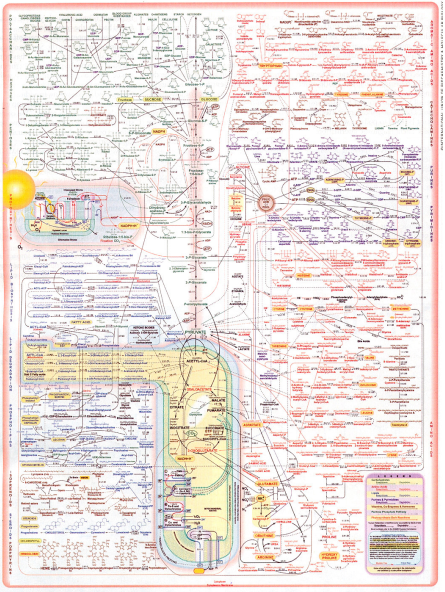
Driving natural systems: Flux balance analysis of metabolism

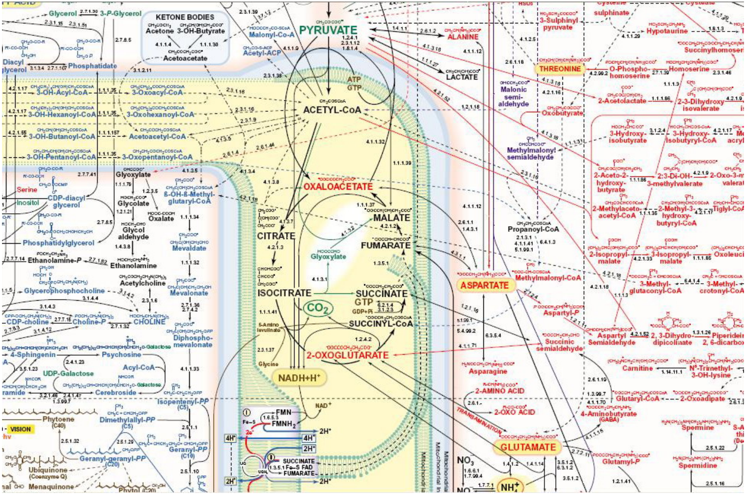
Complicated biochemical networks



- Biological metabolic and regulatory systems are incredibly complicated!
- Even if solvable, at what point does an ODE description of each term lose useful-ness/interpretability? How robust is it to errors in parameters or missing data?

Complicated biochemical networks

- We saw some of this subset yesterday...
- (Regarding these three lectures, knowledge of individual enzymes/reactants is explicitly non-examinable)



Flux balance analysis

- Briefly, FBA is a simple way of simulating and optimising large metabolic systems, including the effects of perturbations, assuming a steady state
- Is a steady state assumption useful?
- Metabolite concentrations equilibrate fast (seconds) with respect to the timescale of genetic regulation (minutes) ... so a qualified yes
- Imagine we have a stochiometry matrix $\underline{\underline{S}}$

$$\begin{bmatrix} s_{11} & s_{12} & s_{13} & \dots \\ s_{21} & s_{22} & s_{23} & \dots \\ s_{31} & s_{32} & s_{33} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

- and a vector $\underline{v}$ describing the fluxes through each reaction

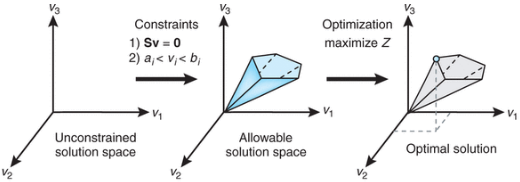
$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \dots \end{bmatrix}$$

Flux balance analysis

- Stoichiometric matrix $\underline{\underline{S}}$, vector of fluxes $\underline{v}$; change in species with time $\frac{d\underline{X}}{dt} = \underline{\underline{S}}\underline{v}$
- (the ith element of the vector $\underline{\underline{S}}\,\underline{v}$ gives the rate of accumulation or loss of species i)
- Balancing the fluxes means imposing $\underline{\underline{S}}\,\underline{v} = \underline{0}$ (finding the steady state)
- We also probably want to bound $\underline{v}$ (for example, $\underline{0} \leqslant \underline{v} \leqslant \underline{v}^{max}$)
- An important feature of $\underline{\underline{S}}$ is that there are generally more reactions than chemical species, so the problem is underdetermined
- The solution space for any system of homogeneous equations and inequalities is a convex polytope

