A general quadratic programming method for the optimisation of genetic contributions using interior point algorithm

R Pong-Wong & JA Woolliams

Introduction

- Inbreeding is a risk and it needs to be controlled

- Optimum contribution selection is an effective tool to control inbreeding (in directional selection or conservation schemes)
  - But… practical intake is low

- Methods need to be improved to exploit dense SNP genotyping
  - Coancestry at different genomic regions constrained separately rather than average

Methods for Optimising Contribution Selection (OCS)

- Relaxed Parameter Space (AKA Langrange multiplier method)
  - Meuwissen, 1997, Grundy et al, 1998
- Evolutionary algorithms
  - Kinghorn et al, 2002
- Semidefinite programming
  - Pong-Wong and Woolliams, 2008

Objective

Propose a new method for OCS

formulated as a quadratic programming

which can accommodate for multiple constraints on coancestry

ROSLIN
THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies
BBSRC
bioscience for the future

Genetic Contribution

- Genetic contribution (c_i):
  - Proportion of genetic material an ancestor *i* passes to the descendant population
  - $c_i \alpha$ number of offspring parent *i* has.

- Expected genetic gain and Inbreeding are functions of genetic contributions
  - g = **c'e** — **e** = ebv
  - F = **c'Gc/2** — **G** = genetic relationship (NRM/GRM)

The OCS problem

- Optimise contribution of candidates
- Objective:
  - Maximise Genetic Gain
  - (or maximise genetic diversity)
- Constrains
  - Coancestry increases at a pre-set rate
    - One or more coancestry restrictions
  - Valid bound of contributions

$$\mathbf{c}$$

$$\begin{cases} \mathbf{e}'\mathbf{c} \\ \mathbf{c}'\mathbf{G}\mathbf{c}/2 \end{cases}$$

$$\mathbf{c}'\mathbf{G}_j\mathbf{c} \leq 2\mathrm{F}_j^*,\ j = 1, p$$

$$\mathbf{s}'\mathbf{c} = 0.5 \quad \mathbf{d}'\mathbf{c} = 0.5$$

$$\underline{\boldsymbol{u}} \leq \mathbf{c} \leq \overline{\boldsymbol{u}}$$

Conditions for optimality

$$\nabla_{\mathbf{c}} h(\mathbf{c}) - \lambda_s \mathbf{s} - \lambda_d \mathbf{d} - \boldsymbol{\lambda}_{\underline{u}} + \boldsymbol{\lambda}_{\overline{u}} + +\Sigma_{j=1}^{p}(\lambda_j \mathbf{G}_j \mathbf{c}) = \mathbf{0}$$

$$0.5 - \mathbf{s}'\mathbf{c} = 0$$

$$0.5 - \mathbf{d}'\mathbf{c} = 0$$

$$\mathbf{y}_{\underline{u}} - \mathbf{c} + \underline{\boldsymbol{u}} = \mathbf{0}$$

$$\mathbf{y}_{\overline{u}} + \mathbf{c} - \overline{\boldsymbol{u}} = \mathbf{0}$$

$$\Sigma_{j=1}^{p}(y_j + \mathbf{c}'\mathbf{G}_j\mathbf{c} - 2\mathrm{F}_j^*) = 0_j \qquad j = 1, p$$

$$(\lambda_{\underline{u}} * y_{\underline{u}})_i = 0_i \qquad i = 1, n$$

$$(\lambda_{\overline{u}} * y_{\overline{u}})_i = 0_i, \qquad i = 1, n$$

$$(\lambda_j * y_j) = 0_j, \qquad j = 1, p$$

$$(\boldsymbol{\lambda}_{\underline{u}}, \boldsymbol{y}_{\underline{u}}) \geq \mathbf{0} \qquad i = 1, n$$

$$(\boldsymbol{\lambda}_{\overline{u}}, \boldsymbol{y}_{\overline{u}}) \geq \mathbf{0} \qquad i = 1, n$$

$$(\lambda_j, y_j) \geq 0, \qquad j = 1, p$$

λ = Lagrangian multipliers. y = slack variables

Conditions for optimality

$$
R(\theta) = \left\{
\begin{array}{ll}
\nabla_{\mathbf{c}} h(\mathbf{c}) - \lambda_s \mathbf{s} - \lambda_d \mathbf{d} - \boldsymbol{\lambda}_{\underline{u}} + \boldsymbol{\lambda}_{\overline{u}} + + \Sigma_{j=1}^{p} (\lambda_j \mathbf{G}_j \mathbf{c}) & \\
0.5 - \mathbf{s}'\mathbf{c} & \\
0.5 - \mathbf{d}'\mathbf{c} & \\
\mathbf{y}_{\underline{u}} - \mathbf{c} + \underline{\boldsymbol{u}} & \\
\mathbf{y}_{\overline{u}} + \mathbf{c} - \overline{\boldsymbol{u}} & \\
\Sigma_{j=1}^{p} (\boldsymbol{y}_j + \mathbf{c}'\mathbf{G}_j\mathbf{c} - 2\mathrm{F}_j^*) & j = 1, p \\
(\lambda_{\underline{u}} * y_{\underline{u}})_i & i = 1, n \\
(\lambda_{\overline{u}} * y_{\overline{u}})_i & i = 1, n \\
(\lambda_j * y_j) & j = 1, p
\end{array}
\right\} = \{0\}
$$

$$(\lambda_{\underline{u}}, y_{\underline{u}})$$
$$(\lambda_{\overline{u}}, y_{\overline{u}})$$
$$(\lambda_j, y_j)$$

$\theta = (c, \lambda, y)$

The OCS problem

$$
R(\boldsymbol{\theta}) = \left\{ \begin{array}{lr}
\nabla_{\mathbf{c}} h(\mathbf{c}) - \lambda_s \mathbf{s} - \lambda_d \mathbf{d} - \boldsymbol{\lambda}_{\underline{u}} + \boldsymbol{\lambda}_{\overline{u}} + +\Sigma_{j=1}^{p}(\lambda_j \mathbf{G}_j \mathbf{c}) & \\
0.5 - \mathbf{s}'\mathbf{c} & \\
0.5 - \mathbf{d}'\mathbf{c} & \\
\mathbf{y}_{\underline{u}} - \mathbf{c} + \underline{\boldsymbol{u}} & \\
\mathbf{y}_{\overline{u}} + \mathbf{c} - \overline{\boldsymbol{u}} & \\
\Sigma_{j=1}^{p}(\boldsymbol{y}_j + \mathbf{c}'\mathbf{G}_j\mathbf{c} - 2\mathrm{F}_j^*) & j = 1, p \\
(\lambda_{\underline{u}} * y_{\underline{u}})_i & i = 1, n \\
(\lambda_{\overline{u}} * y_{\overline{u}})_i & i = 1, n \\
(\lambda_j * y_j) & j = 1, p
\end{array} \right\} = \{\mathbf{0}\}
$$

