Notes of Computational Systems Biology

Naoki Sean Pross

January 11, 2025

Abstract

These (rather rough) notes are based on the contents of the course *Computational Systems Biology* taught by Jörg Stelling at ETH Zürich and the book *System Modelling in Cellular Biology* (ISBN 978-0-262-19548-5). However, at the time of writing the author has practically zero biology knowledge. Therefore, these note are very likely to be skewed towards the mathematical content of the course, since that is the only part that he understands.

Contents

Notation

Symbol	Meaning	Other Commonly Used Symbols
Probability Theory		
$x \in \mathcal{X}, y \in \mathcal{Y}$	Random variables x and y, that can take values from $\mathcal{X}$ and $\mathcal{Y}$ respectively.	
$x \mid y$	Random variable x conditioned on y ("x given y").	
$x \perp y$	The random variables x and y are independent.	
$p_x(\bar{x})$	Probability density function of x evaluated at $\bar{x}$. We may sometime simplify the notation and just write $p(x)$ if it clear from the context.	$P(x), \Pr(x), \mathbb{P}(x)$
$p_{x\|y}(\bar{x} \mid \bar{y})$	Conditional probability of x given y.	
$\mathbb{E}\{x\}$	Expected value of x.	$\mathsf{E}(x), \mathbf{E}(x)$
$x(t) \in \mathcal{X}$	Stochastic process. For fixed t we have that $x(t)$ is a random variable.	
Multivariate Calculus		
$\frac{df}{dx}$	Derivative if $x \in \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$. When $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^m$ then it is the total derivative.	f'
$\frac{\partial f}{\partial x}$	Partial Derivative if $x \in \mathbb{R}$ and $f : \mathbb{R}^n \to \mathbb{R}$. When $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ it is the gradient (row vector) of f, and finally if $f : \mathbb{R}^n \to \mathbb{R}^m$ then it is the Jacobian (matrix) of f.	$\partial_x f, \nabla_x f, J_x f$
Graph Theory		
Stoichiometric Network Analysis		
Dynamical Systems and ODE Models		
Stochastic Systems		

Overview

Model / Algorithm	Applications	Advantages	Disadvantages
Graphical Models			
k-Means	Group unstructured data by similarity	• simple and fast	• k usually unknown • sensitive to noise
k-Cores	Group hierarchical data by similarity		
Network Motifs Sampling	Find motifs,		
Global Properties			
Bayesian Networks			
Stoichiometric Network Analysis			
Dynamical System			
Stochastic System			

1 Graph Theory

Graph theory methods are used to describe or infer the topology of the relations in a biological system.

Definition 1.1 (Graph)**.** A graph is a tuple (V, E) with V being set of vertices or nodes and $E \subseteq V \times V$ a set of edges.

Graphs can be directed if edges $(u, v) \in E$ denote an arrow with head u and tail v. Furthermore, graphs can have weighted edges, in which case $E \subseteq V \times V \times S$ where S contains the weights.

For a concrete example, nodes could be proteins and edges interactions between proteins (non-covalent interaction, i.e. complex association / dissociation). Then inferring the graph structure means understanding the functional units in protein-protein interaction networks (network identification).

Note that the number of possible graphs for a given number of nodes is combinatorial, so identifying the graph from data is not easy.

1.1 Clustering

A clustering algorithm groups together "things" that are "similar" in a dataset. To specify what it means for things to be similar we use a *metric*.

Definition 1.2 (Metric)**.** A metric is a distance function $d(x, y)$ that satisfies

- Definiteness $d(x, y) = 0$ iff $x = y$;
- Symmetry $d(x, y) = d(y, x)$;
- A triangle inequality $d(x, z) \leq d(x, y) + d(y, z)$.

Examples of metrics are the Euclidean metric

$$d(x, y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

or the Manhattan (aka City-block, Taxicab, L^1) metric

$$d(x, y) = \sum_{k=1}^{n} |x_k - y_k|.$$

There are many clustering algorithms, and they are all usually under the broader category of *unsupervised classification* in machine learning.

