University of Vienna

Master Thesis Exposé

Improving Efficiency of Markov Chain Monte Carlo Algorithms for ^{13}C Metabolic Flux Analysis

Richard D. Paul

November 18, 2020

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

1 Introduction

In the field of Metabolomics, Metabolic Flux Analysis (MFA) deals with determining reaction rates in metabolic reaction networks [10]. A metabolic reaction network maps what substrates get turned over into which metabolites. Naively, such a reaction network contains only static information like stoichiometry and possible pathways, but does not contain any dynamic information, i.e. we do not know which reactions happen at what rate or if they happen at all. Figuratively speaking, if we were to feed an organism with some substrate, then we do not expect all products to be there immediately, but instead they get slowly turned over. Also, depending on the substrate and the external stress that we apply on the organism, we do expect the metabolism to behave differently. For example, under certain conditions *Aspergillus nidulans* does start emitting toxins when experiencing nutrient depletion [?].

1.1 An overview

Measuring reaction rates, to which we will from now on refer to as *fluxes*, directly is in general only possible for extracellular fluxes like substrate uptake, CO_2 production or biomass growth. For all other fluxes, it is necessary to infer their values from measurement data. A naming property of ^{13}C MFA is to obtain such measurement data from ^{13}C isotope labelling experiments. The input substrate gets labelled with ^{13}C atoms at predetermined postitions and by time labellings will be distributed over the metabolites. The labellings later can then be measured using nuclear magnetic resonance (NMR) and mass spectrometry (MS). In order to be able to use the isotope labelling experiment data, we need so called atom transition networks. Basic reaction networks usually map molecules to other molecules. Atom transition networks concretize reactions on an atomic level, mapping the single atoms of the substrate molecule to the atoms in the product molecule by position. Given such an atom transition network, an input substrate labelling and a *flux distributions*, which is a vector of all fluxes, this allows for forward simulations of labelling data and hence for computing residuals to quantify the quality of an estimated flux distribution.

Equipped with this residual the problem of determining fluxes becomes an optimization problem, whose solution is also commonly referred to as the *maximum likelihood estimator*. Such point estimators however become rather meaningless in the presence of multiple optima present in the problem at hand. Apart from non-uniqueness of the solution, point estimators fail to come up for measurement noises and hence it is of great interest to also quantify uncertainities that come with likely solutions. A modern and very intuitive approach to obtain such uncertainity quantification is provided within the Bayesian Framework, where credible intervals are constituted using the posterior distribution of the flux distributions given the measurements. We will refer to the particular problem of approximately computing the posterior distribution of the flux distributions as the *flux estimation problem*. This posterior distribution can in general be approximated by using Markov chain Monte Carlo algorithms. These algorithms run a Markov chain constructed such that in the limit it produces samples proportional to the target distribution we desire to compute. In general, Markov chain Monte Carlo algorithms are a well-investigated topic [] with a large variety of different known proposal distributions and approaches to tune them, of which we will discuss some later in this exposé. A distinguishing feature of the parameter estimation problem in ^{13}C Metabolic Flux Analysis is the presence of a linearly constrained parameter space, turning it into a convex polytope on which standard Markov chain Monte Carlo algorithms may work far from optimal [?]. This has led to the development of proposal distributions which are specialized for working on convex polytopes like the Dikin and Vaidya walk samplers [1, 5]. A more detailed overview over proposal distributions and also on the general notion of Markov chain Monte Carlo algorithms is given within Section 2.

1.2 Intended contributions

Apart from the challenge of developing sampling algorithms optimized for working on convex polytopes, the flux estimation problem suffers from costly forward simulations which are needed in the acceptance step of the Markov chain Monte Carlo algorithm. This leads to a natural interest in highly efficient algorithms, where efficiency may be either statiscally or computationally. A quite naive but promising approach seems to be replacing exact but costly forward simulations with rather cheap and fast ones in order to pre-select moves which have a higher acceptance rate, hence reducing the number of costly evaluations for computing steps, which might be rejected anyways. Methods using such cascades of coarse-to-fine models are known as *Multifidelity methods* [3, 8], but to the best of our knowledge have not yet been applied to the field of ^{13}C Metabolic Flux Analysis. A more thorough introduction to this approach will be given in Section 3.

In order to apply the Multifidelity idea, the question arises of how to obtain such lower accuracy models and what pitfalls one may encounter. Some different approaches which seem promising will be outlined in Section 3. On of these approaches includes reducing the measurement data which for which we seek credible parameter sets, possibly leading to the kind of cheaper models we aim for. The key idea behined this approach is that any estimate which is able to explain the complete data should also be able to explain only parts of the data.

Another idea is to use regression models of pre-run samples closely following [2]. The obvious problem when fitting approximate models on generated samples is that the samples taken from a pre-run have no guarantee to be actually representative of the target function if the pre-run did not run until convergence. However, if it ran until convergence, then the problem at hand is already solved, leaving the idea of approximate models superfluous. The issue might be prevented by introducing a trust criterion based on which we decide whether the model fitted on the non-convergent pre-run is safe to use or not. Distrusting the approximate model we switch back to generate proposals without a low fidelity pre-filtering.

The aim of this Master's Thesis is to examine the possibilities of employing Multifidelity methods to the flux estimation problem as well as developing and implementing such methods in detail. This shall further lead to the development of guidelines for employing such models based not only on experimental but also theoretical considerations. Of particular interest will be the trust criterion approach as it seems not yet much investigated. Also, Müller et al. [2], who proposed this approach, apply it only for computing the gradient used in their proposal distribution. To the best of our knowledge there have been no attempts so far to connect this approach with the Multifidelity methods mentioned above. The proclaimed goal of this work is to considerably accelerate existing sampling methods especially for the scenario of isotopically nonstationary measurements, which will be discussed in some more detail in Section 2.3.