OCS=

Find : $\boldsymbol{\theta} = (\mathbf{c}, \boldsymbol{\lambda}, \mathbf{y})$

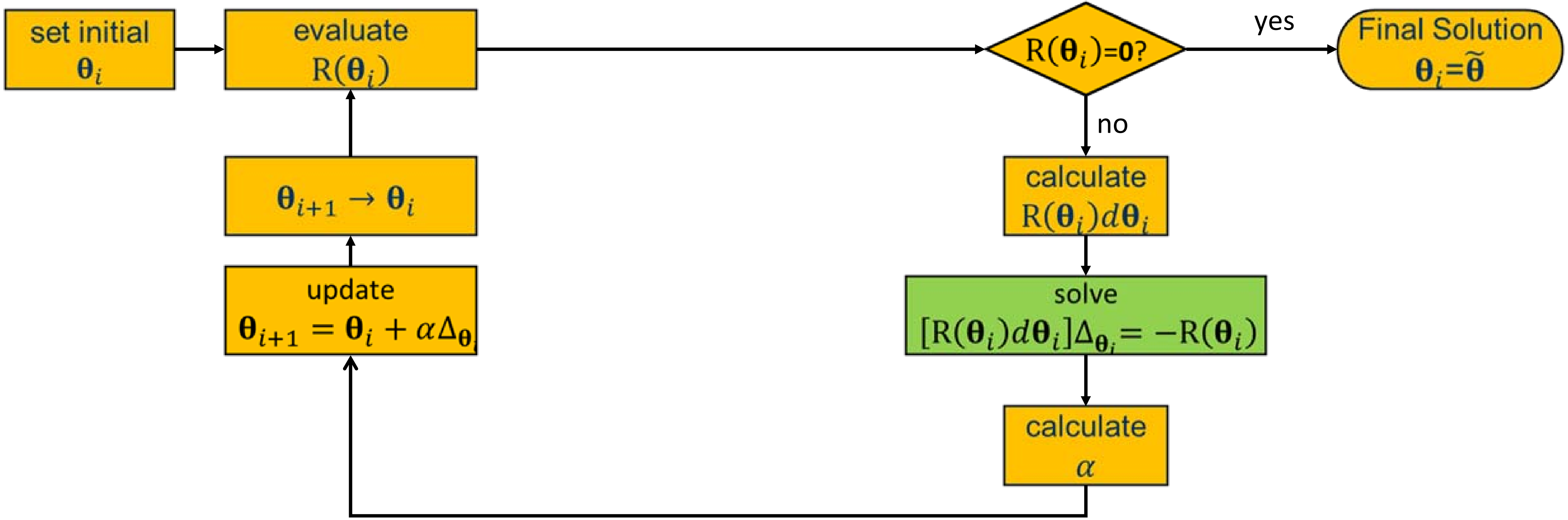
Such as : $R(\boldsymbol{\theta}) = \mathbf{0}$

Solution = root of $R(\boldsymbol{\theta})$

The Newton Rapson method

- $\tilde{\boldsymbol{\theta}}$ is the root of the function $\mathrm{R}(\tilde{\boldsymbol{\theta}})=\mathbf{0}$
- Initial estimate: $\boldsymbol{\theta}_i$
- A better estimate: $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \alpha\Delta_i$
  - $[\mathrm{R}'(\boldsymbol{\theta}_i)d\boldsymbol{\theta}_i]\Delta_i = -\,\mathrm{R}(\boldsymbol{\theta}_i)$
- Repeat until: $\boldsymbol{\theta}_{i+x} = \tilde{\boldsymbol{\theta}}$

The Newton Rapson method



Convergence problems with NR

- Sometimes with NR: $\boldsymbol{\theta}_{i+1}$ is invalid or not better than $\boldsymbol{\theta}_i$

Interior Point algorithm: The Mehrotra's method

- Update of $\boldsymbol{\theta}$ using a perturbed Newton Rapson Step
- $[\mathrm{R}'(\boldsymbol{\theta}_i) d\boldsymbol{\theta}_i]\Delta_i = (-\mathrm{R}(\boldsymbol{\theta}_i) + R^*)$
  - R^* = central path

