Parametric gene expression profiles and optimal Bayesian clustering

Peter Green and John Lau

University of Bristol
P.J.Green@bristol.ac.uk
John.Lau@bristol.ac.uk

BICSS symposium, Warwick, 30 May 2006

Keywords

- Gene expression
- Time course experiments, other numerical covariates
- Dirichlet process and other mixtures
- Asymmetric mixtures
- MCMC samplers for partition models
- Loss functions and optimal clustering

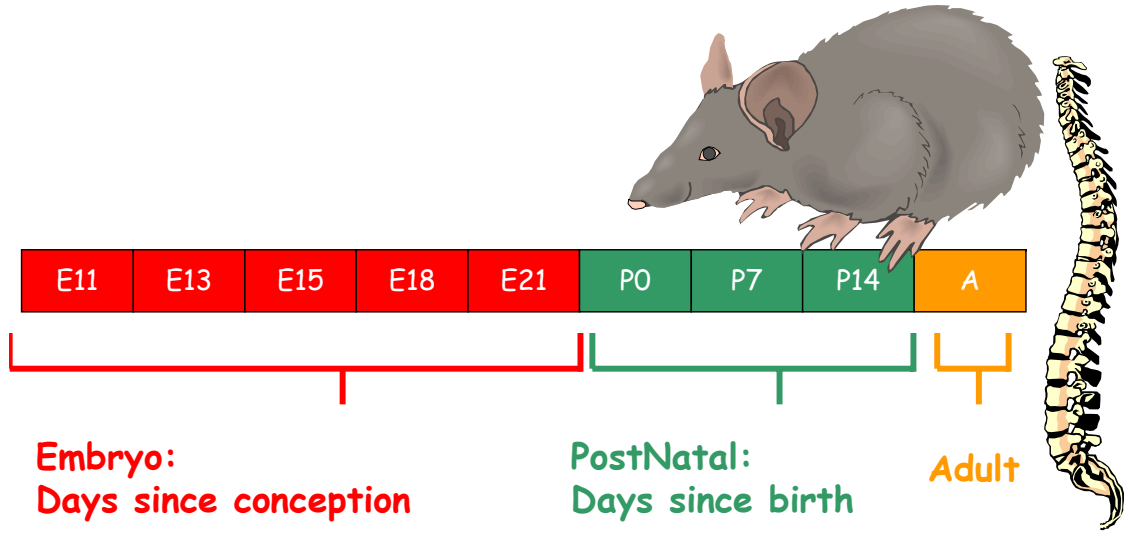
Gene expression data

We work with possibly replicated gene expression measures, often from Affymetrix gene chips. Data are $\{Y_{gsr}\}$, indexed by

- replicates $r = 1, 2, \ldots, R_s$
- conditions $s = 1, 2, \ldots, S$, and
- genes $g = 1, 2, \ldots, n$

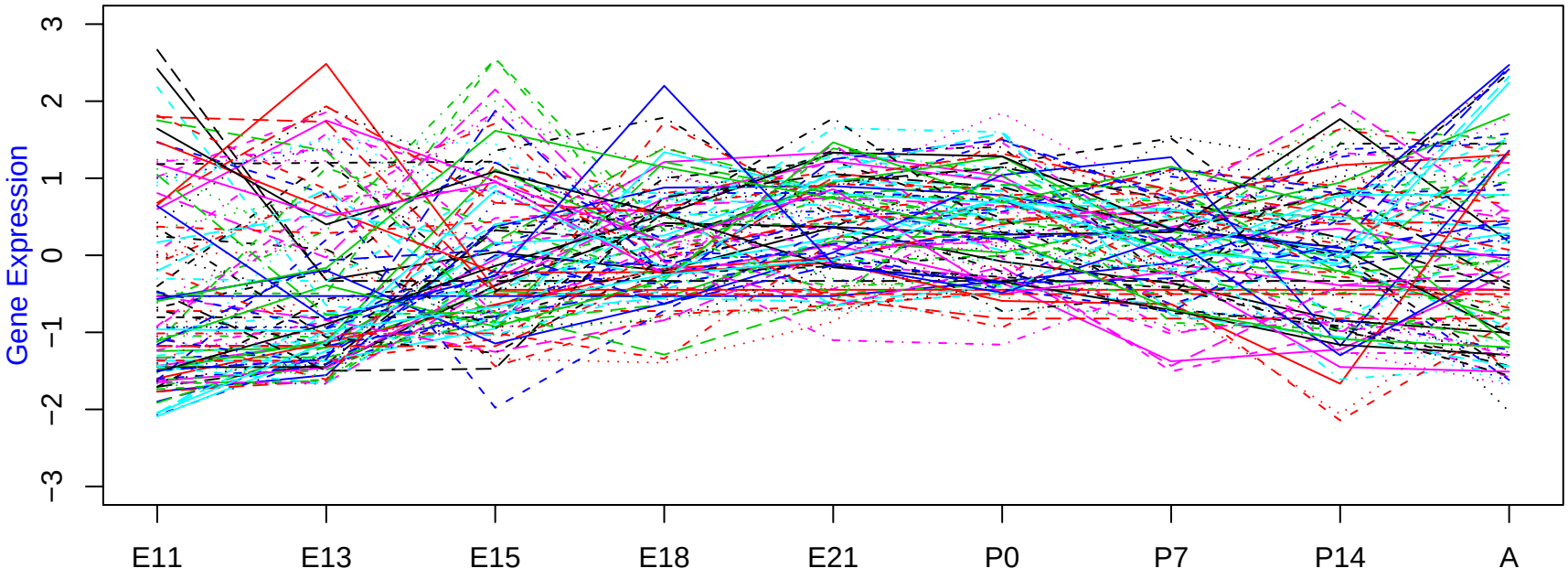
Typically R_s is very small, S is much smaller than n, and the 'conditions' represent different subjects, different treatments, or different experimental settings.

Rats CNS development



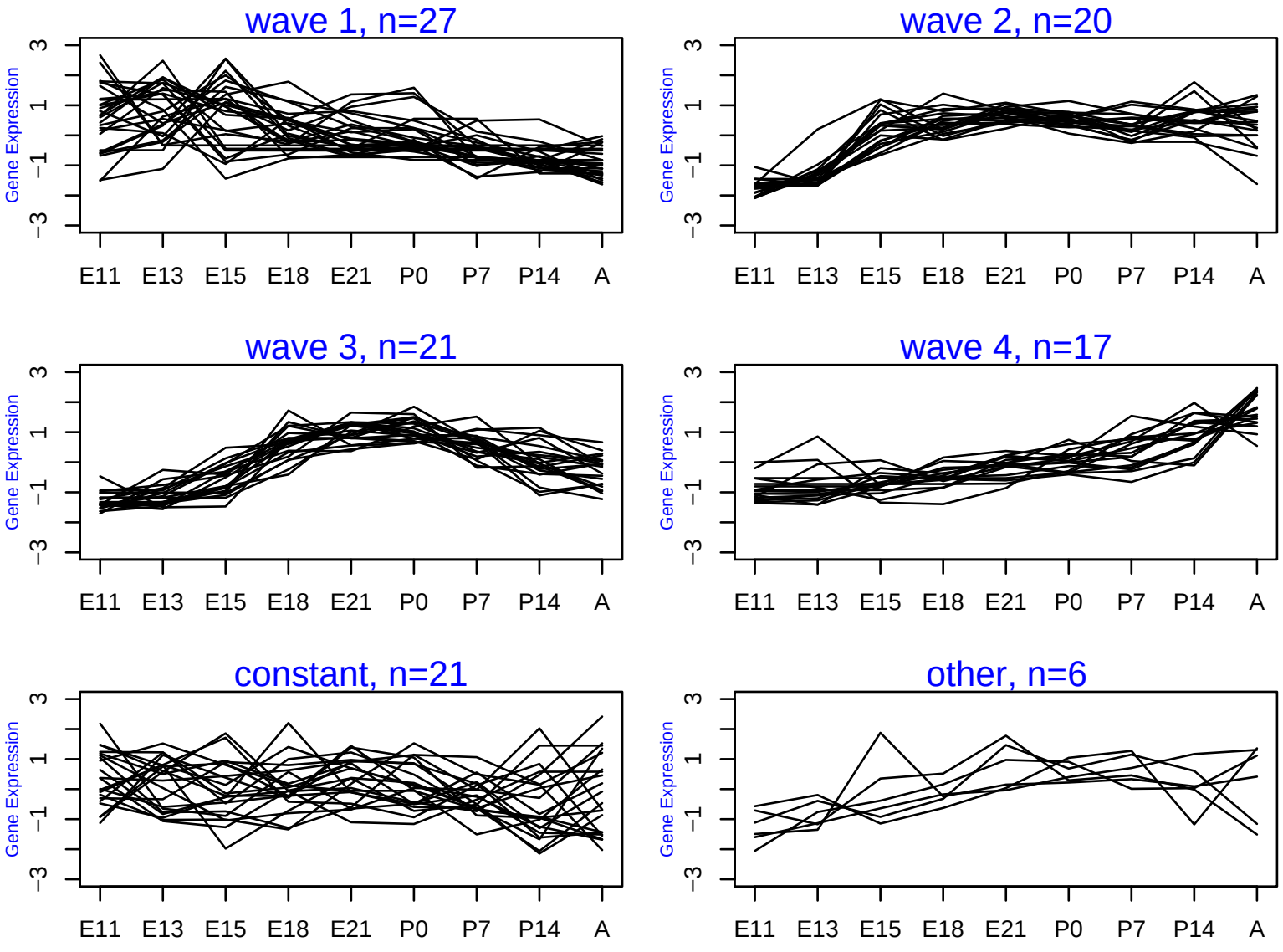
Wen et al (*PNAS*, 1998) studied development of central nervous system in rats: mRNA expression levels of 112 genes at 9 time points.

Rats data, normalised



Wen et al found clusters (waves) characterising distinct phases of development. . .

Rats data: Wen partition

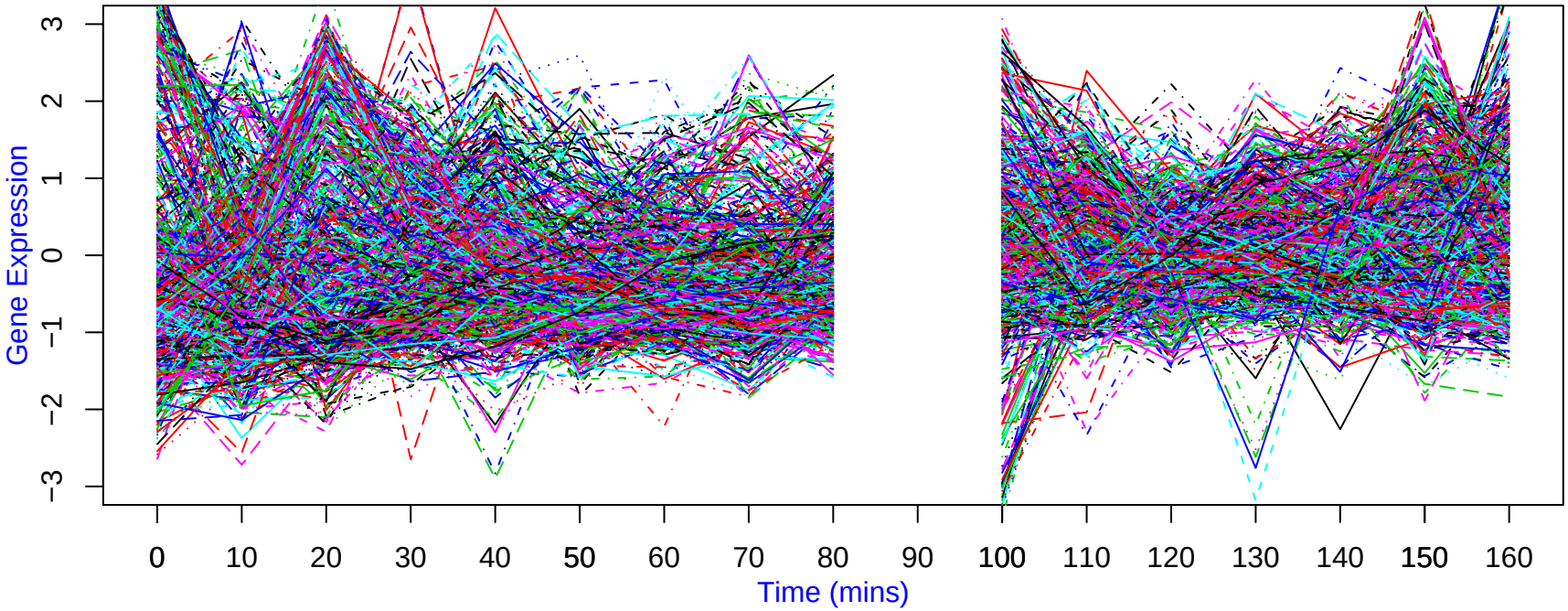


Yeast cell cycle data

Data from Cho et al (*Mol. Cell*, 1998) (can also be found in **R** `som` package). Yeast culture synchronised in G_1, then released and RNA collected at 10 minute intervals over 160 minutes ($\approx$ two cell cycles). 6601 genes $\times$ 17 time points. $t = 90$ excluded because of scaling difficulties.

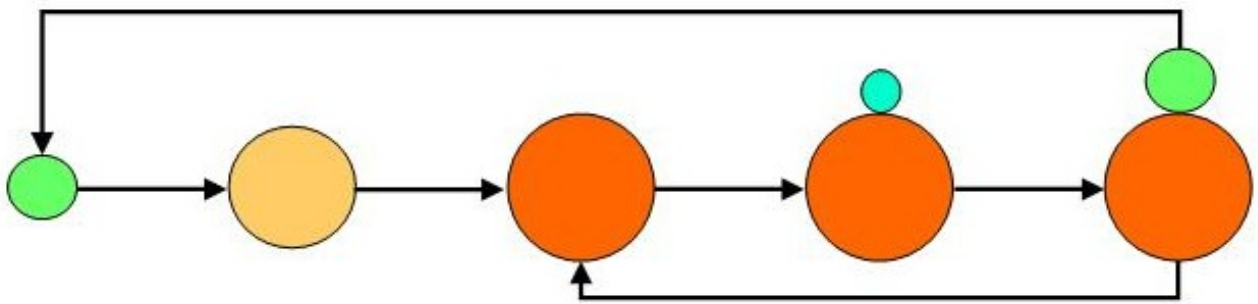
The biological interest is in identifying genes that are up- or down-regulated during the key phases of the cell cycle (early G_1, late G_1, S, G_2 and M), some of which may be involved in controlling the cycle itself.