2 Preliminaries

In the following section we give a simplified and partly superficial overview on the key approaches to model metabolic networks and their reaction rates as well as on the very basics of Bayesian parameter estimation and Markov chain Monte Carlo algorithms.

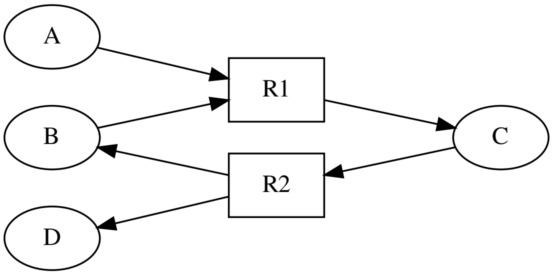

Figure 1: The reaction network described by the chemical reaction equations 1 as a bipartite graph.

2.1 Stoichiometric modelling

The metabolism of an organism is defined by a set of biochemical reactions which turn substrates into energy, proteins, lipids and further "building blocks" of the organism, as well as other possibly desired compounds e.g. toxins. Along so called metabolic pathways different substrates will be turned into new products which then again act as substrates for further reactions. An intuitive way of handling the set of metabolic reactions is via reaction networks. A *reaction network* forms a directed bipratite graph $G = (M \cup R, E)$ consisting of a node set M representing the metabolites, i.e. the chemical compounds included throughout the metabolism, a node set R representing the reactions and an edge set $E \subseteq M \times R$ of directed edges, i.e. $(v, u) \neq (u, v) \in E$. An alternative but equivalent way to interpret the reaction network is to regard it as a hypergraph, where the node set consists of the metabolites only and the directed hyperedges form the reactions. The previously described bipartite graph is then called the incidence or Levi graph of the latter.

We introduce some basic concepts by the means of the following exemplary chemical reaction equations including three not further specified compounds A, B, C and D

$$\begin{aligned} R_1: \quad A + B &\rightarrow C \\ R_2: \quad C &\rightarrow B + D \end{aligned} \tag{1}$$

Its reaction network looks as depictured in Figure 1.

Given such a reaction network described by its reaction equations, we can also set up a *stoichiometric matrix* $\mathbf{S} \in \mathbb{N}^{m \times n}$, where $m = |M|$ is the number of compounds present in the reaction network and $n = |R|$ the number of reactions in the network. The entries s_{ij} of $\mathbf{S}$ are then given as the difference of stoichiometric coefficients on the input and output sides of the i-th compound in the j-th reaction. Note that sometimes the stoichiometric matrix is also defined as the transposed of $\mathbf{S}$. For a general reaction equation with compounds $M = (M_1, M_2, \ldots, M_m)^T$ and n reactions

$$R_j : \sum_{i=1}^{m} k_{ij} M_i \rightarrow \sum_{i_1}^{m} \ell_{ij} M_i, \qquad \forall R_j \in R \tag{2}$$

the entries of the stoichiometric matrix therefore are $s_{ij} = k_{ij} - \ell_{ij}$. The stoichiometric matrix for our exemplary reaction equations is given by

$$\mathbf{S} = \begin{pmatrix} -1 & 0 \\ -1 & 1 \\ 1 & -1 \\ 0 & 1 \end{pmatrix} \tag{3}$$

Note that no catalystic effects can be described by the stoichiometric matrix, since the difference of left and right hand side stoichiometric coefficients will always cancel out catalysts.

We now extend our formalism in order to take into account fluxes. The fluxes we wish to determine are each associated with one of the reactions. In the following we denote the fluxes as the vector $\mathbf{v} = (v_i, v_2, \ldots, v_m)^T \in \mathbb{R}^m$, where v_i is the reaction rate associated to reaction $R_i \in R$. Under the assumption, that all metabolites are available plenty and reaction rates happen at a constant rate, we can define the time derivative of the metabolite concentration $\mathbf{x} = (x_1, \ldots, x_n)^T$ where $x_i \in M$ as

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = \mathbf{S}\mathbf{v} \tag{4}$$

which intuitively describes the rate of change of each metabolite as sum of the in and out fluxes. By introducing the so-called steady state condition, we can restrict the space of possible fluxes to the solution of the linear system

$$\mathbf{S}\mathbf{v} = \mathbf{0} \tag{5}$$

From a biological point of view the steady state condition puts the system into a metabolic steady state also known as homeostasis, where metabolite concentrations do not change anymore because the in and out fluxes cancel each other out. In general, there exist more reactions than compounds in realisitic metabolic models [10, 7], i.e. $m > n$ which immediately leads to the linear system in 5 to be underdetermined. By the rank-nullity theorem, we have that the dimension of the null space of $\mathbf{S}$ is $k = \mathrm{nullity}(S) = m - \mathrm{rank}(S)$. It follows that if we somehow determine k linearily independent flux values, the rest can be computed from them as

$$\mathbf{v} = \mathbf{v}(\boldsymbol{\omega}) = \mathbf{K}\,\boldsymbol{\omega}$$

where $\boldsymbol{\omega} \in \mathbb{R}^k$ are the *free fluxes* and $\mathbf{K}$ is a basis of the nullspace of $\mathbf{S}$.

From a biological perspective, not every flux distribution $\mathbf{v}(\boldsymbol{\omega}) \in \mathbb{R}^m$ is biologically feasible or meaningful, e.g. upper limits on the absolute value or on the directionality of reactions may be known from thermodynamic considerations. Also, usually the substrake uptake and other extracellular fluxes like CO_2 production of the metabolism under consideration are directly measurable, allowing to fix their values or at least bound them within some narrow interval accounting for measurement noise. Taking these linear constraints of the form $\hat{\mathbf{A}}\mathbf{v} \leq \mathbf{b}$ into account, we obtain a convex polytope

$$\mathcal{V} := \left\{ \boldsymbol{\omega} \in \mathbb{R}^k : \mathbf{A}\,\boldsymbol{\omega} \leq \mathbf{b} \right\} \tag{6}$$

with $\mathbf{A} := \hat{\mathbf{A}}\mathbf{K}$, to which we will refer to as the *flux polytope*. In general, this flux polytope has a strictly positive volume, meaning that it usually does still not admit a unique solution to (5).