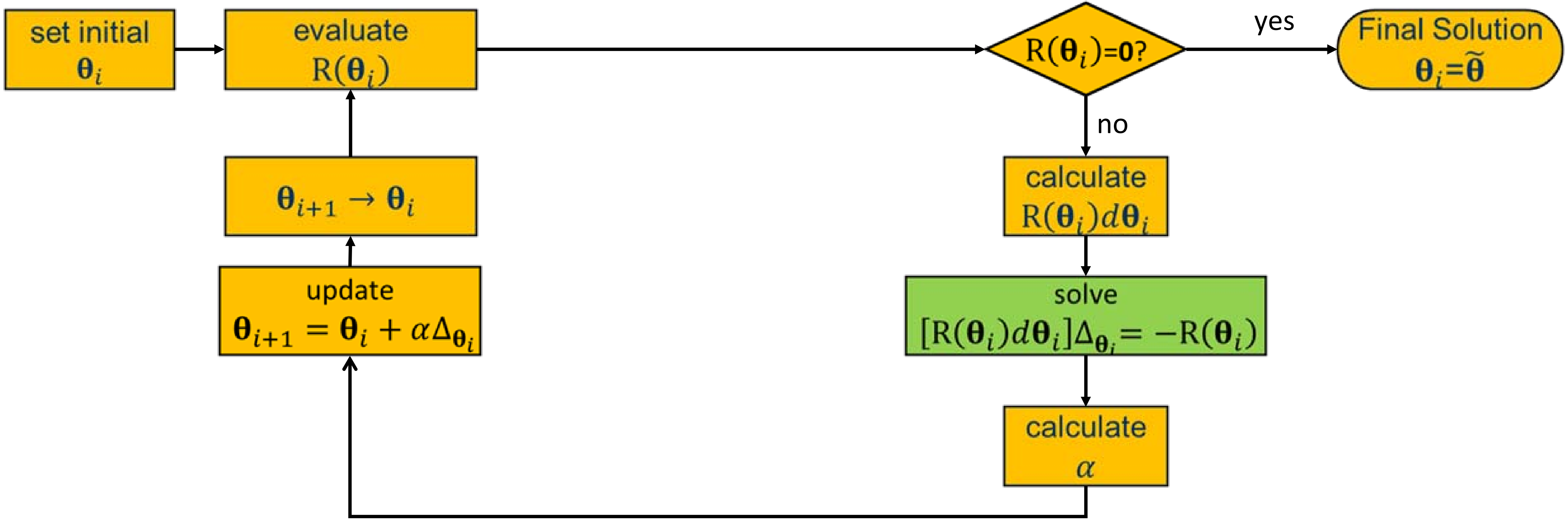
R(θ) and R*

$$R(\boldsymbol{\theta}) = \begin{bmatrix} \nabla_{\mathbf{c}} h(\mathbf{c}) - \lambda_s \mathbf{s} - \lambda_d \mathbf{d} - \boldsymbol{\lambda}_{\underline{u}} + \boldsymbol{\lambda}_{\overline{u}} + + \Sigma_{j=1}^{p} (\lambda_j \mathbf{G}_j \mathbf{c}) & \\ 0.5 - \mathbf{s}'\mathbf{c} & \\ 0.5 - \mathbf{d}'\mathbf{c} & \\ \mathbf{y}_{\underline{u}} - \mathbf{c} + \underline{\boldsymbol{u}} & \\ \mathbf{y}_{\overline{u}} + \mathbf{c} - \overline{\boldsymbol{u}} & \\ \Sigma_{j=1}^{p} (\boldsymbol{y}_j + \mathbf{c}'\mathbf{G}_j\mathbf{c} - 2\mathrm{F}_j^*) & j = 1, p \\ (\lambda_{\underline{u}} * y_{\underline{u}})_i & i = 1, n \\ (\lambda_{\overline{u}} * y_{\overline{u}})_i & i = 1, n \\ (\lambda_j * y_j) & j = 1, p \end{bmatrix} \qquad R^* = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \tau e \\ \tau e \\ \tau e \end{bmatrix}$$

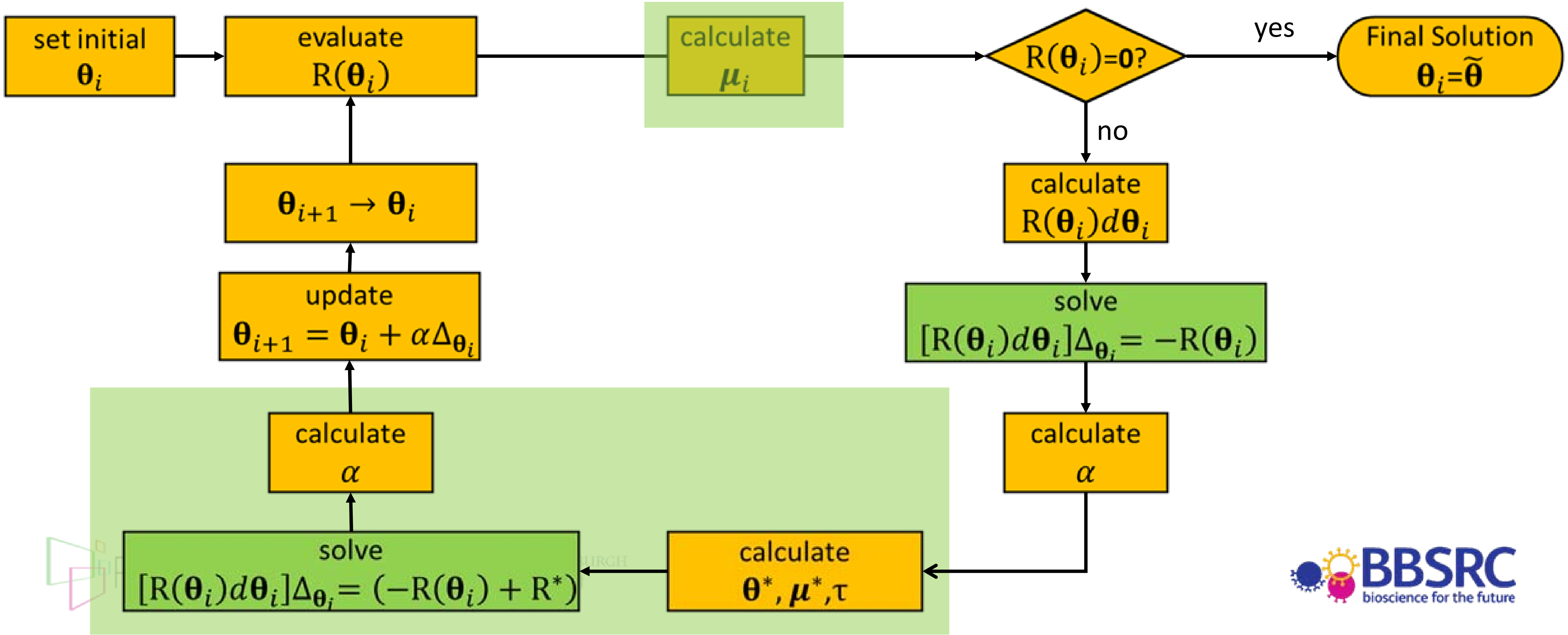
- μ=mean$(\lambda * y)$: $\mu \to 0$, when $\boldsymbol{\theta} \to$ optimum
- $\mu_{i+1} < \mu_i$: $\boldsymbol{\theta}_{i+1}$ is better solution than $\boldsymbol{\theta}_i$
- τ defined by relative value between μ_{i+1} and μ_i

R^* = central path

The Newton Rapson method



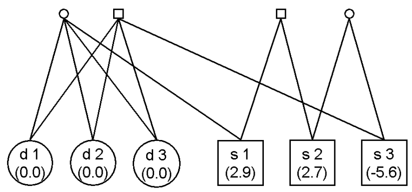
The interior point: The Mehrotra's method



BBSRC
bioscience for the future

Testing the performance of the QP method

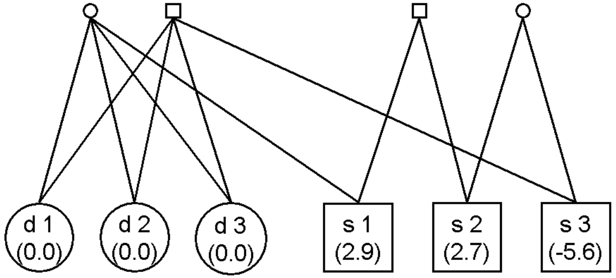
- Small example
  - Example 2 from Pong-Wong and Woolliams (2008)



- Large example
  - 200 candidates
  - GRMs on each chromosome
  - constrains on coancestry of multiple genomic regions