`TODO: hierarchical algo, vs partitioning`

Algorithm 1.1 (k-means Clustering)**.** Given a data set $\{M_j\}_j$, a (guess of the) number of clusters k, and a metric $d(x, y)$ we assign each data point to one cluster C_i such that the clusters are maximally distinct. Formally this means that k-means solves

$$\min \sum_{i=1}^{k} \sum_{j \in C_i} d(M_j, \mu_i), \quad \mu_i = \frac{1}{|C_i|} \sum_{j \in C_i} M_j$$

where the mean μ_i is called *centroid* of the cluster C_i.

function kmeans(k, $\{M_j\}_j$)
  Randomly assign to each data point to a cluster
  repeat
    For each cluster C_i compute its centroid μ_i
    For each data point M_j, assign it the cluster with the closest centroid (closest with respect to the metric)
  until cluster assignments stop changing
  return Clusters $\{C_i\}_{i=1}^{k}$.

Example. Suppose M is a matrix of measurements from a series of experiments for gene expression profiles. The rows of M describe the gene, while the columns are the experimental conditions. Each column is a data point M_j and using k-means clustering, we can partition the experimental conditions into k groups that resulted in similar gene expression levels.

Caveats The number of clusters k is usually unknown. There are ways to estimate it from data. Each data point must be assigned to a cluster, which makes it sensitive to noise.

1.2 Complex Identification

1.2.1 Cliques

This method identifies more complex structures on graphs called cliques.

Definition 1.3 (Clique)**.** A clique of a graph $G = (V, E)$ is a set of vertices $C \subseteq V$ such that every two distinct indices are adjacent, or equivalently, that the subgraph induced by C is complete.

We are interested in finding the cliques of maximal size (largest). However this is an NP-complete problem, and a brute-force approach cannot work because the number of cliques is combinatorial in the number of vertices and edges.

The idea for a heuristic algorithm to find cliques is that since each node is a clique of size 1, successive merging of connected cliques will find large cliques, though there is no guarantee that it will be the largest one.

Moreover, in practice the graphs are usually extracted from experimental data and may thus have missing edges or nodes, which limits the applicability of this method. The next method more robust to these imperfections.

1.2.2 Cores

If instead looking for fully connected subgraph we relax this requirement to have at least k connections we get *cores*.

Definition 1.4 (Degree of a vertex)**.** In a graph $G = (V, E)$ the degree $\deg v$ of a vertex $v \in V$ is the number of edges connected to it.

Definition 1.5 (k-Core)**.** A k-core is a maximum subgraph in which all vertices v have $\deg(v) \geq k$.

Note that cores are not necessarily connected subgraphs.

Algorithm 1.2 (k-Cores)**.** Given a graph $G = (V, E)$ and a (guess on the) number of cores, we find all (nested) cores in G using dynamic programming.

function kcores(k, G)
  Compute the degree of each vertex $v \in V$ and sort V by increasing degree
  for each $v \in V$ **do**
    $\text{core}(v) \leftarrow \deg(v)$
    for each $u \in N(v)$ **do**
      if $\deg(u) > \deg(v)$ **then**
        $\deg(u) \leftarrow \deg(u) - 1$
        Sort V by degree
  return core

The algorithm above has complexity $O(m \log n)$, where $n = |V|$ and $m = |E|$.

Example. Applying k-cores to a graph representing a network of protein-protein interactions in yeast was used to predict functional modules.

The k-cores method can be improved by introducing more local information through *density*. Which roughly speaking is a measure of how much the nodes are connected.

Definition 1.6 (Density of a graph)**.** For a graph G with n vertices and m edges the density of G is $d = \frac{2m}{n(n-1)}$.

An improved version of k-cores weights the vertices by their local density (density of subgraphs).

1.2.3 Network Motifs

Network motifs are patterns of interconnections (subgraphs) that recur in many different parts of a network at frequencies significantly higher than those found in randomized networks. Because these are may be informative structures and it is of interest to be able to count them. For each motif i by counting all (isomorphic) subgraphs in the network we find its number of occurrences N_i and then we can define a density $C_i = N_i/(\sum_j N_j)$.

To extensively search for all occurrences motifs is practically impossible because the number of subgraphs is combinatorial in size of the graph, therefore, since the networks are usually large we use a sampling algorithm to estimate it instead.