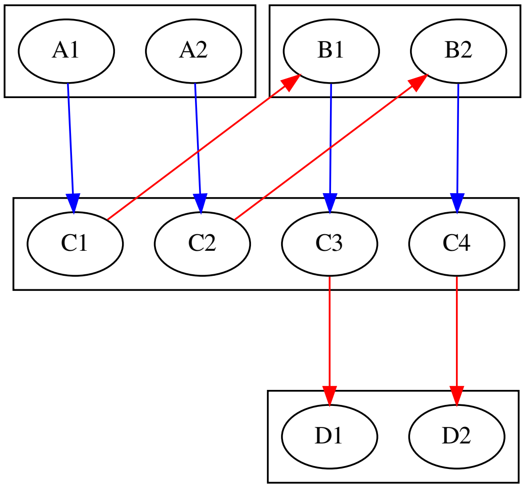

Figure 2: Atom transition network of the toy example 1. Reactions are now color-encoded, blue stands for reaction r_1, red stands for reaction r_2.

2.2 Atom transition networks

The technique of Flux Balance Analysis tries to obtain an unique flux distribution by assuming that the organism under consideration has been optimized by evolution for some specific purpose e.g. growth or substrate uptake. These assumptions translate into linear objective functions on the flux polytope, reducing the problem of obtaining an unique flux distribution to an easy to solve linear program. However the assumption made that the organism under consideration is optimized for some specific purpose is very restrictive and may especially not apply when using unusual cultivation regimes or *engineered* organisms, which may never have felt the pressure of evolution.

Flux Balance Analysis does not need any measurement data at all but only the static reaction network, whereas ^{13}C Metabolic Flux Analysis on the other hand relies heavily on measurement data. A naming property of ^{13}C MFA is to obtain such measurements from ^{13}C isotope labelling experiments. Note that in general labellings are not restricted to carbon atoms only, but forming the basis of life on earth and being the backbone of most biochemical molecules they are a very natural choice. The input substrate gets labelled with ^{13}C atoms at predetermined postitions and by time the labellings will be distributed across the metabolites. The labellings can then be measured using nuclear magnetic resonance (NMR) and mass spectrometry (MS). In order to be able to use the isotope labelling experiment data, we need so called atom transition networks, which as we will see will enable us to simulate isotope labelling experiments and hence lead to a naturally arising residuum, which we can then aim to minimize. Basic reaction networks usually map molecules to other molecules, atom transition networks concretize reactions on an atomic level, mapping the single atoms of the substract molecule to the atoms in the product molecule by position.

Let us reconsider our previous example reaction network and add some carbon atom information to the compounds as well as transition information to the reactions. Assume that the compounds A, B and D consist of two carbon atoms each and C of four and consider the atom transitions as depictured in Figure 2. Similarily to Section 2.1 we are interested in the rates of change of the now labelled metabolites. Let a_i, b_i, d_i and c_k denote the different fractions of labeled metabolites A, B, D and C respectively, where $i = 0, \dots, 3$ and $k = 0, \dots, 15$ since every carbon atom can either be labelled or not, hence yielding a total of 2^κ possible different labelling states for every compound with κ carbon atoms. It is common to assume that the reactions on the isotopomer level follow a law of mass action, meaning that the reaction rates are proportional to the concentrations of the reactants.

Consider compound C and in an abuse of notation let C also denote the size of the metabolite pool. Further let $\mathbf{v} = (v_1, v_2)^\top$ be a flux distribution to our two reaction toy example, then we get the following nonlinear differential equation

$$C \frac{\mathrm{d}}{\mathrm{d}t} c_k = v_1 a_i b_j - v_2 c_k, \qquad i, j = 0, \dots, 3, \quad k = 0, \dots, 15 \tag{7}$$

yielding a total of 16 equations for compound C. The nonlinear term $v_1 a_i b_j$ can be interpreted as the probability that a metabolite unit of the metabolite pool A in labelling state i and a metabolite unit of the pool B in the labelling state j "hit" each other and react scaled by the reaction rate v_1. Similar equations to (7) may be derived for all other metabolite pools A, B and D, which are then known as the *mass balance equations*, which we denote as

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathbf{y} = g(\mathbf{v}, t) \tag{8}$$

in the general case. These equations form a dynamic system of nonlinear ordinary differential equations, which has been proven to always converge towards a unique equilibrium [11]

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathbf{y} = g(\mathbf{v}, t) = 0 \tag{9}$$

where $\mathbf{y} \in I$ is the vector of all isotopomers of every metabolite in the system and we denote the isotop labelling space $I \subset \mathbb{R}^\eta$, where

$$\eta = \sum_i^m 2^{\kappa_i}$$

where κ_i is the number of carbon atoms in metabolite M_i. The condition (9) describes the so called *isotopic steady state.* Notice that the pool sizes as e.g. given in the differential equation (7) are constant since we consider the metabolic steady state. Hence the pool sizes vanish in equation (9) Given an atom transition network, an input substrate labelling and a flux distribution, we are now able to perform simulations of labelling data.

