Uniform sampling a genome-scale model

Author(s): German A. Preciat Gonzalez, Hulda S. Haraldsdóttir, Systems Biochemistry Group, Leiden University.

Reviewer(s): Almut Heinken, Molecular Systems Physiology Group, National University of Ireland, Galway.

INTRODUCTION

The flux space Ω for a given set of biochemical and physiologic constraints is represented by:

$$\Omega = \{v \mid Sv = b; l \leq v \leq u\}$$

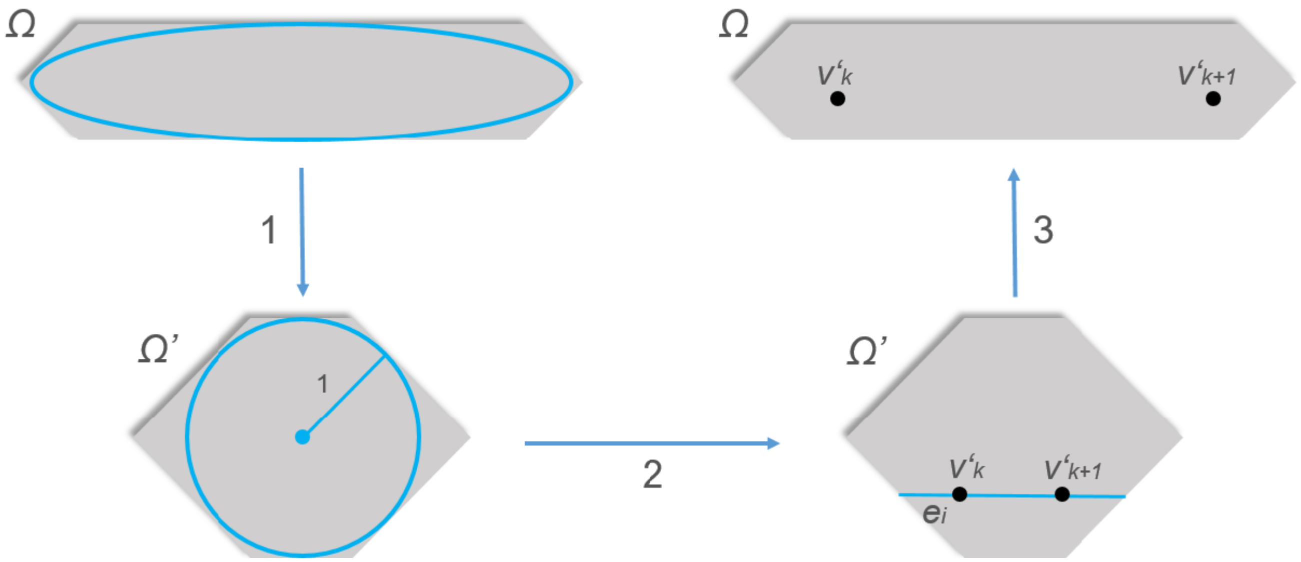
where v represents feasible flux vectors, $S \in \mathcal{Z}^{m\times n}$ the stoichiometric matrix, while l and u are lower and upper bounds on fluxes. These criteria still allow a wide range of admissible flux distributions which, in FBA are commonly further restricted by introducing an objective to optimise, transforming the question of admissible fluxes into an FBA problem [1] of the form

$$\begin{aligned} \min_{v} \quad & c^T v \\ \text{s.t.} \quad & Sv = b, \\ & l \leq v \leq u, \end{aligned}$$

where c is a linear biological objective function (biomass, ATP consumption, HEME production, etc.). Even under these conditions there is commonly a range of optimal flux distributions, which can be investigated using flux variability analysis. If the general capabilities of the model are of interest, however, uniform sampling of the entire flux space Ω is able to provide an unbiased characterization, and therefore, can be used to investigate the biochemical networks. It requires collecting a statistically meaningful number of flux distributions uniformly spread throughout the whole flux space and then analysing their properties. There are three basic steps to perform a uniform sampling for a set of feasible fluxes:

- Define the flux space to be sampled from physical and biochemical constraints
- Randomly sample the defined flux space based on uniform statistical criteria
- If is necessary, section the flux space according to post-sampling.