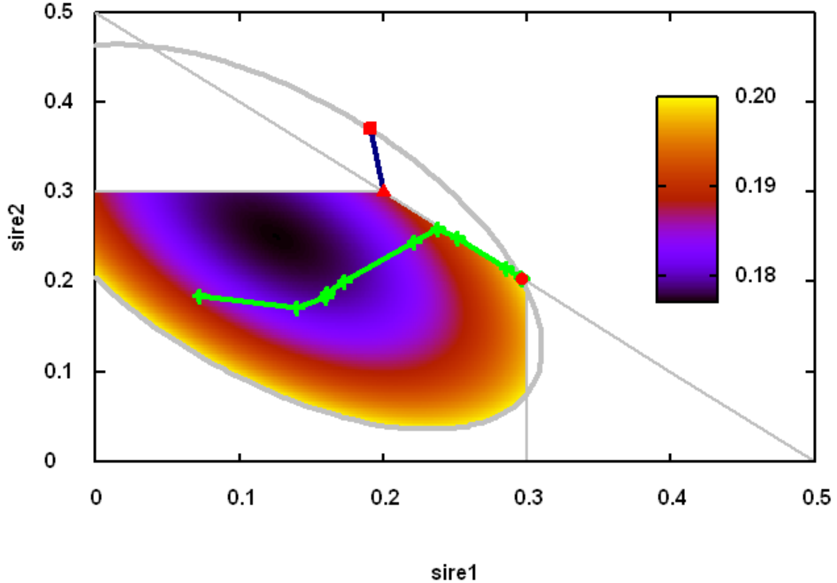
Results compared with SDP

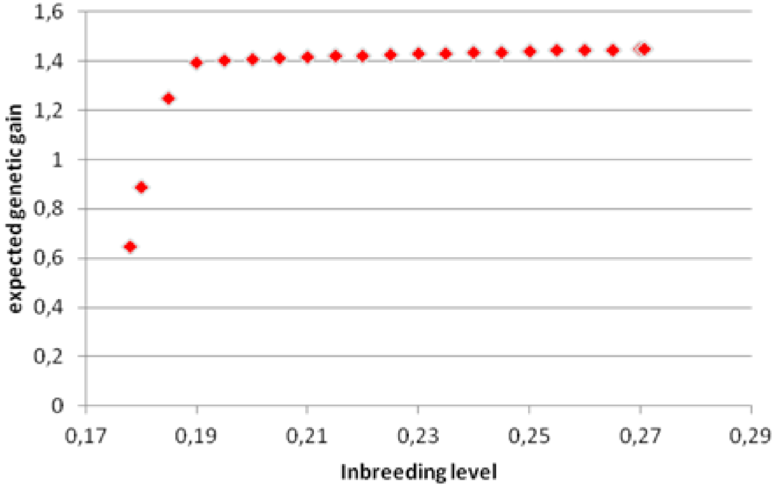
Small example



RPS 1.39

SDP 1.41





QP= SDP

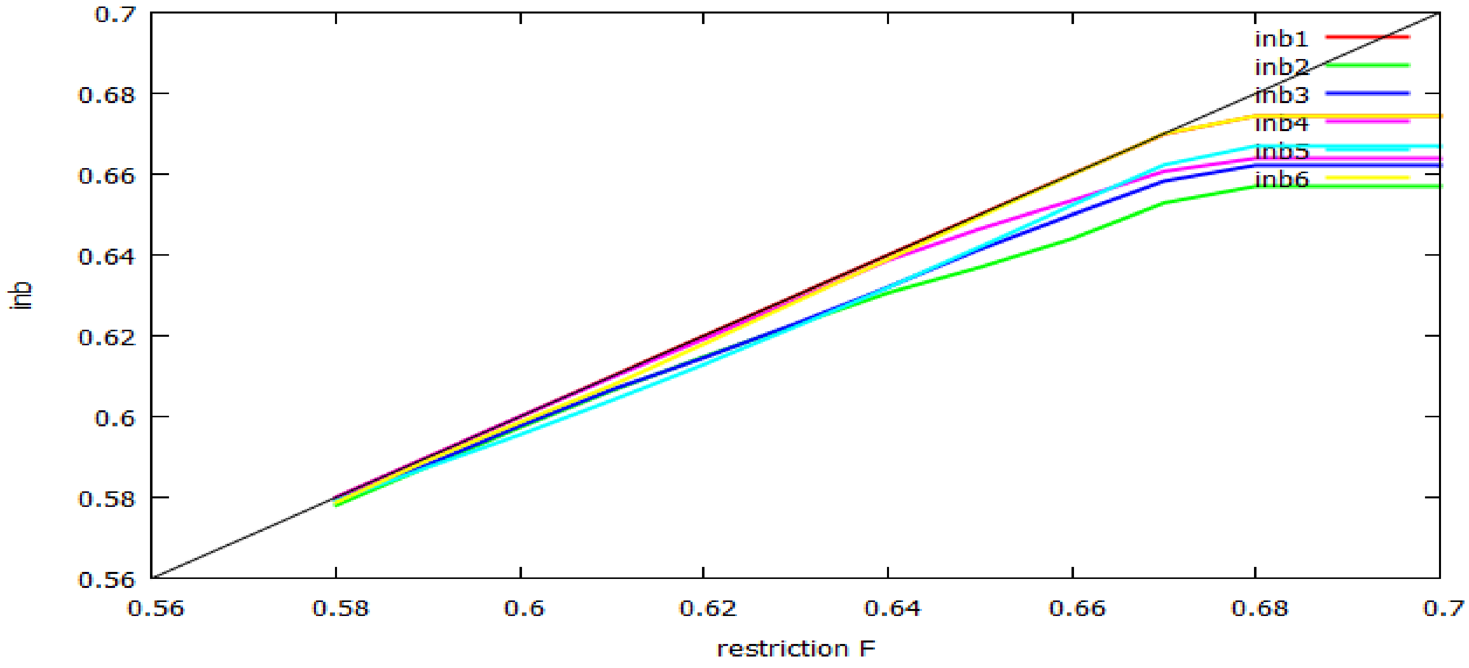
QP finds the true optimum solution

Large Example: two restrictions on coancestry

Restriction on F*		Results in optimum solution: Observed F		Genetic gain
chr 1	**chr 2**	**chr 1**	**chr 2**	
0.63	**0.58**	0.59	0.58	1.85
0.64	**0.59**	0.60	0.59	1.92
0.65	**0.60**	0.61	0.60	1.97
0.66	**0.61**	0.62	0.61	2.01
0.67	**0.62**	0.63	0.62	2.04
0.68	**0.63**	0.64	0.63	2.07
0.69	**0.64**	0.66	0.64	2.09
0.70	**0.65**	0.67	0.65	2.10
0.58	0.63	0.58	0.58	1.82
0.59	0.64	0.59	0.59	1.89
0.60	0.65	0.60	0.60	1.94
0.61	0.66	0.61	0.61	1.98
0.62	0.67	0.62	0.61	2.02
0.63	0.68	0.63	0.62	2.04
0.64	0.69	0.64	0.63	2.06
0.65	0.70	0.65	0.64	2.08

