather than computing.

A key component of computational methods for QTL mapping is the hidden Markov model (HMM) technology for dealing with missing genotype data. We have implemented the main HMM algorithms, with allowance for the presence of genotyping errors, for backcrosses, intercrosses, and phase-known four-way crosses.

The current version of R/qtl includes facilities for estimating genetic maps, identifying genotyping errors, and performing single-QTL genome scans and two-QTL, two-dimensional genome scans, by interval mapping (with the EM algorithm), Haley-Knott regression, and multiple imputation. All of this may be done in the presence of covariates (such as sex, age or treatment). One may also fit higher-order QTL models by multiple imputation.

R/qtl is distributed as source code for Unix or compiled code for Windows or Mac OS X. R/qtl is released under the GNU General Public License. To download the software, you must agree to the terms in that license.

Overview of R

R is an open-source implementation of the S language. As described on the R-project homepage (www.r-project.org):

> R is a system for statistical computation and graphics. It consists of a language plus a run-time environment with graphics, a debugger, access to certain system functions, and the ability to run programs stored in script files.
>
> The core of R is an interpreted computer language which allows branching and looping as well as modular programming using functions. Most of the user-visible functions in R are written in R. It is possible for the user to interface to procedures written in the C, C++, or FORTRAN languages for efficiency. The R distribution contains functionality for a large number of statistical procedures. Among these are: linear and generalized linear models, nonlinear regression models, time series analysis, classical parametric and nonparametric tests, clustering and smoothing. There is also a large set of functions which provide a flexible graphical environment for creating various kinds of data presentations. Additional modules are available for a variety of specific purposes.

R is freely available for Windows, Unix and Mac OS X, and may be downloaded from the Comprehensive R Archive Network (CRAN; cran.r-project.org).

Learning R may require a formidable investment of time, but it will definitely be worth the effort. Numerous free documents on getting started with R are available on CRAN. In additional, several books are available. The most important book on R is Venables and Ripley (2002) *Modern Applied Statistics with S*, 4th edition. Dalgaard (2002) *Introductory Statistics with R* provides a more gentle introduction.

Citation for R/qtl

To cite R/qtl in publications, use

> Broman KW, Wu H, Sen S, Churchill GA (2003) R/qtl: QTL mapping in experimental crosses. Bioinformatics 19:889-890

Selected R/qtl functions

Sample data	badorder	An intercross with misplaced markers
	bristle3	Data on bristle number for Drosophila chromosome 3
	bristleX	Data on bristle number for Drosophila X chromosome
	fake.4way	Simulated data for a 4-way cross
	fake.bc	Simulated data for a backcross
	fake.f2	Simulated data for an F_2 intercross
	hyper	Backcross data on salt-induced hypertension
	listeria	Intercross data on Listeria monocytogenes susceptibility
	map10	A genetic map modeled after the mouse genome (10 cM spacing)
Input/output	read.cross	Read data for a QTL experiment
	write.cross	Write data for a QTL experiment to a file
Simulation	sim.cross	Simulate a QTL experiment
	sim.map	Generate a genetic map
Summaries	geno.table	Create table of genotype distributions
	plot.cross	Plot various features of a cross object
	plot.missing	Plot grid of missing genotypes
	plot.pheno	Histogram or bar plot of a phenotype
	plot.info	Plot the proportion of missing genotype data
	summary.cross	Print summary of QTL experiment
	summary.map	Print summary of a genetic map
	nchr, nind, nmar, nphe, totmar, nmissing	
Data manipulation	clean.cross	Remove intermediate calculations from a cross
	drop.markers	Remove a list of markers
	drop.nullmarkers	Remove markers without data
	fill.geno	Fill in holes in genotype data by imputation or Viterbi
	pull.map	Pull out the genetic map from a cross
	replace.map	Replace the genetic map of a cross
	subset.cross	Select a subset of chromosomes and/or individuals from a cross
	switch.order	Switch the order of markers on a chromosome
	movemarker	Move a marker from one chromosome to another
HMM engine	argmax.geno	Reconstruct underlying genotypes by the Viterbi algorithm
	calc.genoprob	Calculate conditional genotype probabilities
	sim.geno	Simulate genotypes given observed marker data
QTL mapping	scanone	Genome scan with a single QTL model
	scantwo	Two-dimensional genome scan with a two-QTL model
	lodint	Calculate a LOD support interval
	bayesint	Calculate an approximate Bayes credible interval
	plot.scanone	Plot output for a one-dimensional genome scan
	plot.scantwo	Plot output for a two-dimensional genome scan
	summary.scanone	Print summary of scanone output
	summary.scantwo	Print summary of scantwo output
	effectplot	Plot phenotype means of genotype groups defined by 1 or 2 markers
	plot.pxg	Like effectplot, but as a dot plot of the phenotypes
Genetic mapping	est.map	Estimate genetic map
	est.rf	Estimate pairwise recombination fractions
	plot.map	Plot genetic map(s)
	plot.rf	Plot recombination fractions
	ripple	Assess marker order by permuting groups of adjacent markers
	summary.ripple	Print summary of ripple output
Genotyping errors	calc.errorlod	Calculate Lincoln & Lander (1992) error LOD scores
	top.errorlod	List genotypes with highest error LOD values
	plot.geno	Plot observed genotypes, flagging likely errors
Multiple QTL models	makeqtl	Make a qtl object for use by fitqtl
	fitqtl	Fit a multiple QTL model, using multiple imputation
	summary.fitqtl	Get summary of the result of fitqtl
	scanqtl	Perform a multi-dimensional genome scan, using multiple imputation

Preliminaries

Use of the R/qtl package requires considerable knowledge of the R language/environment. We hope that the examples presented here will be understandable with little prior knowledge of R, especially because we neglect to explain the syntax of R. Several books, as well as some free documents, are available to assist the user in learning R; see the R project website cited above. We assume here that the user is running either Windows or Mac OS X.

1. To start R, double-click its icon.

2. To exit, type:

   ```
   q()
   ```

   Click yes or no to save or discard your work.

3. R keeps all of your work in RAM. If R should crash, all will be lost, and you will have to start from the beginning. The function `save.image` can be used to save your work to disk as you go along, so that, should R crash, you won't have to start from scratch. You would type:

   ```
   save.image()
   ```

4. Load the R/qtl package:

   ```
   library(qtl)
   ```

5. View the objects in your workspace:

   ```
   ls()
   ```

6. The best way to get help on the functions and data sets in R (and in R/qtl) is via the html version of the help files. One way to get access to this is to type

   ```
   help.start()
   ```

   This should open a browser with the main help menu. If you then click on Packages → qtl, you can see all of the available functions and datasets in R/qtl. For example, look at the help file for the function `read.cross`.

   An alternative method to view this help file is to type one of the following:

   ```
   help(read.cross)
   ?read.cross
   ```

   The html version of the help files are somewhat easier to read, and allow use of hotlinks between different functions.

7. All of the code in this tutorial is available as a file from which you may copy and paste into R, if you prefer that to typing. Type the following within R to get access to the file:

   ```
   url.show("http://www.rqtl.org/rqtltour.R")
   ```

Data import

A difficult first step in the use of most data analysis software is the import of data. With R/qtl, one may import data in several different formats by use of the function `read.cross`. (Example data files are available at www.rqtl.org/sampledata.) The internal data structure used by R/qtl is rather complicated, and is described in the help file for `read.cross`. (Also see example 6, below.) We won't discuss data import any further here, except to say that the comma-delimited format (`"csv"`) is recommended. If you have trouble importing data, send an email to Karl Broman (`kbroman@jhsph.edu`), attaching examples of your data files. (Such data will be kept confidential.)

Example 1: Hypertension

As a first example, we consider data from an experiment on hypertension in the mouse (Sugiyama et al., Genomics 71:70-77, 2001), kindly provided by Bev Paigen and Gary Churchill.

1. First, get access to the data, see that it is in your workspace, and view its help file. These data are included with the R/qtl package, and so you can get access to the data with the function `data()`.

   ```
   data(hyper)
   ls()
   ?hyper
   ```

2. We will postpone discussion of the internal data structure used by R/qtl until later. For now we'll just say that the data `hyper` has "class" `"cross"`. The function `summary.cross` prints summary information on such data. We can call that function directly, or we may simply use `summary` and the data is sent to the appropriate function according to its class.

   ```
   summary(hyper)
   ```

   Several other utility functions are available for getting summary information on the data. Hopefully these are self-explanatory.

   ```
   nind(hyper)
   nphe(hyper)
   nchr(hyper)
   totmar(hyper)
   nmar(hyper)
   ```

3. Plot a summary of these data.

   ```
   plot(hyper)
   ```

   In the upper left, black pixels indicate missing genotype data. Note that one marker has no genotype data. In the upper right, the genetic map of the markers is shown. In the lower left, a histogram of the phenotype is shown.

   The Windows version of R has a slick method for recording graphs, so that one may page up and down through a series of plots. To initiate this, click (on the menu bar) History → Recording.

   We may plot the individual components of the above multi-plot figure as follows.

   ```
   plot.missing(hyper)
   plot.map(hyper)
   plot.pheno(hyper, pheno.col=1)
   ```

   We can plot the genetic map with marker names, but they can be rather difficult to read. The following code plots the map with marker names for chr 1, 4, 6, 7 and 15.

   ```
   plot.map(hyper, chr=c(1, 4, 6, 7, 15), show.marker.names=TRUE)
   ```

4. Note the odd pattern of missing data; we may make this missing data plot with the individuals ordered according to the value of their phenotype.

   ```
   plot.missing(hyper, reorder=TRUE)
   ```

   We see that, for most markers, only individuals with extreme phenotypes were genotyped. At many markers (in regions of interest), markers were typed only on recombinant individuals.

5. The function `drop.nullmarkers` may be used to remove markers that have no genotype data (such as the marker on chr 14). A call to `totmar` will show that there are now 173 markers (rather than 174, as there were initially).

   ```
   hyper <- drop.nullmarkers(hyper)
   totmar(hyper)
   ```

6. Estimate recombination fractions between all pairs of markers, and plot them. This also calculates LOD scores for the test of H_0: $r = 1/2$. The plot of the recombination fractions can be either with recombination fractions in the upper part and LOD scores below, or with just recombination fractions or just LOD scores. Note that red corresponds to a small recombination fraction or a big LOD score, while blue is the reverse. Gray indicates missing values.

   ```
   hyper <- est.rf(hyper)
   plot.rf(hyper)
   plot.rf(hyper, chr=c(1,4))
   ```

   There are some very strange patterns in the recombination fractions, but this is due to the fact that some markers were typed largely on recombinant individuals.

   For example, on chr 6, the tenth marker shows a high recombination fraction with all other markers on the chromosome, but a plot of the missing data shows that this marker was typed only on a selected number of individuals (largely those showing recombination events across the interval).

   ```
   plot.rf(hyper, chr=6)
   plot.missing(hyper, chr=6)
   ```

7. Re-estimate the genetic map (keeping the order of markers fixed), and plot the original map against the newly estimated one.

   ```
   newmap <- est.map(hyper, error.prob=0.01)
   plot.map(hyper, newmap)
   ```

   We see some map expansion, especially on chr 6, 13 and 18. It is questionable whether we should replace the map or not. Keep in mind that the previous map locations are based on a limited number of meioses. If one wished to replace the genetic map with the estimated one, it could be done as follows:

   ```
   hyper <- replace.map(hyper, newmap)
   ```

   This replaces the map in the `hyper` data with `newmap`.

8. We now turn to the identification of genotyping errors. In the following, we calculate the error LOD scores of Lincoln and Lander (1992). A LOD score is calculated for each individual at each marker; large scores indicate likely genotyping errors.

   ```
   hyper <- calc.errorlod(hyper, error.prob=0.01)
   ```

   This calculates the genotype error LOD scores and inserts them into the `hyper` object.

   The function `top.errorlod` gives a list of genotypes that may be in error. Error LOD scores < 4 can probably be ignored.

   ```
   top.errorlod(hyper)
   ```

   Note that the results will be different, depending on whether you used `replace.map` above. If you did, you will get an indication of potential errors on chr 16. If you didn't, you will get an indication of potential errors on chr 1, 11 and 17.

9. The function `plot.geno` may be used to inspect the observed genotypes for a chromosome, with likely genotyping errors flagged. Of course, it's difficult to look at too many individuals at once. Note that white = AA and black = AB (for a backcross).

   ```
   plot.geno(hyper, chr=16, ind=c(24:34, 71:81))
   ```

   We don't have any utilities for fixing any apparent errors; it would be best to go back to the raw data. (Of course, you should edit a copy of the file; never discard the primary data.)