Flux balance analysis

- Stoichiometric matrix $\underline{\underline{S}}$, vector of fluxes $\underline{v}$
- In general, many different ways exist to balance the fluxes in a system: set of possible $\underline{v}$ forms a convex polytope
- Each point in the polytope represents a set of fluxes at which the system can run at steady state
- We can find the solution that is optimal with respect to some feature that we are interested in
- Introduce a target function described by $\underline{c}$; defined so that the quantity we want to maximise is $\underline{c} \cdot \underline{v}$
- For example: if we're interested in producing chemical X, set $c_i = 1$ for i describing $X \to \emptyset$, $c_i = 0$ otherwise



Flux balance analysis

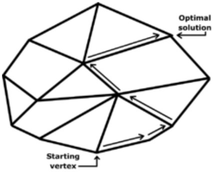
- FBA is computationally cheap, needs only stoichiometric coefficients, no need for kinetic parameters
- FBA doesn't uniquely specify a solution, is difficult to use dynamically, isn't perfect (e.g. regulatory loops), and creating the stochiometric matrix is a non-trivial exercise in curation

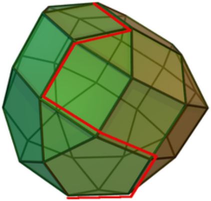
Input, output, accumulation and dummy reactions

- Assume zero accumulation, thus working in a steady-state picture
- Introduce input and output ‘reactions’: flux of reactants and products into and out of the system
- Maximise the dummy output flux of products of interest to maximise production of these products
- Problem is to maximise $\underline{c} \cdot \underline{v}$ subject to $\underline{\underline{S}}\,\underline{v} = \underline{0}$ and $\underline{0} \leqslant \underline{v}$
- This problem falls into *linear programming* (maximise a function within a convex polytope)

Linear programming

- Dantzig's simplex algorithm: 'the algorithm that rules the world' according to New Scientist
- Locate extreme points (vertices) of the polytope
- For a standard linear program, if the objective function has an optimal value in the convex polytope, then it has this value on at least one extreme point of the polytope
- If an extreme point is not a minimum, there is an edge containing that point such that the objective function gets better along that edge away from the point
- If that edge is finite, we take it and find a better point; otherwise, the problem is unbounded and has no solution
- Some problems: potential cycling if points are degenerate; exponential worst-case. Many alternatives: refined pivoting; criss-cross; conic sampling; other algorithms for interior points in more complicated contexts





The simplex algorithm

- Step 1: write problem using ‘slack variables’ to set up equalities.
- Minimise $r = -2x - 3y - 4z$ given

$$\begin{aligned} 3x + 2y + z &\leq 10 \\ 2x + 5y + 3z &\leq 15 \\ x, y, z, &> 0 \end{aligned}$$

becomes

$$\begin{aligned} 3x + 2y + z + s + 0t &= 10 \\ 2x + 5y + 3z + 0s + t &= 15 \\ x, y, z, s, t &> 0 \end{aligned}$$

- Step 2: write the problem in tableau form

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 0 & 0 & 0 \\ 0 & 3 & 2 & 1 & 1 & 0 & 10 \\ 0 & 2 & 5 & 3 & 0 & 1 & 15 \end{bmatrix} \tag{1}$$

The simplex algorithm

- A tableau gives a ‘basic feasible solution’: identify columns with one unit entry

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 0 & 0 & 0 \\ 0 & 3 & 2 & 1 & 1 & 0 & 10 \\ 0 & 2 & 5 & 3 & 0 & 1 & 15 \end{bmatrix} \tag{2}$$

- The corresponding variables take values from the final column; other variables are zero ($x = y = z = 0, s = 10, t = 15$)
- Step 3: identify a ‘pivot’ column – need a positive entry in the top row

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 0 & 0 & 0 \\ 0 & 3 & 2 & 1 & 1 & 0 & 10 \\ 0 & 2 & 5 & 3 & 0 & 1 & 15 \end{bmatrix} \tag{3}$$

The simplex algorithm

- Step 4: identify the pivot row.
- Minimise (entry in final column) / (entry in pivot column) over all rows

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 0 & 0 & 0 \\ 0 & 3 & 2 & 1 & 1 & 0 & 10 \\ 0 & 2 & 5 & 3 & 0 & 1 & 15 \end{bmatrix} \tag{4}$$

- $15/3 > 10/1$: choose row 3.
- Step 5: multiply pivot row by the reciprocal of pivot entry (e.g. 3)

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 0 & 0 & 0 \\ 0 & 3 & 2 & 1 & 1 & 0 & 10 \\ 0 & 2/3 & 5/3 & 1 & 0 & 1/3 & 5 \end{bmatrix} \tag{5}$$