2.3 Forward simulations

Let $\hat{\mathcal{V}} = \{\mathbf{K}\boldsymbol{\omega} : \boldsymbol{\omega} \in \mathcal{V}\}$ be the space of all feasible flux distributions. Forward simulations in ^{13}C MFA are maps $f : \hat{\mathcal{V}} \times [0, \infty] \to \mathcal{I}$ from the feasible flux space $\hat{\mathcal{V}}$ and a given time $t \in [0, \infty]$ to the isotope labelling space $\mathcal{I}$. If a flux distribution $\mathbf{v} \in \hat{\mathcal{V}}$ is given, then the forward simulation allows us to predict isotope labelling values at any time $t \in [0, \infty]$ and to compare the result with actual measurements to assess the quality of our flux distribution $\mathbf{v}$. In the following we denote $\mathbf{y} = f(\mathbf{v}, t) \in \mathcal{I}$. For isotopic steady-state scenarios, we set $t = \infty$ and write $\mathbf{y} = f(\mathbf{v}, \infty) = f(\mathbf{v})$, which is why we included ∞ explicitly in the previous definition of f. A forward simulation can then be computed by solving the algebraic system

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y} = \frac{\mathrm{d}}{\mathrm{d}t} f(\mathbf{v}) = g(\mathbf{v}, t) = 0$$

where $\mathrm{d}\mathbf{y}/\mathrm{d}t$ are the time derivatives of the isotope labellings, given by the mass balance equations $g(\mathbf{v}, t)$ defined as in Equation 8. As stated before, this nonlinear system admits a unique solution for the kind of problems which we are dealing here [11].

For instationary scenarios, i.e. problems where the real measurements where taken before the isotopically steady state was reached, the measurements are complemented with timestamps of the measurement. Hence, to obtain a forward simulation at the particular time t_0 of a measurement, the time derivatives have to be integrated up the desired time t_0 to obtain

$$\mathbf{y}(t_0) = \int_0^{t_0} g(\mathbf{v}, t)\,\mathrm{d}t$$

which is the isotope labelling data predicted at time t_0.

Since, as discussed before, the flux distribution is uniquely determined by the k free fluxes $\boldsymbol{\omega}$, we can defined $\mathbf{y}$ also as a function of the free fluxes, i.e.

$$\mathbf{y}(t) = f(\mathbf{K}\,\boldsymbol{\omega}, t)$$

which abusing notation we will from here on also simply denote as $\mathbf{y}(t) = f(\boldsymbol{\omega}, t)$.

2.4 Estimating flux distributions

Now that we are equipped with the ability to simulate isotope labelling experiments, suppose we are given an isotope labelling measurement $\boldsymbol{\mu}$ and its respective timestamp $t_{\boldsymbol{\mu}}$. Intuitively speaking, we want to assess if the simulated measurements will be close to the actual measurements using a given flux dsitribution $\mathbf{v}(\boldsymbol{\omega})$. If we now pick an arbitrary flux distribution $\boldsymbol{\omega}$ from our flux polytope $\mathcal{V}$, we can perform the forward simulation and receive a measurement prediction $\mathbf{y} = f(\boldsymbol{\omega}, t_{\boldsymbol{\mu}}) = f(\boldsymbol{\omega})$, where we use the latter as a short hand notation when $t_{\boldsymbol{\mu}}$ is clear from context. By taking the sum of squares residual

$$r_{\boldsymbol{\mu}}(\boldsymbol{\omega}) \;=\; \big\| f(\boldsymbol{\omega}) - \boldsymbol{\mu} \big\|^2_{\boldsymbol{\Sigma}^{-1}} \;=\; \big\| \mathbf{y} - \boldsymbol{\mu} \big\|^2_{\boldsymbol{\Sigma}^{-1}} \;=\; \big(\mathbf{y} - \boldsymbol{\mu}\big)^{\top} \boldsymbol{\Sigma}^{-1} \big(\mathbf{y} - \boldsymbol{\mu}\big) \tag{10}$$

we obtain a measure for the quality of our flux distribution, where $\boldsymbol{\Sigma}$ contains the squared standard errors of the measurements on the diagional axis. Since measurements may be assumed to be independent from each other, $\boldsymbol{\Sigma}$ is diagonal in our case and computing the inverse is trivial. The problem

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega} \in \mathcal{V}} r_{\boldsymbol{\mu}}(\boldsymbol{\omega}) \tag{11}$$

is equivalent to the maximum likelihood estimation problem, if we consider the sum of squares residual as proportional to the negative log-likelihood. Such point estimators have to be considered with caution as the measurement data we are provided comes with measurement noise for which a point estimator cannot come up for. Intuitively speaking, the maximum likelihood estimator provides us the flux most likely to explain the data, assuming the data is correct. This being said, it is a natural goal to quantify the uncertainity introduced by the noise as well. Bayesian statistics, which have been on the rise in the last decades, give us an intuitive way of quantifying the uncertainity related to various issues, that are found within our modelling approach.

In a Bayesian framework, we are interested in the *posterior distribution* $\mathbb{P}(\boldsymbol{\omega} \,|\, \boldsymbol{\mu})$ which gives us the probability of a parameter based on the data. From Bayes' Theorem follows that we can compute this posterior distribution as

$$\mathbb{P}(\boldsymbol{\omega} \,|\, \boldsymbol{\mu}) = \frac{\mathbb{P}(\boldsymbol{\mu} \,|\, \boldsymbol{\omega})\, \mathbb{P}(\boldsymbol{\omega})}{\mathbb{P}(\boldsymbol{\mu})} \propto \mathbb{P}(\boldsymbol{\mu} \,|\, \boldsymbol{\omega})\, \mathbb{P}(\boldsymbol{\omega}) \tag{12}$$

where $\mathbb{P}(\boldsymbol{\mu} \,|\, \boldsymbol{\omega})$ is the *likelihood function* and $\mathbb{P}(\boldsymbol{\omega})$ is our *prior* belief. In our case, this prior belief is that the flux distribution should lie inside the flux polytope $\mathcal{V}$, i.e.