Yeast data, normalised

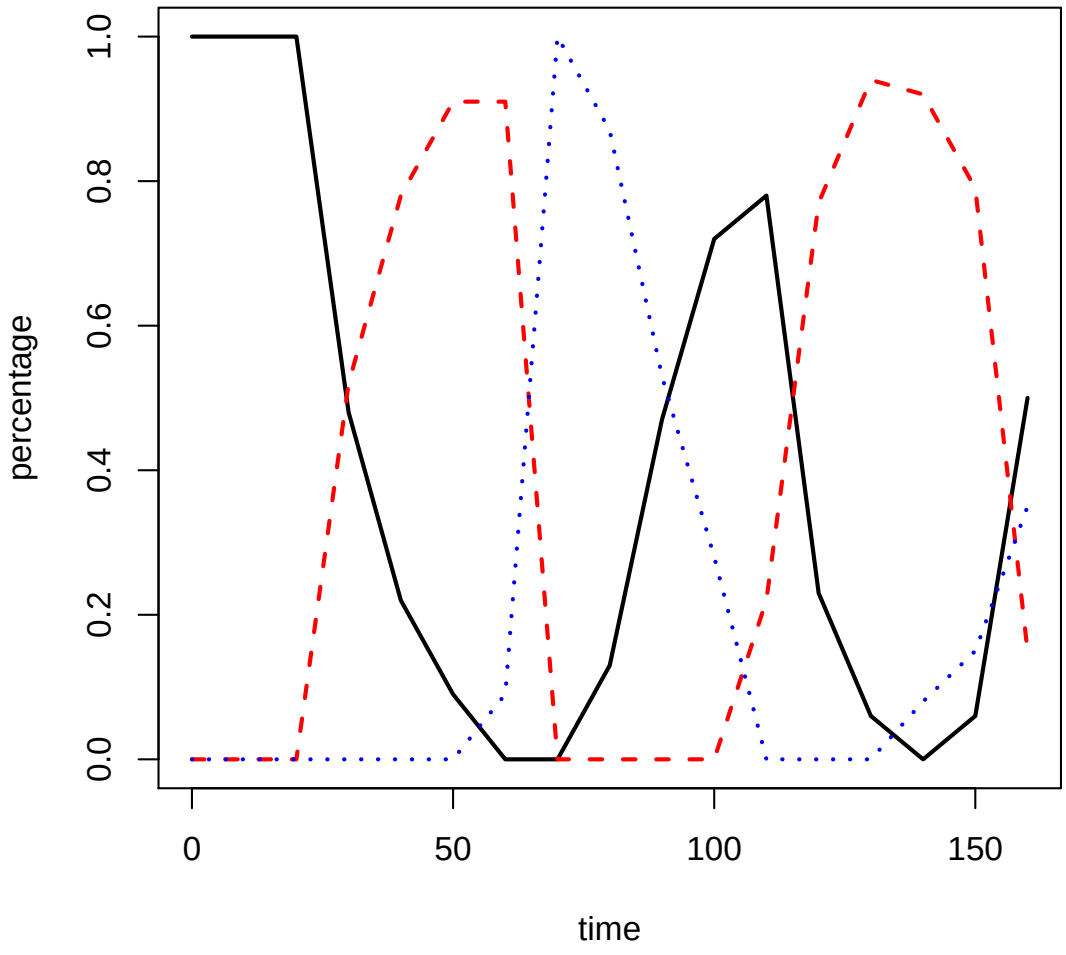


Yeast cell cycle

We have data on percentages of cells in each of three phases of growth (unbudded/small-budded/large-budded).



Yeast data, cell phase statistics basis



Parametric expression profiles

We suppose there is a k-dimensional ($k \leq S$) covariate vector x_s describing each condition, and model parametric dependence of Y on x, whilst regarding genes as *a priori* exchangeable, seeking common patterns across s under a nonparametric model for clustering.

Although other variants are easily envisaged (and we see a generalisation later), we suppose initially that

$$Y_{gsr} \sim N(x_s' \beta_g, \tau_g^{-1}), \quad \text{independently}$$

where $\theta_g = (\beta_g, \tau_g) \in \mathcal{R}^{k+1}$ are drawn i.i.d. from a distribution G, where in turn G has a Dirichlet process prior:

$$G \sim DP(\alpha, G_0)$$

The Dirichlet process - view 0

Given a probability distribution G_0 on an arbitrary measure space Ω, and a positive real α, we say the random distribution G on Ω follows a Dirichlet process,

$$G \sim DP(\alpha, G_0)$$

if for all partitions $\Omega = \bigcup_{j=1}^{m} B_j$ ($B_j \cap B_k = \emptyset$ if $j \neq k$), and for all m,

$$(G(B_1), \ldots, G(B_m)) \sim \text{Dirichlet}(\alpha G_0(B_1), \ldots, \alpha G_0(B_m))$$

Even if G_0 is continuous, G is a.s. discrete, so i.i.d. draws $\{\theta_g, g = 1, 2, \ldots, n\}$ from G exhibit ties.

The Dirichlet process - view 0(ctd.)

α measures *concentration*: given i.i.d. draws $\{\theta_g, g = 1, 2, \ldots, n\}$ from G,

- As $\alpha \to 0$, all θ_g are equal, drawn from G_0.
- As $\alpha \to \infty$, the θ_g are drawn i.i.d. from G_0.

The Dirichlet process - view 1

Sethuraman and Tiwari's 'stick-breaking' construction:

- draw $\theta_j^\star \sim G_0$, i.i.d., $j = 1, 2, \ldots$
- draw $V_j \sim \text{Beta}(1, \alpha)$, i.i.d., $j = 1, 2, \ldots$
- define G to be the discrete distribution putting probability $(1 - V_1)(1 - V_2) \ldots (1 - V_{j-1})V_j$ on $\theta_j^\star$
- draw θ_g i.i.d from G, $g = 1, 2, \ldots, n$.

The Dirichlet process - view 2

Finite mixture model $\sum_j w_j g_0(\cdot|\theta_j^\star)$ with Dirichlet weights:

- Draw $(w_1, w_2, \ldots, w_k) \sim$ Dirichlet$(\delta, \ldots, \delta)$
- Draw $c_g \in \{1, 2, \ldots, k\}$ with $P\{c_g = j\} = w_j$, i.i.d., $g = 1, \ldots, n$
- Draw $\theta_j^\star \sim G_0$, i.i.d., $j = 1, \ldots, k$
- Set $\theta_g = \theta_{c_g}^\star$

Let $k \to \infty$, $\delta \to 0$ such that $k\delta \to \alpha$.

G is invisible in view 2.

The Dirichlet process - view 3

Partition model: partition $\{1, 2, \ldots, n\} = \bigcup_{j=1}^{d} C_j$ at random, so that

$$p(C_1, C_2, \ldots, C_d) = \frac{\alpha^d \Gamma(\alpha) \prod_{j=1}^{d} (n_j - 1)!}{\Gamma(\alpha + n)}$$

where $n_j = \#C_j$. (NB preference for unequal cluster sizes!) Draw $\theta_j^\star \sim G_0$, i.i.d., $j = 1, \ldots, d$, and set $\theta_g = \theta_j^\star$ if $g \in C_j$.

G is also invisible in view 3.

The Dirichlet process - reprise

Genes are clustered, according to a tractable distribution parameterised by $\alpha > 0$, and within each cluster the regression parameter/precision pair $\theta = (\beta, \tau)$ is drawn i.i.d. from G_0.

We take a standard normal–inverse Gamma model: $\theta = (\beta, \tau) \sim G_0$ means

$$\tau \sim \Gamma(a_0, b_0) \quad \text{and} \quad \beta|\tau \sim \mathsf{N}_k(m_0, (\tau t_0)^{-1} I)$$

This is a conjugate set-up, so that (β, τ) can be integrated out *in each cluster*.

How nonparametric is that?

Multiple notations for partitions

- $\boldsymbol{c}$ is a partition of $\{1, 2, \ldots, n\}$
- clusters of partition are $C_1, C_2, \ldots, C_d$
  (d is the *degree* of the partition):
  $\bigcup_{j=1}^{d} C_j = \{1, 2, \ldots, n\}$, $C_j \cap C_{j'} = \emptyset$ if $j \neq j'$
- c is the allocation vector: $c_g = j$ if and only if $g \in C_j$

We are abusing notation by mixing up allocations and partitions: labelling of C_j is arbitrary, likewise values of $\{c_g\}$.

'Micro-posterior' and marginal likelihoods

Within-cluster parameter posteriors:

$$\tau_j^\star | Y \sim \Gamma(a_j, b_j)$$
$$\beta_j^\star | \tau_j^\star, Y \sim \mathsf{N}_k(m_j, (\tau_j^\star t_j)^{-1})$$

where

$$a_j = a_0 + 1/2 \#\{gsr : c_g = j\}$$
$$b_j = b_0 + 1/2 (Y_{C_j} - X_{C_j} m_0)' (X_{C_j} t_0^{-1} X'_{C_j})^{-1} (Y_{C_j} - X_{C_j} m_0)$$
$$m_j = (X'_{C_j} X_{C_j} + t_0 I)^{-1} (X'_{C_j} Y_{C_j} + t_0 m_0)$$
$$t_j = X'_{C_j} X_{C_j} + t_0 I$$

Marginal likelihoods $p(Y_{C_j})$ are multivariate t distributions.

DP mixture MCMC

There is a huge literature on MCMC for Dirichlet mixture models (Escobar, West, MacEachern, Mueller, Neal, Green, Richardson, . . .).

Non-conjugate cases demand keeping (β, τ) pairs in state vector, handled through various augmentation or reversible jump schemes.

In the conjugate case, it is obviously appealing to target Markov chain on posterior solely of the partition, and generate (β, τ) pairs from micro-posterior as needed. See also recent work by Nobile and Fearnside.

The incremental algorithm (Gibbs sampler/Pólya urn/ Weighted Chinese restaurant process)

MCMC on posterior for partition, limited to re-allocating single gene at a time (single-variable Gibbs sampler for c_g).

We allocate Y_g to a new cluster $C_\star$ with probability

$$\propto p(\boldsymbol{c}^{g\to\star}|\alpha) \times p(Y_g|\psi),$$

$\boldsymbol{c}^{g\to\star}$ denotes the current partition $\boldsymbol{c}$ with g moved to $C_\star$.

and to cluster C_j^{-g} with probability

$$\propto p(\boldsymbol{c}^{g \to j}|\alpha) \times p(Y_{C_j^{-g} \cup \{g\}}|\psi)/p(Y_{C_j^{-g}}|\psi).$$

$\boldsymbol{c}^{g \to j}$ denotes the partition $\boldsymbol{c}$, with g moved to cluster C_j.

The ratio of marginal likelihoods $p(Y|\psi)$ can be interpreted as the posterior predictive distribution of Y_g given those observations already allocated to the cluster, i.e. $p(Y_g|Y_{C_j^{-g}}, \psi)$ (= multivariate t for NIG setup).

For Dirichlet mixtures, we have

$$p(\boldsymbol{c}|\alpha) = \frac{\alpha^d \Gamma(\alpha) \prod_{j=1}^{d} (n_j - 1)!}{\Gamma(\alpha + n)}$$

where $n_j = \#C_j$ and $\boldsymbol{c} = (C_1, C_2, \ldots, C_d)$, so the re-allocation probabilities are explicit, and take a simple form.

But the same sampler can be used for many other partition models.

When the incremental sampler applies

All we require of the model are that

(a) A partition $\boldsymbol{c}$ of $\{1, 2, \ldots, n\}$ is drawn from a distribution with parameter α

(b) Conditionally on $\boldsymbol{c}$, parameters $(\theta_1, \theta_2, \ldots, \theta_d)$ are drawn independently from a distribution G_0 (possibly with a hyperparameter ψ)

(c) Conditional on $\boldsymbol{c}$ and on $\theta = (\theta_1, \theta_2, \ldots, \theta_d)$, $\{y_1, y_2, \ldots, y_n\}$ are drawn independently, from not necessarily identical distributions $p(y_i|\boldsymbol{c}, \theta) = f_i(y_i|\theta_j)$ for $i \in C_j$.

Examples

$p(\boldsymbol{c}^{g\to\star}|\alpha)$ and $p(\boldsymbol{c}^{g\to j}|\alpha)$ are simply proportional to

- α and $\#C_j^{-g}$ for the DP mixture model
- $(k - d(\boldsymbol{c}^{-g}))\delta$ and $\#C_j^{-g} + \delta$ for the Dirichlet-multinomial finite mixture model
- $\theta + \alpha d(\boldsymbol{c}^{-g})$ and $\#C_j^{-g} - \alpha$ for the Pitman–Yor Poisson–Dirichlet process

So the ease of using the Pólya urn/Gibbs sampler is not a reason to use DPM!

Simultaneous re-allocation

There is no need to stick to updating only one c_g at a time: the idea extends to simultanously re-allocating any subset of genes *currently in the same cluster*.

The notation is a bit cumbersome, but again the subset forms a new cluster, or moves to an existing cluster, with relative probabilities that are each products of two terms:

- the relative (new) partition prior probabilities, and
- the predictive density of the moved set of gene expressions, given those already in the receiving cluster

An asymmetric Dirichlet process mixture ('top table' model)

In gene expression, natural to suppose a 'background' cluster that is not *a priori* exchangeable with the others.

Take 'limit of finite mixture' view, and adapt it:

- Draw $(w_0, w_1, w_2, \ldots, w_k) \sim$ Dirichlet$(\gamma, \delta, \ldots, \delta)$
- Draw $c_g \in \{0, 1, \ldots, k\}$ with $P\{c_g = j\} = w_j$, i.i.d., $g = 1, \ldots, n$
- Draw $\theta_0^\star \sim G_{00}$, $\theta_j^\star \sim G_0$, i.i.d., $j = 1, \ldots, k$
- Set $\theta_g = \theta_{c_g}^\star$

Let $k \to \infty$, $\delta \to 0$ such that $k\delta \to \alpha$, but leave γ fixed.

Top-table Dirichlet process incremental sampler

When re-allocating gene g, there are three kinds of choice: a new cluster $C_\star$, the 'top table' C_0, or a regular cluster $C_j, j \neq 0$: the corresponding prior probabilities

$p(\boldsymbol{c}^{g\to\star}|\alpha)$, $p(\boldsymbol{c}^{g\to 0}|\alpha)$ and $p(\boldsymbol{c}^{g\to j}|\alpha)$

are proportional to

α, $(\gamma + \#C_0^{-g})$ and $\#C_j^{-g}$

for the asymmetric DP mixture model.

The model and sampler can be extended to have several classes of cluster, with allocations exchangeable within classes, and different parameter priors G_0 in each class.

Bayesian inference about partitions

The full posterior distribution – computed by sampling – tells us all about the partition and parameters: how to report a point estimate of the partition alone?

The posterior mode (MAP) partition is a common choice: but why? We would usually shy away from using posterior models in such a high-dimensional problem.

Here we consider going the extra mile – and obtaining optimal Bayesian clustering under a pairwise coincidence loss function.

Loss functions for clustering

So long as our formulation is exchangeable with respect to labelling of both items and clusters, we are confined to loss functions with the same invariances. These constraints, and issues of tractability, lead us to a pairwise coincidence loss function: if the true partition is $\boldsymbol{c}$ and you declare it to be $\widehat{\boldsymbol{c}}$ you incur a loss $L(\boldsymbol{c}, \widehat{\boldsymbol{c}}) =$

$$\sum_{1 \leq g < g' \leq G} \{aI[c_g = c_{g'}, \widehat{c}_g \neq \widehat{c}_{g'}] + bI[c_g \neq c_{g'}, \widehat{c}_g = \widehat{c}_{g'}]\}$$

The posterior expected loss is $E(L(\boldsymbol{c}, \widehat{\boldsymbol{c}})|\boldsymbol{y}) =$

$$\sum \{aP(c_g = c_{g'}|\boldsymbol{y})I[\widehat{c}_g \neq \widehat{c}_{g'}] + bP(c_g \neq c_{g'}|\boldsymbol{y})I[\widehat{c}_g = \widehat{c}_{g'}]\}$$

Loss functions for clustering (2)

After a little manipulation, we find minimising expected loss is the same as maximising

$$\ell(\widehat{\boldsymbol{c}}) = \sum_{1 \le g < g' \le G} \{(\rho_{gg'} - K) I[\widehat{c}_g = \widehat{c}_{g'}]\}$$

$$= \sum_{j} \sum_{g,g' \in \widehat{C}_j} (\rho_{gg'} - K)$$

where $K = b/(a+b) \in [0,1]$ and $\rho_{gg'} = P(c_g = c_{g'}|\boldsymbol{y})$. Note this requires only saving the posterior pairwise coincidence probabilities from the MCMC run.

How to optimise this?

