Sep 20, 2024

Introduction to flux balance analysis (FBA)

This protocol is a draft, published without a DOI.

Cailean Carter[1], Dipali Singh[1], Gemma Langridge[1]

[1]Quadram Institute Biosciences

Linear programming

Cailean Carter
Quadram Institute

Protocol Citation: Cailean Carter, Dipali Singh, Gemma Langridge 2024. Introduction to flux balance analysis (FBA). **protocols.io** **https://protocols.io/view/introduction-to-flux-balance-analysis-fba-daz22f8e**

Protocol status: Working
We use this protocol and it's working

Created: March 21, 2024

Last Modified: September 20, 2024

Protocol Integer ID: 97050

Keywords: FBA, LP, metabolic model

Abstract

Flux balance analysis (FBA) is a mathematical approach to finding an optimal net flow of mass through a metabolic network that follows a set of instructions defined by the user. This protocol covers the mathematical principles behind FBA and provides coding examples that can be followed using Python3. A basic understanding of the following aspects of linear algebra is advantageous: vectors, matrices, transposition, matrix multiplication, and matrix decomposition.

Materials

A computer

Background

1 Flux balance analysis (FBA) is a mathematical approach to finding an optimal net flow of mass through a metabolic network that follows a set of instructions defined by the user. This protocol covers the mathematical principles behind FBA and provides coding examples that can be followed using Python3. A basic understanding of the following aspects of linear algebra is advantageous: vectors, matrices, transposition, matrix multiplication, and matrix decomposition.

Software

python	NAME
3.6	OS
Guido van Rossum	DEVELOPER

Command

Install Python packages

```
pip install numpy
pip install scipy
```

Linear programming

2 FBA is built on a mathematical technique called linear programming (LP); a well-established method for solving optimisation problems that is applicable to any discipline (Dantzig, 1963). The main constituents of LP are functional units called “activities” which represent the behaviours being investigated like units of materials or rates of change (Watson, 1986). And it is the proportions of these activities that are optimised to either minimise or maximise the output of one or more activity called the objective function (z). Activities can have an associated cost coefficient ($\boldsymbol{c}$), either literal cost or physical work, included in the objective

function which is taken into consideration when calculating an optimum solution. Therefore, the objective function is to either maximise or minimise the equation

$$z = \sum_{i=1}^{x} c_i \cdot x_i$$

commonly expressed as

$$z = \boldsymbol{c}^T \boldsymbol{x}$$

where $\boldsymbol{x}$ is a vector of all activities (Dantzig, 1963). While the objective function provides a purpose, the problem being addressed must have a list of requirements (called constraints) which limit the possible combinations of values for the activities (otherwise solutions could be 0 or infinite for all activities). Constraints represent the known or imposed limitations of a system that can be presented in mathematical form and must have linear relationships with the activity value. For example, an activity might have a quota to meet ($x_i = 1$) or cannot exceed a budget ($x_i \leq 1$). These concepts are called linear equalities and inequalities, respectively. It is plausible that there are none or an almost infinite number of feasible combinations of $\boldsymbol{x}$ that satisfy the constraints, yet the objective function tasks the programme with finding an optimum solution. This step can be performed using the simplex method which jumps between feasible solutions until the optimum solution is found (Dantzig, 1963). An example of how LP can be applied to a real-world scenario is discussed in **Box 1**.

Box 1. Example of linear programming (LP) – the minimising UTI prescribing cost problem.

3 **Scenario:** A pharmacy wants to minimise its expenditure on first-line treatments for urinary tract infections (UTIs) to keep up with its monthly budget of £10,000. A full dose of trimethoprim costs £0.75 while nitrofurantoin costs £3.43. On average, the pharmacy prescribes 7,300 antibiotics for UTIs each month. However, at least 20% of cases will be prescribed nitrofurantoin due to high incidents of resistance to trimethoprim, thus, trimethoprim should make up at most 80% of the stock order. What is the optimum number of trimethoprim and nitrofurantoin doses the pharmacy should order to minimise cost?