- Step 6: add linear multiples of pivot row to tableau such that pivot column entry is zero (e.g. row 1 = row 1 - $4/3\times$ row 3; row 2 = row 2 - row 3)

$$\begin{bmatrix} 1 & -2/3 & -11/3 & 0 & 0 & -4/3 & -20 \\ 0 & 7/3 & 1/3 & 0 & 1 & -1/3 & 5 \\ 0 & 2/3 & 5/3 & 1 & 0 & 1/3 & 5 \end{bmatrix} \tag{6}$$

The simplex algorithm

- Repeat until we have no positive entries corresponding to variables in the top row.

$$\begin{bmatrix} 1 & -2/3 & -11/3 & 0 & 0 & -4/3 & -20 \\ 0 & 7/3 & 1/3 & 0 & 1 & -1/3 & 5 \\ 0 & 2/3 & 5/3 & 1 & 0 & 1/3 & 5 \end{bmatrix} \tag{7}$$

- We're done!
- Basic feasible solution:

$$\begin{bmatrix} 1 & -2/3 & -11/3 & 0 & 0 & -4/3 & -20 \\ 0 & 7/3 & 1/3 & 0 & 1 & -1/3 & 5 \\ 0 & 2/3 & 5/3 & 1 & 0 & 1/3 & 5 \end{bmatrix} \tag{8}$$

$\rightarrow z = 5, s = 5, x = y = t = 0.$

- Solution $r = -2x - 3y - 4z = -20.$

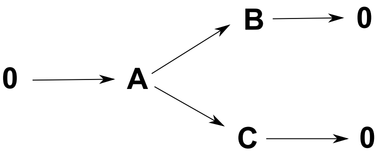
Evolution and biological optimisation

- Is it reasonable to expect natural systems to find optimal solutions?
- Evolution is a powerful optimiser, particularly in systems with large populations and small generation times (for example, bacteria)
- Growth rate is often explored with FBA: a reasonable proxy for 'fitness' in bacterial populations
- How is growth rate predicted? Often, biomass production; also arguments for ATP production.
- e.g. *E. coli*:

  $0.33G6P + 0.07F6P + 0.96R5P + 0.36E4P + 0.36GA3P + 0.863PG + 0.77PEP + 2.94PYR + 2.41ACCOA + 1.65OA + 1.28AKG + 15.7NADPH + 40.2ATP \rightarrow BM + 3NADH$

  (Schuetz *et al.* MSB 3 119 (2007))
- Can also consider nonlinear and/or nonconvex objectives: need more general solvers.
- As we will see, bacterial systems can evolve to find optimal solutions on reasonable human timescales
- Variants of flux balance analysis also allow us to explore transient phenomena

A simple example of FBA



- All reactions irreversible
- Sources and sinks allow balance

$$\underline{\underline{S}}^T = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

- Trivial example: maximise production of B

$$\underline{c} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

A simple example of FBA

$$\underline{\underline{S}}^T = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} ; \ \underline{c} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

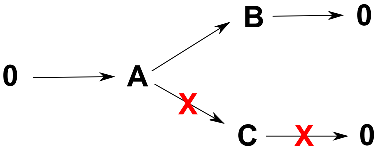
- Find the $\underline{v}$ that maximises $\underline{c} \cdot \underline{v}$ such that $\underline{\underline{S}}\underline{v} = \underline{0}$ and $\underline{0} \leqslant \underline{v} \leqslant \underline{1}$
- Worth knowing the appropriate MATLAB command: `linprog`
- `v = linprog(-c, [], [], S, zeros(1,N), a, b)`
- First argument $-\underline{c}$ because `linprog` is a minimisation tool
- Empty arguments 2 & 3 allow constraints of the form $\underline{\underline{S}}\underline{v} \leqslant \underline{w}$
- Arguments 4 & 5 impose $\underline{\underline{S}}\underline{v} = \underline{0}$
- Arguments 6 & 7 bound $\underline{v}$ with $\underline{a} \leqslant \underline{v} \leqslant \underline{b}$

A simple example of FBA

$$\underline{\underline{S}}^T = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}; \ \underline{c} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

- In this case `v = linprog(-c, [], [], S, zeros(1,N), a, b)` with $\underline{a} = \underline{1}$, $\underline{b} = \underline{0}$ straightforwardly gives

$$\underline{v} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

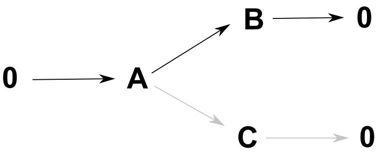


A simple example of FBA

$$\underline{\underline{S}}^T = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}; \; \underline{c} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