In COBRA v3 there are three different sampling algorithms: coordinate hit-and-run with rounding (CHRR), artificial centring hit-and-run (ACHR) and the minimum free energy (MFE). In this tutorial, we will use the CHRR algorithm [2] to uniformly sample a high dimensionally constraint-based model of the differentiation of induced pluripotent stem cells to dopaminergic neurons (iPSC_dopa). The algorithm consists of rounding the anisotropic flux space Ω using a maximum volume ellipsoid algorithm [3] and then performs a uniform sampling based on the provably efficient hit-and-run random walk [4]. Below is a high-level illustration of the process to uniformly sample a random metabolic flux vector v from the set Ω of all feasible metabolic fluxes (grey). **1)** Apply a rounding transformation T to Ω. The transformed set $\Omega' = T\Omega$ is such that its maximal inscribed ellipsoid (blue) approximates a unit ball. **2)** Take q steps of coordinate hit-and-run. At each step, i) pick a random coordinate direction e_i, and ii) move from current point $v'_k \in \Omega'$ to a random point $v'_{k+1} \in \Omega'$ along $v'_k + \alpha e_i \cap \Omega'$. **3)** Map samples back to the original space by applying the inverse transformation, e.g., $v_k = T^{-1}v'_k$.



Equipment setup

Please note that some of the plotting options in the tutorial require Matlab 2016a or higher. Moreover, the tutorial requires a working installation of the Parallel Computing Toolbox.

Please set a solver, e.g., gurobi. Note that the solver ibm_cplex is required for the function fastFVA.

In this tutorial, we will perform FVA using the function `fluxVariability`. Change the variable `options.useFastFVA = 1` to use `fastFVA`.

Modelling

We will investigate ATP energy production with limited and unlimited oxygen uptake, following closely the flux balance analysis (FBA) tutorial published with [1].

We start by loading the model with its flux bounds and the objective function (ATP demand reaction). We set the maximum glucose uptake rate to 18.5 mmol/gDW/hr. To explore the entire space of feasible steady state fluxes we also remove the cellular objective.

```
options.useFastFVA = false; % to use fluxVariability
tutorialPath = fileparts(which('tutorial_uniformSampling.mlx'));
longFileName=[tutorialPath filesep 'data' filesep 'iPSC_DA.mat'];
if 0
    model = readCbModel(longFileName,'modelName','modelUptClosed');
else
    load(longFileName)
    model = modelUptClosed;
end
model = changeRxnBounds(model, 'EX_glc(e)', -18.5, 'l');
model.c = 0 * model.c; % clear the objective
```

We allow unlimited and limited oxygen uptake in the models creating two distinct models based on the input model.

```
unlimitedOx = changeRxnBounds(model, 'EX_o2(e)', -1000, 'l');
limitedOx = changeRxnBounds(model, 'EX_o2(e)', -4, 'l');
```

Flux variability analysis

Flux variability analysis (FVA) returns the minimum and maximum possible flux through every reaction in a model.

```
if options.useFastFVA
    [minUn, maxUn] = fastFVA(unlimitedOx, 100);
    [minLim, maxLim] = fastFVA(limitedOx, 100);
else
    [minUn, maxUn] = fluxVariability(unlimitedOx);
    [minLim, maxLim] = fluxVariability(limitedOx);
end
```

```
Starting parallel pool (parpool) using the 'local' profile ...
Connected to the parallel pool (number of workers: 2).
```

FVA predicts faster maximal ATP production with unlimited than with limited oxygen uptake conditions.

```
ATP = 'DM_atp_c_';  % Identifier of the ATP demand reaction
ibm = find(ismember(model.rxns, ATP));  % column index of the ATP demand reaction
fprintf('Max. ATP energy production with an unlimited oxygen uptake: %.4f/h.\n', maxUn(
fprintf('Max. ATP energy production with a limited oxygen uptake: %.4f/h.\n\n', maxLim(
```