Algorithm 1.3 (Network Motifs Sampling)**.** Given a graph $G = (V, E)$ the idea is to pick (sample) a random starting edge and iteratively expand to its neighbors until a n-node subgraph is obtained.

function sample-motif(G, n)
  Let $V_s = \emptyset$ and $E_s = \emptyset$
  repeat
    Pick a random edge $e = \{u, v\} \in E$ and update $E_s \leftarrow \{e\}$, $V_s \leftarrow \{u, v\}$
    repeat
      Let L be the list of edges neighboring E_s excluding edges incident to the nodes in V_s, that is
      $L \leftarrow \{\{u, v\} : \{u, v\} \in E_s \text{ and } u, v \notin V_s\}$
      Pick a random edge $e = \{u, v\} \in L$ and update $E_s \leftarrow E_s \cup \{e\}$, $V_s \leftarrow V_s \cup \{u, v\}$
    until (V_s, E_s) is a n-node subgraph of G
    return Subgraph (V_s, E_s)
  until $L \neq \emptyset$

`TODO: fix control flow and return in function above`

function estimate-motif-density(G, n)
  For each n-node subgraph of type i compute the probability P_i of sampling it from edge e_j based on the permutations S_m of the topology
  $P_i \leftarrow \sum_{\sigma \in S_m} \prod_{E_j \in \sigma} \Pr(E_j = e_j \mid E_k = e_k \forall k \neq j)$
  repeat
    Sample subgraph $H \leftarrow$ sample-motif(G, n)
    Determine motif type of H, and increment counter of associated to motif $S_i \leftarrow S_i + 1/P_i$
  until collected enough samples
  For each motif i compute empirical probability $p_i \approx S_i/(\sum_j S_j)$
  return empirical probabilities $\{p_i\}_i$

1.3 Global Characterizations

For very large networks it is also interesting to look at more more global properties, such as whether the network has a hierarchical or scale-free (or random) structure. We consider a graph $G = (V, E)$.

Degree The average degree

$$\langle k \rangle = \frac{1}{|V|} \sum_{v \in V} \deg(v)$$

and the degree distribution $p(k)$. For random networks $p(k)$ is Poisson distributed, because nodes with a degree that is far from the average are rare. For scale-free networks $p(k)$ is a power law so $p(k) \sim k^{-\gamma}$, because the network is composed of hubs with high connectivity and short paths.

Distance The average shortest path length $\langle \ell \rangle$ is a global network property that indicates navigability. The shortest path length $\ell(u, v)$ between two nodes u, v can be found using algorithms such as breath first search or Dijkstra's algorithm.

Clustering The clustering coefficient $C(u)$ for a node $u \in V$ is the ratio between the number k_u of edges linking nodes adjacent to u and the total possible number of edges among them, so

$$C(u) = \frac{2k_u}{k_u(k_u - 1)}, \quad \text{and} \quad \langle C \rangle = \frac{1}{|V|} \sum_{v \in V} C(v)$$

is the average clustering coefficient which is a measure for the tendency of the network to form groups or clusters. For clustering it also is possible to define a clustering distribution of the nodes $p(c)$.

Example. Metabolic networks have been analyzed using using the above characterizations and through the degree distribution it has been found that many organism have a scale-free metabolic network, which implies the existence of hubs (Water, ADP, Orthophosphate, ATP, $NADP^+$, Pyrophosphate, NAD^+, NADPH, ...). Similarly using path lengths it can be inferred that most metabolic pathways are short. `TODO: what is network diameter.`

1.4 Caveats / Challenges

1. Biochemical reactions may use multiple substrates to generate multiple products. These cannot be modelled with graphs, instead one must use a generalization called hypergraphs. Though for some simple reactions it is possible to decouple the substrates and / or products in the reaction.
2. `TODO: small world characteristics`
3. In some cases such as metabolism the power law (scale-free network properties) emerge from a combination of many underlying distribution. So it could be that the actual structure is "scale-rich" instead of "scale-free", but these models cannot capture it.
4. Data from real world experiments is sampled. The effects of sampling (incomplete information) distorts the distributions.

2 Probabilistic Graphical Models

2.1 Probability Recap

`TODO: definitions, inference, conditioning, joint, independence, baysed theorem`

2.2 Bayesian Networks

A Bayesian network is a graphical probability model that represents a joint probability distribution. It consists of a graph representing the relations between random variables and conditional distributions for each variable. The model is formulated (simplified) by specifying which variables are conditionally independent.