Toy example

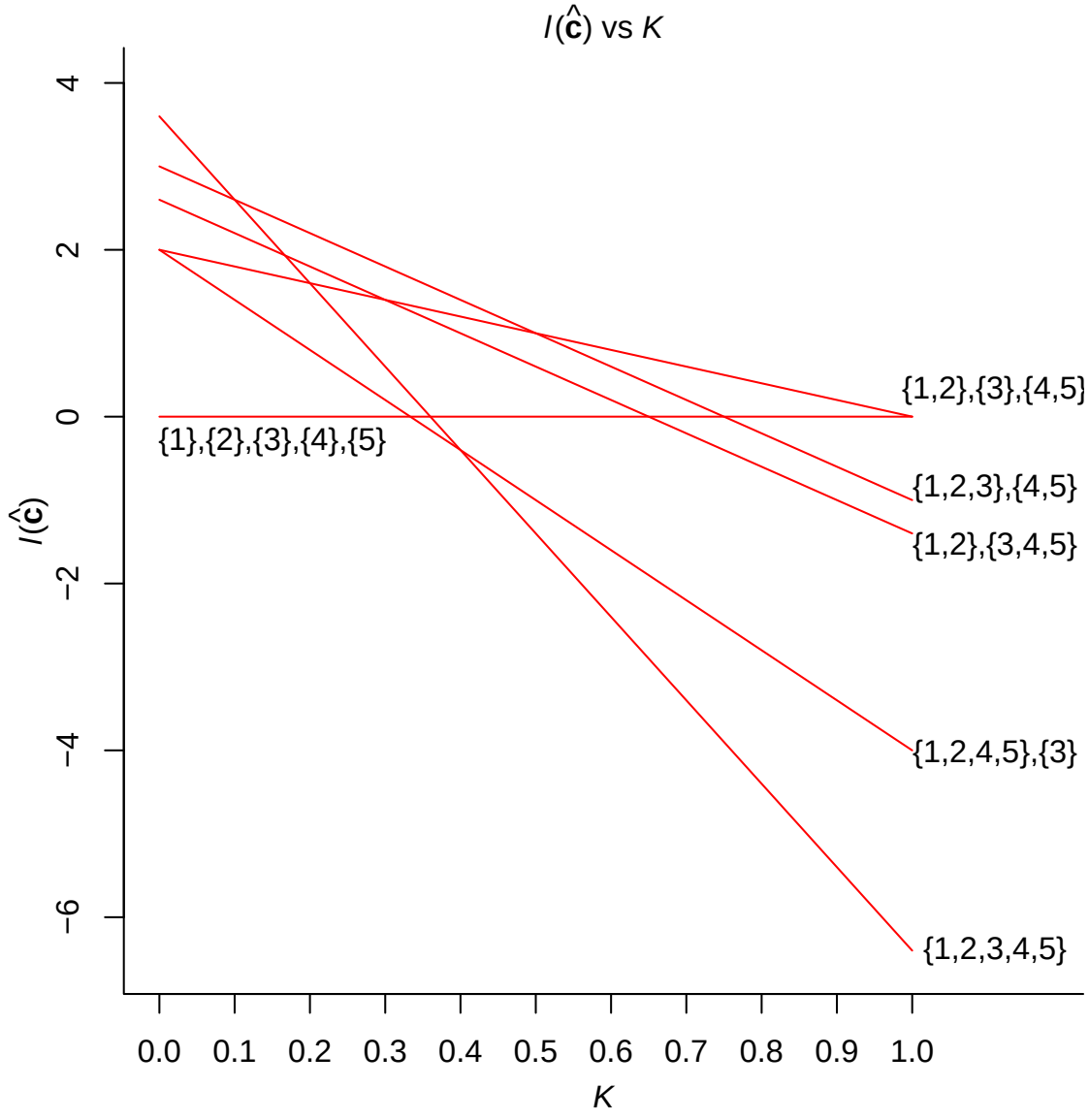
Suppose there are $n = 5$ items/elements, and that the partitions and corresponding probabilities are

$$\begin{aligned} \boldsymbol{c}_1 &= \{\{1,2,3\},\{4,5\}\} & P(\boldsymbol{c}_1) &= 0.5 \\ \boldsymbol{c}_2 &= \{\{1,2\},\{3\},\{4,5\}\} & P(\boldsymbol{c}_2) &= 0.2 \\ \boldsymbol{c}_3 &= \{\{1,2\},\{3,4,5\}\} & P(\boldsymbol{c}_3) &= 0.3 \end{aligned}$$

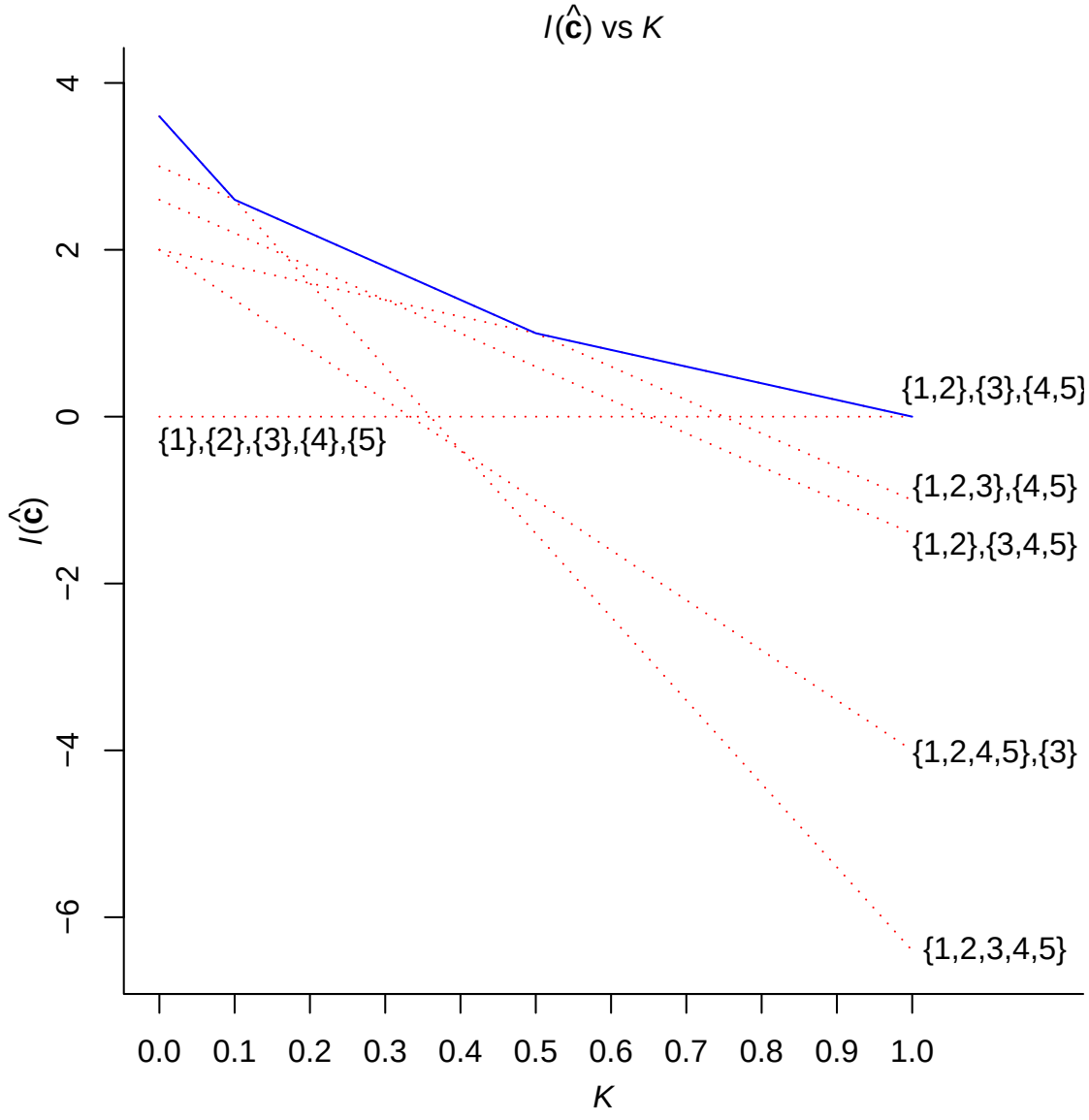
The ρ_{ij} matrix is

	1	2	3	4	5
1	–	1	0.5	0	0
2	–	–	0.5	0	0
3	–	–	–	0.3	0.3
4	–	–	–	–	1
5	–	–	–	–	–

Toy example


$l(\hat{\mathbf{c}})$ vs K
$l(\hat{\mathbf{c}})$
4
2
0
−2
−4
−6
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
K
{1,2},{3},{4,5}
{1},{2},{3},{4},{5}
{1,2,3},{4,5}
{1,2},{3,4,5}
{1,2,4,5},{3}
{1,2,3,4,5}

Toy example


$l(\hat{\mathbf{c}})$ vs K
$l(\hat{\mathbf{c}})$
4
2
0
−2
−4
−6
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
K
{1,2},{3},{4,5}
{1},{2},{3},{4},{5}
{1,2,3},{4,5}
{1,2},{3,4,5}
{1,2,4,5},{3}
{1,2,3,4,5}

Binary integer programming

To maximise

$$\ell(\widehat{\boldsymbol{c}}) = \sum_{1 \le g < g' \le G} \{(\rho_{gg'} - K) I[\widehat{c}_g = \widehat{c}_{g'}]\}$$

over choice of partitions $\widehat{\boldsymbol{c}}$, we first treat this as a mathematical programming problem in the binary variables $x_{gg'} = I[\widehat{c}_g = \widehat{c}_{g'}]$. We aim to maximise $\sum(\rho_{gg'} - K)x_{gg'}$ subject to numerous constraints ensuring $\widehat{\boldsymbol{c}}$ is a partition. It is necessary and sufficient that for all triples $\{g, g', g''\}$, $x_{gg'} = 1$ implies $x_{gg''} = x_{g'g''}$, and these constraints can be represented as algebraic inequalities $x_{gg'} + x_{gg''} - x_{g'g''} \le 1$ for all $\{g, g', g''\}$.

Binary integer programming (2)

The optimisation is now in the form of a standard linear integer programme in binary variables:

maximise

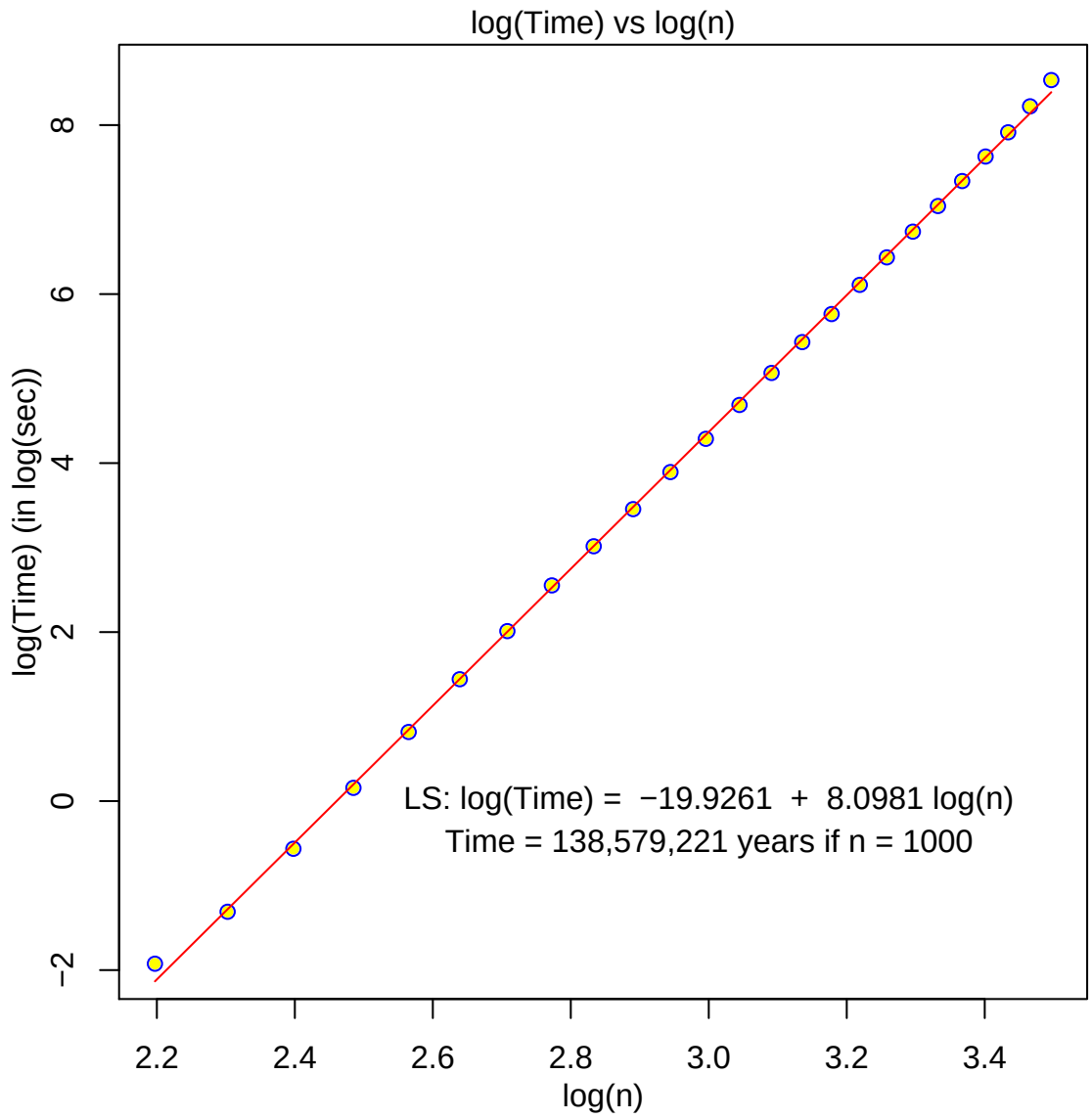
$$\ell(\widehat{\boldsymbol{c}}) = \sum (\rho_{gg'} - K) x_{gg'}$$

subject to all

$x_{gg'} \in \{0, 1\}$ and $x_{gg'} + x_{gg''} - x_{g'g''} \leq 1$ for all $\{g, g', g''\}$,

and on a small scale can easily be solved with standard (free) software.

This solution scales very badly with number of items (genes)! In fact, the problem is known to be NP hard.


log(Time) vs log(n)
log(Time) (in log(sec))
log(n)
LS: log(Time) = −19.9261 + 8.0981 log(n)
Time = 138,579,221 years if n = 1000
−2
0
2
4
6
8
2.2
2.4
2.6
2.8
3.0
3.2
3.4

A simple heuristic

We have had some success with a very simple heuristic – iteratively removing items (genes) from the partition one-by-one and reallocating them so as to maximise the objective function $\ell(\widehat{\boldsymbol{c}}) = \sum(\rho_{gg'} - K)x_{gg'}$ at each step.

Simultaneous approximate optimisation for all K

K is the ratio of elementary costs $b/(a+b)$, and we would be interested in finding the optimal partition for all K, at least in some interval. As would be anticipated, the optimum varies with K but remains constant on intervals of K. (But there need be no monotonicity of the optimal partition with respect to K).

To get the flavour of our approach to simultaneous optimisation, note the form of our objective function $\ell(\widehat{\boldsymbol{c}}) = \sum(\rho_{gg'} - K)x_{gg'}$ – this is a non-increasing linear function of K.

Simultaneous approximate optimisation for all K (2)

$\sum(\rho_{gg'} - K)x_{gg'}$ is a non-increasing linear function of K, and so the maximum of such functions over any candidate set $\mathcal{C}$ of partitions $\widehat{\boldsymbol{c}}$ is a non-increasing convex polygonal function of K, that is non-decreasing in $\mathcal{C}$ with respect to set inclusion.