Note

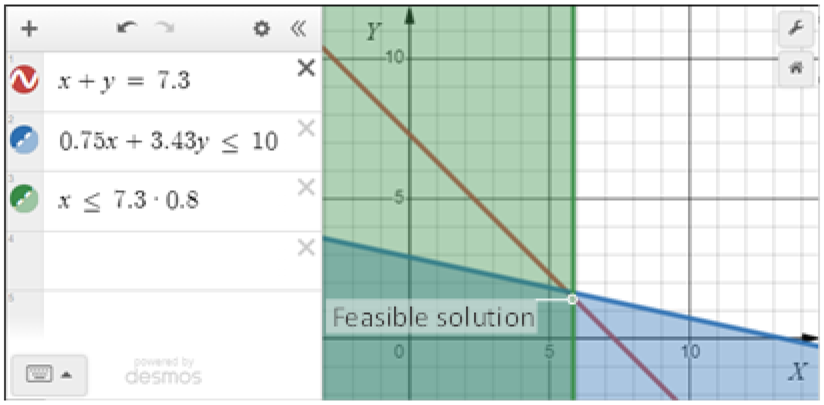
Firstly, packets of trimethoprim and nitrofurantoin are the activities and can be represented by x and y, respectively, and have an associated cost coefficient (cost per dose). Thus, the objective function is to find the optimum combination of $0.75x$ and $3.43y$ that has the smallest total cost. The pharmacy also has a list of requirements that must be met - the constraints. The combination of x and y must meet their monthly demand of 7,300 prescriptions, thus: $x + y = 7.3$ (thousand). Also, the objective function must be below their monthly budget, thereby: $0.75x + 3.43y \leq 10$ (thousand). The final constraint was that trimethoprim must make at most 80% of the order: $x \leq 7.3 \cdot 0.8$. This information can be expressed in a mathematical representation shown in **Eq. 1**. Python code can be used to solve this equation **(A)**. Yet, with a small number of constraints it is possible to also identify the optimum solution using a graphics calculator **(B)**.

$$\text{Minimise: } 0.75x + 3.43y$$
$$\text{Subject to } \begin{cases} x + y = 7.3 \\ 0.75x + 3.43y \leq 10 \\ x \leq 7.3 \cdot 0.8 \end{cases} \qquad (1)$$

Command

A) Solve LP problem

```
# Python3
import numpy as np
from scipy.optimize import linprog

# x = packets of trimethoprim;
# y = packets of nitrofurantoin

# Objective function: minimise 0.75x+3.43y
z = [0.75, 3.43]

# Equality constraints
# x+y=7.3
eq_lhs = [[1, 1]]
eq_rhs = [7.3]

# Inequality constraints
# 0.75x + 3.43y <= 10
# x <= 7.3*0.8
in_lhs = [[0.75, 3.43],
          [1, 0]]
in_rhs = [10, 7.3*0.8]

# Solve for optimum solution
linprog(c      = z,
        A_eq   = eq_lhs,
        b_eq   = eq_rhs,
        A_ub   = in_lhs,
        b_ub   = in_rhs,
        method = 'revised simplex')
# Solution: x=5.84, y=1.46
```



B) Graphics calculator visualising the constraints with the feasible solution annotated.

The solution to this problem is 5,840 doses of trimethoprim and 1,460 doses of nitrofurantoin.

Flux balance analysis

4 FBA uses the same mathematical principles as LP but includes some new concepts for analysing metabolic models. Firstly, the activities in FBA are chemical, enzymatic, and transportation reactions denoted by v. In this context, the activities are irrespective of enzyme and metabolite properties and environmental conditions, instead it is the combination of these factors as an economic process called flux (Orth et al., 2010). For ease of use, all the metabolic information of an organism is stored in a mathematical matrix; where each column represents a reaction containing the stoichiometric coefficients of metabolites in rows and is called the stoichiometry matrix $(\boldsymbol{N})$.