An overall comparison of the FVA results can be obtained by computing the Jaccard index for each reaction. The Jaccard index is here defined as the ratio between the intersection and union of the flux ranges in the unlimitedOx and limitedOx models. A Jaccard index of 0 indicates completely disjoint flux ranges and a Jaccard index of 1 indicates completely overlapping flux ranges. The mean Jaccard index gives an indication of the overall similarity between the models.

```
J = fvaJaccardIndex([minUn, minLim], [maxUn, maxLim]);
fprintf('Mean Jaccard index = %.4f.\n', mean(J));
```

To visualise the FVA results, we plot the flux ranges as errorbars, with reactions sorted by the Jaccard index.

```
E = [(maxUn - minUn)/2 (maxLim - minLim)/2];
Y = [minUn minLim] + E;
X = [(1:length(Y)) - 0.1; (1:length(Y)) + 0.1]';

[~, xj] = sort(J);

f1 = figure;
if strcmp(version('-release'), '2016b')
    errorbar(X, Y(xj, :), E(xj, :), 'linestyle', 'none', 'linewidth', 2, 'capsize', 0);
else
    errorbar(X, Y(xj, :), E(xj, :), 'linestyle', 'none', 'linewidth', 2);
```

```
end
set(gca, 'xlim', [0, length(Y) + 1])
legend('Unlimited oxygen uptake', 'Limited oxygen uptake', 'location', 'northoutside',
       'orientation', 'horizontal')
xlabel('Reaction')
ylabel('Flux range (mmol/gDW/h)')

yyaxis right
plot(J(xj))
ylabel('Jaccard index')
```

Sampling

CHRR can be called via the function `sampleCbModel`. The main inputs to `sampleCbModel` are a COBRA model structure, the name of the selected sampler and a parameter struct that controls properties of the sampler used. In the instance of CHRR, two parameters are important: the sampling density (`nStepsPerPoint`) and the number of samples (`nPointsReturned`). The total length of the random walk is `nStepsPerPoint*nPointsReturned`. The time it takes to run the sampler depends on the total length of the random walk and the size of the model [2]. However, using sampling parameters that are too small will lead to invalid sampling distributions, e.g.,

```
options.nStepsPerPoint = 1;
options.nPointsReturned = 500;
```

An additional on/off parameter (`toRound`) controls whether or not the polytope is rounded. Rounding large models can be slow but is strongly recommended for the first round of sampling. Below we show how to get around this step in subsequent rounds.

```
options.toRound = 1;
```

The method outputs two results. First, the model used for sampling (in case of `toRound = 1` this would be the rounded model), and second, the samples generated. To sample the unlimitedOx and limitedOx iPSC_dopa models, run,

```
[P_un, X1_un] =  sampleCbModel(unlimitedOx, [], [], options);
[P_lim, X1_lim] = sampleCbModel(limitedOx, [], [], options);
```

The sampler outputs the sampled flux distributions (X_un and X_lim) and the rounded polytope (P_un and P_lim). Histograms of sampled ATP synthase show that the models are severely undersampled, as evidenced by the presence of multiple sharp peaks.

```
nbins = 20;
[yUn, xUn] = hist(X1_un(ibm, :), nbins);
[yLim, xLim] = hist(X1_lim(ibm, :), nbins);

f2 = figure;
plot(xUn, yUn, xLim, yLim);
legend('Unlimited oxygen uptake', 'Limited oxygen uptake')
xlabel('Flux (mmol/gDW/h)')
```

```
ylabel('# samples')
```

Undersampling results from selecting too small sampling parameters. The appropriate parameter values depend on the dimension of the polytope Ωdefined by the model constraints (see intro). One rule of thumb says to set $\mathrm{nSkip} = 8 * \dim(\Omega)^2$ to ensure the statistical independence of samples. The random walk should be long enough to ensure convergence to a stationary sampling distribution [2].

```
options.nStepsPerPoint = 8 * size(P_lim.A, 2);
options.nPointsReturned = 1000;
```

This time, we can avoid the rounding step by inputting the rounded polytope from the previous round of sampling.

```
options.toRound = 0;
[~, X2_un] = sampleCbModel(unlimitedOx, [], [], options, P_un);
[~, X2_lim] = sampleCbModel(limitedOx, [], [], options, P_lim);
```