$$\mathbb{P}(\boldsymbol{\omega}) = \begin{cases} \varphi & \text{if } \boldsymbol{\omega} \in \mathcal{V} \\ 0 & \text{else} \end{cases} \tag{13}$$

but since any of these flux distributions seems equally likely *a priori* we choose φ such that $\int_{\mathcal{V}} \mathbb{P}(\boldsymbol{\omega}) \mathrm{d}\, \boldsymbol{\omega} = 1$. In words, we assume that the flux distribution is uniformly distributed within $\mathcal{V}$. The likelihood function is a model which relates the unknown parameters and the given data. Intuitively speaking it translates how likely it would be to observe the outcome of the data μ given that $\boldsymbol{\omega}$ where the true parameters. A common choice is to assume that measurements are normally distributed, resulting in the likelihood function

$$\mathbb{P}(\boldsymbol{\mu} \,|\, \boldsymbol{\omega}) = \frac{1}{Z} \exp\left\{ -\frac{1}{2} r_{\boldsymbol{\mu}}(\boldsymbol{\omega}) \right\} = \frac{1}{Z} \exp\left\{ -\frac{1}{2} \big(f(\boldsymbol{\omega}, t_{\boldsymbol{\mu}}) - \boldsymbol{\mu} \big)^{\top} \boldsymbol{\Sigma}^{-1} \big(f(\boldsymbol{\omega}, t_{\boldsymbol{\mu}}) - \boldsymbol{\mu} \big) \right\} \tag{14}$$

where $Z = \sqrt{(2\pi)^{\eta} \det \Sigma}$ and m is the dimension of the isotope labelling space. To point out that the likelihood function is a function of the parameter vector $\boldsymbol{\omega}$ and not of the measurement $\boldsymbol{\mu}$, we denote it also as $\ell_{\boldsymbol{\mu}}(\boldsymbol{\omega}) \coloneqq \mathbb{P}(\boldsymbol{\mu} \,|\, \boldsymbol{\omega})$. Note that the likelihood function in (14) is not a closed-form expression, since f requires computation of either an algebraic system of equations or an integral. Also, it is therefore not possible to directly draw samples from this distribution. Instead such problems are tackled by employing Markov Chain Monte Carlo algorithms. The key idea is to construct a Markov Chain, whose stationary distribution converges to the desired target function. Some key concepts and the derivation of the particular Markov chain for our purposes are given in the following section.

2.5 Markov chain Monte Carlo algorithms

A Markov chain is a stochastic process $(X_n)_{n \geq 0}$ on a state space S, which is defined by a starting distribution λ and a transition kernel $\mathcal{T} : S \times S \to [0, 1]$, for which

$$\sum_{j \in S} \mathcal{T}(i, j) = 1, \quad \forall i \in S$$

holds and which contains the transition probabilities, i.e.

$$\mathbb{P}(X_{n+1} = j | X_n = i, X_{n-1} = x_{n-1}, \ldots, X_0 = x_0) = \mathcal{T}(i, j)$$

which intuitively means that the transition probability from the state at time n to $n+1$ does not dependent on the chains history before n. A distribution $\pi : S \to [0,1]$ is said to be an *invariant* or *stationary measure* of $\mathcal{T}$, if

$$\pi \mathcal{T} = \pi \quad \Leftrightarrow \quad \sum_{i \in S} \pi(i)\, \mathcal{T}(i,j) = \pi(j)$$

and *invariant* or *stationary distribution* if also $\sum_{i \in S} \pi(i) = 1$ holds. A sufficient condition for π to be invariant, is

$$\pi(i)\, \mathcal{T}(i,j) = \pi(j)\, \mathcal{T}(j,i) \tag{15}$$

which is widely known as the *detailed balance condition.* Assuming detailed balance and taking the sum over $i \in S$ on both sides we obtain

$$\sum_{i \in S} \pi(i)\, \mathcal{T}(i,j) \; = \sum_{i \in S} \pi(j)\, \mathcal{T}(j,i) \; = \; \pi(j) \sum_{i \in S} \mathcal{T}(j,i) \; = \; \pi(j)$$

which shows that π indeed is invariant under $\mathcal{T}$. Our goal in order to approximate the posterior distribution $\mathbb{P}(\boldsymbol{\omega} \,|\, \boldsymbol{\mu})$ is to construct a Markov chain such that it has $\mathbb{P}(\boldsymbol{\omega} \,|\, \boldsymbol{\mu})$ as its unique invariant distribution, to which the detailed balance equation is a key concept. Hastings [4] made the following ansatz,

$$\mathcal{T}(i,j) := \begin{cases} p(i,j)a(i,j) & \text{if } i \neq j \\ 1 - \sum\limits_{k \neq i} p(i,k)a(i,k) & \text{if } i = j \end{cases}$$

where $p(i,j)$ is the *proposal distribution* and $a(i,j)$ the *acceptance probability.* Note that detailed balance holds trivially for $\mathcal{T}(i,i)$. Plugging the ansatz for $i \neq j$ into the the detailed balance equation, we obtain

$$\pi(i)p(i,j)a(i,j) = \pi(j)p(j,i)a(j,i) \quad \Longleftrightarrow \quad \frac{a(i,j)}{a(j,i)} = \frac{\pi(j)p(j,i)}{\pi(i)p(i,j)}$$

where using the *Metropolis criterion*

$$a(i,j) := \min \left\{ 1, \frac{\pi(j)p(j,i)}{\pi(i)p(i,j)} \right\} \tag{16}$$

indeed satisfies detailed balance. The distribution for which we want to generate samples is our posterior distribution as defined in (12) and hence for $\boldsymbol{\omega}_n \in \mathcal{V}$ we obtain the Metropolis criterion for our Markov chain as

$$\begin{aligned} a(\boldsymbol{\omega}_n, \boldsymbol{\omega}_{n+1}) &= \min \left\{ 1, \frac{\mathbb{P}(\boldsymbol{\omega}_{n+1} \,|\, \boldsymbol{\mu})\, p(\boldsymbol{\omega}_{n+1}, \boldsymbol{\omega}_n)}{\mathbb{P}(\boldsymbol{\omega}_n \,|\, \boldsymbol{\mu})\, p(\boldsymbol{\omega}_n, \boldsymbol{\omega}_{n+1})} \right\} \\ &= \min \left\{ 1, \frac{\ell_{\boldsymbol{\mu}}(\boldsymbol{\omega}_{n+1})\, p(\boldsymbol{\omega}_{n+1}, \boldsymbol{\omega}_n)}{\ell_{\boldsymbol{\mu}}(\boldsymbol{\omega}_n)\, p(\boldsymbol{\omega}_n, \boldsymbol{\omega}_{n+1})} \right\} \end{aligned} \tag{17}$$