FBA involves a steady state assumption whereby the quantity of metabolites within the system cannot change. This is to prevent metabolites from having unrealistic quantities particularly when the actual quantity of metabolites present in a sample is unknown (Holzhütter, 2004). This may seem counterintuitive as an organism needs to catabolise metabolites for energy and growth. To address this, some metabolites have external equivalents (denoted by the prefix *X*) and are not included in the stoichiometry matrix. Instead, the transporters are included with the stoichiometry coefficient for the internal metabolite only, thereby defining the inputs and outputs of the network. Also, steady state provides the first set of constraints for the LP as the rate of change for all internal metabolites must equal zero. Therefore, the flux of all reactions involved in the production/consumption of a metabolite (S) must equal zero:
$\frac{dS}{dt} = v_1 + v_2 = 0$
Thus, if v_1 carried 1 unit of flux then v_2 must carry -1 unit of flux for the equation to be balanced - hence flux balance analysis. Further, the list of equations derived from the steady state assumption is called the mass balance equations. The flux values carried by a reaction represent the net conversion of the metabolites as reactions can be reversible (carrying flux in both directions) and be the amalgamation of several reactions (or pathways) into one activity (Watson, 1986). For simplicity, steady state is represented by $\boldsymbol{N} \cdot \boldsymbol{v} = 0$ in the LP formulation and all the combinations of $\boldsymbol{v}$ that satisfy this equation is called the null space. It is possible to

identify what combinations of reactions can carry flux under steady state by calculating the null space of the stoichiometry matrix; this includes feasible routes from inputs to outputs and cyclical pathways called conserved moieties **(Box 2)**. However, null space analysis is limited by not taking the directionality of reactions into consideration, whereas FBA does take this into account.

The objective function for FBA is to maximise or minimise the absolute total flux through one or more reactions - *absolute* because reactions can carry negative flux. A weighting attribute (denoted by ω) is used instead of a cost coefficient to better reflect its function in the analysis and is given the default value of 1. Therefore, the objective value in FBA is to minimise or maximise the equation:
$\sum_{i=1}^{n} |v_i \cdot \omega_i|$

Whether the choice of objective function is representative of the metabolism within an organism is debatable but does, nonetheless, provide valuable insights into the metabolic processes (Schuetz et al., 2007). Constraints relating to the investigation are added as (in)equality constraints on the flux of a reaction and/or by applying a weighting factor other than 1. LP is then used to find a combination of flux values that most likely represents the metabolism of the organism. A demonstration of this is provided in **Box 3**.

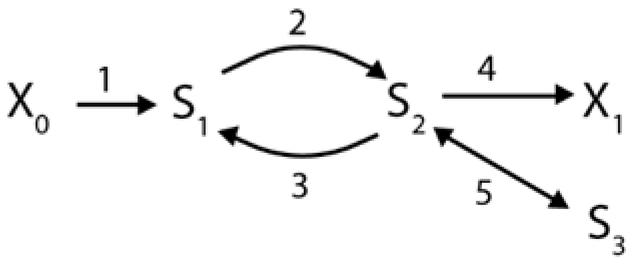
Note

Although the code examples here used the Python package SciPy for LP, practical applications would use a dedicated LP package like the GNU LP kit (GLPK).

Box 2. Finding the null space of a stoichiometry matrix.

5 **(A)** Small network map containing internal metabolites (S_i), external metabolites (X_i), and arrows representing the reactions (R_i) connecting metabolites. The stoichiometry matrix of this network is
shown in **(B)** and contains the stoichiometric coefficients of the metabolites involved: a positive value denotes produced, negative consumed, and zero for not involved. Null space can be calculated using a computer directly from the stoichiometry matrix. In the Python code **(C)**, a method called single value
decomposition is used to find the null space which outputs a matrix of fractions. This matrix is then normalised with lower-upper (LU) decomposition and made human readable by removing sign ambiguity and rounding of the numbers. The result is a matrix called the kernel **(D)** where each column contains a
combination of reactions that can carry flux under steady state. The output in **(C)** contains fractions for the first column as it envelopes the conserved moiety solution identified in the

second solution/column. Therefore, the result in column 1 recognises that R3 can be part of this solution.