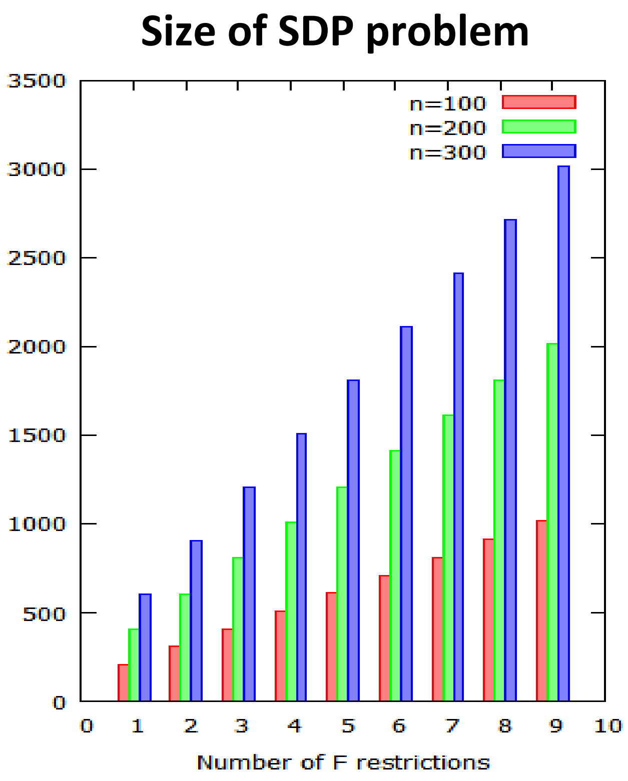
Solutions are the same as SDP method

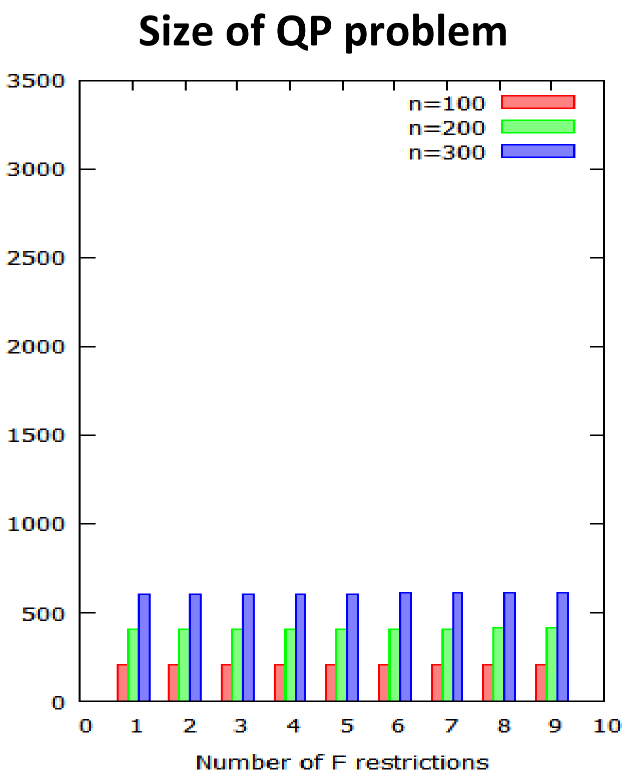
Large Example: Six coancestry restrictions