The converged sampling distributions for the ATP synthase reaction are much smoother, with a single peak at zero flux.

```
nbins = 20;
[yUn, xUn] = hist(X2_un(ibm, :), nbins);
[yLim, xLim] = hist(X2_lim(ibm, :), nbins);

f3 = figure;
p1 = plot(xUn, yUn, xLim, yLim);
legend('Unlimited oxygen uptake', 'Limited oxygen uptake')
xlabel('Flux (mmol/gDW/h)')
ylabel('# samples')
```

Adding the FVA results to the plot shows that the sampling distributions give more detailed information about the differences between the two models. In particular, we see that the flux minima and maxima are not equally probable. The number of samples from both the unlimitedOx and limitedOx models peaks at the minimum flux of zero, and decreases monotonically towards the maximum. It decreases more slowly in the unlimitedOx model, indicating that higher ATP production is more probable under unlimited oxygen uptake conditions. It is interesting to see that maximum ATP production is highly improbable in both models.

```
ylim = get(gca, 'ylim');
cUn = get(p1(1), 'color');
cLim = get(p1(2), 'color');

hold on
p2 = plot([minUn(ibm), minUn(ibm)], ylim, '--', [maxUn(ibm), maxUn(ibm)], ylim, '--');
set(p2,'color', cUn)
p3 = plot([minLim(ibm), minLim(ibm)], ylim, '--', [maxLim(ibm), maxLim(ibm)], ylim, '--
set(p3, 'color', cLim)
hold off
```

Finally, plotting sampling distributions for six selected iPSC_dopa reactions shows how oxygen availability affects a variety of metabolic pathways.

```
f4 = figure;
position = get(f4, 'position');
set(f4, 'units', 'centimeters', 'position', [position(1), position(2), 18, 27])

sampledRxns = {'r2139', 'GLNSERNaEx', 'HMR_9791', 'r1616', 'r1578', 'r2537'};
rxnsIdx = findRxnIDs(model, sampledRxns);

%ridx = randi(size(model.rxns,1), 1,6);

for i = rxnsIdx
    nbins = 20;
    [yUn, xUn] = hist(X2_un(i, :), nbins);
    [yLim, xLim] = hist(X2_lim(i, :), nbins);

    subplot(3, 2, find(rxnsIdx==i))
    h1 = plot(xUn, yUn, xLim, yLim);
    xlabel('Flux (mmol/gDW/h)')
    ylabel('# samples')
    title(sprintf('%s (%s)', strjoin(model.subSystems{i},';'), model.rxns{i}), 'FontWei

    if find(rxnsIdx==i)==2
        legend('Unlimited oxygen uptake', 'Limited oxygen uptake')
    end

    ylim = get(gca, 'ylim');

    hold on
    h2 = plot([minUn(i), minUn(i)], ylim, '--', [maxUn(i), maxUn(i)], ylim, '--');
    set(h2,'color',cUn)
    h3 = plot([minLim(i), minLim(i)], ylim, '--', [maxLim(i), maxLim(i)], ylim, '--');
    set(h3, 'color', cLim)
    hold off
end
```

References

1. Orth, J. D., Thiele I., and Palsson, B. Ø. What is flux balance analysis? *Nat. Biotechnol.* 28(3), 245-248 (2010).

2. Haraldsdóttir, H. S., Cousins, B., Thiele, I., Fleming, R.M.T., and Vempala, S. CHRR: coordinate hit-and-run with rounding for uniform sampling of constraint-based metabolic models. *Bioinformatics*. 33(11), 1741-1743 (2016).

3. Zhang, Y. and Gao, L. On Numerical Solution of the Maximum Volume Ellipsoid Problem. *SIAM J. Optimiz*. 14(1), 53-76 (2001).

4. Berbee, H. C. P., Boender, C. G. E., Rinnooy Ran, A. H. G., Scheffer, C. L., Smith, R. L., Telgen, J. Hit-and-run algorithms for the identification of nonredundant linear inequalities. *Math. Programming*, 37(2), 184-207 (1987).