We remark some convenient properties of the Metropolis criterion. The normalization constant $\mathbb{P}(\boldsymbol{\mu})$, the prior $\mathbb{P}(\boldsymbol{\omega})\big|_{\mathcal{V}} = \varphi$ as well as the constant Z introduced in (14) are cancelled out here. This is especially interesting as we avoid explicit computation of the high-dimensional integrals

$$\mathbb{P}(\boldsymbol{\mu}) = \int_{\mathcal{V}} \mathbb{P}(\boldsymbol{\mu} \,|\, \boldsymbol{\omega})\, \mathbb{P}(\boldsymbol{\omega}) \mathrm{d}\, \boldsymbol{\omega} \qquad \text{and} \qquad \mathbb{P}(\boldsymbol{\omega})\Big|_{\mathcal{V}} = \varphi = \frac{1}{\int_{\mathcal{V}} \mathrm{d}\, \boldsymbol{\omega}} \tag{18}$$

However note that every evaluation of the Metropolis criterion (17) requires performing a forward simulation in order to compute the likelihood function $\ell_{\boldsymbol{\mu}}(\boldsymbol{\omega}_{n+1})$, even for moves that might be rejected, which is an issue we will revisit later. In general, our Metropolis criterion (17) becomes undefined for $\boldsymbol{\omega}_n \notin \mathcal{V}$ as then $\mathbb{P}(\boldsymbol{\omega}_n) = 0$ (cf. (13)). Assuming that $\boldsymbol{\omega}_n \in \mathcal{V}$ however, we have that for every $\boldsymbol{\omega}_{n+1} \notin \mathcal{V}$,

$$\mathbb{P}(\boldsymbol{\omega}_{n+1}) = 0 \quad \Longrightarrow \quad a(\boldsymbol{\omega}_n, \boldsymbol{\omega}_{n+1}) = 0$$

and hence if we require

$$\operatorname{supp}(\lambda) \subseteq \mathcal{V}$$

for our starting distribution λ, meaning that we will definitely start the Markov chain inside our parameter space $\mathcal{V}$, we can guarantee that it will never escape from $\mathcal{V}$. More formally, any *closed communicating classes* of our Markov chain will lie within $\mathcal{V}$, where we refrain from giving precise definitions and instead refer to [6].

A Markov chain is said to be *irreducible*, if its state space forms on single closed communicating class, which basically means that every state is reachable from every other state in a finite number of steps. If a Markov chain is irreducible, then its invariant measure is unique up to scaling. Irreducibility is achieved by using an appropriate proposal distribution. In fact proposal distributions that satisfy irreducibility are easily obtained and most naive approaches will do so out of the box. In the following we quickly present some frequently used proposal distributions along with their strengths and weaknesses.

2.5.1 Proposal distributions

quickliy present a hand full of proposal distributions and issues with rejection-based moves

Ball Walk The ball walk uniformly picks a point y from the k-dimensional ball $\mathcal{B}$ with radius δ and midpoint x, which is our current state. If y lies outside $\mathcal{V}$, which can happen when $\mathcal{B} \cap \mathcal{V} \neq \emptyset$, the move gets automatically rejected. If $y \in \mathcal{V}$, then the move gets accepted or rejected with probability $a(x, y)$, where $a(\cdot, \cdot)$ is the Metropolis criterion from (17).

This proposal move is in a sense rejection based as it does not take into account the structure of the state space $\mathcal{V}$ at all and creates infeasible moves. In an unfortunate situation, e.g. when x sits close to the boundaries in a pointy corner of a high-dimensional poltyope, the fraction of $\mathcal{B}$ which lies outside $\mathcal{V}$ can be vast hence also most probably leading to a vast number of infeasible proposals. This problem is especially striking in high dimensions and very much comparable to the phenomenon of the ratio of the volume of an inscibed hypersphere to the volume of a hypercube exponentially decaying to zero as dimensions increase.

Hit and Run Hit-and-Run proposals "shoot" a chord from its current position $\mathbf{x}$ into some uniformly randomly picked direction $\mathbf{d}$ and move along this chord $\mathbf{y} = mbx + \alpha\mathbf{d}$. The size of the step α can be controlled by controlling the distribution with which a point along the ray is picked. Apart from using an uniform distribution on the chord, one can also choose an one-dimensional normal distribution with the current state as mean. This will promote shorter steps, which usually contain less risk of being rejected at the cost of being more correlated and thus resulting in less statiscal efficiency. Note that computing the intersections of the chord $\mathbf{x} + \alpha\mathbf{d}$ with the boundaries of the polytope can be easily achieved by taking the minimal and maximal values from the vector

$$\frac{\mathbf{b} - \mathbf{A}\mathbf{x}}{\mathbf{A}\mathbf{d}}$$

for the parameter α, where the fraction denotes an elementwise fraction and $\mathbf{A}$ and $\mathbf{b}$ describe the linear constraints of our polytope $\mathcal{V}$ as in (6).