Definition 2.1 (Parents and descendant)**.** For node $v \in V$ of a directed graph $G = (V, E)$ the sets of parents and descendants of v are respectively

$$\text{pa}(v) = \{u : (u, v) \in E\} \subset V,$$
$$\text{de}(v) = \{w : (v, w) \in E\} \subset V.$$

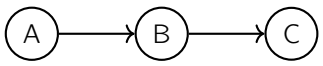
Definition 2.2 (Bayesian Network)**.** Let $G = (V, E)$ be a directed acyclic graph (DAG) and let $\mathbb{X} = \{X_v\}_{v \in V}$ be a set of random variables indexed by V. Then, $\mathbb{X}$ is a Bayesian network with respect to G if it satisfies the local Markovian property

$$p(\bar{X}_v \mid \{\bar{X}_u\}_{u \in V \setminus \text{de}(v)}) = p(\bar{X}_v \mid \{\bar{X}_w\}_{w \in \text{pa}(v)}),$$

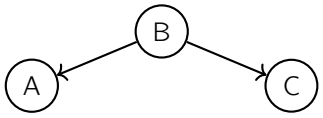
or equivalently for all $v \in V$

$$X_v \perp \{X_u\}_{u \in V \setminus \text{de}(v)} \mid \{X_w\}_{w \in \text{pa}(v)}.$$

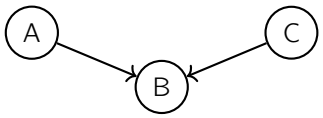
Put into words the local Markov property states that each variable X_v is conditionally independent of its non-descendants $V \setminus \text{de}(v)$ given its parents $\text{pa}(v)$. We consider some simple examples. For a serial connection shown below we have that $X_C \perp X_A \mid X_B$.



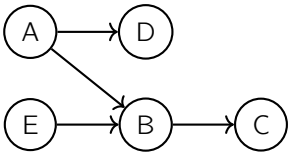
The same condition ($X_C \perp X_A \mid X_B$) can also arise from the following graph with a divergent connection. It is therefore important to note that there can be different graphs that give the same set of independences (formally there is an equivalence class).



The opposite is a convergent connection, in which case both A and C are parents of B, so there are no other non-descendent variables that can be conditionally independent.



Finally a more involved example that describes the independence relations ($X_A \perp X_E$), ($X_B \perp X_D \mid X_A, X_E$), ($X_C \perp X_A, X_D, X_E \mid X_B$), ($X_D \perp X_B, X_C, X_E \mid X_A$).



The resulting expression for the joint probability is then

$$p(\{\bar{X}_v\}_{v \in V}) = p(\bar{X}_A)p(\bar{X}_E)p(\bar{X}_B \mid \bar{X}_A, \bar{X}_E) \cdot p(\bar{X}_C \mid \bar{X}_B)p(\bar{X}_D \mid \bar{X}_A).$$

2.3 Maximum Likelihood Estimator

In general, given a random variable x with a probability distribution[1] $p(\bar{x} \mid \theta)$ that depends on an unknown parameter θ, we can estimate the value of θ from observations of x using the maximum likelihood principle. Suppose we

[1] This abuse of notation of conditioning with respect to a non-random variable θ makes sense if you consider the ML to be a special case of MAP.

have some observations $\{\bar{x}_i\}_{i=1}^N$ of x, then we define the likelihood and log-likelihood functions to be

$$L(\theta) = \prod_{i=1}^{N} p(\bar{x}_i \mid \theta) \text{ and}$$

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{N} \log p(\bar{x}_i \mid \theta)$$

respectively. To find an estimate $\hat{\theta}$ of θ we compute

$$\hat{\theta} = \arg\max_{\theta} L(\theta) = \arg\max_{\theta} \ell(\theta).$$

2.4 Maximum A Posteriori Estimator

`TODO: map`

2.5 Estimating Bayesian Networks

If $G = (V, E)$ with $\{X_v\}_{v \in V}$ is a Bayesian network and we have some observations $\{\bar{X}_v^i\}_{i=1}^N$ from the parametric distributions $p_{X_v}(\bar{X}_v, \theta)$ for each $v \in V$ (fully observable, no hidden variables), we can use the maximum likelihood estimator to estimate the unknown θ form the observations. The likelihood function can be decomposed into local likelihood functions using the independence relations in G:

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{N} p(\{\bar{X}_v^i\}_{v \in V} \mid \theta) \\ &= \prod_{i=1}^{N} \prod_{v \in V} p(\bar{X}_v^i \mid \{\bar{X}_w\}_{w \in \mathrm{pa}(v)}, \theta) \\ &= \prod_{v \in V} \prod_{i=1}^{N} p(\bar{X}_v^i \mid \{\bar{X}_w\}_{w \in \mathrm{pa}(v)}, \theta) = \prod_{v \in V} L_v(\theta_{v|\mathrm{pa}(v)}). \end{aligned}$$

This decomposition can be further improved to reduce the number of computational constraints.