Final solutions always fulfill all six restrictions on coancestry

Effect of number of coancestry constraints on the size of the problem



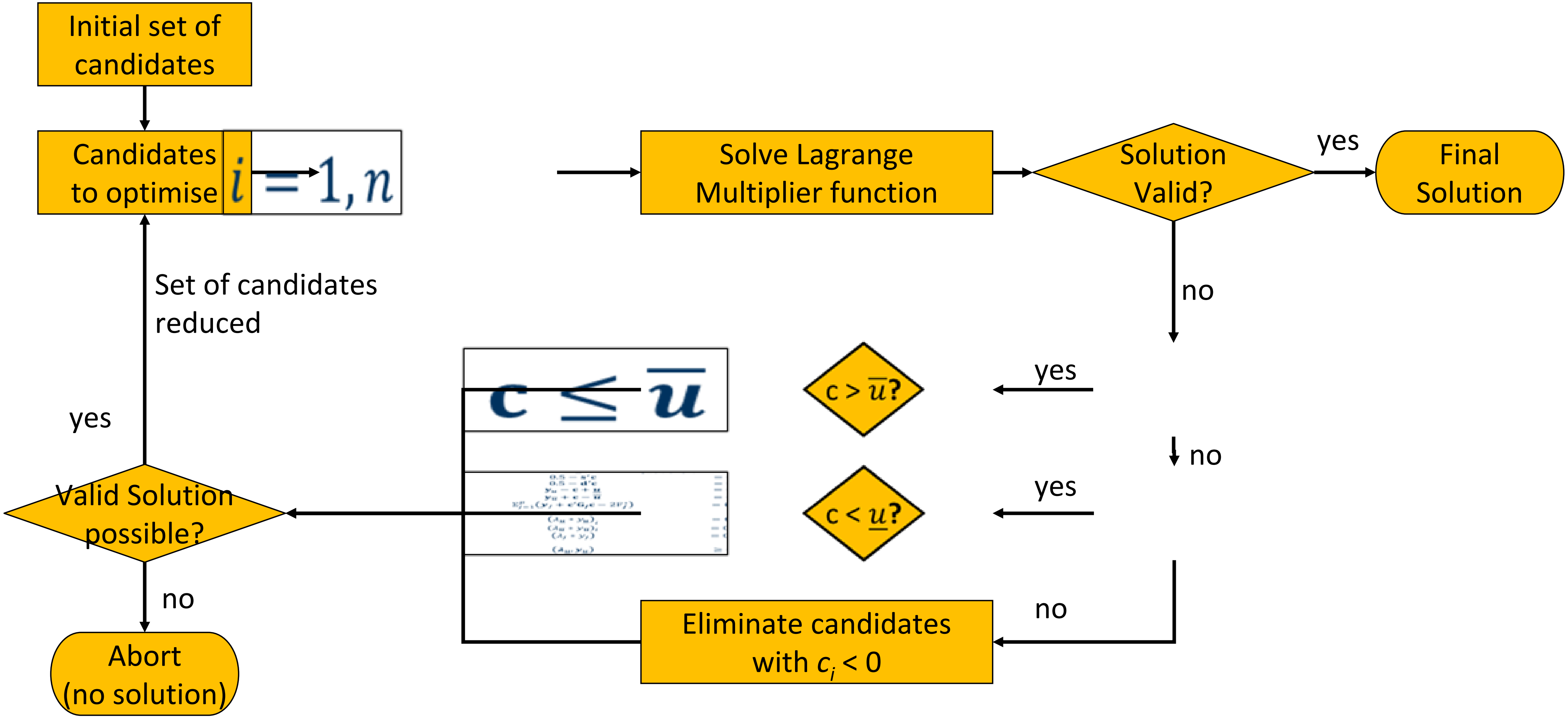


SDP increases by n (number of candidates)
QP increases by 1

Conclusions

- A OCS method formulated as quadratic programming
- Allow the inclusion of several restrictions on coancestry
- Like SDP, it guarantees that results are optimum
  - – but expected to be more computationally efficient

RPS (AKA Langrange multiplier method)



Semidefinite programming method

$$\textbf{Min} \quad \mathbf{a}'\mathbf{x}$$
$$\textbf{s.t.} \quad \mathbf{Y}(\mathbf{x}) \geq \mathbf{0}$$

$$\textbf{Min} \quad -\mathbf{c}'\mathbf{e}$$

s.t.

$$\mathbf{Y} = \begin{bmatrix} \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{c} \\ \mathbf{c}^T & 2F^* \end{bmatrix} & & & & & & \\ & [\mathbf{c}^T\mathbf{s} - 0.5] & & & & & \\ & & [-\mathbf{c}^T\mathbf{s} + 0.5] & & & & \\ & & & [\mathbf{c}^T\mathbf{d} - 0.5] & & & \\ & & & & [-\mathbf{c}^T\mathbf{d} + 0.5] & & \\ & & & & & [\mathbf{diag}(\mathbf{c} - \underline{\mathbf{u}})] & \\ & & & & & & [\mathbf{diag}(\bar{\mathbf{u}} - \mathbf{c})] \end{bmatrix} \geq \mathbf{0}$$

Solve it using a general purpose software

ROSLIN

THE UNIVERSITY of EDINBURGH
Royal (Dick) School of Veterinary Studies

BBSRC
bioscience for the future

The derivative

$$
J\left(c,\lambda_s,\lambda_d,\boldsymbol{\lambda}_{\underline{u}},\boldsymbol{\lambda}_{\overline{u}},\lambda_j,\boldsymbol{y}_{\underline{u}},\boldsymbol{y}_{\overline{u}},y_j\right)
= \begin{bmatrix}
\left[\nabla_{\mathbf{c}}(\nabla_{\mathbf{c}}h(\mathbf{c})) + \Sigma_{j=1}^{p}\left(\lambda_j\mathbf{G}_j\right)\right]_{(nxn)} & -\mathbf{s} & -\mathbf{d} & -\mathbf{I}_{(nxn)} & \mathbf{I}_{(nxn)} & \mathcal{H}_{(nxp)} & & & \\
-\mathbf{s}' & 0 & & & & & & & \\
-\mathbf{d}' & & 0 & & & & & & \\
-\mathbf{I}_{(nxn)} & & & 0 & & & \mathbf{I}_{(nxn)} & & \\
\mathbf{I}_{(nxn)} & & & & 0 & & & \mathbf{I}_{(nxn)} & \\
\mathcal{H}'_{(pxn)} & & & & & 0 & & & \mathbf{I}_{(pxp)} \\
 & & & \mathbf{Y}_{\underline{u}(nxn)} & & & \boldsymbol{\Lambda}_{\underline{u}(nxn)} & & \\
 & & & & \mathbf{Y}_{\overline{u}(nxn)} & & & \boldsymbol{\Lambda}_{\overline{u}(nxn)} & \\
 & & & & & \mathbf{Y}_{\mathbf{j}(pxp)} & & & \boldsymbol{\Lambda}_{j(pxp)}
\end{bmatrix}
$$