10. The function `plot.info` plots a measure of the proportion of missing genotype information in the genotype data. The missing information is calculated in two ways: as entropy, or via the variance of the conditional genotypes, given the observed marker data. (See the help file, using `?plot.info`.)

    ```
    plot.info(hyper)
    plot.info(hyper, chr=c(1,4,15))
    plot.info(hyper, chr=c(1,4,15), method="entropy")
    plot.info(hyper, chr=c(1,4,15), method="variance")
    ```

11. We now, finally, get to QTL mapping.

    The core of R/qtl is a set of functions which make use of the hidden Markov model (HMM) technology to calculate QTL genotype probabilities, to simulate from the joint genotype distribution and to calculate the most likely sequence of underlying genotypes (all conditional on the observed marker data). This is done in a quite general way, with possible allowance for the presence of genotyping errors. Of course, for convenience we assume no crossover interference.

    The function `calc.genoprob` calculates QTL genotype probabilities, conditional on the available marker data. These are needed for most of the QTL mapping functions. The argument `step` indicates the step size (in cM) at which the probabilities are calculated, and determines the step size at which later LOD scores are calculated.

    ```
    hyper <- calc.genoprob(hyper, step=1, error.prob=0.01)
    ```

    We may now use the function `scanone` to perform a single-QTL genome scan with a normal model. We may use maximum likelihood via the EM algorithm (Lander and Botstein 1989) or use Haley-Knott regression (Haley and Knott 1992).

    ```
    out.em <- scanone(hyper)
    out.hk <- scanone(hyper, method="hk")
    ```

    We may also use the multiple imputation method of Sen and Churchill (2001). This requires that we first use `sim.geno` to simulate from the joint genotype distribution, given the observed marker data. Again, the argument `step` indicates the step size at which the imputations are performed and determines the step size at which LOD scores will be calculated.

The n.draws indicates the number of imputations to perform. Larger values give more precise results but require considerably more computer memory and computation time.

```
hyper <- sim.geno(hyper, step=2, n.draws=16, error.prob=0.01)
out.imp <- scanone(hyper, method="imp")
```

12. The output of scanone has class "scanone"; the function summary.scanone displays the maximum LOD score on each chromosome for which the LOD exceeds a specified threshold.

```
summary(out.em)
summary(out.em, threshold=3)
summary(out.hk, threshold=3)
summary(out.imp, threshold=3)
```

13. The function max.scanone returns just the highest peak from output of scanone.

```
max(out.em)
max(out.hk)
max(out.imp)
```

14. We may also plot the results. plot.scanone can plot up to three genome scans at once, provided that they conform appropriately. Alternatively, one may use the argument add.

```
plot(out.em, chr=c(1,4,15))
plot(out.em, out.hk, out.imp, chr=c(1,4,15))
plot(out.em, chr=c(1,4,15))
plot(out.hk, chr=c(1,4,15), col="blue", add=TRUE)
plot(out.imp, chr=c(1,4,15), col="red", add=TRUE)
```

15. The function scanone may also be used to perform a permutation test to get a genome-wide LOD significance threshold. For Haley-Knott regression, this can be quite fast.

```
operm.hk <- scanone(hyper, method="hk", n.perm=1000)
```

The permutation output has class "scanoneperm". The function summary.scanoneperm can be used to get significance thresholds.

```
summary(operm.hk, alpha=0.05)
```

In addition, if the permutations results are included in a call to summary.scanone, you can estimated genome-scan-adjusted p-values for inferred QTL, and can get a report of all chromosomes meeting a certain significance level, with the corresponding LOD threshold calculated automatically.

```
summary(out.hk, perms=operm.hk, alpha=0.05, pvalues=TRUE)
```

16. We should mention at this point that the function save.image may be used to save your workspace to disk. If R crashes, you will wish you had used this.

```
save.image()
```

17. The function scantwo performs a two-dimensional genome scan with a two-QTL model. For every pair of positions, it calculates a LOD score for the full model (two QTL plus interaction) and a LOD score for the additive model (two QTL but no interaction). This be quite time consuming, and so you may wish to do the calculations on a coarser grid.

```
hyper <- calc.genoprob(hyper, step=5, error.prob=0.01)
out2.hk <- scantwo(hyper, method="hk")
```

One can also use method="em" or method="imp", but they are even more time consuming.

18. The output of scantwo has class "scantwo"; there are functions for obtaining summaries and plots, of course.

The summary function considers each pair of chromosomes, and calculates the maximum LOD score for the full model (M_f) and the maximum LOD score for the additive model (M_a). These two models are allowed to be maximized at different positions. We futher calculate a LOD score for a test of epistasis, $M_i = M_f - M_a$, and two LOD scores that concern evidence for a second QTL: M_{fv1} is the LOD score comparing the full model to the best single-QTL model and M_{av1} is the LOD score comparing the additive model to the best single-QTL model.

In the summary, we must provide five thresholds, for M_f, M_{fv1}, M_i, M_a, and M_{av1}, respectively. Call these T_f, T_{fv1}, T_i, T_a, and T_{av1}. We then report those pairs of chromosomes for which at least one of the following holds:

- $M_f \geq T_f$ and ($M_{fv1} \geq T_{fv1}$ or $M_i \geq T_i$)
- $M_a \geq T_a$ and $M_{av1} \geq T_{av1}$

The thresholds can be obtained by a permutation test (see below), but this is extremely time-consuming. For a mouse backcross, we suggest the thresholds (6.0, 4.7, 4.4, 4.7, 2.6) for the full, conditional-interactive, interaction, additive, and conditional-additive LOD scores, respectively. For a mouse intercross, we suggest the thresholds (9.1, 7.1, 6.3, 6.3, 3.3) for the full, conditional-interactive, interaction, additive, and conditional-additive LOD scores, respectively. These were obtained by 10,000 simulations of crosses with 250 individuals, markers at a 10 cM spacing, and analysis by Haley-Knott regression.

```
summary(out2.hk, thresholds=c(6.0, 4.7, 4.4, 4.7, 2.6))
```

The appropriate decision rule is not yet completely clear. I am inclined to ignore M_i and to choose genome-wide thresholds for the other four based on a permutation, using a common significance level for all four. M_i would be ignored if we gave it a very large threshold, as follows.

```
summary(out2.hk, thresholds=c(6.0, 4.7, Inf, 4.7, 2.6))
```

19. Plots of `scantwo` results are created via `plot.scantwo`.

```
plot(out2.hk)
plot(out2.hk, chr=c(1,4,6,15))
```

By default, the upper-left triangle contains epistasis LOD scores and the lower-right triangle contains the LOD scores for the full model. The color scale on the right indicates separate scales for the epistasis and joint LOD scores (on the left and right, respectively).

20. The function `max.scantwo` returns the two-locus positions with the maximum LOD score for the full and additive models.

```
max(out2.hk)
```

21. One may also use `scantwo` to perform permutation tests in order to obtain genome-wide LOD significance thresholds. These can be extremely time consuming, though with the Haley-Knott regression and multiple imputation methods, there is a trick that may be used in some cases to dramatically speed things up. So we'll try 100 permutations by the Haley-Knott regression method and hope that your computer is sufficiently fast.

```
operm2.hk <- scantwo(hyper, method="hk", n.perm=100)
```

We can again use `summary` to get LOD thresholds.

```
summary(operm2.hk)
```

And again these may be used in the summary of the `scantwo` output to calculate thresholds and p-values. If you want to ignore the LOD score for the interaction in the rule about what chromosome pairs to report, give $\alpha = 0$, corresponding to a threshold $T = \infty$.

```
summary(out2.hk, perms=operm2.hk, pvalues=TRUE,
        alphas=c(0.05, 0.05, 0, 0.05, 0.05))
```

You can't really trust these results. Haley-Knott regression performs poorly in the case of selective genotyping (as with the `hyper` data). Standard interval mapping or imputation would be better, but Haley-Knott regression has the advantage of speed, which is the reason we use it here.

22. Finally, we consider the fit of multiple-QTL models. Currently, only the use of multiple imputation has been implemented. We first create a QTL object using the function `makeqtl`, with five QTL at specified, fixed positions.

```
chr <- c(1, 1, 4, 6, 15)
pos <- c(50, 76, 30, 70, 20)
qtl <- makeqtl(hyper, chr, pos)
```

Finally, we use the function `fitqtl` to fit a model with five QTL, and allowing the QTL on chr 6 and 15 to interact.

```
my.formula <- y ~ Q1 + Q2 + Q3 + Q4 + Q5 + Q4:Q5
out.fitqtl <- fitqtl(hyper$pheno[,1], qtl, formula=my.formula)
summary(out.fitqtl)
```

23. You may wish to clean up your workspace before we move on to the next example.

```
ls()
rm(list=ls())
```

Example 2: Genetic mapping

R/qtl includes some utilities for estimating genetics maps and checking marker orders. In this example, we describe the use of these utilities.

1. Get access to some sample data. This is simulated data with some errors in marker order.

   ```
   data(badorder)
   summary(badorder)
   plot(badorder)
   ```

2. Estimate recombination fractions between all pairs of markers, and plot them.

   ```
   badorder <- est.rf(badorder)
   plot.rf(badorder)
   ```

   It appears that markers on chr 2 and 3 have been switched.

   Also note that, if we look more closely at the recombination fractions for chr 1, there seem to be some errors in marker order.

   ```
   plot.rf(badorder, chr=1)
   ```

3. Re-estimate the genetic map.

   ```
   newmap <- est.map(badorder, verbose=TRUE)
   plot.map(badorder, newmap)
   ```

   This really shows the problems on chr 2 and 3.

4. Fix the problems on chr 2 and 3. First, we look more closely at the recombination fractions for these chromosoems

   ```
   plot.rf(badorder, chr=2:3)
   ```

   We need to move the sixth marker on chr 2 to chr 3, and the fifth marker on chr 3 to chr 2. We need to figure out which markers these are.

   ```
   pull.map(badorder, chr=2)
   pull.map(badorder, chr=3)
   ```

   Now we can use the function `movemarker` to move the markers. It seems like they should be exactly switched.

   ```
   badorder <- movemarker(badorder, "D2M937", 3, 48)
   badorder <- movemarker(badorder, "D3M160", 2, 28.8)
   ```

   Now look at the recombination fractions again.

   ```
   plot.rf(badorder, chr=2:3)
   ```

5. We can check the marker order on chr 1. The function `ripple` will consider all permutations of a sliding window of adjacent markers. A quick-and-dirty approach is to count the number of obligate crossovers for each possible order, to find the order with the minimum number of crossovers. A more refined, but also more computationally intensive, approach is to re-estimate the genetic map for each order, calculating LOD scores (log$_{10}$ likelihood ratios) relative to the initial order. (This may be done with allowance for the presence of genotyping errors.) The default approach is the quick-and-dirty method.

   The following checks the marker order on chr 1, permuting groups of six contiguous markers.

   ```
   rip1 <- ripple(badorder, chr=1, window=6)
   summary(rip1)
   ```

   In the summary output, markers 9–11 clearly need to be flipped. There also seems to be a problem with the order of markers 4–6.

6. The following performs the likelihood analysis, permuting groups of three adjacent markers, assuming a genotyping error rate of 1%. It's considerably slower, but more trustworthy.

   ```
   rip2 <- ripple(badorder, chr=1, window=3, err=0.01, method="likelihood")
   summary(rip2)
   ```

   Note that positive LOD scores indicate that the alternate order has a higher likelihood than the original.

7. We can switch the order of markers 9–11 with the function `switch.order` (which works only for a single chromosome) and then re-assess the order. Note that the second row of `rip1` corresponds to the improved order.

   ```
   badorder.rev <- switch.order(badorder, 1, rip1[2,])
   rip1r <- ripple(badorder.rev, chr=1, window=6)
   summary(rip1r)
   ```

   It looks like the marker pairs (5,6) and (1,2) should each be inverted. We use `switch.order` again, and then check marker order using the likelihood method.

   ```
   badorder.rev <- switch.order(badorder.rev, 1, rip1r[2,])
   rip2r <- ripple(badorder.rev, chr=1, window=3, err=0.01)
   summary(rip2r)
   ```

   It's probably best to start out using the quick-and-dirty method, with a large window size, to find the marker order with the minimum number of obligate crossovers, and then refine that order using the slower, but more trustworthy, likelihood method.