If the random variables are discrete and the entire distribution is the unknown, provided that the model is fully observable, the probabilities can be estimated using conditional counting

$$p(\bar{X}_v \mid \{X_w\}_{w \in \mathrm{pa}(v)}) \approx \frac{N(X_v, \mathrm{pa}(v))}{\sum_{u \in V} N(X_u, \mathrm{pa}(u))},$$

`TODO: explain` N `and then MCMC`

2.6 Network Inference

Now suppose that we have random variables $\{X_v\}_{v \in V}$ for a set of nodes V of a directed graph $G = (V, E)$ but we do not know the structure (its topology), that is E, and we would like to infer it from a set of observations $D = \{\bar{X}_v^i\}_{v,i}$. To do so the idea is to construct a space of candidate models and assign a score to each model, then optimize to find the highest scoring models (this however is NP hard). The family is constructed using prior (biological) knowledge (e.g. "a gene as at most n regulators"). For there score the *Bayesian score* is defined to be

$$s(G) = \log p(G \mid D) = \log(p(D \mid G)) + \log(p(G))$$

where the marginal likelihood is taken from a prior θ

$$p(D \mid G) = \int p(D \mid G, \theta) p(\theta \mid G) \, d\theta.$$

Algorithm 2.1 (Greedy structure search)**.** Given a set of observations $D = \{\bar{X}_v^i\}_{i,v}$ and an initial guess for the graph $G_0 = (V, E)$.

function greedy-search(G, D)
  $\hat{G} \leftarrow G_0$
  repeat
    $G \leftarrow \hat{G}$
    ▷ $\mathfrak{o}$ *can be edge addition, removal or reversal*
    for each operation $\mathfrak{o}$ **do**
      $G' \leftarrow \mathfrak{o}(G)$
      if G' is not cyclic **then**
        if score(G') > score($\hat{G}$) **then**
          $\hat{G} \leftarrow G'$
  until $\hat{G} = G$

2.7 Dynamic Bayesian Networks

The model of Bayesian networks can be extended by introducing a time dimension: let $G = (V, E)$ be a directed graph and $\{X_v(t)\}_{v \in V}$ be a set of discrete-time stochastic processes with $t \in \{1, \ldots, T\}$ indexed by V, then similar to Bayesian networks there is a factorization that additionally involves time

$$p(\{X_v(t)\}_{v \in V}) = \prod_{v \in V} \prod_{t=2}^{T} p(X_v(t) \mid \{X_u(t-1)\}_{u \in \mathrm{pa}(v)}).$$

2.8 Caveats / Challenges

3 Stoichiometric Network Analysis

Motivated by metabolic networks the idea is to use a structural analysis from first principles: conservation of mass (and energy) combined with well-characterized reaction stoichiometries and reversibilities.

3.1 Metabolic Networks

In a metabolic network we use the vocabulary

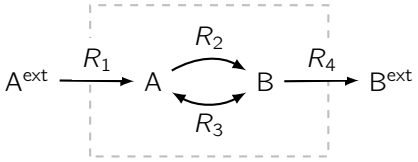
Metabolism enzyme-catalyzed reaction;

Metabolites educts (consumed) and products.

Given the reaction stoichiometry (ratios of products / educts) and reaction directionalities (reversible or irreversible) we seek to compute metabolic fluxes (rates of metabolic reactions). Further, we distinguish between external and internal metabolites, so that external metabolites are assumed to be sources / sinks.

To represent metabolic networks we use the stoichiometric matrix $N \in \mathbb{R}^{n \times q}$, wherein on the rows there are the internal metabolites and in the columns the reactions. An element n_{ij} of N is then the stoichiometric coefficient for the metabolite i in reaction j.