- What if the reaction $A \to C$ keeps us alive?
- We can solve with a lower bound on flux through this reaction: $a_3 = 0.1$.
- In this case `v = linprog(-c, [], [], S, zeros(1,N), a, b)` unsurprisingly gives

$$\underline{v} = \begin{bmatrix} 1 \\ 0.9 \\ 0.1 \\ 0.9 \\ 0.1 \end{bmatrix}$$

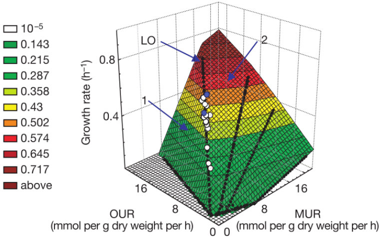


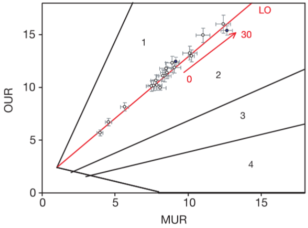
Deleting and restricting reactions

- `v = linprog(-c, [], [], S, zeros(1,N), a, b)` finds $\underline{v}$ that maximises $\underline{c} \cdot \underline{v}$ such that $\underline{\underline{S}}\underline{v} = \underline{0}$ and $\underline{a} \leqslant \underline{v} \leqslant \underline{b}$
- Removing single reactions in turn allows identification of the key reactions for production of a target (e.g. biomass)
- Many reactions are catalysed by specific enzymes: identification of critical reactions can thus be used to choose drug targets
- Extension to multiple reaction deletions (e.g. pairwise) for promiscuous enzymes
- Bounds in linear programming can be used to “knock down” rather than “knock out” reactions
- Set appropriate $b_i = 0$ or $b_i = b_i^{max}$

Evolution and optimality

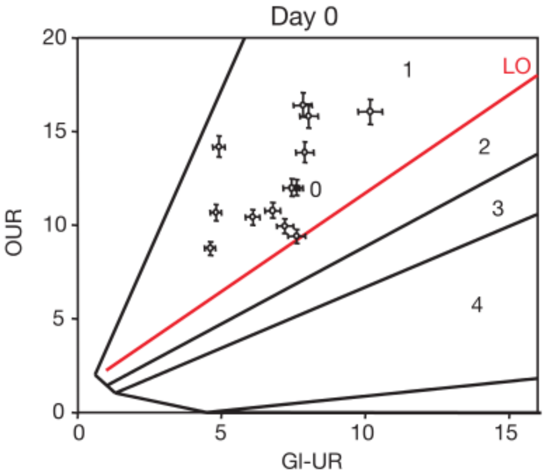
- *E. coli* wildtype evolved to use several different food sources
- When grown on malate,
  - Growth rates against oxygen and malate uptake closely match FBA predictions
  - Initial positions lie on optimal extreme path; evolution drives further increase along this path





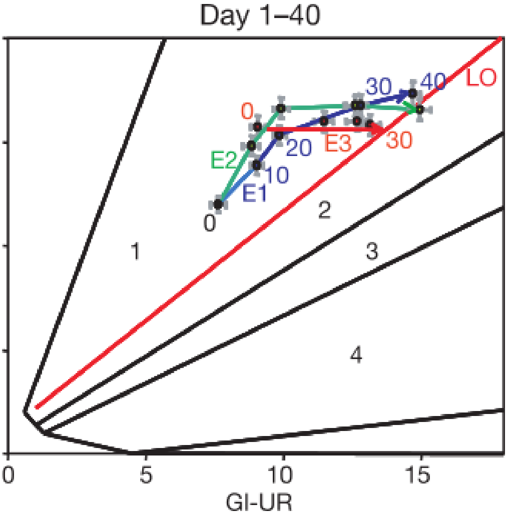
Evolution and optimality

- When grown on glycerol – an unnatural, but passable, sole foodstuff – metabolism is initially far from optimal...



Evolution and optimality

- ... but evolution locates the optimal (and accurately predicted) metabolic profile after 40 days (∼700 generations)



Extensions of FBA

- Flux variability analysis: for degenerate optima, determines the range of parameters that gives rise to the optimal solution
- Minimisation of metabolic adjustment: attempts to predict transient behaviour in response to perturbation by minimising distance between standard and perturbed fluxes
- Quadratic programming: minimise $||\underline{v}_{WT} - \underline{v}_{\mu}||^2$ such that $\underline{\underline{S}}\underline{v}_{\mu} = \underline{0}$
- Other approaches to dynamics: regulatory on-off minimisation; iterated dynamic FBA