A) Small network map

B)
$$\boldsymbol{N} = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{ccccc} R_1 & R_2 & R_3 & R_4 & R_5 \end{array} \\ \left[\begin{array}{ccccc} 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}$$

Command

C) Calculate null space

```
# Python3
import numpy as np
from scipy import linalg

# Stoichiometry matrix
N = [[1, -1, 1, 0, 0],
     [0, 1, -1, -1, -1],
     [0, 0, 0, 0, 1]]

# Compute null space
ns = linalg.null_space(np.matrix(N))

# Normalise with lower-upper decomposition
p, l, u = linalg.lu(ns)
# Make human-readable
K = l * np.sign(l)
np.around(K, 1)
# array([[1. , 0. ],
#        [0.4, 1. ],
#        [0.6, 1. ],
#        [1. , 0. ],
#        [0. , 0. ]])
```

D) Null space : $\mathcal{N}(\boldsymbol{N}) = \{\boldsymbol{v} | \boldsymbol{N}\boldsymbol{v} = 0\}$

$$\boldsymbol{K} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{matrix} R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \end{matrix}$$

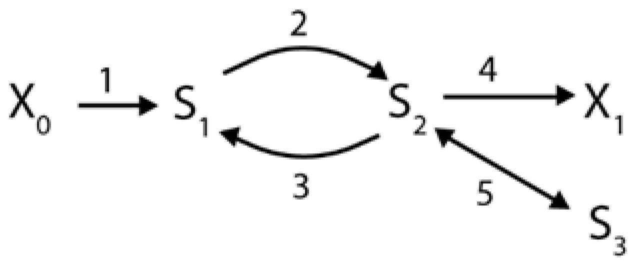
Note

Glossary:

$\boldsymbol{N}$	Stoichiometry matrix	R	Reaction
$\boldsymbol{v}$	Flux vector	$\mathcal{N}(\boldsymbol{N})$	Null space
S	Internal metabolite	$\boldsymbol{K}$	Kernal
X	External metabolite	LU	Lower-upper decomposition

Box 3. Example of flux balance analysis (FBA) on a small metabolic network.

6 **(A)** The network contains internal metabolites (S_i) whose concentration must remain balanced,
external metabolites (X_i) whose abundance can be unbalanced (like growth media and biomass components), and arrows representing the reactions (R_i) connecting metabolites.
(B) The network is
represented in a mathematical grid called the stoichiometry matrix $(\boldsymbol{N})$, with columns of reactions $(\boldsymbol{n})$ and rows of internal metabolites $(\boldsymbol{m})$. The contents of the grid denote whether a metabolite is produced (positive), consumed (negative), or not involved (zero) in each reaction. **(Eq. 2)** An LP formulation is then formed to address the research question, including the objective function of minimising absolute total flux and the steady state constraints. Additional constraints are included that represent the demands imposed by the research question. Also, the steady state mass balance equations represent the rows in the stoichiometry matrix. **(C)** Python code can be used to solve this equation with an LP function using the simplex method. The output is the optimum solution to the research question.



A) Simple network map

B)
$$\boldsymbol{N}_{\boldsymbol{m}\times\boldsymbol{n}} = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{ccccc} R_1 & R_2 & R_3 & R_4 & R_5 \end{array} \\ \begin{bmatrix} 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Note

Query: What is the most efficient route through the network to synthesise one unit of X_1 ?

The flux through $R_4(v_4)$ will be 1 to produce 1 units of $X_1 : v_4 = 1$.

From the literature, R_3 and R_4 are known to share an enzyme; for every unit of $S_2 \rightarrow X_1$, the enzyme returns half a unit of $S_2 \rightarrow S_1$: $v_3 = 0.5$.

$$\text{Minimise: } \sum_{i=1}^{n} |v_i \cdot \omega_i|$$
$$\text{Subject to } \left\{ \begin{array}{l} \left. \mathbf{N} \cdot \mathbf{v} = 0 \left\{ \begin{array}{r} v_1 + v_3 - v_2 = 0 \\ v_2 - v_3 - v_4 - v_5 = 0 \\ v_5 = 0 \end{array} \right. \right\} \begin{array}{l} \text{Under steady state:} \\ \frac{dS_i}{dt} = \sum v_{inflow} - \sum v_{outflow} = 0 \end{array} \\ \text{Additional constraints} \left\{ \begin{array}{l} v_4 = 1 \\ v_3 = 0.5 \end{array} \right. \end{array} \right. \qquad (2)$$