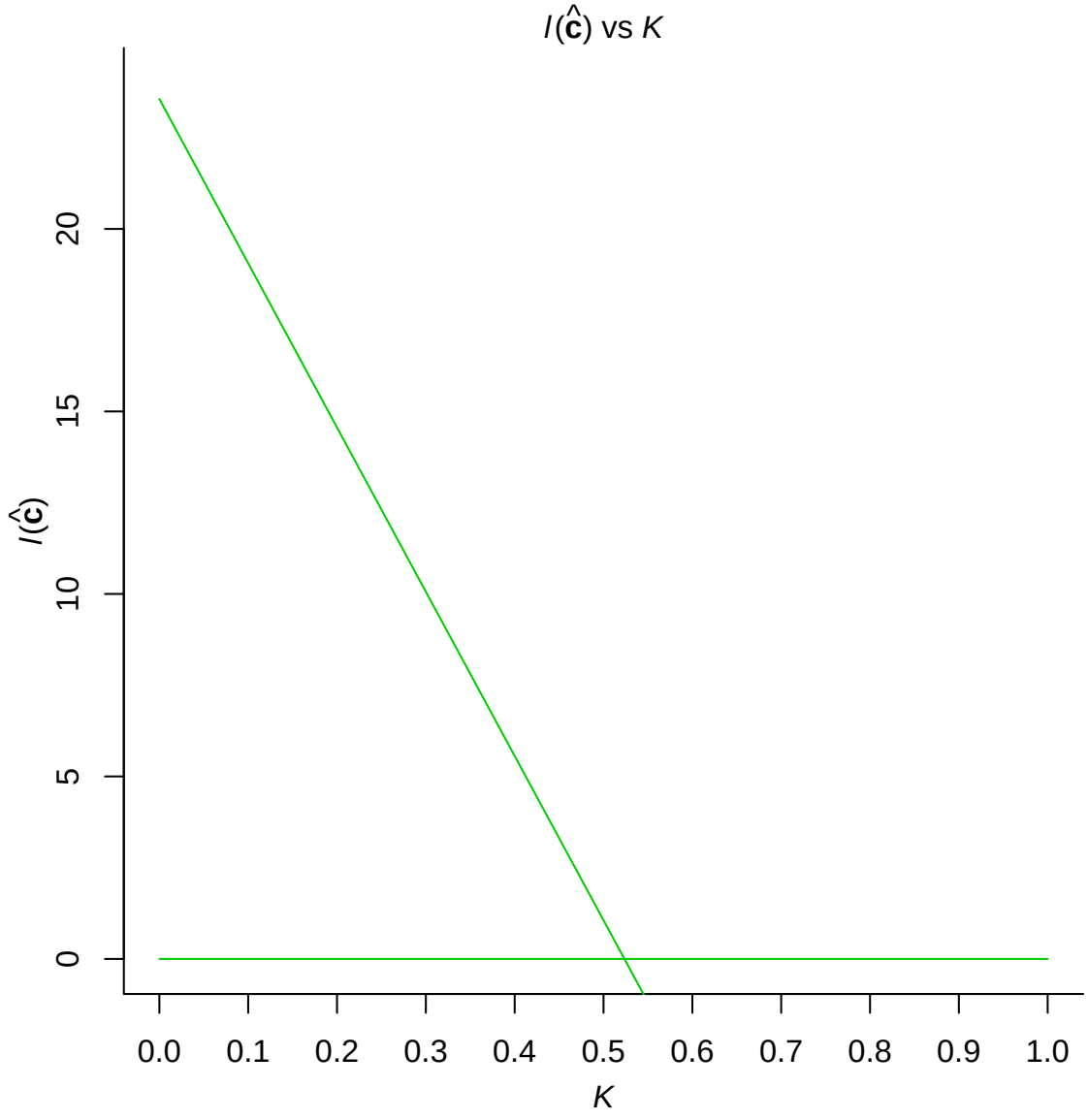
For any $\mathcal{C}$, $\sup_{\widehat{\boldsymbol{c}}\in\mathcal{C}} \sum(\rho_{gg'} - K)x_{gg'}$ is characterised by a smaller subset $\partial\mathcal{C} \in \mathcal{C}$ of 'active' partitions that define the convex hull, and our algorithm proceeds by iteratively adding new partitions to $\mathcal{C}$, updating its representation $\partial\mathcal{C}$ as needed. The new partitions for consideration are generated by single-gene reallocations to partitions in the current active set, as in the simple heuristic.

A 10-item example

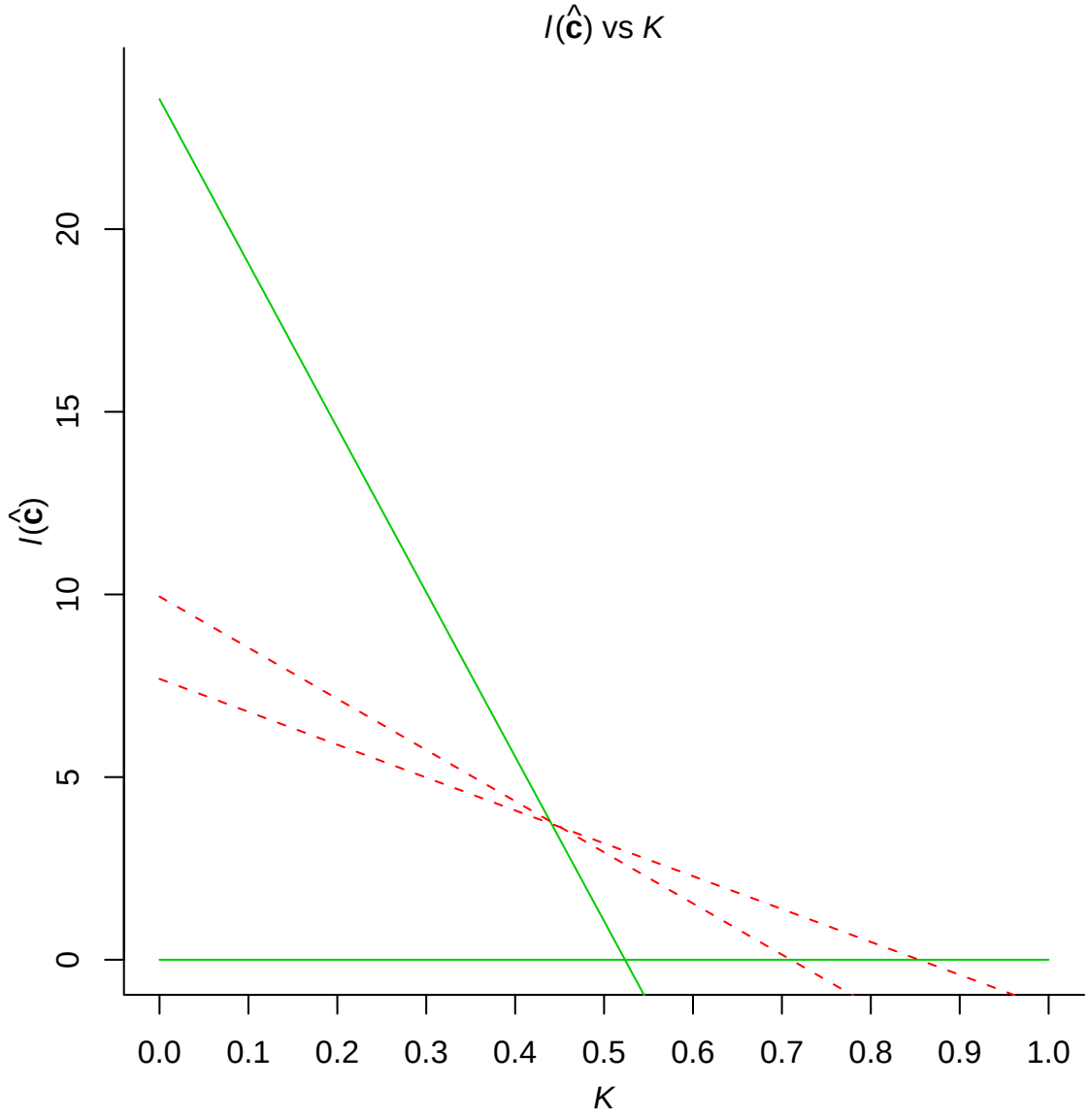
The ρ_{ij} matrix is

	1	2	3	4	5	6	7	8	9	10
1	–	0.1551	0.7426	0.2610	0.3069	0.9572	0.0990	0.6533	0.3563	0.4436
2	–	–	0.8502	0.5840	0.8307	0.8146	0.4835	0.8732	0.1441	0.3150
3	–	–	–	0.9333	0.4686	0.5256	0.0269	0.3754	0.4597	0.9801
4	–	–	–	–	0.5194	0.7063	0.9799	0.4827	0.9398	0.0252
5	–	–	–	–	–	0.8931	0.7275	0.3576	0.8293	0.3555
6	–	–	–	–	–	–	0.4834	0.0305	0.4134	0.6203
7	–	–	–	–	–	–	–	0.1808	0.8813	0.6506
8	–	–	–	–	–	–	–	–	0.0306	0.1819
9	–	–	–	–	–	–	–	–	–	0.6311