8. We can look again at the recombination fractions for this chromosome.

   ```
   badorder.rev <- est.rf(badorder.rev)
   plot.rf(badorder.rev, 1)
   ```

Example 3: Listeria susceptibility

In order to demonstrate further uses of the function `scanone`, we consider some data on susceptibility to *Listeria monocytogenes* in mice (Boyartchuk et al., Nature Genetics 27:259-260, 2001). These data were kindly provided by Victor Boyartchuk and Bill Dietrich.

1. Get access to the data and view some summaries.

   ```
   data(listeria)
   summary(listeria)
   plot(listeria)
   plot.missing(listeria)
   ```

   Note that in the missing data plot, gray pixels are partially missing genotypes (e.g., a genotype may be known to be either AA or AB, but not which).

   The phenotype here is the survival time of a mouse (in hours) following infection with *Listeria monocytogenes*. Individuals with a survival time of 264 hours are those that recovered from the infection.

2. We'll use the log survival time, rather than survival time, so we first need to create a new phenotype, which will end up as the third phenotype (after `sex`).

   ```
   listeria$pheno$logSurv <- log(listeria$pheno[,1])
   plot(listeria)
   ```

3. Estimate pairwise recombination fractions.

   ```
   listeria <- est.rf(listeria)
   plot.rf(listeria)
   plot.rf(listeria, chr=c(5,13))
   ```

4. Re-estimate the genetic map.

   ```
   newmap <- est.map(listeria, error.prob=0.01)
   plot.map(listeria, newmap)
   listeria <- replace.map(listeria, newmap)
   ```

5. Investigate genotyping errors; nothing gets flagged with a cutoff of 4, but one genotype is indicated with error LOD ∼3.6.

   ```
   listeria <- calc.errorlod(listeria, error.prob=0.01)
   top.errorlod(listeria)
   top.errorlod(listeria, cutoff=3.5)
   plot.geno(listeria, chr=13, ind=61:70, cutoff=3.5)
   ```

   Note that in the plot given by `plot.geno`, for an intercross, white = AA, gray = AB, black = BB, green = AA or AB, and orange = AB or BB.

6. Now on to the QTL mapping. Recall that the phenotype distribution shows a clear departure from the standard assumptions for interval mapping; 30% of the mice survived longer than 264 hours, and were considered recovered from the infection.

   One approach for these data is to use the two-part model considered by Boyartchuk et al. (2001). In this model, a mouse with genotype g has probability p_g of surviving the infection. If it does die, its log survival time is assumed to be distributed normal(μ_g,σ^2). Analysis proceeds by maximum likelihood via an EM algorithm. Three LOD scores are calculated. LOD(p, μ) is for the test of the null hypothesis $p_g \equiv p$ and $\mu_g \equiv \mu$. LOD(p) is for the test of the hypothesis $p_g \equiv p$ but the μ are allowed to vary. LOD(μ) is for the test of the hypothesis $\mu_g \equiv \mu$ but the p are allowed to vary.

   The function `scanone` will fit the above model when the argument `model="2part"`. One must also specify the argument `upper`, which indicates whether the spike in the phenotype is the maximum phenotype (as it is with this phenotype; take `upper=TRUE`) or the minimum phenotype (take `upper=FALSE`). For this model, only the EM algorithm has been implemented so far.

   ```
   listeria <- calc.genoprob(listeria, step=2)
   out.2p <- scanone(listeria, pheno.col=3, model="2part", upper=TRUE)
   ```

   Note that, because this model has three extra parameters, the appropriate LOD threshold is higher—around 4.5 rather than 3.5. The three different LOD curves are in columns 3–5 of the output. We can use the `lodcolumn` argument to `plot.scanone` to plot these other LOD scores.

   ```
   summary(out.2p)
   summary(out.2p, threshold=4.5)
   ```

   Alternatively, we may use `format="allpeaks"`, in which case it displays the maximum LOD score or each column, with the position at which each was maximized. You may provide either one threshold, which would be applied to all LOD score columns, or a separate threshold for each column.

   ```
   summary(out.2p, format="allpeaks", threshold=3)
   summary(out.2p, format="allpeaks", threshold=c(4.5,3,3))
   ```

7. By default, `plot.scanone` will plot the first LOD score column. Alternatively, we may indicate another column to plot with the `lodcolumn` argument. Or we can plot up to three LOD scores at once by giving a vector.

   ```
   plot(out.2p)
   plot(out.2p, lodcolumn=2)
   plot(out.2p, lodcolumn=1:3, chr=c(1,5,13,15))
   ```

   Note that the locus on chr 1 shows effect mostly on the mean time-to-death, conditional on death; the locus on chr 5 shows effect mostly on the probability of survival; and the loci on chr 13 and 15 shows some effect on each.

8. Permutation tests may be performed as before. The output will have three columns, corresponding to the three LOD scores.

   ```
   operm.2p <- scanone(listeria, model="2part", pheno.col=3,
                       upper=TRUE, n.perm=25)
   summary(operm.2p, alpha=0.05)
   ```

   We may again use the permutation results in `summary.scanone` to have thresholds calculated automatically and to obtain genome-scan-adjusted p-values, but of course we would want to have performed more than 25 permutations.

   ```
   summary(out.2p, format="allpeaks", perms=operm.2p,
           alpha=0.05, pvalues=TRUE)
   ```

9. Alternatively, one may perform separate analyses of the log survival time, conditional on death, and the binary phenotype survival/death. First we set up these phenotypes.

   ```
   y <- listeria$pheno$logSurv
   my <- max(y, na.rm=TRUE)
   z <- as.numeric(y==my)
   y[y==my] <- NA
   listeria$pheno$logSurv2 <- y
   listeria$pheno$binary <- z
   plot(listeria)
   ```

   We use standard interval mapping for the log survival time conditional on death; the results are slightly different from LOD(μ).

```
out.mu <- scanone(listeria, pheno.col=4)
plot(out.mu, out.2p, lodcolumn=c(1,3), chr=c(1,5,13,15), col=c("blue","red"))
```

We can use scanone with model="binary" to analyze the binary phenotype. Again, the results are only slight different from LOD(p).

```
out.p <- scanone(listeria, pheno.col=5, model="binary")
plot(out.p, out.2p, lodcolumn=c(1,2), chr=c(1,5,13,15), col=c("blue","red"))
```

10. A further approach is to use a non-parametric form of interval mapping. R/qtl uses an extension of the Kruskal-Wallis test statistic. Use scanone with model="np". In this case, the argument method is ignored; the analysis method is much like Haley-Knott regression. If the argument ties.random=TRUE, tied phenotypes are ranked at random. If ties.random=FALSE, tied phenotypes are given the average rank and a correction is applied to the LOD score.

```
out.np1 <- scanone(listeria, model="np", ties.random=TRUE)
out.np2 <- scanone(listeria, model="np", ties.random=FALSE)
```

```
plot(out.np1, out.np2, col=c("blue","red"))
plot(out.2p, out.np1, out.np2, chr=c(1,5,13,15))
```

Note that the significance threshold for the non-parametric genome scan will be quite a bit smaller than that for the two-part model. The two approaches for dealing with ties give basically the same results. Randomizing ties for the non-parametric approach can give quite variable results in the case of a great number of ties, and so we would recommend the use of ties.random=FALSE in this case.

Example 4: Covariates in QTL mapping

As a further example, we illustrate the use of covariates in QTL mapping. We consider some simulated backcross data.

1. Get access to the data.

```
data(fake.bc)
summary(fake.bc)
plot(fake.bc)
```

2. Perform genome scans for the two phenotypes without covariates.

```
fake.bc <- calc.genoprob(fake.bc, step=2.5)
out.nocovar <- scanone(fake.bc, pheno.col=1:2)
```

3. Perform genome scans with sex as an additive covariate. Note that the covariates must be numeric. Factors may have to be converted.

```
sex <- fake.bc$pheno$sex
out.acovar <- scanone(fake.bc, pheno.col=1:2, addcovar=sex)
```

Here, the average phenotype is allowed to be different in the two sexes, but the effect of the putative QTL is assumed to be the same in the two sexes.

4. Note that the use of sex as an additive covariate resulted in an increase in the LOD scores for phenotype 1, but resulted in a decreased LOD score at the chr 5 locus for phenotype 2.

```
summary(out.nocovar, threshold=3, format="allpeaks")
summary(out.acovar, threshold=3, format="allpeaks")
```

```
plot(out.nocovar, out.acovar, chr=c(2, 5))
plot(out.nocovar, out.acovar, chr=c(2, 5), lodcolumn=2)
```

5. Let us now perform genome scans with sex as an interactive covariate, so that the QTL is allowed to be different in the two sexes.

```
out.icovar <- scanone(fake.bc, pheno.col=1:2, addcovar=sex, intcovar=sex)
```

6. The LOD score in the output is for the comparison of the full model with terms for sex, QTL and QTL×sex interaction to the reduced model with just the sex term. Thus, the degrees of freedom associated with the LOD score is 2 rather than 1, and so larger LOD scores will generally be obtained.

```
summary(out.icovar, threshold=3, format="allpeaks")
```

```
plot(out.acovar, out.icovar, chr=c(2,5), col=c("blue", "red"))
plot(out.acovar, out.icovar, chr=c(2,5), lodcolumn=2,
     col=c("blue", "red"))
```

7. The difference between the LOD score with sex as an interactive covariate and the LOD score with sex as an additive covariate concerns the test of the QTL × sex interaction: does the QTL have the same effect in both sexes? The differences, and a plot of the differences, may be obtained as follows.

   ```
   out.sexint <- out.icovar - out.acovar
   plot(out.sexint, lodcolumn=1:2, chr=c(2,5), col=c("green", "purple"))
   ```

   The green and purple curves are for the first and second phenotypes, respectively.

8. To test for the QTL × sex interaction, we may perform a permutation test. This is not perfect, as the permutation test eliminates the effect of the QTL, and so we must assume that the distribution of the LOD score for the QTL × sex interaction is the same in the presence of a QTL as under the global null hypothesis of no QTL effect.

   The permutation test requires some care. We must perform separate permutations with sex as an additive covariate and with sex as an interactive covariate, but we must ensure, by setting the "seed" for the random number generator, that they use matched permutations of the data.

   For the sake of speed, we will use Haley-Knott regression, even though the results above were obtained by standard interval mapping. Also, we will perform just 100 permutations, though 1000 would be preferred.

   ```
   seed <- ceiling(runif(1, 0, 10^8))
   set.seed(seed)
   operm.acovar <- scanone(fake.bc, pheno.col=1:2, addcovar=sex,
                           method="hk", n.perm=100)
   set.seed(seed)
   operm.icovar <- scanone(fake.bc, pheno.col=1:2, addcovar=sex,
                           intcovar=sex, method="hk", n.perm=100)
   ```

   Again, the differences concern the QTL × sex interaction.

   ```
   operm.sexint <- operm.icovar - operm.acovar
   ```

   We can use `summary` to get the genome-wide LOD thresholds.

   ```
   summary(operm.sexint, alpha=c(0.05, 0.20))
   ```

   We can also use these results to look at evidence for QTL × sex interaction in our initial scans.

   ```
   summary(out.sexint, perms=operm.sexint, alpha=0.1,
           format="allpeaks", pvalues=TRUE)
   ```

Example 5: Multiple QTL mapping

We return to the `hyper` data to illustrate some of the more advanced methods for exploring multiple QTL models. Note that the multiple QTL mapping features are currently implemented only for the multiple imputation method, and some aspects remain quite cumbersome. Also, we will rely here on functions that are not yet available in the released version of R/qtl. These functions are available at www.rqtl.org/multqtlfunc.R.

The multiple-QTL aspects of R/qtl are under active development (as they should be!), and so the methods used below will hopefully be improved in the near future. Our aim here is to give a flavor of what is possible.

1. First, let us delete everything in our workspace and then re-load the `hyper` data.

   ```
   rm(list=ls())
   data(hyper)
   ```

2. Now let's load the additional, developmental functions for multiple QTL mapping.

   ```
   source("http://www.rqtl.org/multqtlfunc.R")
   ```

3. We will be using the multiple imputation method throughout this example, and so we first need to perform the imputations. Recall that more imputations give more precise results, but take more time and memory. To speed things along, we will use only 32 imputations, even though much more would be needed for a definitive analysis.

   ```
   hyper <- sim.geno(hyper, step=2.5, n.draw=32, err=0.01)
   ```

4. We first perform a single-QTL genome scan and inspect the results.

   ```
   out1 <- scanone(hyper, method="imp")
   plot(out1)
   ```

   As you'll recall from the results in Example 1, we have clear evidence for a QTL on chr 4, and strong evidence for a QTL on chr 1. The LOD curve on chr 1 has an interesting double peak, suggestive of possibly two QTL.