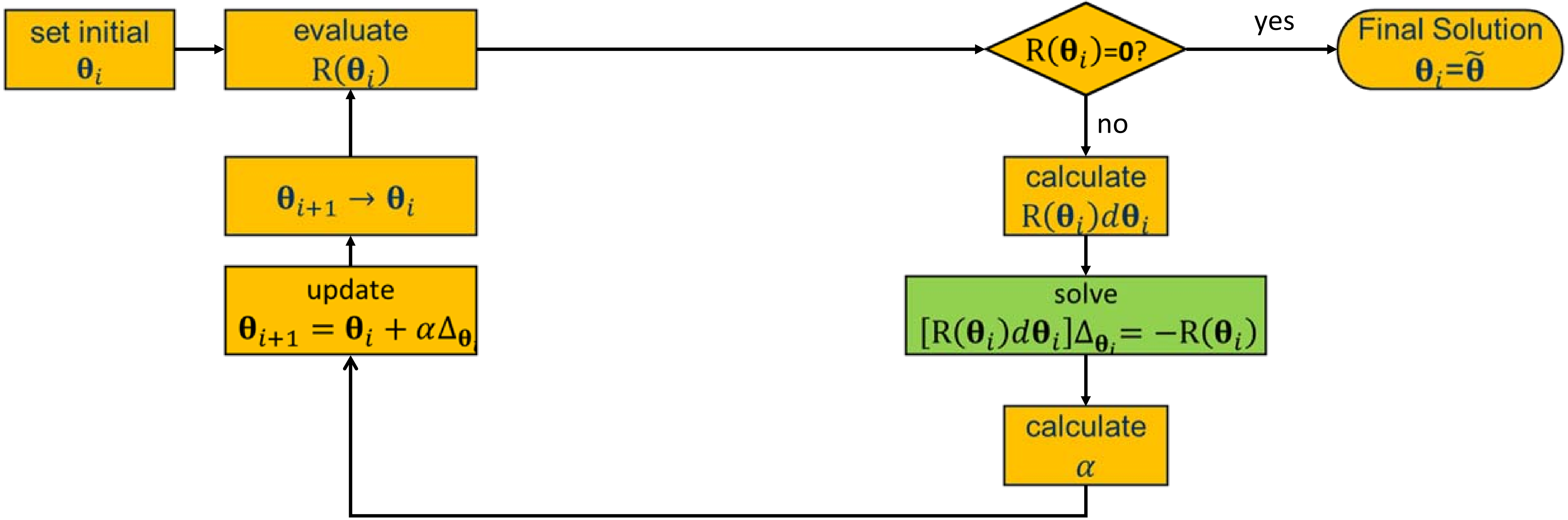
Absorbing the slack variables

$$
\begin{bmatrix}
\nabla^2_{\mathbf{c}}\mathcal{L}(\dots) & -\mathbf{s} & -\mathbf{d} & -\mathbf{I}_{(nxn)} & \mathbf{I}_{(nxn)} & \mathcal{H}_{(nxp)} \\
-\mathbf{s}' & 0 & & & & \\
-\mathbf{d}' & & 0 & & & \\
-\mathbf{I}_{(nxn)} & & & -\boldsymbol{\Lambda}_{\underline{u}}^{-1}\mathbf{Y}_{\underline{u}} & & \\
\mathbf{I}_{(nxn)} & & & & -\boldsymbol{\Lambda}_{\overline{u}}^{-1}\mathbf{Y}_{\overline{u}} & \\
\mathcal{H}'_{(pxn)} & & & & & -\boldsymbol{\Lambda}_{j}^{-1}\mathbf{Y}_{j}
\end{bmatrix}
\begin{bmatrix}
\Delta\mathbf{c} \\
\Delta\lambda_s \\
\Delta\lambda_d \\
\Delta\boldsymbol{\lambda}_{\underline{u}} \\
\Delta\boldsymbol{\lambda}_{\overline{u}} \\
\begin{bmatrix} \Delta\lambda_1 \\ \vdots \\ \Delta\lambda_p \end{bmatrix}
\end{bmatrix}
=
\begin{bmatrix}
\boldsymbol{r}_c \\
r_s \\
r_d \\
\boldsymbol{r}_{\underline{u}} - \boldsymbol{\Lambda}_{\underline{u}}^{-1}\boldsymbol{r}_{\lambda\underline{u}} \\
\boldsymbol{r}_{\overline{u}} - \boldsymbol{\Lambda}_{\overline{u}}^{-1}\boldsymbol{r}_{\lambda\overline{u}} \\
\begin{bmatrix} r_{y1} \\ \vdots \\ r_{yp} \end{bmatrix} - \boldsymbol{\Lambda}_{j}^{-1}\begin{bmatrix} r\lambda_{y1} \\ \vdots \\ r_{\lambda yp} \end{bmatrix}
\end{bmatrix}
$$

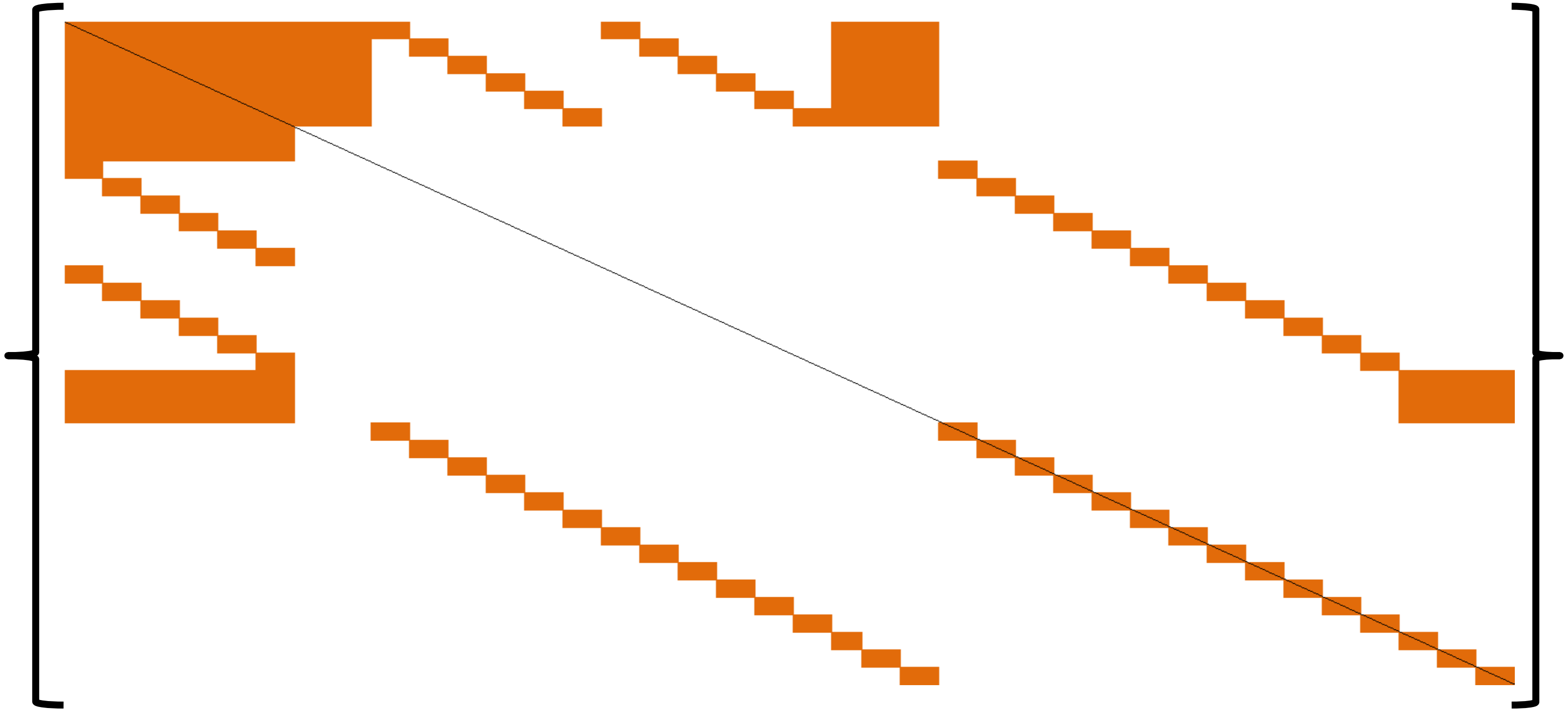
The interior point method

- Newton Rapson can lead to unfeasible solutions
- Keeping to the interior point
- Central path
  - Corrector (how much to move to central path
  - Predictor
- The algorithm

The Newton Rapson method



The Jacobian matrix



Sparse

Indefinite

Non Symmetric

Non-diagonal dominant

Value of τ and the complementarity measure (μ)

- Complementarity measure : μ=mean($\lambda * y$)
  - As $\mu \rightarrow 0$: $\boldsymbol{\theta}$ closer to optimum solution