Dikin Walk

CSmMALA

2.6 The challenge

In general, the process of solving the inverse problem of estimating flux distributions using Markov chain Monte Carlo algorithms is well-defined. One major difficulty however is the usage of isotopically instationary measurements within this framework, because, as we mentioned before, every step of the Markov chain requires an evaluation of the forward simulation. For isotopically instationary scenarios, this forward simulation turns into the problem of numerically integrating a system of ordinary differential equations, which for some flux distributions turns into a *stiff* problem. Although numerical solutions to ordinary differential equations are a well-investigated topic, in the presence of stiff problems the simulation time can go up significantly [?], making flux estimation in isotopicaly instationary scenarios expensive. Obviously speeding up forward simulations would be a natural target, however regarding the long presence of numerical solvers for ordinary differential equation and the fact that dedicated work has already been put into accelerating solvers and methods for ordinary differential equations derived from reaction networks [9], this approach does not appear too fruitful.

Another approach, on which we will focus within this exposé is to instead reduce the number of forward simulations needed.

from here on still very much a draft

One possible way to avoid its evaluation is to approximate its value e.g. by using regression models. The natural question, which arises is on what data to fit the regression model. Creating samples is not only costly, but the samples need to represent the true underlying function and hence would have to be taken from a convergent Markov chain. This again contradicts our goal to run exactly such a Markov chain using the regression model.

On the other hand, computing approximate models online, i.e. while sampling, would violate the fundamental Markov property destroying the theoretical basis for the convergence of our chain. The field of adaptive Markov chains deals with this issue, where the proposal distribution is adapted online from the chains history, but in a fashion that keeps convergence guaranteed.

3 Multifidelity methods

3.1 Reduced networks from reduced measurements

Reduce networks by reducing measurements and hence omit metabolites which do not affect the measured pools at all. Less measurements however lead to fluxes, which are only found in the backtrace of the omitted measurements to become structurally non-identifiable. The question arises how many measurements should be omitted to obtain a good tradeoff between computational cost and drawbacks from increased non-identifiabilities. It has to be investigated more closely how non-identifiabilities propagate and what effect they have on the adapted proposal step using the low-fidelity model.

Identifying the reduction in size of the network from omitting measurements is most probably feasible. Doing so allows to control speedup and degree of non-identifiability introduced.

3.2 Stationary and instationary scenarios

Combine stationary and instationary scenarios, where stationary scenarios act as low-fidelity method. Often only one or the other is available from an isotope labelling experiment. Stationary values however might be extrapolated using time-series analysis from instationary data.

3.3 Regression models on pre-run data

Pre-runs allow to generate data points from the posterior distribution on which approximate models can be computed. Such pre-runs will not be run until convergence as this would make any follow up run superfluous. Therefore the samples produced will have no guarantee of being distributed according to the posterior distribution. This means, close to samples of the pre-runs the approximate model will reflect the function $\ell_{\boldsymbol{\mu}}(\boldsymbol{\omega})$ quite well, further away arbitrary large errors might be introduced.

[2] propose to use a trust criterion where the approximate model is not used if moves got rejected more than N times in a row and used again if M moves got accepted in a row using the accurate MALA proposal step. In the case where the approximate model is distrusted, the N rejected moves get removed from the set of samples. [2] do not use their approach in a multifidelity framework but in an alternating manner. The multifidelity method using an intermediate low-fidelity step if the approximate model is trusted and no intermediate step if the approximate model is currently distrusted can be seen as two different transition kernels which have the same stationary distribution as a target distribution but are used mixed. It seems provable that this mixed usage of these transition kernels is permitive under mild conditions such as irreducibility of both transition kernels and given that a low-fidelity approximation may be arbitrarily bad without threatening convergence.

Gaussian process regression delivers standard deviation intervals at every evaluated point and hence is basically already shipped with a natural trust criterion, where one can for example set a hard threshold on which to decoed if the model is used or not. Also it could be randomly decided whether the regression model is used or not with probability proportional to the precision of the estimate.

3.4 Combination with multistage approaches

The cheap evaluation of coarse model functions also allows for delayed-rejection schemes, where a proposal rejected by the coarse model can be corrected a few times before being assessed by the high-fidelity model. These stacked proposal corrections are even allowed to use the previously rejected proposals in order to optimize the current proposal. In the end however a single new sample is produced and hence too many corrections may increase computational cost over what is sensible to actually decrease overall computational costs.

3.5 Theoretical considerations

What conditions do low-fidelity models have to meet to actually allow for convergence? It might well be that low-fidelity models are allowed to be arbitrarily bad without threatening theoretical convergence properties at least if some basic properties like irreducibility are maintained. Results in [3] hint that the support of the target distribution being contained in the support of the low-fidelity model is a sufficient condition. In practice low-fidelity models with large approximation errors will most likely lead to a reduction of computational efficiency. What conditions should be met to allow for an increase of computational efficiency? Probably conserving monotonicity properties might be a promising condition, since this will lead to a bias in the proposal moves somewhat proportional to the acceptance probability of the accurate model.

from here on still very much a draft

Multi-Stage (or apparently also sometimes called Delayed-Acceptance/Rejection) algorithms use differently costly forward simulations cascaded after each other [4,5]. The idea is to quickly compute a coarse value as an approximate to the finer ones and only consider promising looking areas. A first proposal is accepted or not using the energy function of the coarse simulation result and only if the move gets accepted by the first stage the next finer simulation is evaluated. This saves costly computations of exact solutions for proposal steps which might have been rejected anyway.

The natural questions which arises here is what kind of coarse and fine versions of the model functions to apply. In our case, where a forward simulation consist of solving a system of nonlinear ordinary differential equations using numerical integration a natural choice would be the integration step size or - when using adaptive schemes - the error tolerance. In particular, one could arange a sequence of different error tolerances where each proposal stage solves the system again using a lower tolerance. According to [3], a decrease of accuracy from 1e-3 to 1e-9 might lead to an increase in computational time taken in an order of magnitude.