   There is a hint of further loci on chr 6 and 15 and elsewhere.

5. In the presence of a large-effect QTL, as seen on chr 4, one may wish to repeat the scan, controlling for that locus. This can make the loci with more modest effect more apparent.

   A simple (but rough) approach is to pull out the genotypes for a marker near the peak locus, and use that marker as an additive covariate in a single-QTL scan. The peak marker for these data was D4Mit164:

   ```
   max(out1)
   ```

   If the peak LOD score is not at a marker, we may use `find.marker` to identify the marker closest to the LOD peak.

   ```
   find.marker(hyper, 4, 29.5)
   ```

6. The function `pull.geno` may be used to pull out the genotype data for that marker, but we'll see that most individuals were not typed at D4Mit164.

   ```
   g <- pull.geno(hyper)[,"D4Mit164"]
   mean(is.na(g))
   ```

   We may fill in the genotype data using a single imputation, and then use those imputed genotypes as if they were observed. This is not ideal; we'll do this analysis properly (though with more complex code) below.

   ```
   g <- pull.geno(fill.geno(hyper))[,"D4Mit164"]
   ```

7. Now we perform the genome scan, controlling for the chr 4 locus. (Note that in an intercross, we would have to re-code the genotype data to be a two-column numeric matrix.)

   ```
   out1.c4 <- scanone(hyper, method="imp", addcovar=g)
   ```

   We can plot the results together with the original genome scan.

   ```
   plot(out1, out1.c4, col=c("blue", "red"))
   ```

   The LOD curve on chr 1 went up quite a bit. (And, of course, the LOD curve on chr 4 went down to near 0.) To see the effect of controlling for the chr 4 locus more clearly, we can plot the differences between the LOD scores.

   ```
   plot(out1.c4 - out1, ylim=c(-3,3))
   abline(h=0, lty=2, col="gray")
   ```

8. We may also look for loci that interact with the chr 4 locus, by including marker D4Mit164 as an interactive covariate.

   ```
   out1.c4i <- scanone(hyper, method="imp", addcovar=g, intcovar=g)
   ```

   The difference between these LOD scores and those obtained with D4Mit164 as a strictly additive covariate indicates evidence for an interaction with the chr 4 locus.

   ```
   plot(out1.c4i - out1.c4)
   ```

   There is nothing particularly interesting here.

9. Now let us perform a 2d scan. This will take a few minutes, as we're doing the scan at a 2.5 cM step size.

   ```
   out2 <- scantwo(hyper, method="imp")
   ```

10. Let us look at some summaries for the `scantwo` results. Recall that we need to provide five thresholds (see Example 1). We'll ignore the threshold on the epistasis LOD score, T_i, and use the thresholds suggested above.

    ```
    summary(out2, thr=c(6.0, 4.7, Inf, 4.7, 2.6))
    ```

    Your results may be different from mine, since we are using so few imputations, but I see evidence for loci on chr 1 and 4 (which don't appear to interact) and loci on chr 6 and 15 (which do show evidence of epistasis).

    This didn't pick up evidence for two QTL on chr 1; we can look directly at the chr 1 results as follows.

    ```
    summary( subset(out2, chr=1) )
    ```

    The LOD score for a second, additive QTL on chr 2 (LOD$_{av1}$) is $\sim$1.6; not strong, but not uninteresting.

Evidence for an interaction between loci on chr 7 and 15 had been previously reported. Those results may be inspected as follows.

```
summary( subset(out2, chr=c(7,15)) )
```

Again, this is interesting but not strong.

11. Let us look at some plots of the `scantwo` results. First we make the standard plot with selected chromosomes; the upper triangle contains LOD$_i$ and the lower triangle contains LOD$_f$.

    ```
    plot(out2, chr=c(1,4,6,7,15))
    ```

    The arguments `lower` and `upper` may be used to change what is plotted in the upper and lower triangles. For example, with `lower="cond-int"` , LOD$_{fv1}$ (evidence for a second QTL, allowing for epistasis) is displayed in the lower triangle, while with `lower="cond-add"`, LOD$_{av1}$ (evidence for a second QTL, assuming no epistasis) is displayed.

    ```
    plot(out2, chr=1, lower="cond-add")
    plot(out2, chr=c(6,15), lower="cond-int")
    plot(out2, chr=c(7,15), lower="cond-int")
    ```

    Again, evidence for a second QTL on chr 1 is not strong. Evidence for interacting QTL on chr 6 and 15 is quite strong; the 7×15 interaction is not.

12. We can also perform the 2d scan conditional on the chr 4 locus. We'll do this just for chr 1, 6, 7, and 15, to save time.

    ```
    out2.c4 <- scantwo(hyper, method="imp", addcovar=g, chr=c(1,6,7,15))
    ```

    If we look at the same summaries as before, we see decreased evidence for a second QTL on chr 1 and for the 7×15 interaction, but increased evidence for the 6×15 interaction.

    ```
    summary(out2.c4, thr=c(6.0, 4.7, Inf, 4.7, 2.6))
    summary( subset(out2.c4, chr=1) )
    summary( subset(out2.c4, chr=c(7,15)) )
    ```

    The sort of plots we made before remain interesting.

    ```
    plot(out2.c4, chr=c(1,4,6,7,15))
    plot(out2.c4, chr=1, lower="cond-int")
    plot(out2.c4, chr=c(6,15), lower="cond-int")
    plot(out2.c4, chr=c(7,15), lower="cond-int")
    ```

    We can also look at the differences in the LOD scores, to see how much conditioning on D4Mit164 has affected the results. We need to subset our original results, since we only scanned selected chromosomes in the conditional analysis. The `allow.neg` argument is used to allow negative LOD scores in the `scantwo` plot, as they would generally be replaced with 0.

    ```
    out2sub <- subset(out2, chr=c(1,6,7,15))
    plot(out2.c4 - out2sub, allow.neg=TRUE, lower="cond-int")
    ```

13. Now let us turn to the fit of multiple-QTL models. The function `fitqtl` is used to fit a specific model.

    One must first pull out the data on fixed QTL locations using `makeqtl`. We will consider the possibility of two QTL on chr 1, but will ignore the putative QTL on chr 7. Also note that `fitqtl` takes a vector of phenotypes as input, and so we pull that from the `hyper` data to make things simpler.

    ```
    qc <- c(1, 1, 4, 6, 15)
    qp <- c(43.3, 78.3, 30.0, 62.5, 18.0)
    qtl <- makeqtl(hyper, chr=qc, pos=qp)
    phe <- hyper$pheno[,1]
    ```

    We also create a "formula" which indicates which QTL are to be included in the fit and which interact.

    ```
    myformula <- y ~ Q1+Q2+Q3+Q4+Q5 + Q4:Q5
    ```

    We can now fit a model, including the 6×15 interaction, and get a summary of the results.

    ```
    out.fq <- fitqtl(phe, qtl, formula = myformula)
    summary(out.fq)
    ```

    The first part of the summary describes the overall fit; the LOD score of $\sim$23 is the log$_{10}$ likelihood ratio comparing the full model to the null model.

The second part of the summary gives results dropping one term at a time from the model. In the presence of an interaction, if a term included in the interaction is omitted, the interaction is also omitted, and so the rows for the loci on chr 6 and 15 indicate 2 degrees of freedom.

14. One may also use `fitqtl` to get estimated effects of the QTL in the context of the multiple-QTL model. We can use `drop=FALSE`, so that the "drop one at a time" part of the analysis is not performed, and `get.ests=TRUE` to get the estimated effects.

    ```
    out.fq <- fitqtl(phe, qtl, formula = myformula, drop=FALSE, get.ests=TRUE)
    summary(out.fq)
    ```

    The estimated effects are the differences between the heterozygote and homozygote groups. The interaction effect is of the difference between the differences.

15. The function `refineqtl` (developmental code in the `"multqtlfunc.R"` file that we loaded earlier) can be used to refine the estimated positions of the QTL in the context of the multiple-QTL model.

    ```
    out.rq <- refineqtl(hyper, chr=qc, pos=qp, formula = myformula)
    ```

    The output has two columns: the chromosome IDs and new positions of the QTL. For me, a couple of the QTL moved, but very slightly:

    ```
    qp - out.rq[,2]
    ```

    We can re-run `makeqtl` and `fitqtl` to get a fit with the new positions; the overall LOD score should have increased slightly. (For me, it increased from 23.0 to 23.7.)

    ```
    qp2 <- out.rq[,2]
    qtl2 <- makeqtl(hyper, chr=qc, pos=qp2)
    out.fq2 <- fitqtl(phe, qtl2, formula=myformula)
    summary(out.fq2)
    ```

16. The `scanqtl` function is used to perform general genome scans in the context of a multiple QTL model. It is quite flexible, but not simple to use.

    We will first use `scanqtl` to perform a more precise version of our genome scan, conditional on the chr 4 locus. Previously, we had conditioned on imputed genotypes at a marker near the LOD peak on chr 4. With `scanqtl` we can do this properly: take proper account of the missing genotype information at the chr 4 locus, rather than taking genotypes from a single imputation as if they had been observed.

    Like `makeqtl`, the `scanqtl` function takes the chromosome and positions of a set of QTL, as well as a formula indicating which QTL interact. If the formula is omitted, all loci are assumed to be additive. The QTL positions may be a single number (in which case the QTL location is fixed) or an interval (in which case a scan over that region is performed.

    And so, the following performs a scan on all of chr 1 (indicated by `(-Inf,Inf)`) with a QTL on chr 4 fixed at 29.5 cM.

    ```
    out1.sq <- scanqtl(hyper, chr=c(1,4), pos = list( c(-Inf,Inf), 29.5) )
    ```

    The output contains LOD scores comparing the two-QTL model to the null model. If we want the LOD score comparing the two-QTL model to the model with just the chr 4 locus, we need to subtract off the LOD score for the latter, single-QTL model.

    The output of `scanqtl` is not simple to work with (yet), but the `"multqtlfunc.R"` file we loaded earlier contains a function `convert.scanqtl` that will convert the output to an object of the form produced by `scanone` or `scantwo`.

    And so, we first calculate the LOD score for the model with a single QTL on chr 4, and then use the function `convert.scanqtl` to convert the `scanqtl` output to a more useable form.

    ```
    null <- scanqtl(hyper, chr=4, pos=list(29.5))
    out1.c4r <- convert.scanqtl(out1.sq, null)
    ```

    We may now plot these results with those obtained earlier. The results are not actually too different.

    ```
    plot(out1.c4, out1.c4r, col=c("blue", "red"), chr=1)
    ```

17. The same approach may be used to perform a 2d scan on chr 1, conditioning on the locus on chr 4. We need to use `scanqtl` twice, once with an additive model and once with the full model (two QTL plus interaction).

    ```
    out2.sq.add <- scanqtl(hyper, chr=c(1,1,4),
                           pos=list(c(-Inf,Inf), c(-Inf,Inf), 29.5))
    out2.sq.full <- scanqtl(hyper, chr=c(1,1,4),
    ```

```
                        pos=list(c(-Inf,Inf), c(-Inf,Inf), 29.5),
                        formula=y~Q1+Q2+Q3+Q1:Q2)
```

We again use `convert.scanqtl` to convert the output to a more useable form.

```
out2.c4r <- convert.scanqtl(out2.sq.full, null, out2.sq.add)
```

We can plot the difference between these results and our previous results; we first need to subset the old results, since here we have just looked at chr 1.

```
out2.c4sub <- subset(out2.c4, chr=1)
plot(out2.c4sub - out2.c4r, lower="cond-add", allow.neg=TRUE)
```

Again, things have hardly changed.

18. Finally, let us use `scanqtl` to scan for additional loci. Let us take the five-QTL model (with the loci on 6 and 15 interacting) as fixed, and look to add a further locus. Here `out.fq2` is taken as the null model, and we must scan each chromosome, one at a time, for a further locus. We'll skip the X chromosome.

    The syntax of the QTL positions is perhaps most tricky. Well, without much knowledge of R, this is all likely mysterious.

    ```
    newpos <- c( as.list(qp2), list(c(-Inf, Inf)) )
    out.sq <- NULL
    for(i in 1:19) {
      temp <- scanqtl(hyper, chr=c(qc,i), pos=newpos,
                      formula = y ~ Q1+Q2+Q3+Q4+Q5 + Q4:Q5 + Q6)
      out.sq <- rbind(out.sq, convert.scanqtl(temp, out.fq2))
    }
    ```

    The result, `out.sq`, is just like the output from `scanone`, and so we may plot it as follows:

    ```
    plot(out.sq)
    ```

19. We may use the same approach to look for additional loci that might interact with the locus on chr 15. The code is the same, but we add the additional interaction to the formula.