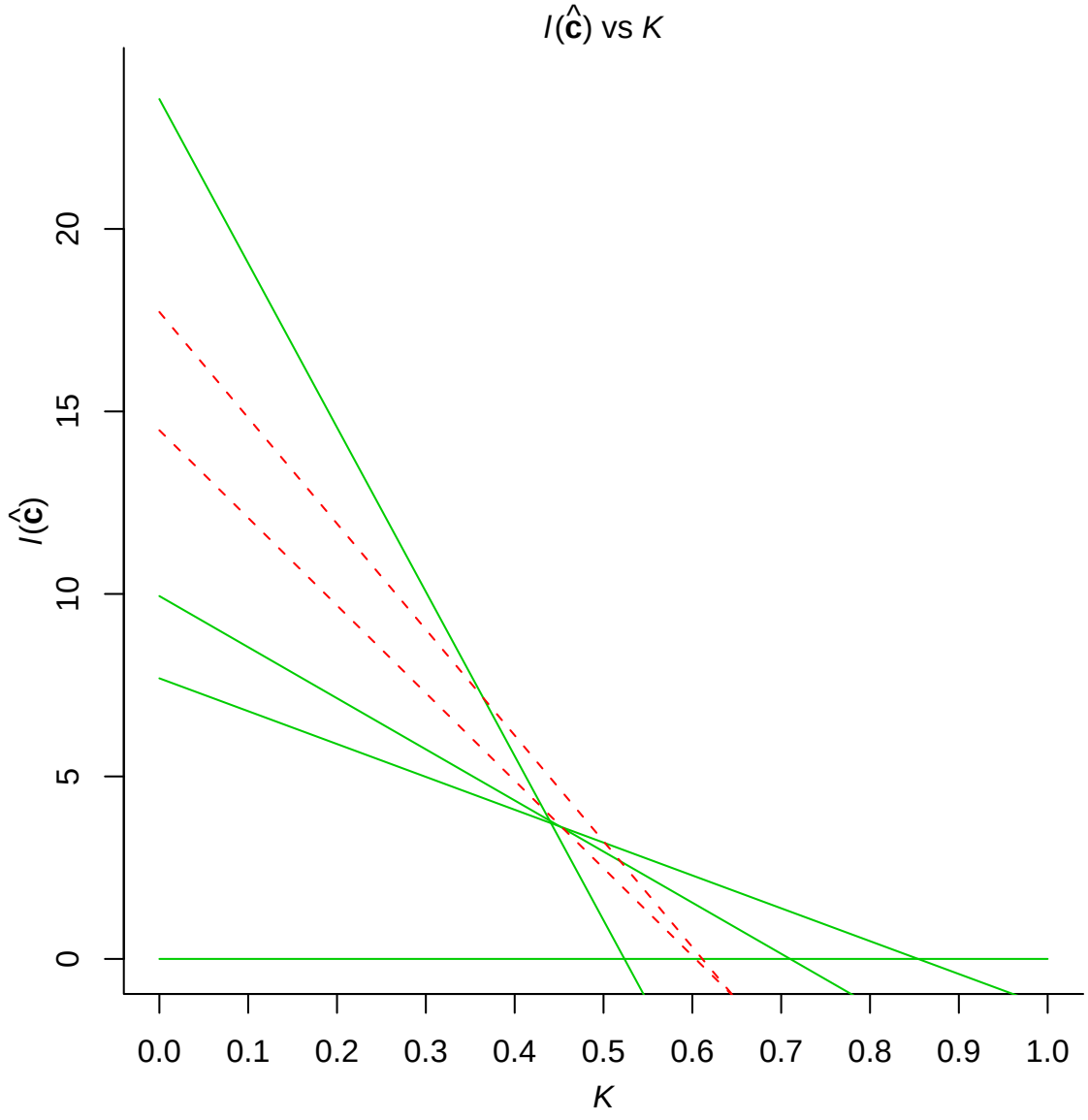
A 10-item example



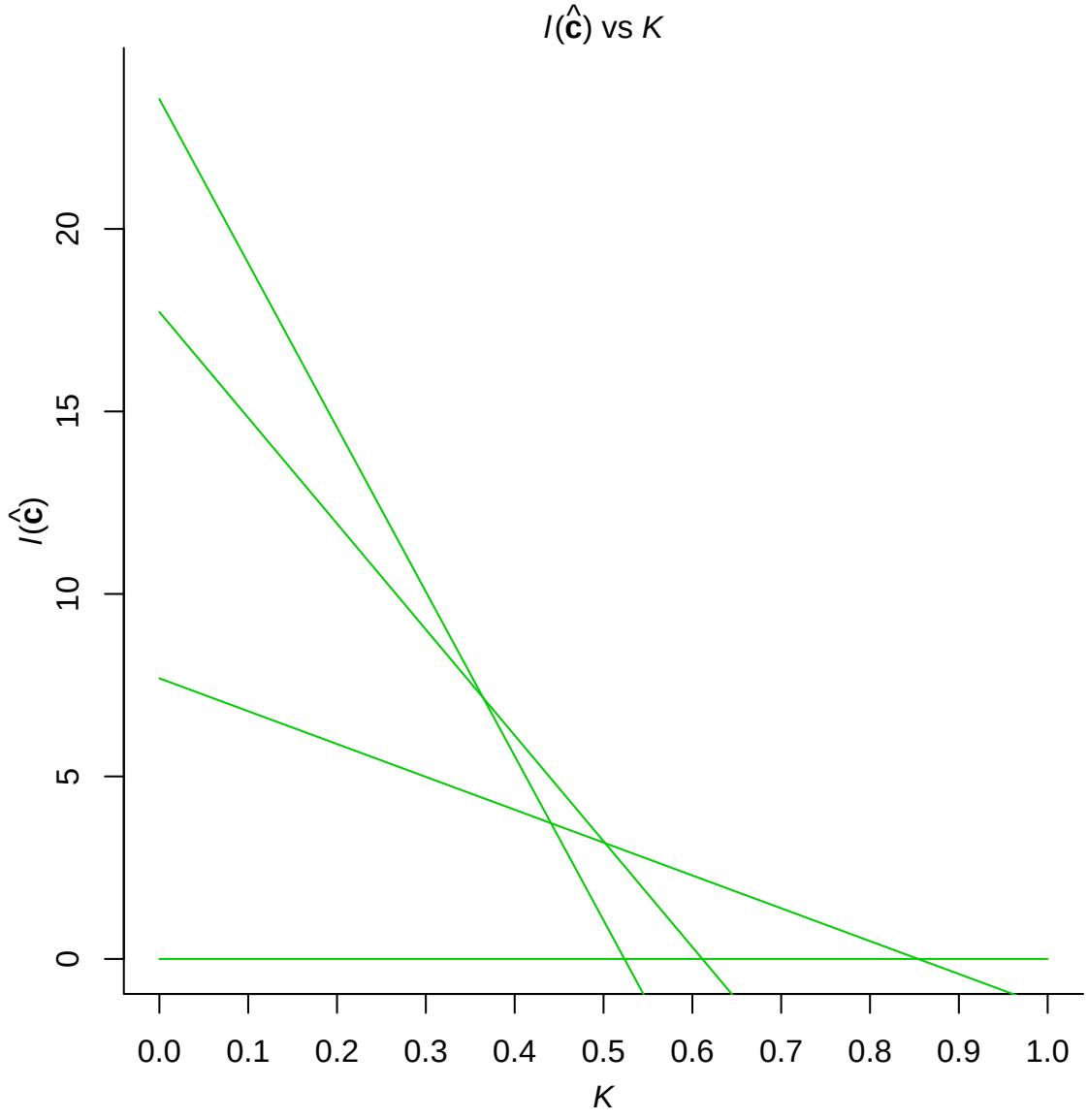
A 10-item example



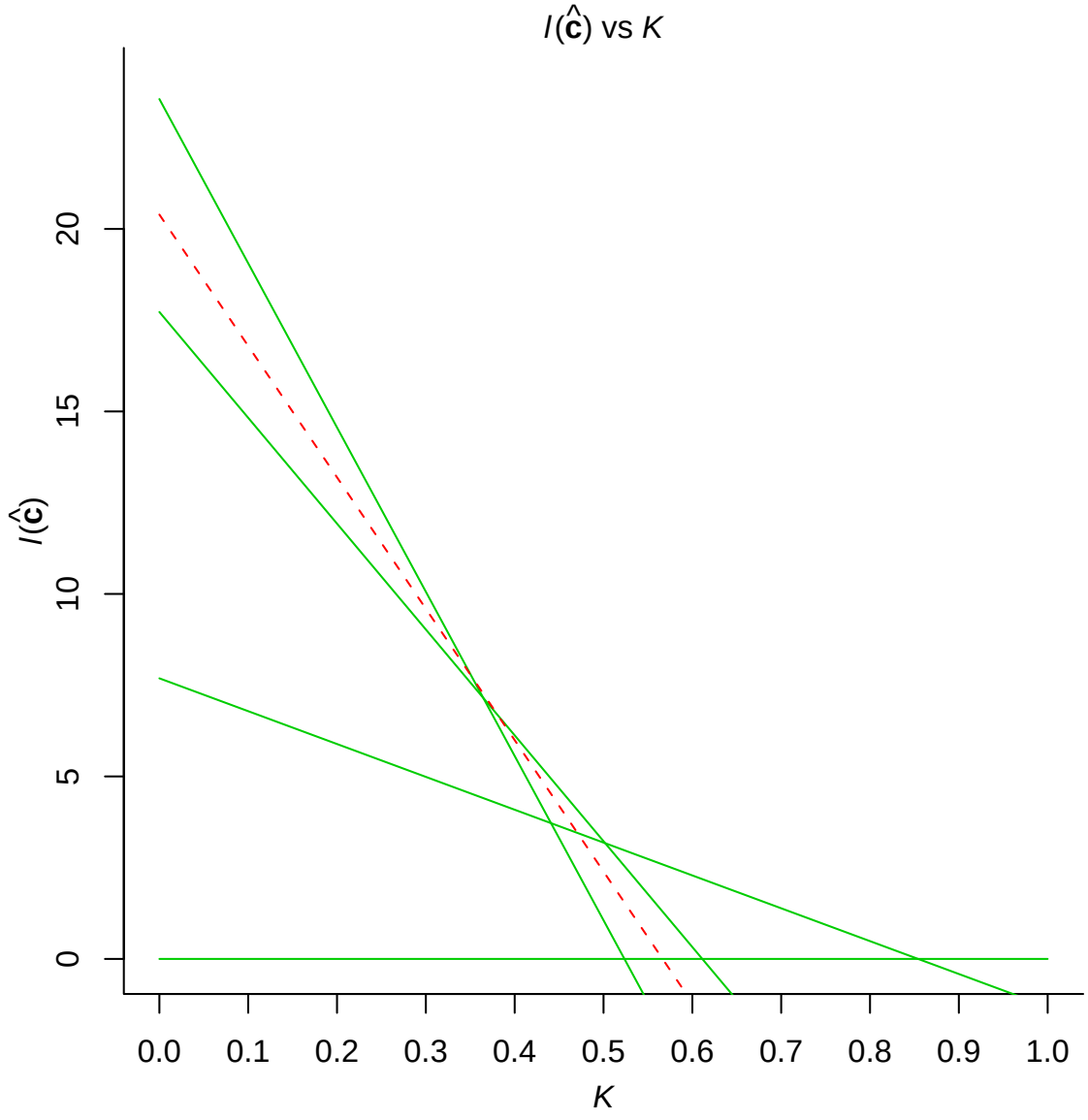
A 10-item example



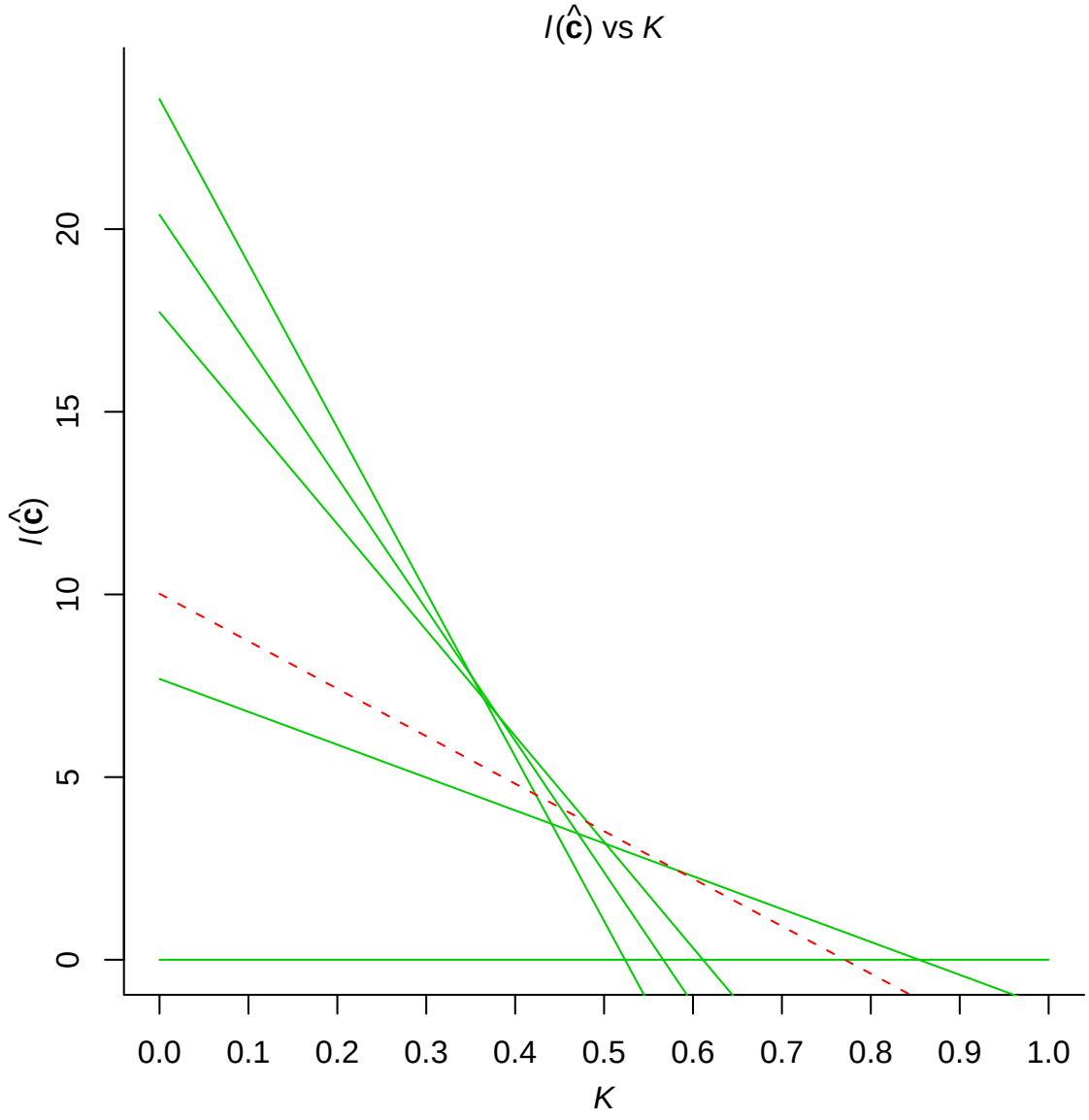
A 10-item example



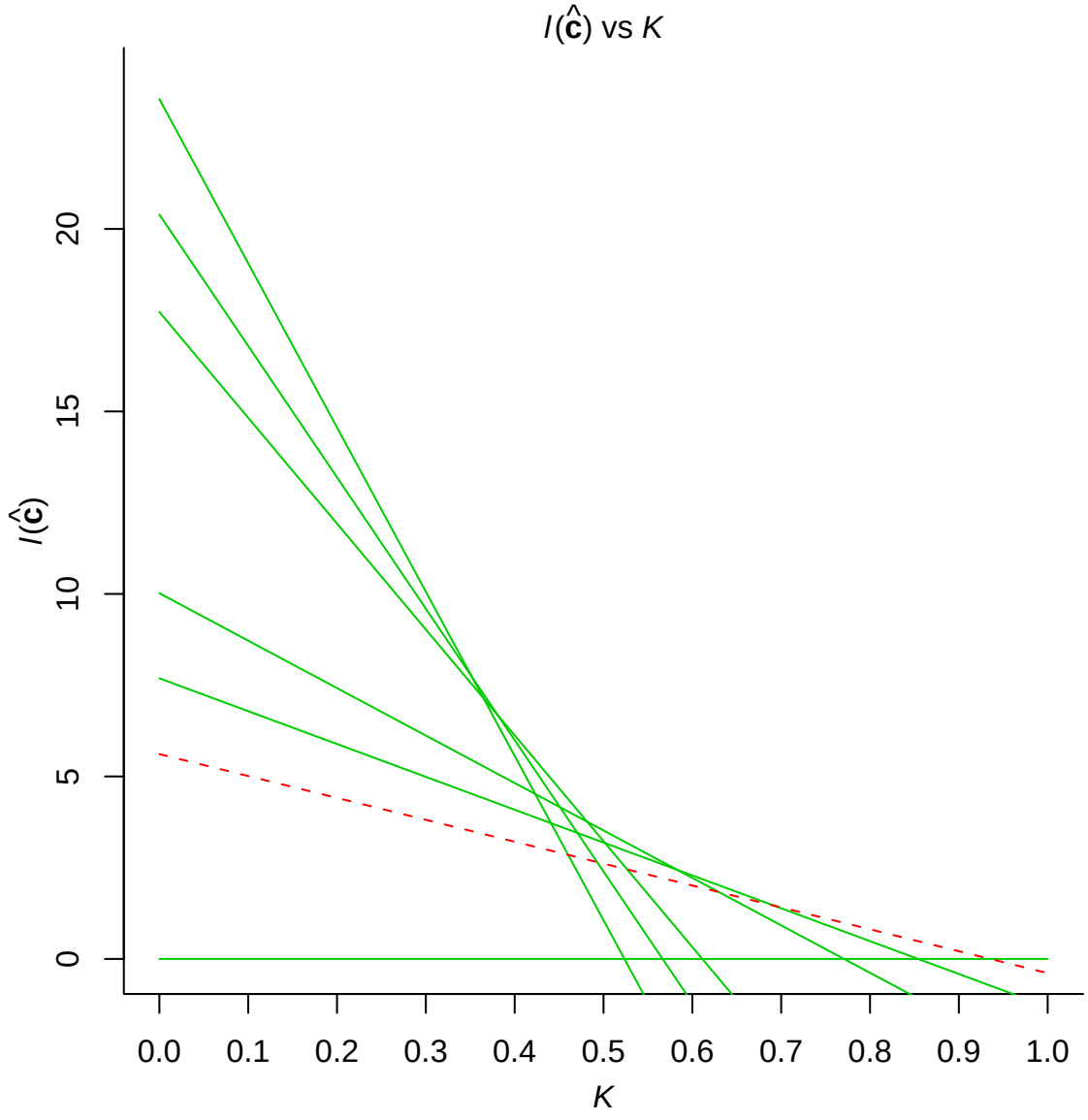
A 10-item example



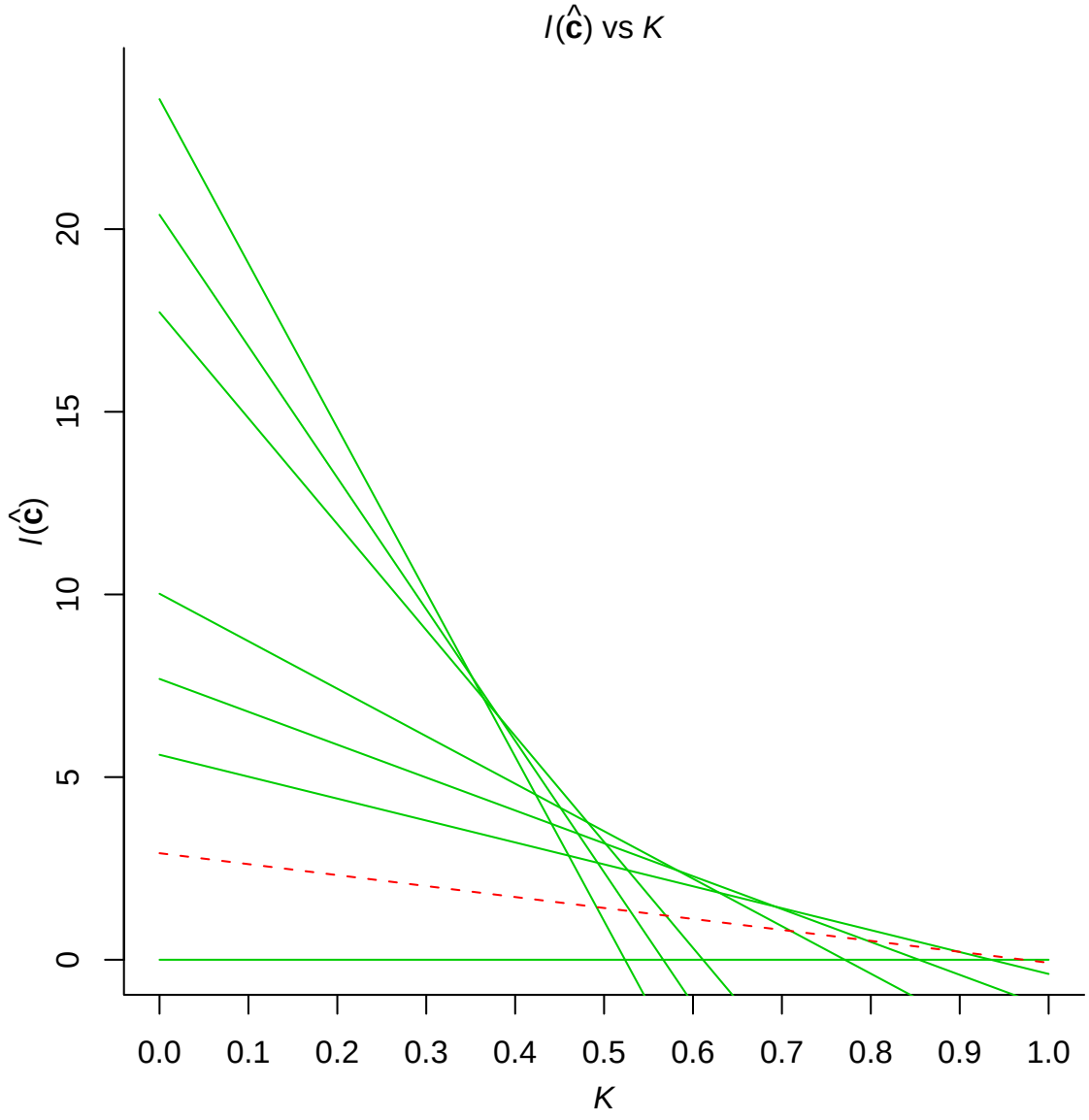
A 10-item example



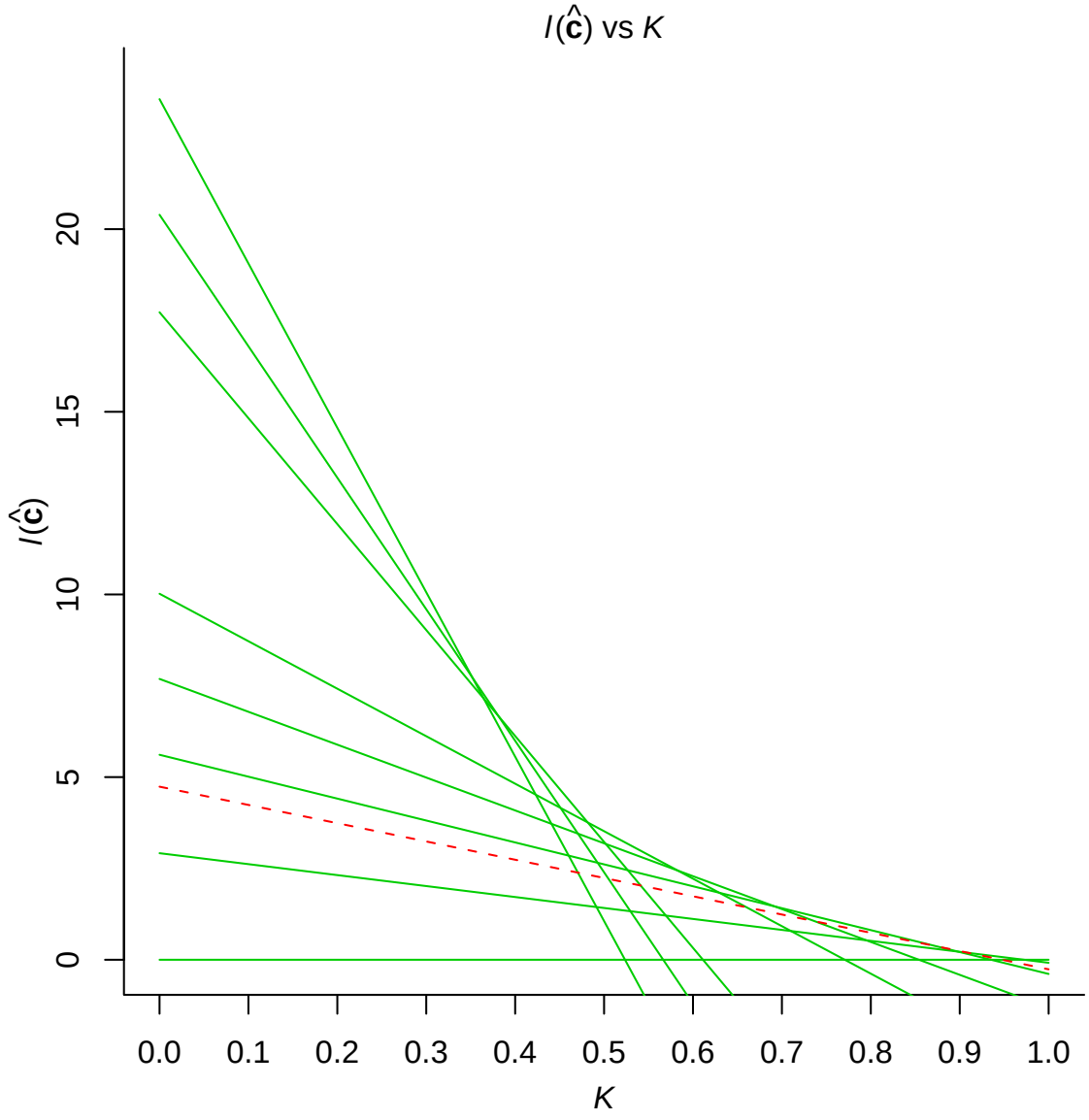
A 10-item example



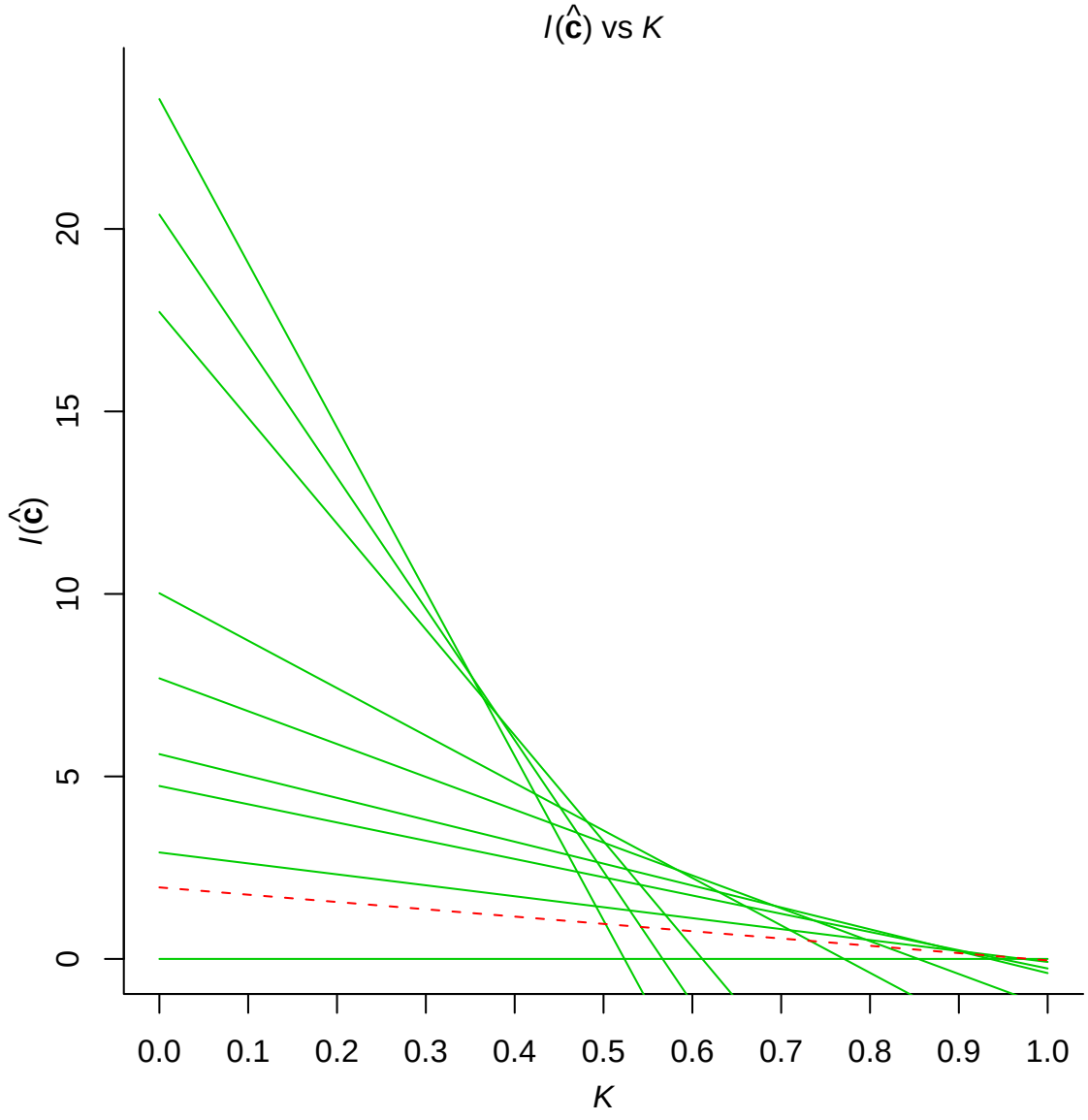
A 10-item example



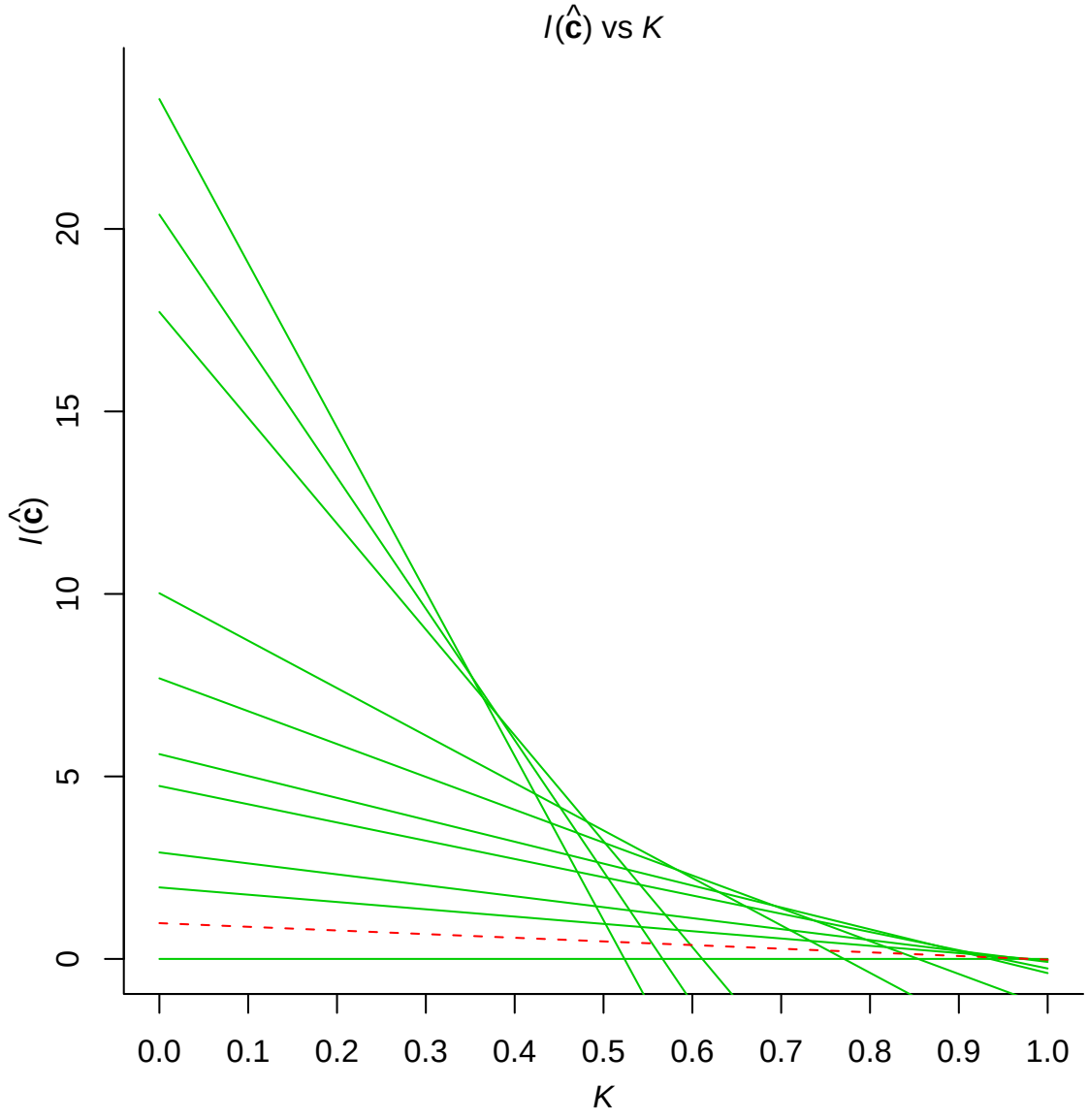
A 10-item example



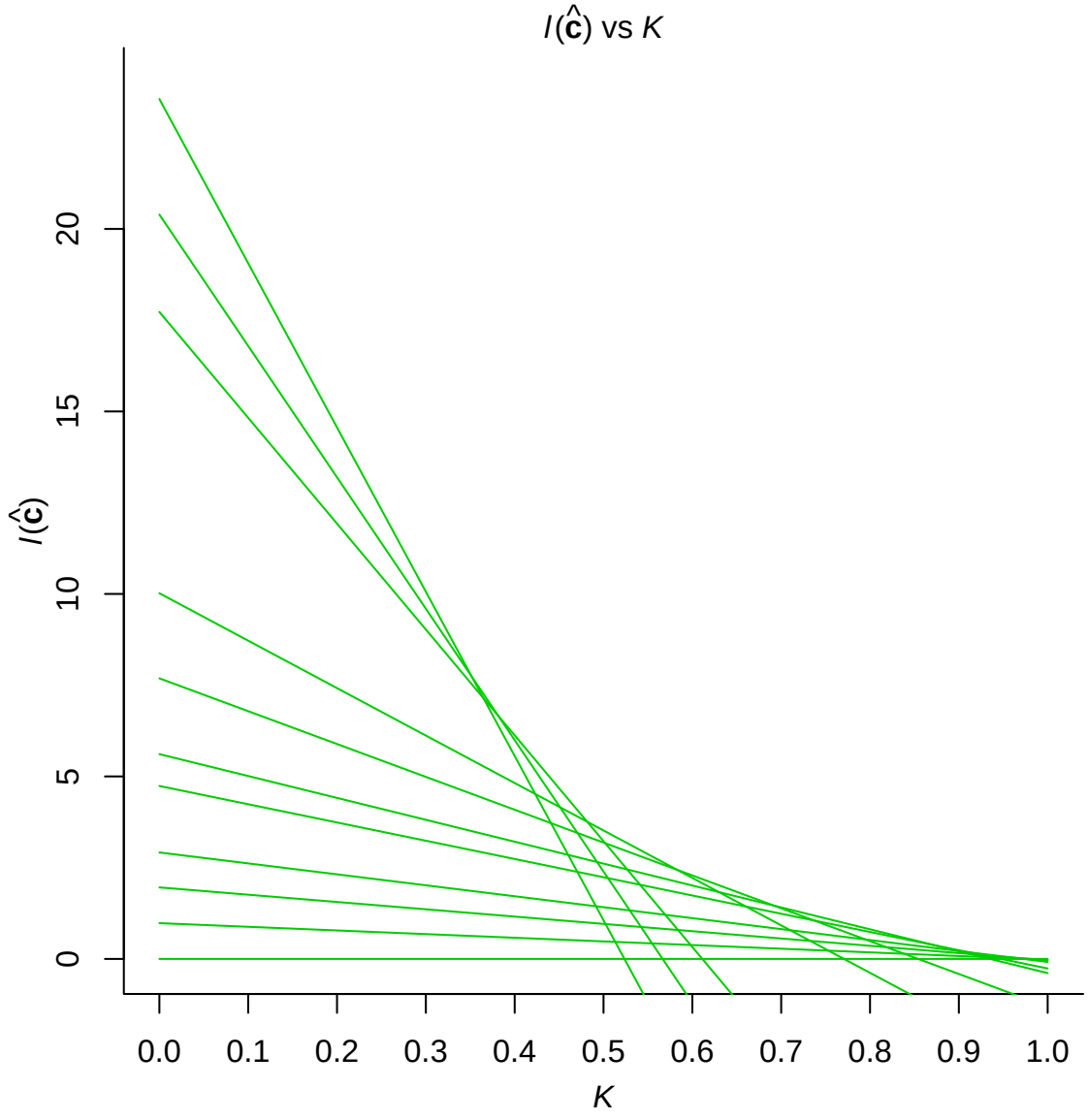
A 10-item example



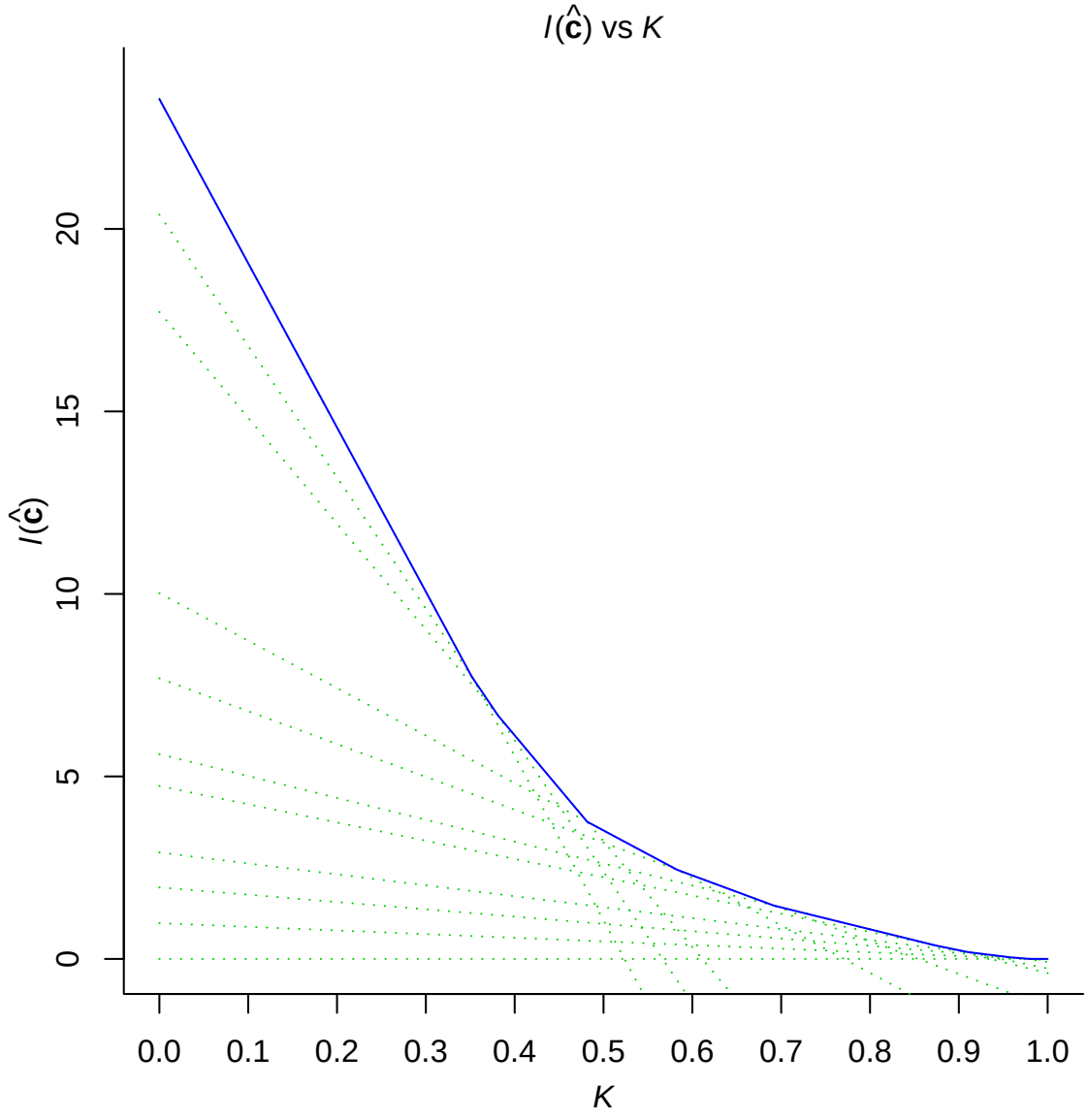
A 10-item example