Example. Consider the following metabolic network. Nodes represent metabolites while arrows reactions. Reversible reactions have double headed arrows.



The stoichiometric matrix is

$$N = \begin{bmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & 1 & -1 \end{bmatrix}.$$

A flux distribution is a vector $r \in \mathbb{R}^q$ of reaction rates. We say r is feasible if $r_i \geq 0$ for all irreversible reactions. Then, Nr is a mass concentration $c \in \mathbb{R}^n$ so the reaction kinetics are given by the balancing equation

$$\frac{dc}{dt} = Nr(t) \stackrel{!}{=} 0,$$

which we have set to zero because we are interested in knowing a quasi-steady state. Physically this means a constant comsumption / production from the network. Because there are more reactions that metabolites, $n \gg q$, there is not a unique solution (underdetermined system of equations), but rather an infinite number of them in the kernel (null space) of N.

Linear Algebra Recap Recall that the kernel

$$\ker(N) = \{r \in \mathbb{R}^q : Nr = 0\}$$

has dimension $q - \text{rank}(N)$. Recall that this subspace can be parametrized by finding $k := q - \text{rank}(N)$ linearly independent solutions $\{\tilde{r}_i\}_{i=1}^k$ to the balancing equation $Nr = 0$. Then any solution can be written as a linear combination $r = \sum_i w_i \tilde{r}_i$ for some coefficients w_i. If we use the $\tilde{r}_i$ as columns of a matrix, we have the kernel matrix K, which by grouping the coefficients into $w \in \mathbb{R}^k$ gives that $r = Kw$. Note that the kernel matrix is *not* unique.

3.2 Enzyme Subsets

The rows of the kernel matrix K correspond to the reactions. If two rows of K differ only by a scalar factor, it means that the corresponding reactions are coupled, which means that they must always operate together with a fixed ratio in their rates. Typically (but not necessarily) this happens in linear pathways.

`TODO: slides on limitations`
`TODO: conservation relations`

3.3 Flux Balance Analysis

Real metabolic networks usually have a very large number of reactions, so in practice it is impossible to solve for solution of fluxes by hand. To solve this we incorporate additional biological knowledge by assuming again quasi-steady state and that

1. there is an *objective* for instance maximize growth / energy (ATP) production / product yield that is calculated with $w\mathsf{T}r = \sum_i w_i r_i$ with some coefficients $\{w_i\}_{i=1}^q$,
2. for each reaction r_i we (roughly) know a capacity $\alpha_i \leq r_i \leq \beta_i$ (e.g. for irreversible reactions $\alpha_i > 0$).

Then we can formulate a *linear program* to perform a flux balance analysis (FBA)

$$\max_{r \in \mathbb{R}^q} w^\mathsf{T} r \quad \text{such that} \quad Nr = 0, \quad \alpha_i \leq r_i \leq \beta_i$$

which can be solved using the well studied and computationally efficient *simplex method*.

Example. `TODO: Prediction of Phenotypes`

Example. `TODO: Prediction of Mutant behaviour`

Example. `TODO: evolution of metabolism`

3.4 Flux Variability Analysis

`TODO: why we need it`

Algorithm 3.1 (Flux Variability Analysis, FVA)**.** Given the stoichiometric matrix N, the reactions $\{R_i\}_i$ and the bounds on the fluxes $\{\alpha_i\}_i, \{\beta_i\}_i$.

function fva(N, $\{\alpha_i\}_i, \{\beta_i\}_i$)

3.5 Caveats / Challenges

4 Dynamic Systems Fundamentals

This section is interested in modelling biochemical reaction kinetics using ODE models. To construct the model we use the law of mass action from reaction kinetics:

> At constant temperature without catalyst chemical reaction rates are proportional to products of substrate concentrations taken to the power of stoichiometric coefficients (reaction order).