    ```
    out.sqi <- NULL
    for(i in 1:19) {
      temp <- scanqtl(hyper, chr=c(qc,i), pos=newpos,
                      formula = y ~ Q1+Q2+Q3+Q4+Q5 + Q4:Q5 + Q6 + Q5:Q6)
      out.sqi <- rbind(out.sqi, convert.scanqtl(temp, out.fq2))
    }
    ```

    We can plot the results (which indicate evidence for an additional QTL, allowing for epistasis), or the differences between these and our previous ones, which concern just the interaction.

    ```
    plot(out.sqi)
    plot(out.sqi - out.sq)
    ```

    The possible 7×15 interaction is by far the most interesting thing going on here.

Example 6: Internal data structure

Finally, let us briefly describe the rather complicated data structure that R/qtl uses for QTL mapping experiments. This will be rather dull, and will require a good deal of familiarity with the R (or S) language. The choice of data structure required some balance between ease of programming and simplicity for the user interface. The syntax for references to certain pieces of the internal data can become extremely complicated.

1. Get access to some sample data.

   ```
   data(fake.bc)
   ```

2. First, the object has a "class," which indicates that it corresponds to data for an experimental cross, and gives the cross type. By having class `cross`, the functions `plot` and `summary` know to send the data to `plot.cross` and `summary.cross`.

   ```
   class(fake.bc)
   ```

3. Every `cross` object has two components, one containing the genotype data and genetic maps and the other containing the phenotype data.

   ```
   names(fake.bc)
   ```

4. The phenotype data is simply a matrix (more strictly a data.frame) with rows corresponding to individuals and columns corresponding to phenotypes.

   ```
   fake.bc$pheno[1:10,]
   ```

5. The genotype data is a list with components corresponding to chromosomes. Each chromosome has a name and a class. The class for a chromosome is either `"A"` or `"X"`, according to whether it is an autosome or the X chromosome.

   ```
   names(fake.bc$geno)
   sapply(fake.bc$geno, class)
   ```

6. Each component of `geno` contains two components, `data` (containing the marker genotype data) and `map` (containing the positions of the markers, in cM).

   ```
   names(fake.bc$geno[[3]])
   fake.bc$geno[[3]]$data[1:5,]
   fake.bc$geno[[3]]$map
   ```

   That's it for the raw data.

7. When one runs `calc.genoprob`, `sim.geno`, `argmax.geno` or `calc.errorlod`, the output is the input cross object with the derived data attached to each component (the chromosomes) of the `geno` component.

   ```
   names(fake.bc$geno[[3]])
   fake.bc <- calc.genoprob(fake.bc, step=10, err=0.01)
   names(fake.bc$geno[[3]])
   fake.bc <- sim.geno(fake.bc, step=10, n.draws=8, err=0.01)
   names(fake.bc$geno[[3]])
   fake.bc <- argmax.geno(fake.bc, step=10, err=0.01)
   names(fake.bc$geno[[3]])
   fake.bc <- calc.errorlod(fake.bc, err=0.01)
   names(fake.bc$geno[[3]])
   ```

8. Finally, when one runs `est.rf`, a matrix containing the pairwise recombination fractions and LOD scores is added to the cross object.

   ```
   names(fake.bc)
   fake.bc <- est.rf(fake.bc)
   names(fake.bc)
   ```

A brief tour of R/qtlbim

Brian S. Yandell
Departments of Statistics and Horticulture, UW–Madison
www.qtlbim.org

June 7, 2007

Abstract

Bayesian interval mapping of QTL library R/qtlbim provides Bayesian analysis of multiple quantitative trait loci (QTL) models. This includes posterior estimates of the number and location of QTL, and of their main and epistatic effects. This tutorial assumes the reader has read *A brief tour of R/qtl* by Karl Broman, available at www.rqtl.org. We extend his hypertension example by analyzing the same data with Bayesian methods. Some familiarity with Bayesian methods is helpful but not required.

1 Overview of R/qtlbim

R/qtlbim is an extensible, interactive environment for mapping quantitative trait loci (QTL) in experimental crosses using Bayesian methods. It builds on R/qtl (www.rqtl.org), which in turn builds on the widely used statistical language system R (www.r-project.org). R/qtlbim is distributed in the same manner as R/qtl, and can be installed similarly.

This tutorial describes the MCMC sampling routines and some of the plotting facilities available through the `R/qtlbim` package. The purpose of these plots is to provide graphical tools for

1. exploring putative single and multiple QTL,
2. producing interpretable graphics of the relative evidence in favor of a set of putative QTL,
3. visual diagnostics of the MCMC model selection algorithm.

The package provides graphical diagnostics that can help investigate several "better" models. It also provides a 1-D and 2-D genome scan. The `R/qtlbim` package provides plotting facilities for results generated by the analytical tools in the `R/qtlbim` package. These plotting facilities include time series plots of QTL model charactacteristics as basic MCMC diagnostic plots, visual tools for comparison of putative QTL models and exploratory plots whose purpose is the aid in the identification of likely QTL.

This package is currently in "beta" release. That is, most of the basic features are stable, but we expect a learning curve. We would like feedback from experienced QTL mappers and R users especially. Please note that the command `qb.mcmc` that creates the MCMC samples produces external files in an output directory. These files are tens of Mb large. They are integral to `R/qtlbim` diagnostics. The proper way to remove a `qb` object created by `qb.mcmc` is to use the `qb.remove` routine, as indicated below.

This document walks through the `R/qtlbim` package by demonstrating the following major functions: creation of Bayesian samples from the posterior using MCMC sampling; use of plot and summary tools to examine genetic architecture; data management in R/qtlbim.

2 Citation of R/qtlbim

To cite R/qtl in publications, use the following:

> Yandell BS, Mehta T, Banerjee S, Shriner D, Venkataraman R, Moon JY, Neely WW, Wu H, von Smith R, Yi N (2007) R/qtlbim: QTL with Bayesian interval mapping in experimental crosses. *Bioinformatics 23*: 641-643.

The methodology is described in the following paper:

> Yi N, Yandell BS, Churchill GA, Allison DB, Eisen EJ, Pomp D (2005) Bayesian model selection for genome-wide epistatic QTL analysis. *Genetics 170*: 1333–1344.

3 Preliminaries

The Preliminaries of Broman's brief tour, as well as steps 1, 16 and 23 of his Example 1, provide important information on careful use of R.

This tutorial focuses on the hyper dataset from R/qtl. Please complete the R/qtl Tutorial for Hypertension in *A brief tour of R/qtl* available at (www.rqtl.org). Steps 1-4, 11-14 and 17-20 of Example 1 provide an overview of the core analysis in R/qtl.

Some other steps and examples might be skipped in the interest of time. Steps 15 and 21 show how to estimate permutation thresholds, which can take considerable time on slower machines. Step 22 of Example 1 and Example 5 develop a strategy for multiple QTL mapping. Example 4 shows how to incorporate covariates into R/qtl analysis.

The other skipped steps of Example 1 (5-10) concern further investigation of the marker genotypes and map construction. In addition, Example 2 provides further detail on marker order. Example 6 shows the internal data construct for cross objects for those familiar with R who want to dig deeper.

All of the code for this tutorial is available in a file. You can view this as

```
> url.show("http://www.stat.wisc.edu/~yandell/qtl/software/qtlbim/rqtlbimtour.R")
```

4 Hypertension Example

1. Run steps 1-4, 11-14 and 17-20 of Example 1 of Broman's brief tour. This provides an overview of R/qtl.
2. Load R/qtlbim package.

   ```
   > library(qtlbim)
   ```

3. Remove the X chromosome. R/qtlbim does not currently handle the X chromosome properly.

   ```
   > data(hyper)
   > hyper <- subset(hyper, chr = 1:19)
   ```

4. Calculate genotype probabilities.

   ```
   > hyper <- qb.genoprob(hyper, step = 2)
   ```

   This is essentially calc.genoprob of Broman's step 11, but with variable step width required for R/qtlbim.

5. The time-consuming part of R/qtlbim involves creating the MCMC samples. We will NOT do this step in the tutorial. The random seed of 1616 is included to allow reproducible samples. To obtain different MCMC samples, simply use a different seed or drop the seed argument.

   ```
   ## The following command is commented out.
   ## qbHyper <- qb.mcmc(hyper, pheno.col = 1, seed = 1616)
   ```

   Note that this step creates a uniquely named directory containing flat (text) files with the MCMC samples, as well as constructing the qb object.

6. Alternatively, we can load already prepared MCMC samples.

   ```
   > qb.load(hyper, qbHyper)
   ```

   This step actually loads the hyper dataset with the X chromosome removed and genotype probabilities properly calculated, as well as the qb object qbHyper.

7. Show detailed summary of MCMC samples. This includes how the MCMC samples were constructed, where they were stored, etc.

   ```
   > summary(qbHyper)
   ```

   The diagnostic summaries characterize the number of QTL samples (nqtl), the posteriors for the mean and environmental variance (envvar), the explained variance components (varadd and varaa) and the total variance (var). In addition, the percentages of samples for number of QTL, number of epistatic pairs, and the most common epistatic pairs are shown.

8. A collection of diagnostic plots and summaries can be shown with the plot command:

   ```
   > plot(qbHyper)
   ```

   These include the following, which are identified by the separate routine that can be used to get that particular plot.

- Time series of mcmc runs. R/coda trace of MCMC samples to assess the Markov chain mixing.

  ```
  > tmp <- qb.coda(qbHyper)
  > summary(tmp)
  > plot(tmp)
  ```

- Jittered plot of quantitative trait loci by chromosome. A plot of samples loci across chromosomes (separated by main loci, epistatic loci and any GxE loci).

  ```
  > tmp <- qb.loci(qbHyper)
  > summary(tmp)
  > plot(tmp)
  ```

- Bayes Factor selection plots. Posteriors and Bayes factor ratios for number of QTL, pattern of QTL across chromosomes, chromosomes and epistatic pairs.

  ```
  > tmp <- qb.BayesFactor(qbHyper)
  > summary(tmp)
  > plot(tmp)
  ```

- HPD regions and best estimates. One dimensional scan of major QTL for test statistic (2logBF) and means by genotype.

  ```
  > tmp <- qb.hpdone(qbHyper)
  > summary(tmp)
  > plot(tmp)
  ```

- Epistatic effects. Size of epistatic effects for most common pairs of chromosomes.

  ```
  > tmp <- qb.epistasis(qbHyper)
  > summary(tmp)
  > plot(tmp)
  ```

- Summary diagnostics as histograms and boxplots by number of QTL. Posterior distribution overall and separately by number of QTL sampled for the overall mean, environmental variance, explained variance and heritability.

  ```
  > tmp <- qb.diag(qbHyper)
  > summary(tmp)
  > plot(tmp)
  ```

9. Perform log posterior density (LPD) scan of entire genome. This is analogous to R/qtl's `scanone`, which produces the LOD. However there are marginal LPD, adjusting for all other possible QTL, rather than one QTL summaries.

   ```
   > one <- qb.scanone(qbHyper, type = "LPD")
   ```

10. The plot for `qb.scanone` has separate LPD curves for overall (black), main effects (blue), epistatic effects (purple) and QTL by environment (dark red).

    ```
    > plot(one)
    ```

11. The summary shows the estimated peak by chromosome. There are two positions, `m.pos` for position of main effect peak and `e.pos` for position of epistatic effect peak.

    ```
    > summary(one)
    ```

12. We can filter the summary to only pick up chromosomes with large main effects and/or epistasis. We can then save those chromosome IDs.

    ```
    > sum.one <- summary(one, sort = "sum", threshold = c(sum = 4,
    +     epistasis = 4))
    > sum.one
    > chrs <- sort(sum.one$chr)
    > chrs
    ```

13. Now we can show a plot with this subset of chromosomes.

    ```
    > plot(one, chr = chrs)
    ```

14. Now look at cell means by genotype. We restrict attention to the key chromosomes.

    ```
    > onemean <- qb.scanone(qbHyper, chr = chrs, type = "cellmean")
    > plot(onemean)
    > summary(onemean)
    ```

15. An alternative way to filter the chromosomes is to use the highest posterior density (HPD) region. Here we ask for an LPD profile, rather than the default `2logBF`.

    ```
    > hpd <- qb.hpdone(qbHyper, profile = "LPD")
    > summary(hpd)
    > plot(hpd)
    ```

    The summary includes the limits of the HPD interval for each chromosome. The HPD region is computed across the entire genome.