Command

C) Solve LP problem

```
# Python3
import numpy as np
from scipy.optimize import linprog

# Objective function : minimise absolute total flux:
# v1*w1 + v2*w2 + v3*w3 + v4*w4 + v5*w5
reactions = np.ones(shape=5)    # v
weights = np.ones(shape=5)      # w
z = reactions * weights         # objective function

# Subject to:
# 1. Steady state assumption: Nv = 0
N = [[1, -1, 1, 0, 0],
     [0, 1, -1, -1, -1],
     [0, 0, 0, 0, 1]]
# Rate of change for internal metabolites equals 0
N_rhs = [0, 0, 0]

# 2. Additional defined constraints:
# v4 = 1 and v3 = 0.5
constraints = [[0, 0, 0, 1, 0],
               [0, 0, 1, 0, 0]]
constraints_rhs = [1, 0.5]

# Solve for optimum solution
linprog(c       = z,
        A_eq    = N + constraints,
        b_eq    = N_rhs + constraints_rhs,
        method  = 'revised simplex')
# Solution: v1=1, v2=1.5, v3=0.5, v4=1, v5=0
```

Solution is $v_1 = 1,\ v_2 = 1.5,\ v_3 = 0.5,\ v_4 = 1,\ v_5 = 0$

Further reading & applications of FBA

7 The examples in this protocol (**Boxes 1 & 3**) were intended to provide an introduction into the applications of LP and FBA that could be taken towards understanding the literature and/or

pursuing metabolic modelling. As such, we recommend exploring some of the early literature utilising FBA:

1. Watson (1986) described the reconstruction of metabolic pathways into a computerised format and the interrogation of core metabolic pathways.
2. Fell & Small (1986) explored how glucose was converted to fat in rat adipose tissues.
3. Varma & Palsson (1994) investigated a model of *E. coli* and incorporated experimental data as constraints.

Current applications of metabolic modelling have been covered extensively by others (Gu *et al.,* 2019). But, briefly, metabolic modelling using FBA can be invaluable in identifying drug targets (Hartman *et al.,* 2014), optimising bio-production (Trinh *et al*., 2008), and there is recent progress towards modelling bacterial communities from metagenomes (Frioux *et al*., 2020). The breadth of applications is indicative of FBA's effectiveness in helping researchers and industries to understand complex biological networks and tackle wicked problems.

Ideally, this protocol has explained why and how FBA has been used in the literature and encourages more researchers to try FBA.

Protocol references

Dantzig GB, Linear Programming and Extensions. 1963, RAND Corporation.

Fell DA and Small JR. Fat synthesis in adipose tissue. An examination of stoichiometric constraints. Biochem. J., 1986; 238: 781-786.

Frioux C, Singh D, Korcsmaros T, et al. From bag-of-genes to bag-of-genomes: metabolic modelling of communities in the era of metagenome-assembled genomes. Computational and Structural Biotechnology Journal, 2020; 18: 1722-1734.

Gu C, Kim GB, Kim WJ, et al. Current status and applications of genome-scale metabolic models. Genome Biology, 2019; 20: 121.

Hartman HB, Fell DA, Rossell S, et al. Identification of potential drug targets in Salmonella enterica sv. Typhimurium using metabolic modelling and experimental validation. Microbiology, 2014; 160: 1252-1266.

Holzhütter HG. The principle of flux minimization and its application to estimate stationary fluxes in metabolic networks. Eur. J. Biochem., 2004; 271: 2905-22.

Orth JD, Thiele I, and Palsson BØ. What is flux balance analysis? Nat. Biotechnol., 2010; 28: 245-248.

Schuetz R, Kuepfer L, and Sauer U. Systematic evaluation of objective functions for predicting intracellular fluxes in Escherichia coli. Mol. Syst. Biol., 2007; 3: 119.

Trinh CT, Unrean P, and Srienc F. Minimal *Escherichia coli* Cell for the Most Efficient Production of Ethanol from Hexoses and Pentoses. Appl. Environ. Microbiol., 2008; 74: 3634-3643.

Varma A and Palsson BO. Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type Escherichia coli W3110. Appl. Environ. Microbiol., 1994; 60: 3724-3731.

Watson MR. A discrete model of bacterial metabolism. Bioinformatics, 1986; 2: 23-27.