Further choices, which are more closely related to our domain, would be stationary and instationary measurements, which would form coarse and fine model respectively. However often this data is not available in mixed form, meaning that a isotope labelling experiment only either contains measures from the stationary or instationary phase of the experiment.

Also, MS measurements are easier to simulate than MS-MS measurements, hence one could also apply a two-stage proposal scheme here.

Another, again more general approach, is the application of linear or higher order models to approximate the function at least locally. The question arises what data to use to setup such models. Evaluation of the gradient locally is expensive in general, since it needs exact evaluations of the forward simulation. Such evaluations are obviously available from the previous samples. However, using this data to obtain a model of the forward simulation, we would violate the Markov property.

[2] proposed a method where gradient evaluations are taken from a neural network trained on a pre-run of the chain. The approximate evaluations are used within some trust-criteria, meaning that as soon as one starts to distrust the approximation (e.g. because to many rejections happened), the exact evaluation is used again and as soon as the approximation has matched the

exact evaluation well enough, one switches back to using the approximation. The overall concept can most probably be used with any approximate model of the gradient, not necessarily a neural network, but for instance regression models applied to the prerun data.

An issue one might encounter using this approach, is that a model fitted on a prerun of a chain, which did not run until convergence, may fail to approximate the target function well in some non-covered areas. Although precautions are taken above, by not solely relying on the model, but only within trust criteria, this may obstruct convergence to the correct target function.

4 Adaptive Markov Chains

The previous point about using online models fitted on the chains data leads to the broad field of Adaptive MCMC methods, which enable usage of the chains history to tune the proposal distribution s.t. less proposals get rejected, but sill the whole chain remains convergent.

5 Active Subspaces

Informally speaking, active subspace methods try to identify the subspaces of the parameter space which explain most of the variation of some target function $g : U \to \mathbb{R}$, with $U \subset \mathbb{R}^n$. I presume, that they can also be considered as performing a principal component analysis on the samples generated by a chain and hence identifying the dimensions which explain most of the variance in the data. The value in this information lies in the fact, that a direction, along which the target function does not (or only minimally) change might not be worth exploring at all, since no information can be gained. Parameters, for which its partial derivative is zero over the whole domain, are called non-identifiable. In the case, where the active subspace may not be parallel to any parameter axes, a basis transformation towards an orthogonal basis (like the eigenbasis) might be considered, yielding linear combinations of non-identifiable paramters.

The information on the active subspaces could either be collected and used in an offline or online fashion, again meaning that one either tries to gain this information a priori using e.g. preruns of the chain or other techniques, or on the other hand by computing it while running the Markov chain and integrate the gained knowledge in an adaptive way, s.t. convergence may still be guaranteed.

6 Action Plan

As a starter, an easy and straight-forward strategy could be to consider, implement and test the multi-stage MCMC method using different error thresholds for the adaptive ODE solver as a first proof of concept. Careful software design could enable us to replace the coarse/fine models with arbitrary ones, allowing us to test more combinations from what was mentioned previously.

Preconditioning of MCMC methods using a priori computed data like active subspaces and approximate models combined with the trust criterion approach used in [1] also seem like a powerful, but rather simply implemented approach. Using approximate models of the function to cheaply evaluate its gradient could also be used to improve computational efficiency of gradient-based approaches like the CSmMALA proposal move. However, this most likely seems to remain compatible with the multi-stage approach, possibly allowing again for arbitrary combinations of techniques.

Generally, I consider the techniques above very powerful, since they work on a meta plane above the considerations of efficient proposal distributions on convex polytopes. This to consider the techniques more as a framework, where arbitrary proposal moves, which suffice our needs, may be used.

The true excitement however obviously lies in the field of the adaptive methods, which however might leave the scope of a Master Thesis, considering the broad possiblities already outlined before.

References

[1] Y. Chen, R. Dwivedi, M. J. Wainwright, and B. Yu, *Vaidya walk: A sampling algorithm based on the volumetric barrier*, in 2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2017, pp. 1220–1227.

[2] M. Christian, D. Holger, M. Thomas, and S. Andreas, *A neural network assisted metropolis adjusted langevin algorithm*, Monte Carlo Methods and Applications, 26 (2020), pp. 93–111.

[3] Y. Efendiev, T. Hou, and W. Luo, *Preconditioning markov chain monte carlo simulations using coarse-scale models*, SIAM Journal on Scientific Computing, 28 (2006), pp. 776–803.

[4] W. K. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, 57 (1970), pp. 97–109.

[5] R. Kannan and H. Narayanan, *Random walks on polytopes and an affine interior point method for linear programming*, Mathematics of Operations Research, 37 (2012), pp. 1–20.

[6] J. R. Norris, *Markov Chains*, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 1997.

[7] J. D. Orth, I. Thiele, and B. Ø. Palsson, *What is flux balance analysis?*, Nature Biotechnology, 28 (2010), pp. 245–248.

[8] B. Peherstorfer, K. Willcox, and M. Gunzburger, *Survey of multifidelity methods in uncertainty propagation, inference, and optimization*, SIAM Review, 60 (2018), pp. 550–591.

[9] A. Stratmann and M. Grajewski, *Effiziente numerische simulation von kaskadierten ode-systemen: Design, implementierung und test anhand von zellmodellen in der biotechnologie.* print, 2019. DE-A96: 947301; Besitznachweise: 61 P 19/20-34B; Aachen, FH, Bachelorarbeit, 2020.

[10] W. Wiechert, S. Niedenführ, and K. Nöh, *A Primer to 13C Metabolic Flux Analysis*, John Wiley & Sons, Ltd, 2015, ch. 5, pp. 97–142.

[11] W. Wiechert and M. Wurzel, *Metabolic isotopomer labeling systems: Part i: global dynamic behavior*, Mathematical Biosciences, 169 (2001), pp. 173 – 205.