16. Perform a two-dimensional scan on the key chromosomes.

    ```
    > two <- qb.scantwo(qbHyper, chr = chrs, type = "LPD")
    ```

17. Summarize the 2-D scan, sorting by the upper triangle, which contains epistasis by default. Threshold to include only values above 4.

    ```
    > summary(two, sort = "upper", threshold = c(upper = 4))
    ```

18. Plot to visualize epistatic chromosome pairs.

    ```
    > plot(two)
    ```

19. Slice along ridge relative to chromosome 15.

    ```
    > plot(two, chr = c(4, 6, 7), slice = 15)
    ```

20. Slice to examine cell mean for epistasis with chr 15. Plot shows profile of means for chromosome 6 and 7 when genotype on chr 15 is A (top) and H (bottom).

    ```
    > slice <- qb.sliceone(qbHyper, type = "cellmean", chr = c(4, 6,
    +     7), slice = 15)
    > summary(slice)
    > plot(slice, chr = 6:7)
    ```

21. Perform detailed slice at peak on chr 6 and 15. Rightmost plots are from R/qtl at nearest marker to peak.

    ```
    > slice = qb.slicetwo(qbHyper, c(6, 15), c(59, 19.5))
    > plot(slice)
    > summary(slice)
    ```

5 Hypertension Demo

An alternative demo of R/qtlbim run on the hypertension data can be run as

```
> library(qtlbim)
> demo(qb.hyper.tour)
```

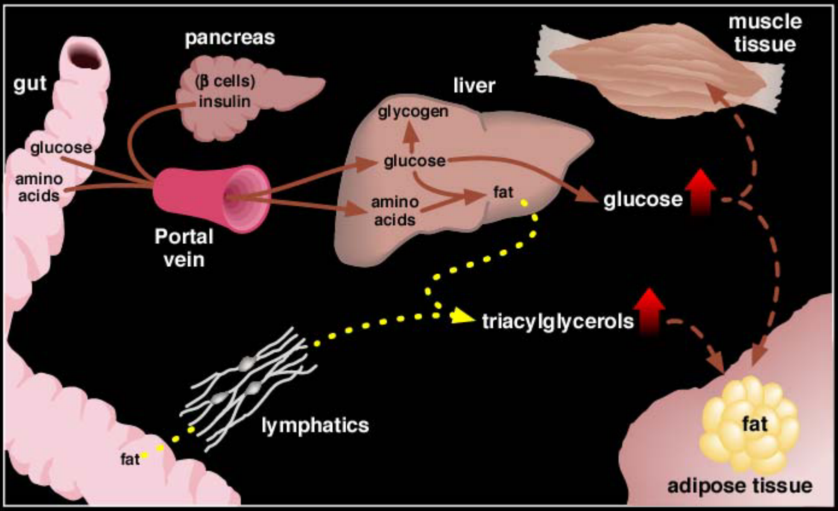
Multiple Traits & Microarrays

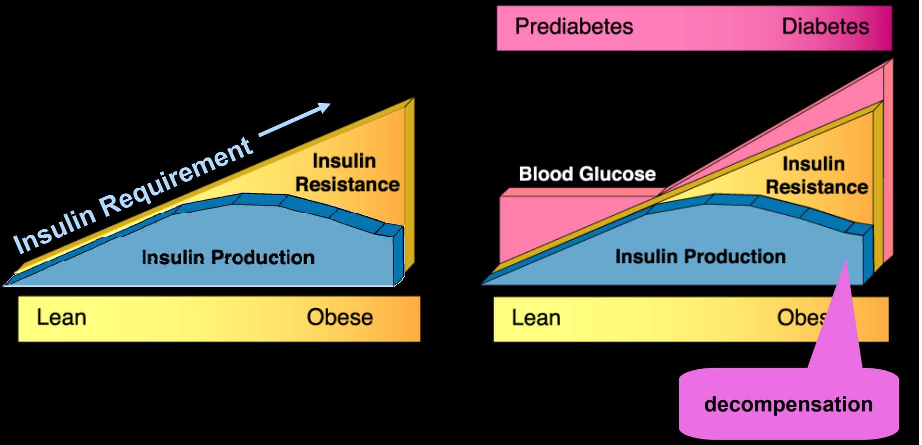
1. why study multiple traits together? 2-10
   – diabetes case study
2. design issues 11-13
   – selective phenotyping
3. why are traits correlated? 14-17
   – close linkage or pleiotropy?
4. modern high throughput 18-31
   – principal components & discriminant analysis
5. graphical models 32-36
   – building causal biochemical networks

1. why study multiple traits together?

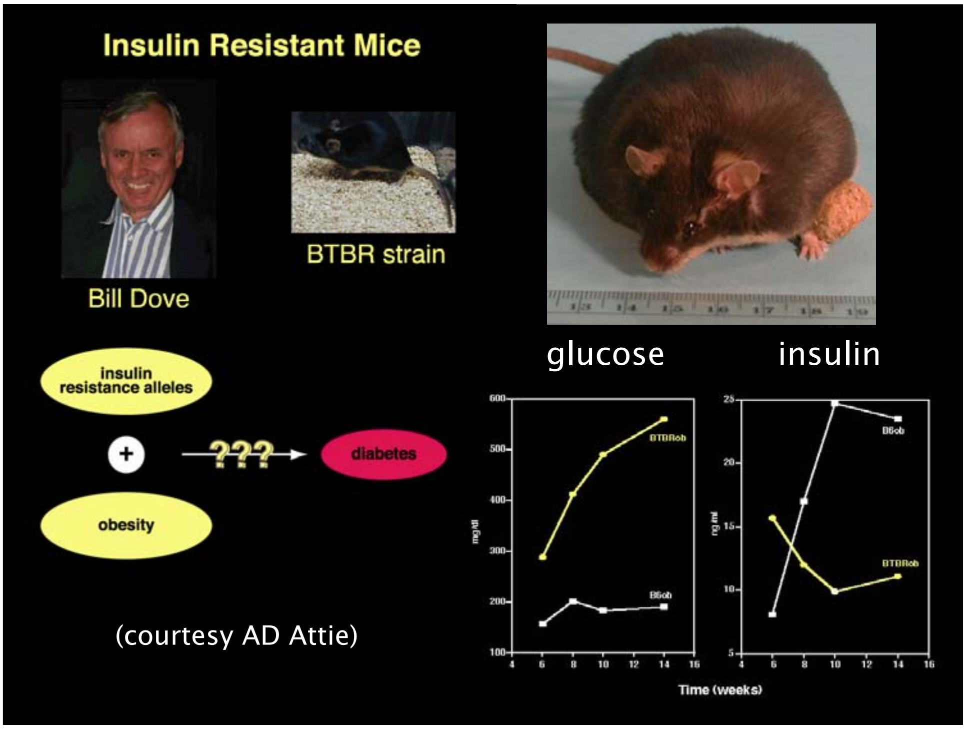
- avoid reductionist approach to biology
  - address physiological/biochemical mechanisms
  - Schmalhausen (1942); Falconer (1952)
- separate close linkage from pleiotropy
  - 1 locus or 2 linked loci?
- identify epistatic interaction or canalization
  - influence of genetic background
- establish QTL x environment interactions
- decompose genetic correlation among traits
- increase power to detect QTL

Type 2 Diabetes Mellitus





from Unger & Orci *FASEB J.* (2001) 15,312



Insulin Resistant Mice

Bill Dove

BTBR strain

insulin resistance alleles

+ obesity → ??? → diabetes

glucose

insulin

(courtesy AD Attie)

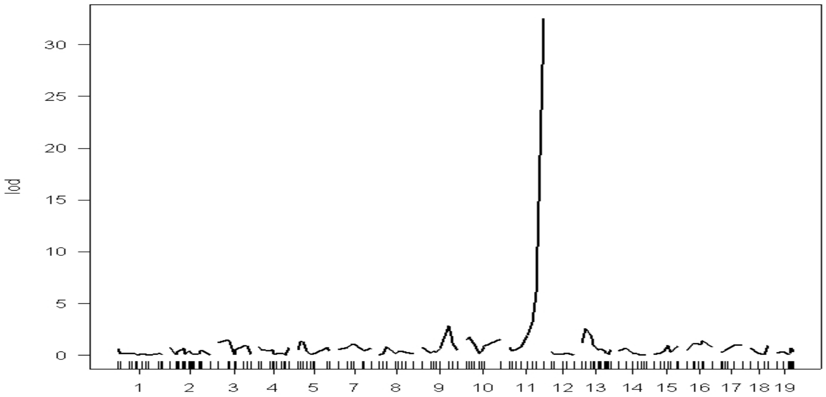
studying diabetes in an F2

- segregating cross of inbred lines
  - B6.ob x BTBR.ob → F1 → F2
  - selected mice with ob/ob alleles at leptin gene (chr 6)
  - measured and mapped body weight, insulin, glucose at various ages (Stoehr et al. 2000 *Diabetes*)
  - sacrificed at 14 weeks, tissues preserved
- gene expression data
  - Affymetrix microarrays on parental strains, F1
    - (Nadler et al. 2000 *PNAS*; Ntambi et al. 2002 *PNAS*)
  - RT-PCR for a few mRNA on 108 F2 mice liver tissues
    - (Lan et al. 2003 *Diabetes;* Lan et al. 2003 *Genetics*)
  - Affymetrix microarrays on 60 F2 mice liver tissues
    - design (Jin et al. 2004 *Genetics* tent. accept)
    - analysis (work in prep.)

why map gene expression as a quantitative trait?

- *cis*- or *trans*-action?
  - does gene control its own expression?
  - or is it influenced by one or more other genomic regions?
  - evidence for both modes (Brem et al. 2002 Science)
- simultaneously measure all mRNA in a tissue
  - ~5,000 mRNA active per cell on average
  - ~30,000 genes in genome
  - use genetic recombination as natural experiment
- mechanics of gene expression mapping
  - measure gene expression in intercross (F2) population
  - map expression as quantitative trait (QTL)
  - adjust for multiple testing

LOD map for PDI: *cis*-regulation (Lan et al. 2003)



mapping microarray data

- single gene expression as trait (single QTL)
  - Dumas et al. (2000 *J Hypertens*)
- overview, wish lists
  - Jansen, Nap (2001 *Trends Gen*); Cheung, Spielman (2002); Doerge (2002 *Nat Rev Gen*); Bochner (2003 *Nat Rev Gen*)
- microarray scan via 1 QTL interval mapping
  - Brem et al. (2002 *Science*); Schadt et al. (2003 *Nature*); Yvert et al. (2003 *Nat Gen*)
  - found putative *cis*- and *trans*- acting genes
- multivariate and multiple QTL approach
  - Lan et al. (2003 *Genetics*)

2. design issues for expensive phenotypes (thanks to CF “Amy” Jin)

- microarray analysis ~ $1000 per mouse
  - can only afford to assay 60 of 108 in panel
  - wish to not lose much power to detect QTL
- selective phenotyping
  - genotype all individuals in panel
  - select subset for phenotyping
  - previous studies can provide guide

selective phenotyping

- emphasize additive effects in F2
  - F2 design: 1QQ:2Qq:1qq
  - best design for additive only: 1QQ:1Qq
  - drop heterozygotes (Qq)
  - reduce sample size by half with no power loss
- emphasize general effects in F2
  - best design: 1QQ:1Qq:1qq
  - drop half of heterozygotes (25% reduction)
- multiple loci
  - same idea but care is needed
  - drop 7/16 of sample for two unlinked loci

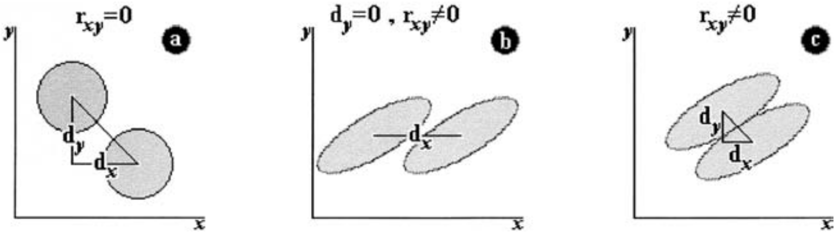
is this relevant to large QTL studies?

- why not phenotype entire mapping panel?
  - selectively phenotype subset of 50-67%
  - may capture most effects
  - with little loss of power
- two-stage selective phenotyping?
  - genotype & phenotype subset of 100-300
    - could selectively phenotype using whole genome
  - QTL map to identify key genomic regions
  - selectively phenotype subset using key regions

3. why are traits correlated?