- $\mu_{i+1} < \mu_i$: $\boldsymbol{\theta}_{i+1}$ is better solution to $\boldsymbol{\theta}_i$

- Size of τ related to relative value between μ_{i+1} and μ_i

- μ=mean($\lambda * y$) $\mu \rightarrow 0$: $\boldsymbol{\theta}$ closer to optimum
- $\mu_{i+1} < \mu_i$: $\boldsymbol{\theta}_{i+1}$ is better solution to $\boldsymbol{\theta}_i$

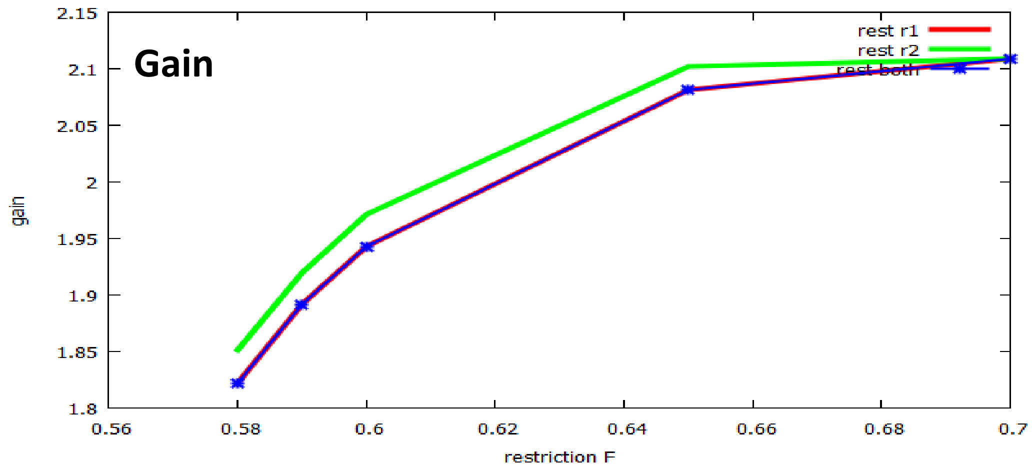
- τ related to relative value between μ_{i+1} and μ_i

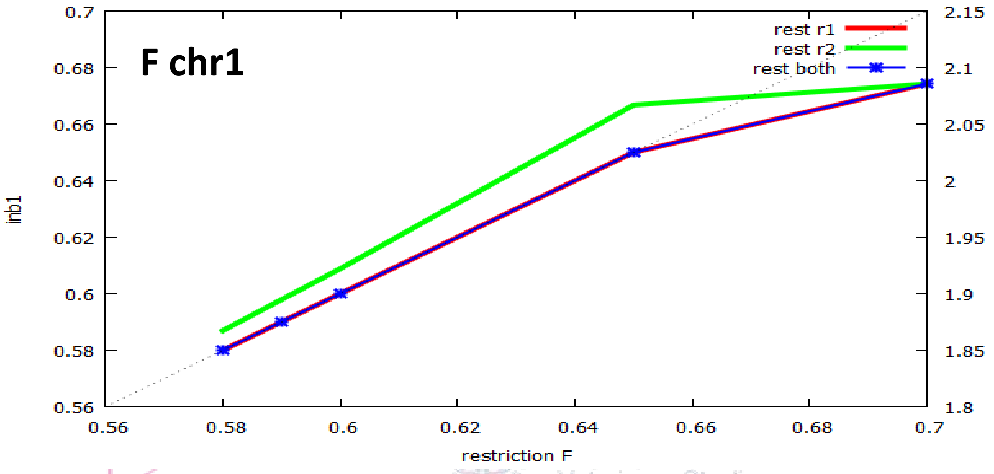
Large Example: Two genomic region of interest

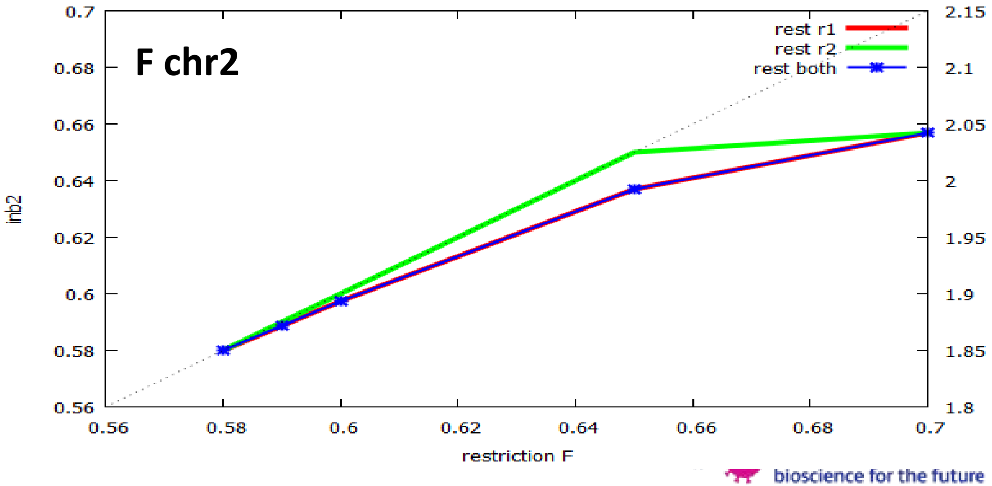
Restricting both regions

Restricting region 1

Restricting region 1







The OCS problem

Min	$h(\mathbf{c})$	$h(\mathbf{c}) = \begin{cases} -\mathbf{e}'\mathbf{c} \\ \mathbf{c}'\mathbf{G}\mathbf{c}/2 \end{cases}$	
s.t.	$\mathbf{c}'\mathbf{G}_j\mathbf{c} \leq 2\mathrm{F}_j^*,\ j = 1, p$	$\mathbf{s}'\mathbf{c} = 0.5$	$\mathbf{c} \geq \underline{\boldsymbol{u}}$
		$\mathbf{d}'\mathbf{c} = 0.5$	$\mathbf{c} \leq \overline{\boldsymbol{u}}$

Lagrange function

$$h(\mathbf{c}) - \lambda_s(\mathbf{s}'\mathbf{c} - 0.5) - \lambda_d(\mathbf{d}'\mathbf{c} - 0.5) - \boldsymbol{\lambda}'_{\underline{u}}(\mathbf{c} - \underline{\boldsymbol{u}}) + \boldsymbol{\lambda}'_{\overline{u}}(\mathbf{c} - \overline{\boldsymbol{u}}) + \sum_{j=1}^{p} \lambda_j\left(\mathbf{c}'\mathbf{G}_j\mathbf{c} - 2\mathrm{F}_j^*\right)$$