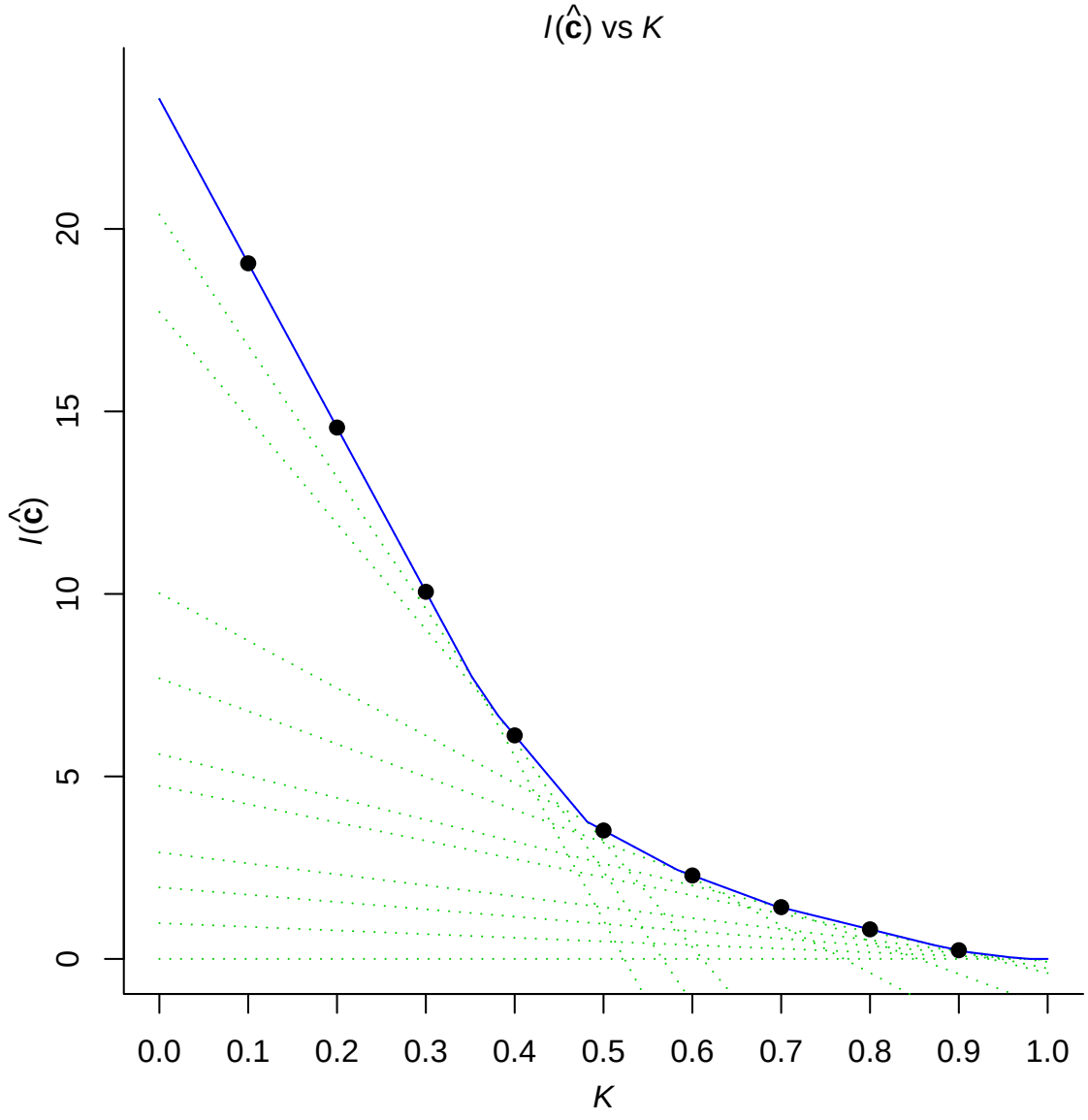
A 10-item example



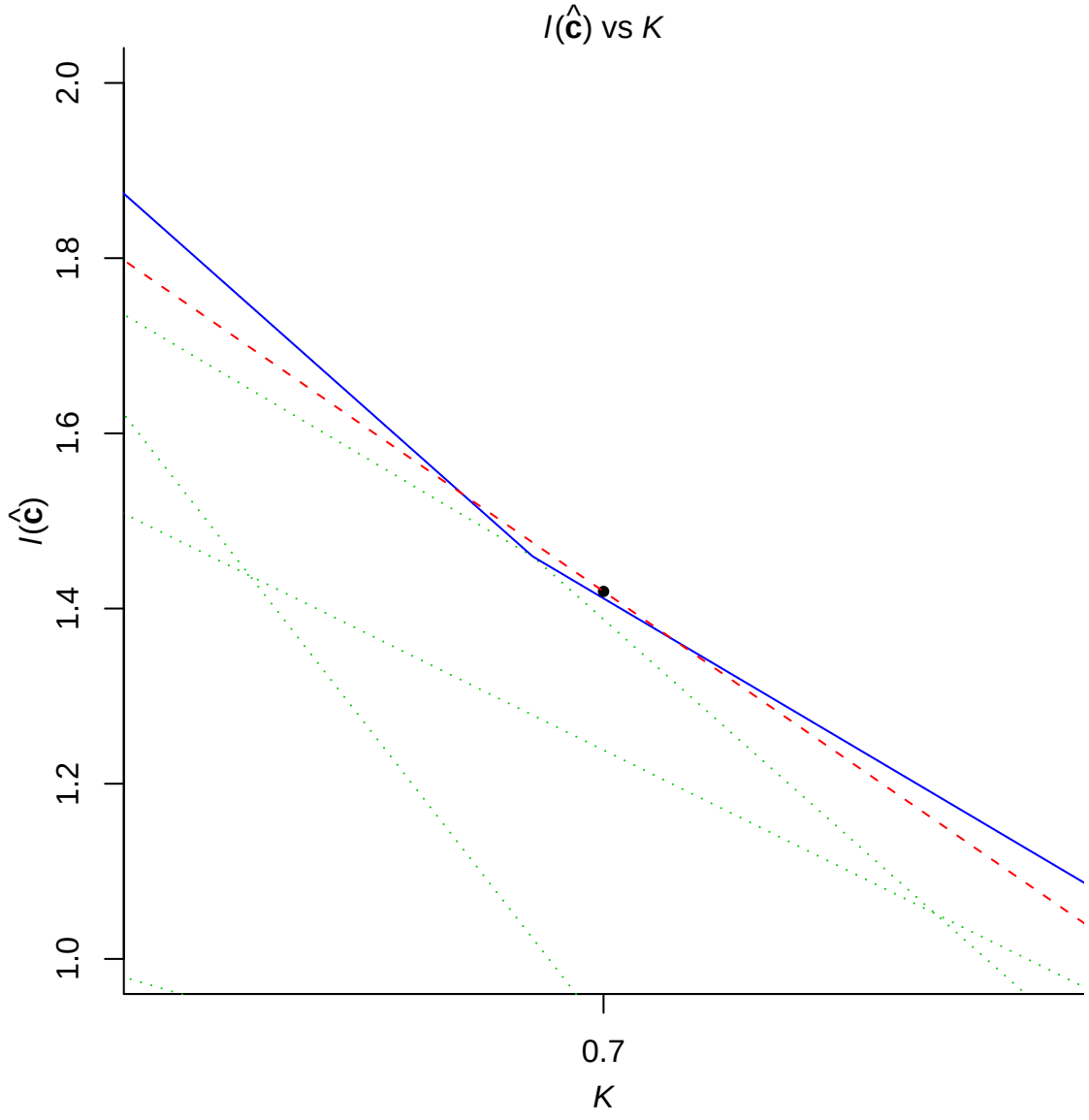
A 10-item example



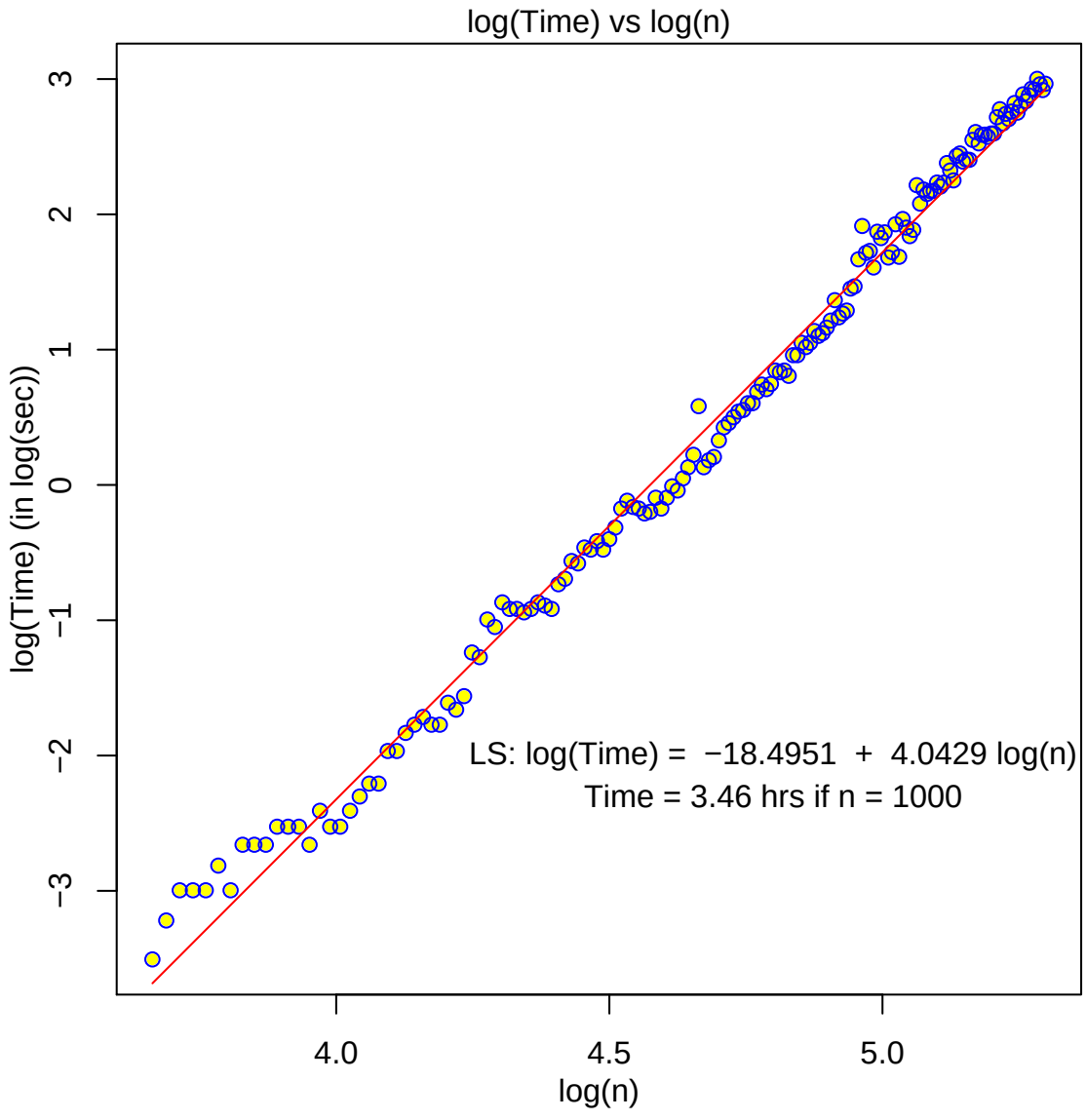
A 10-item example



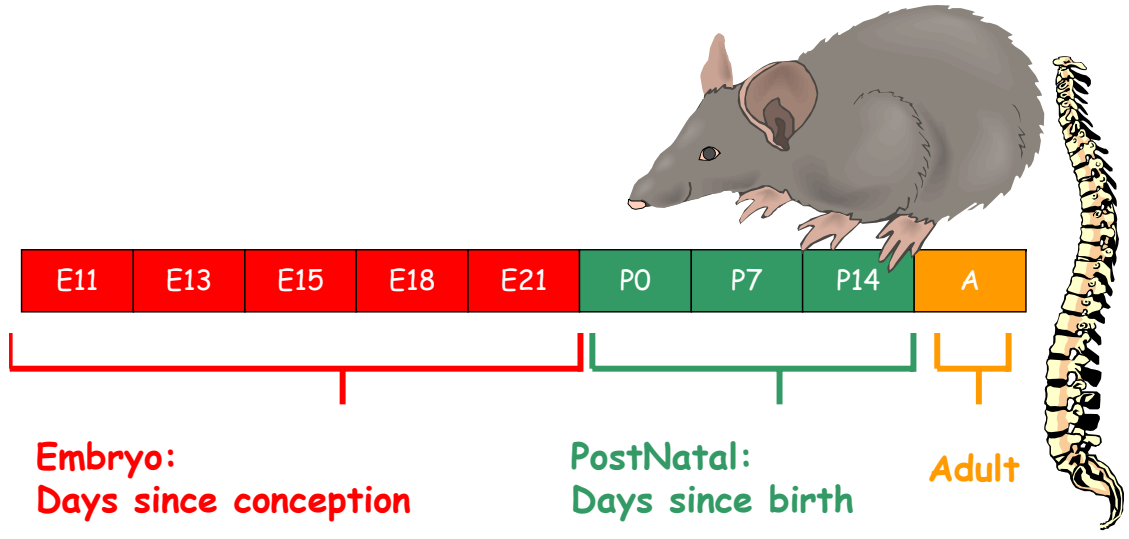
A 10-item example



Now we could afford to cluster 1000 items

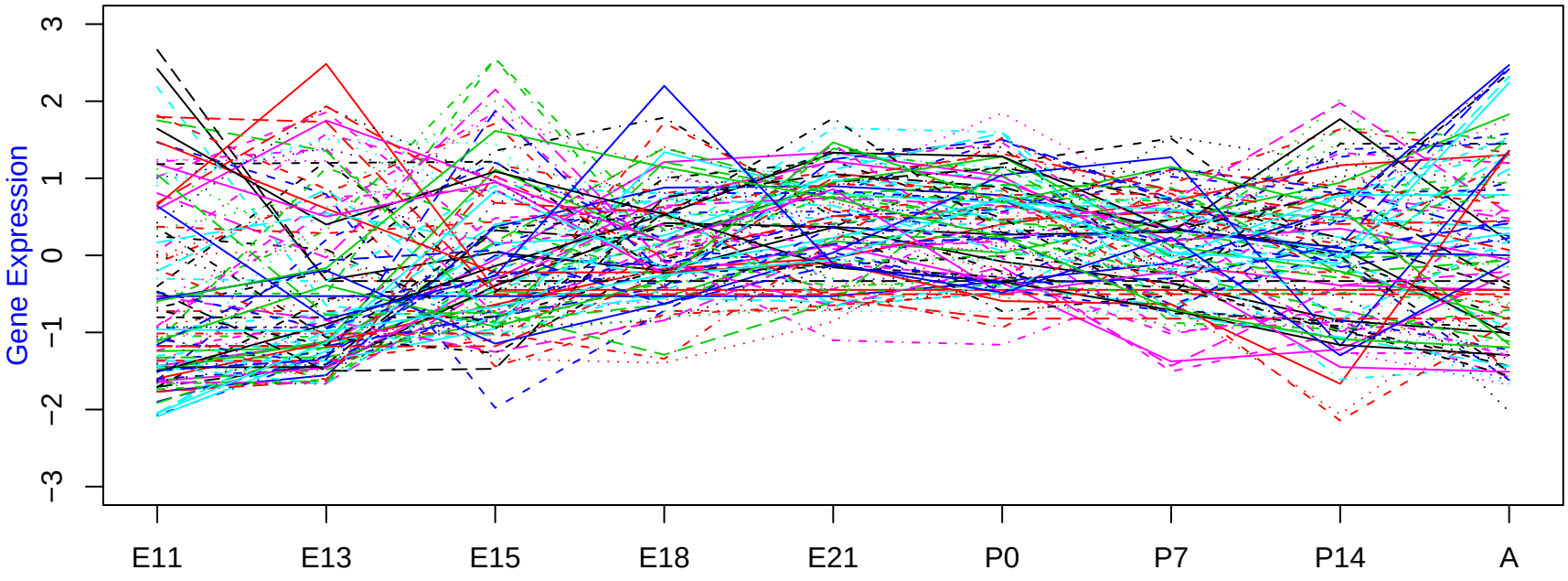


Rats CNS development



Wen et al (*PNAS*, 1998) studied development of central nervous system in rats: mRNA expression levels of 112 genes at 9 time points.

Rats data, normalised

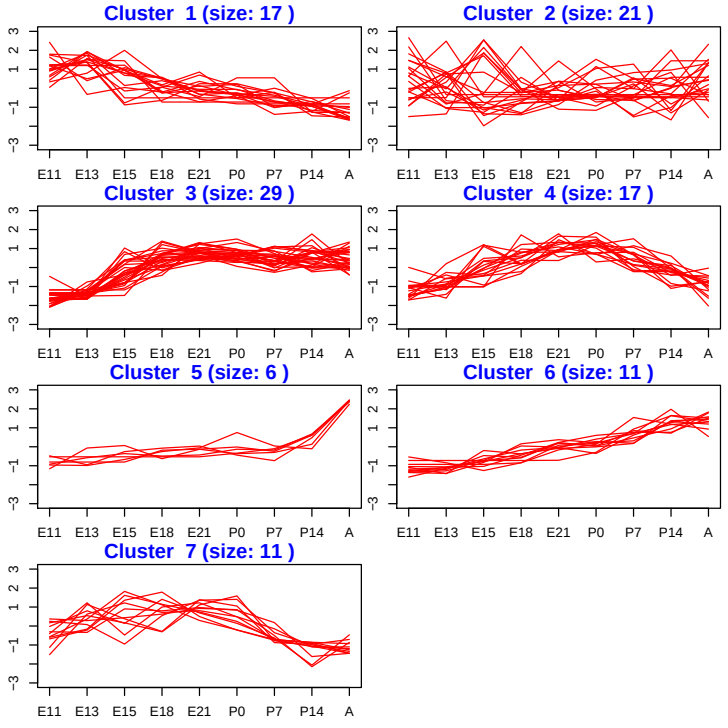


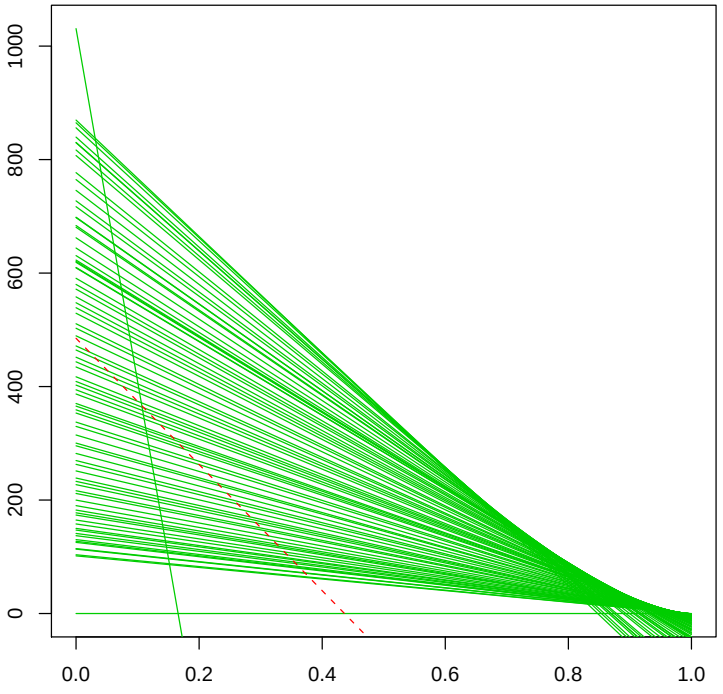
Rats: `stage+stage:day` model

Piecewise linear time dependence:

$$X = \begin{pmatrix} 1 & 11 & 0 & 0 & 0 \\ 1 & 13 & 0 & 0 & 0 \\ 1 & 15 & 0 & 0 & 0 \\ 1 & 18 & 0 & 0 & 0 \\ 1 & 21 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 7 & 0 \\ 0 & 0 & 1 & 14 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Rats: `stage+stage:day` model





Wen's partition is substantially worse than optimal for any K.

MAP vs. optimal clustering

How optimal is MAP partition?

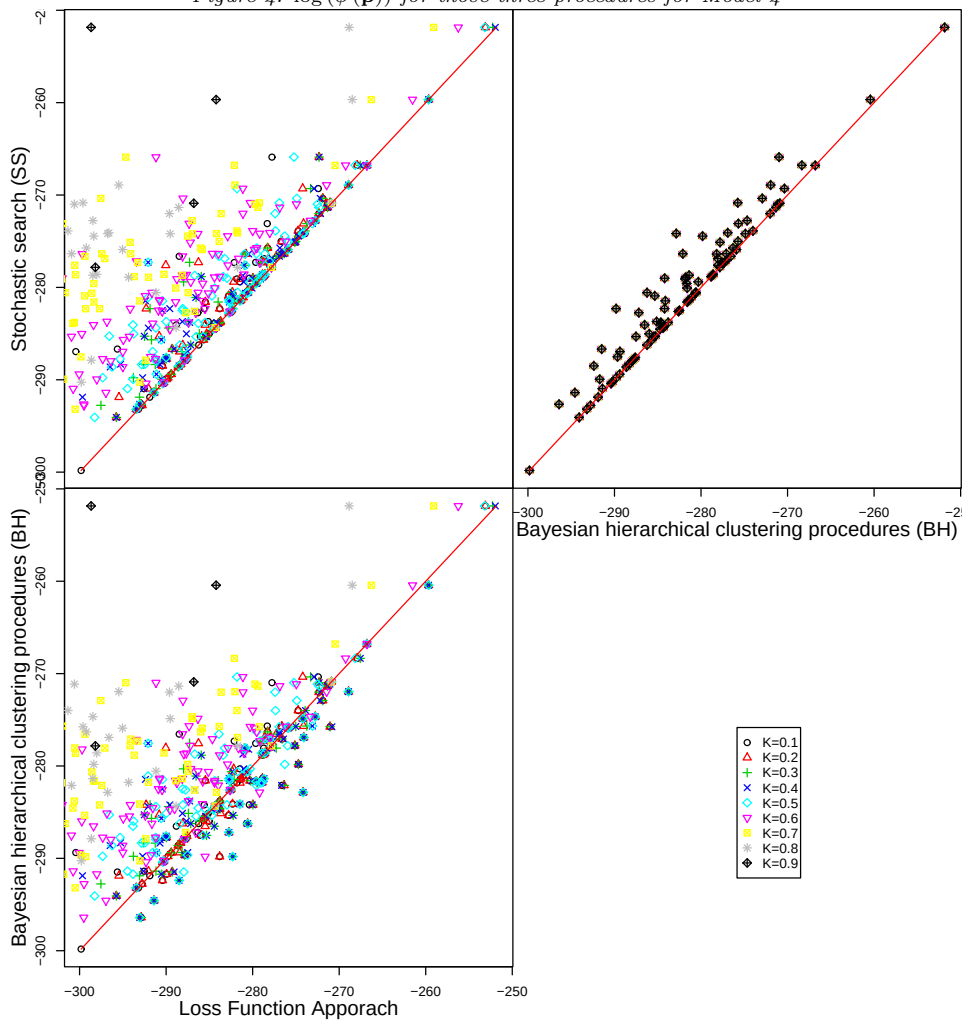
How probable is optimal partition?

Simulation (100 replicates) of samples of size $n = 100$ from 4–component bivariate normal mixture, no covariates, $S = k = 2$. We use DPM prior.