- environmental correlation
  - non-genetic, controllable by design
  - historical correlation (learned behavior)
  - physiological correlation (same body)
- genetic correlation
  - pleiotropy
    - one gene, many functions
    - common biochemical pathway, splicing variants
  - close linkage
    - two tightly linked genes
    - genotypes *Q* are collinear

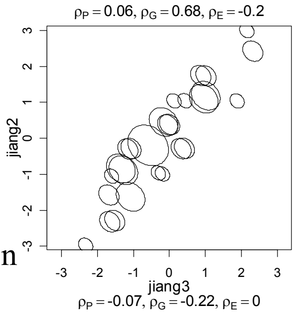
interplay of pleiotropy & correlation



pleiotropy only correlation only both

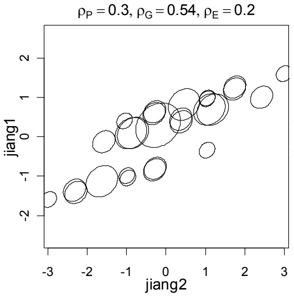
Korol et al. (2001)

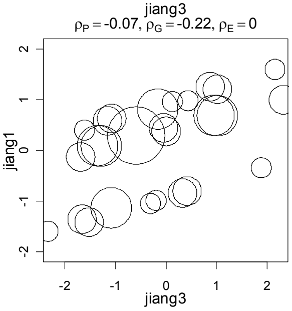
3 correlated traits (Jiang Zeng 1995)



ellipses centered on genotypic value
width for nominal frequency
main axis angle environmental correlation
3 QTL, F2
27 genotypes

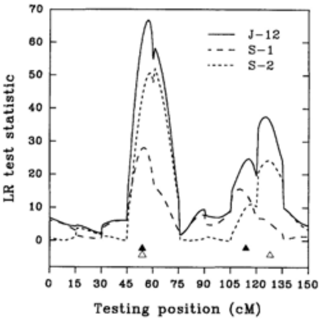
note signs of genetic and environmental correlation





pleiotropy or close linkage?

2 traits, 2 qtl/trait
pleiotropy @ 54cM
linkage @ 114,128cM
Jiang Zeng (1995)



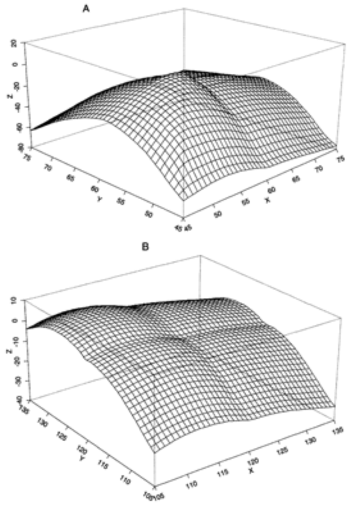
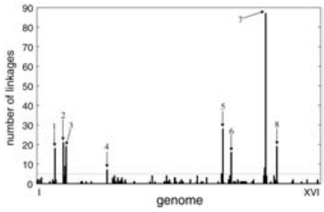


FIGURE 2.—Two-dimensional log-likelihood surfaces (expressed as deviations from the maximum of the log-likelihoods on the diagonal) for the test of pleiotropy *vs.* close linkage are presented for two regions: the region between 45 and 75 cM of Figure 1 (A) and the region between 105 and 135 cM (B). *X* is the testing position for a QTL affecting trait 1 and *Y* is the testing position for a QTL affecting trait 2. On the diagonal of *X-Y* plane, two QTL are located in the same position and statistically are treated as one pleiotropic QTL. *Z* is the likelihood ratio test statistic scaled to zero at the maximum point of the diagonal.

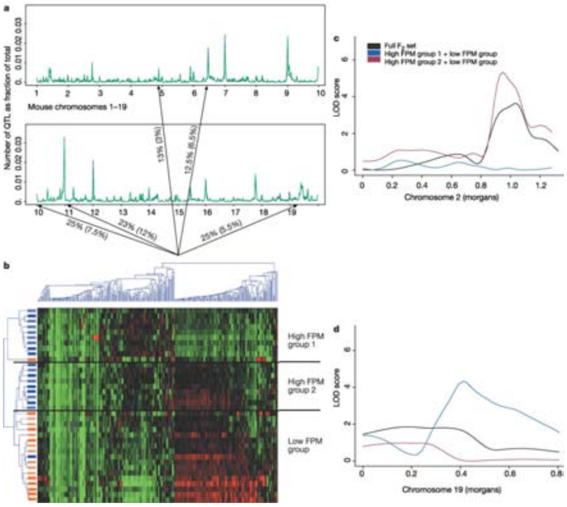
4. modern high throughput biology

- measuring the molecular dogma of biology
  - DNA → RNA → protein → metabolites
  - measured one at a time only a few years ago
- massive array of measurements on whole systems (“omics”)
  - thousands measured per individual (experimental unit)
  - all (or most) components of system measured simultaneously
    - whole genome of DNA: genes, promoters, etc.
    - all expressed RNA in a tissue or cell
    - all proteins
    - all metabolites
- systems biology: focus on network interconnections
  - chains of behavior in ecological community
  - underlying biochemical pathways
- genetics as one experimental tool
  - perturb system by creating new experimental cross
  - each individual is a unique mosaic

expression pleiotropy in yeast genome (Brem et al. 2002)



coordinated expression in mouse genome (Schadt et al. 2003)



finding heritable traits

(from Christina Kendziorski)

- reduce 30,000 traits to 300-3,000 heritable traits
- probability a trait is heritable

  $\text{pr}(H|Y,Q) = \text{pr}(Y|Q,H)\ \text{pr}(H|Q)\ /\ \text{pr}(Y|Q)$ Bayes rule

  $\text{pr}(Y|Q) = \text{pr}(Y|Q,H)\ \text{pr}(H|Q) + \text{pr}(Y|Q, \text{not } H)\ \text{pr}(\text{not } H|Q)$
- phenotype averaged over genotypic mean μ

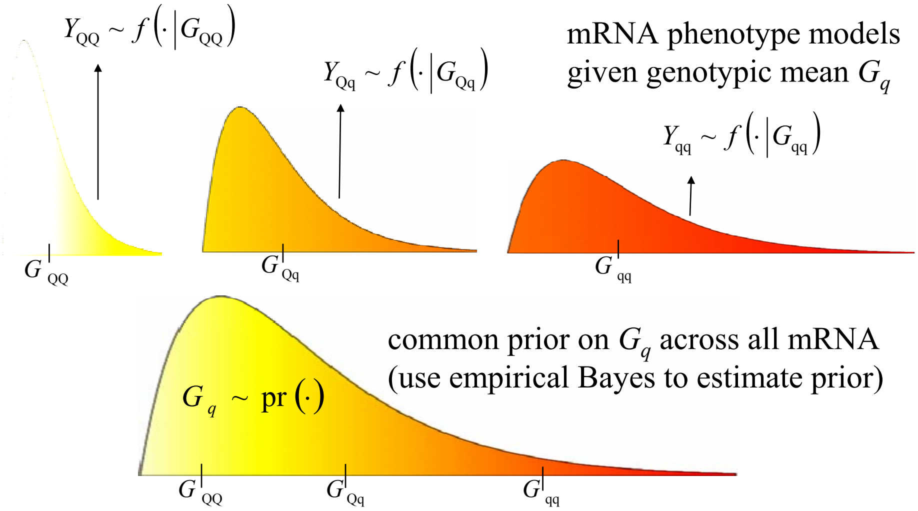
  $\text{pr}(Y|Q, \text{not } H) = f_0(Y) = \int f(Y|G)\ \text{pr}(G)\ \text{d}G$ if not H

  $\text{pr}(Y|Q, H) = f_1(Y|Q) = \prod_q f_0(Y_q)$ if heritable

  $Y_q = \{Y_i \mid Q_i = q\}$ = trait values with genotype $Q=q$

hierarchical model for expression phenotypes

(EB arrays: Christina Kendziorski)



$Y_{QQ} \sim f\left(\cdot \middle| G_{QQ}\right)$

$Y_{Qq} \sim f\left(\cdot \middle| G_{Qq}\right)$

$Y_{qq} \sim f\left(\cdot \middle| G_{qq}\right)$

mRNA phenotype models given genotypic mean G_q

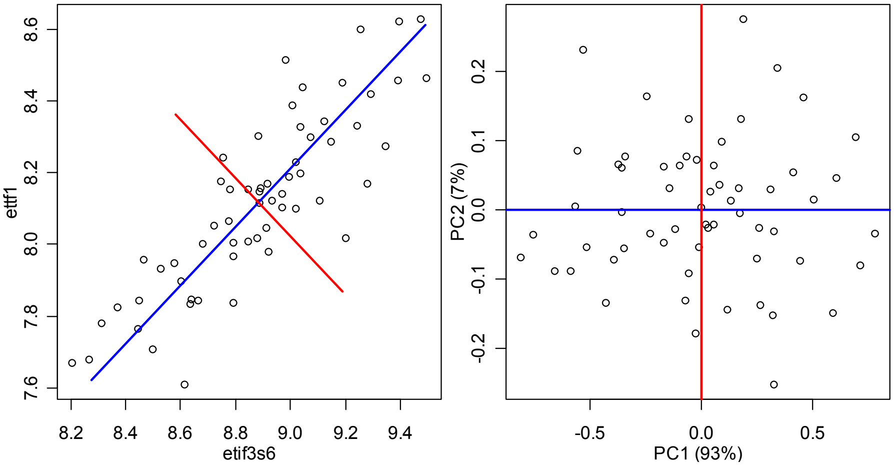
$G_q \sim \mathrm{pr}\left(\cdot\right)$

common prior on G_q across all mRNA (use empirical Bayes to estimate prior)

expression meta-traits: pleiotropy

- reduce 3,000 heritable traits to 3 meta-traits(!)
- what are expression meta-traits?
  - pleiotropy: a few genes can affect many traits
    - transcription factors, regulators
  - weighted averages: $Z = YW$
    - principle components, discriminant analysis
- infer genetic architecture of meta-traits
  - model selection issues are subtle
    - missing data, non-linear search
    - what is the best criterion for model selection?
  - time consuming process
    - heavy computation load for many traits
    - subjective judgement on what is best

PC for two correlated mRNA



PC across microarray functional groups

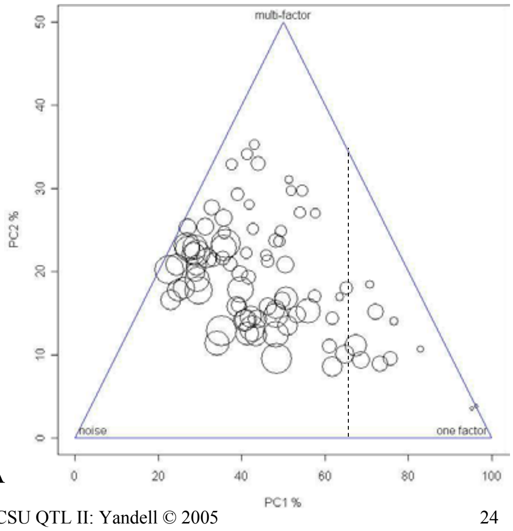
Affy chips on 60 mice
~40,000 mRNA

2500+ mRNA show DE
(via EB arrays with
marker regression)

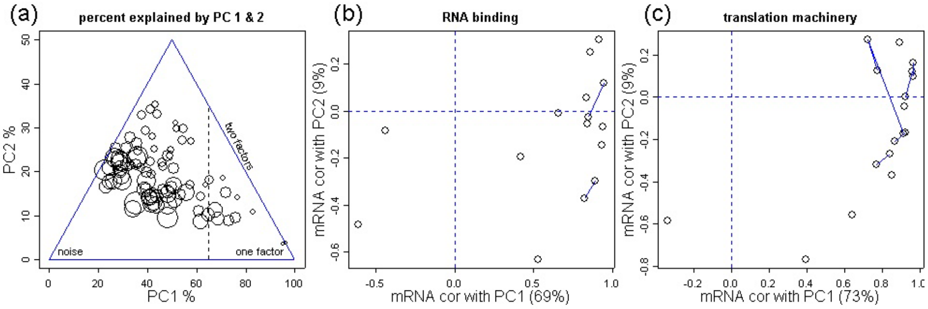
1500+ organized in
85 functional groups
2-35 mRNA / group

which are interesting?
examine PC1, PC2

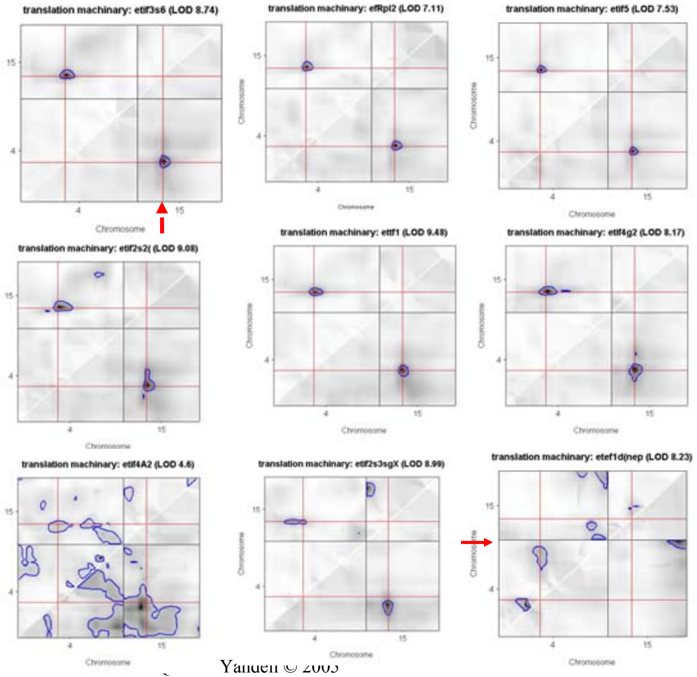
circle size = # unique mRNA



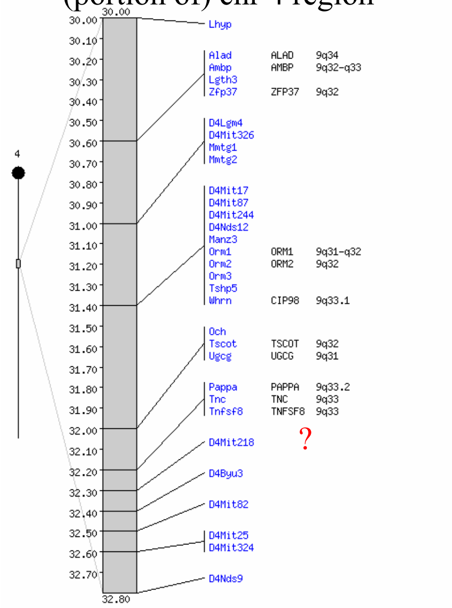
84 PC meta-traits by functional group focus on 2 interesting groups



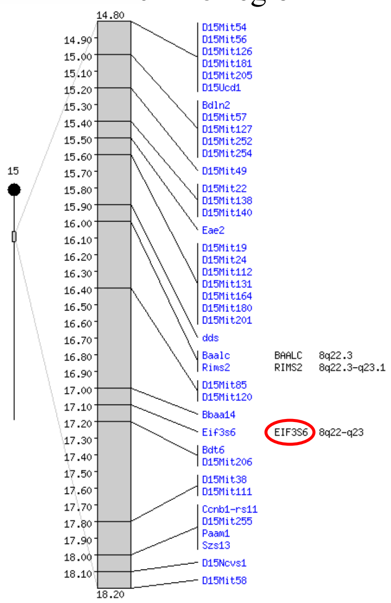
red lines: peak
for PC meta-trait
black/blue: peaks
for mRNA traits
arrows: *cis*-action?



(portion of) chr 4 region



chr 15 region



interaction plots for DA meta-traits

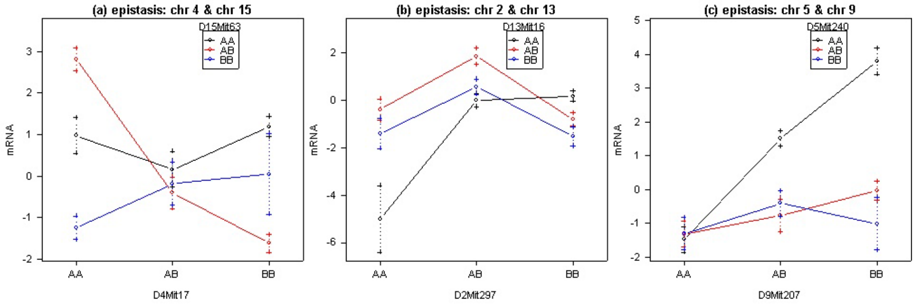
DA for all pairs of markers:

separate 9 genotypes based on markers

(a) same locus pair found with PC meta-traits

(b) Chr 2 region interesting from biochemistry (Jessica Byers)

(c) Chr 5 & Chr 9 identified as important for insulin, SCD

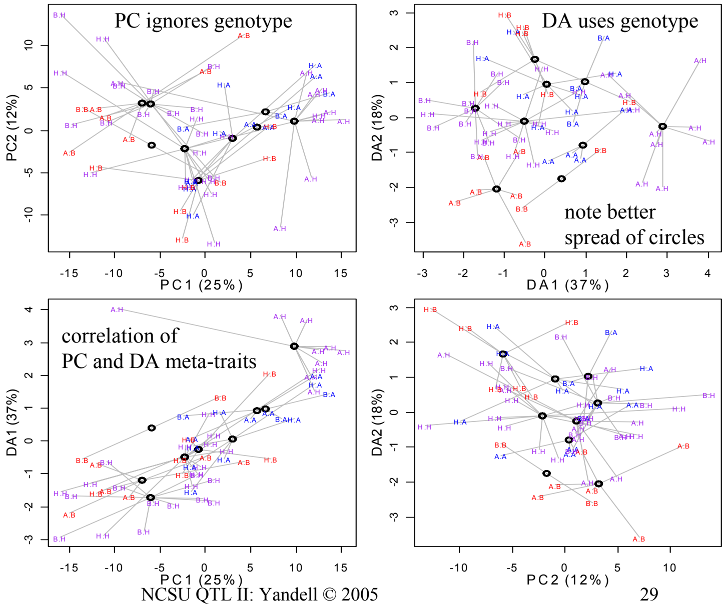


comparison of PC and DA meta-traits on 1500+ mRNA traits

genotypes from Chr 4/Chr 15 locus pair (circle=centroid)

PC captures spread without genotype

DA creates best separation by genotype


PC ignores genotype

DA uses genotype

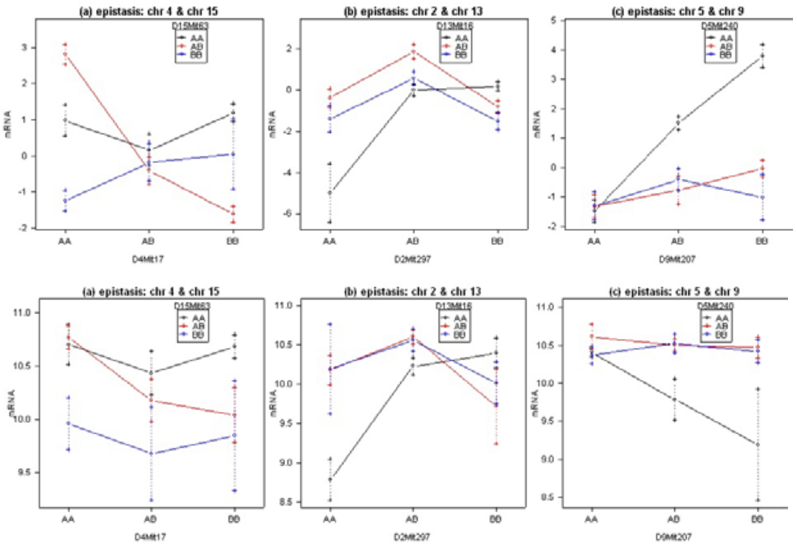
note better spread of circles

correlation of PC and DA meta-traits

relating meta-traits to mRNA traits

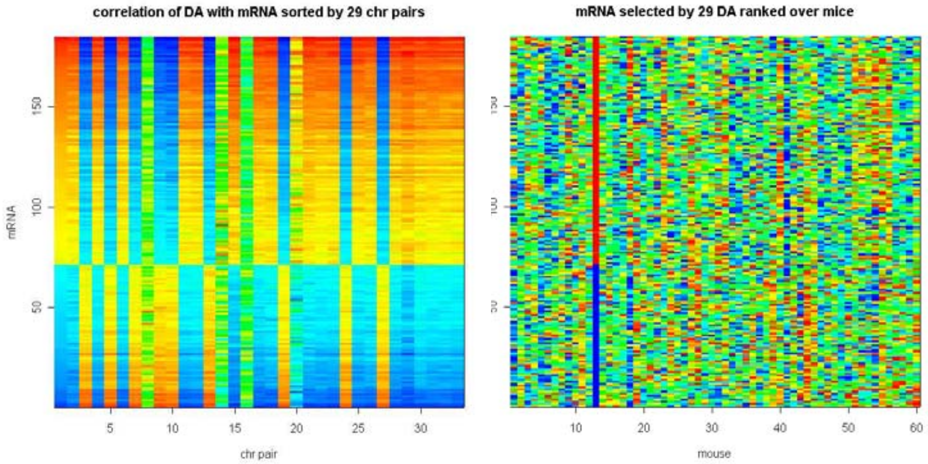
DA meta-trait standard units

SCD trait log2 expression



DA: a cautionary tale

(184 mRNA with |cor| > 0.5; mouse 13 drives heritability)



building graphical models

- infer genetic architecture of meta-trait
  - $E(Z \mid Q, M) = \mu_q = \beta_0 + \Sigma_{\{q \text{ in } M\}}\ \beta_{qk}$
- find mRNA traits correlated with meta-trait
  - $Z \approx \underline{YW}$ for modest number of traits $\underline{Y}$
- extend meta-trait genetic architecture
  - $\underline{M}$ = genetic architecture for $\underline{Y}$
  - expect subset of QTL to affect each mRNA
  - may be additional QTL for some mRNA

posterior for graphical models

•posterior for graph given multivariate trait & architecture

$\mathrm{pr}(G \mid \underline{Y}, Q, \underline{M}) = \mathrm{pr}(\underline{Y} \mid Q, G)\ \mathrm{pr}(G \mid \underline{M})\ /\ \mathrm{pr}(\underline{Y} \mid Q)$

–$\mathrm{pr}(G \mid \underline{M})$ = prior on valid graphs given architecture

•multivariate phenotype averaged over genotypic mean μ

$\mathrm{pr}(\underline{Y} \mid Q, G) = f_1(\underline{Y} \mid Q, G) = \prod q\ \ f_0(\underline{Y}q \mid G)$

$f_0(\underline{Y}_q \mid G) = \int f(\underline{Y}_q \mid \underline{\mu}, G)\ \mathrm{pr}(\underline{\mu})\ \mathrm{d}\underline{\mu}$

•graphical model G implies correlation structure on $\underline{Y}$

•genotype mean prior assumed independent across traits

$\mathrm{pr}(\underline{\mu}) = \prod_t \mathrm{pr}(\mu_t)$

from graphical models to pathways

- build graphical models

  QTL → RNA1 → RNA2

  – class of possible models

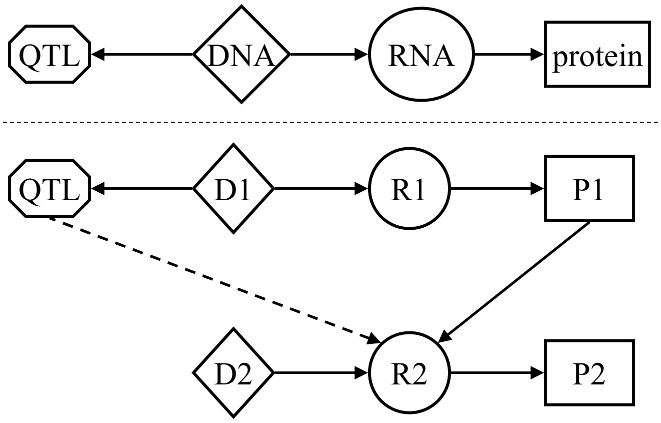
  – best model = putative biochemical pathway
- parallel biochemical investigation

  – candidate genes in QTL regions

  – laboratory experiments on pathway components

graphical models (with Elias Chaibub)

$$f_1(\underline{Y} \mid Q, G=g) = f_1(Y_1 \mid Q)\; f_1(Y_2 \mid Q, Y_1)$$



unobservable
meta-trait

observable
cis-action?

observable
trans-action

summary

- expression QTL are complicated
  - need to consider multiple interacting QTL
- coherent approach for high-throughput traits
  - identify heritable traits
  - dimension reduction to meta-traits
  - mapping genetic architecture
  - extension via graphical models to networks
- many open questions
  - model selection
  - computation efficiency
  - inference on graphical models