MAP approximated by a naive stochastic search (SS) and by (deterministic) Bayesian hierarchical clustering procedure of Heard, Holmes and Stephens (BH).

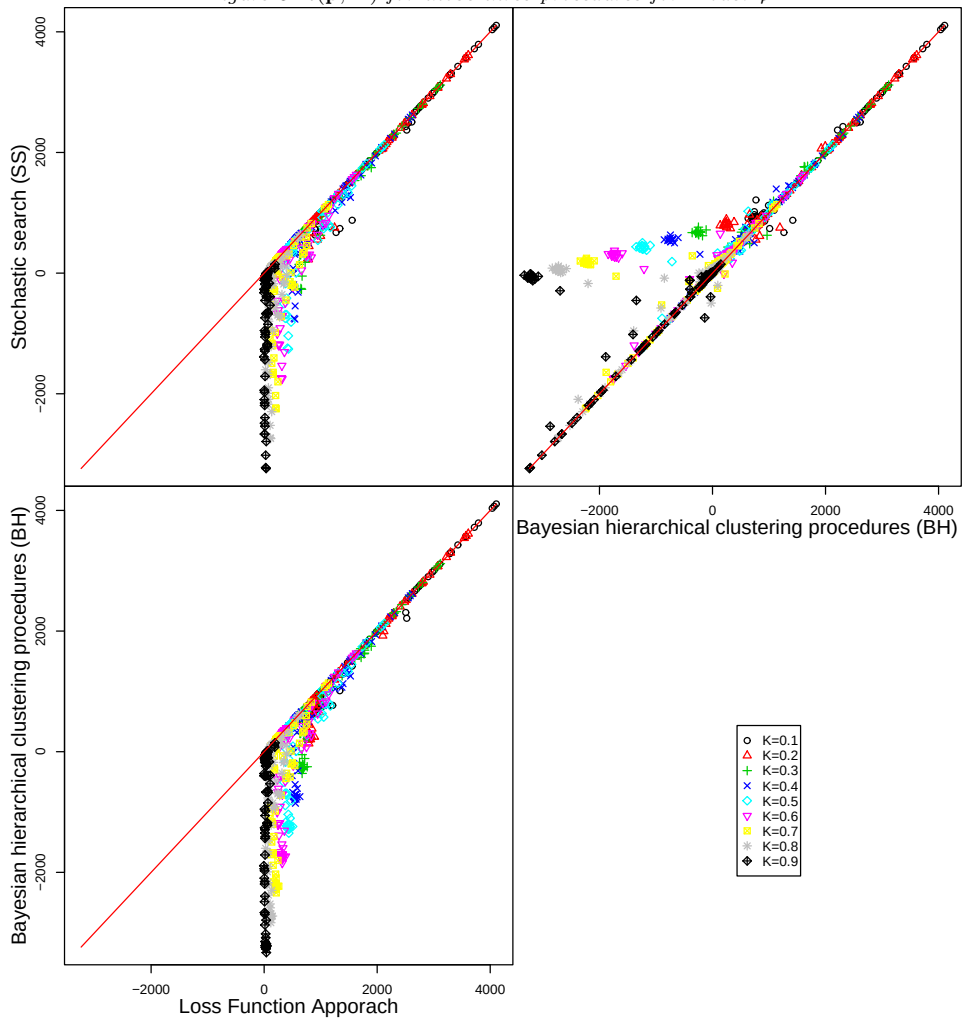
MAP vs. optimal clustering

Figure 4: $\log(\phi(\mathbf{p}))$ *for those three procedures for Model 4*



MAP vs. optimal clustering

Figure 8: $\ell(\mathbf{p}, K)$ for those three procedures for Model 4



Summary

- flexible model that combines
  - parametric dependence on condition-specific covariates
  - non-parametric clustering of genes, allowing baseline category
- conjugate specification greatly facilitates computation
- wider applicability of 'incremental' samplers
- possibility to approximate optimal clustering for certain loss functions