This assumes that there is a large number of molecules and that the system is "well-mixed", i.e. there are no spatial heterogeneities. Hence, given a set of q reactions ($1 \leq j \leq q$) of n reactants

$$\alpha_{1j} X_1 + \cdots + \alpha_{nj} \xrightarrow{k_j} \beta_{1j} X_1 + \cdots + \beta_{nj} X_n$$

by the law of mass action we obtain the system of ordinary differential equations for the concentrations c_i for each X_i

$$\frac{dc_i}{dt} = \sum_{j=1}^{q} k_j (\beta_{ij} - \alpha_{ij}) \prod_{l=1}^{n} c_l^{\alpha_{lj}},$$

The above can also be written more compactly in vector form by defining $c(t) \in \mathbb{R}^n$, the (general) reaction rates $r(c(t), u(t), p)$ and using the stoichiometric matrix $N \in \mathbb{R}^{n \times q}$

$$\frac{dc}{dt} = Nr(c(t), u(t), p).$$

Herein N contains all α_{ij} and β_{ij}, while r depends on $p \in \mathbb{R}^{n_p}$ for *parameters* which encapsulates all the $k_1, \ldots, k_q$ reaction constants. In an even more general setting we will write a model in terms of a set of "states" $x \in \mathbb{R}^{n_x}$ (hitherto c), external "inputs" $u(t) \in \mathbb{R}^{n_u}$ and a "right-hand side" of the dynamics $f(x(t), u(t), p)$, so that

$$\frac{dx}{dt} = f(x(t), u(t), p)$$

4.1 Linear Dynamics

`TODO`: $\dot{x} = Ax + Bu$

4.2 Michaelis-Menten Enzyme Kinetics

To model an enzymatic reaction from a substrate S to a product P we will assume the following reaction

$$\mathrm{E} + \mathrm{S} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \mathrm{E} \cdot \mathrm{S} \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} \mathrm{E} + \mathrm{P}$$

And then we will make the following assumptions

1. There is no feedback by the product, so $k_{-2} = 0$.
2. There is a constant total quantity (concentration) of enzyme $[\mathrm{E}]^t = [\mathrm{E}] + [\mathrm{E} \cdot \mathrm{S}]$. This reduces the number of variables in the ODE model.
3. The reaction is in quasi steady-state because of time-scale separation ($k_1, k_{-1} \gg k_2$) which means that $\frac{d}{dt}[\mathrm{E} \cdot \mathrm{S}] \approx 0$.
4. There is an excess of substrate over the enzyme, so $[\mathrm{S}] \approx [\mathrm{S}]_0$ is a constant, making $\frac{d}{dt}[\mathrm{S}] = 0$ (e.g. justified in metabolic networks).

When applied to an ODE model of the first reaction these assumptions will lead to the following system of algebraic equations

$$\begin{aligned} 0 &= -k_1[\mathrm{E}]^t[\mathrm{S}]_0 + (k_1[\mathrm{S}]_0 + k_{-1})[\mathrm{E} \cdot \mathrm{S}] \\ 0 &= k_1[\mathrm{E}]^t[\mathrm{S}]_0 - (k_1[\mathrm{S}]_0 + k_{-1} + k_2)[\mathrm{E} \cdot \mathrm{S}] \end{aligned}$$

that can be solved for $[\mathrm{S} \cdot \mathrm{E}]$ yielding

$$[\mathrm{S} \cdot \mathrm{E}] = \frac{[\mathrm{S}]_0[\mathrm{E}]^t}{[\mathrm{S}]_0 + \frac{k_2 + k_{-1}}{k_1}}$$

Then, inserted this result into the next reaction model gives

$$\frac{d[\mathrm{P}]}{dt} = k_2[\mathrm{S} \cdot \mathrm{E}] = \frac{k_2[\mathrm{S}]_0[\mathrm{E}]^t}{[\mathrm{S}]_0 + \frac{k_2 + k_{-1}}{k_1}} = \nu,$$

wherein since everything on the left hand side is constant we defined ν to be the so-called reaction velocity, which itself is described by two parameters

$$\nu = \frac{\nu_{\max}[\mathrm{S}]_0}{[\mathrm{S}]_0 + K_M},$$

the maximal reaction rate $\nu_{\max}$ and the Michaelis-Menten constant K_M. The latter can be understood as a measure for the affinity between the enzyme and the substrate. Physically, it is the substrate concentration at half of the maximal rate.

4.2.1 Competitive Inhibition

For competitive inhibition we assume the reaction

$$\mathrm{E} + \mathrm{S} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \mathrm{E} \cdot \mathrm{S} \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} \mathrm{E} + \mathrm{P}$$
$$\mathrm{E} + \mathrm{I} \underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}} \mathrm{E} \cdot \mathrm{I}$$

and further assume

1. The inhibitor is also conserved.
2. Quasi steady-state for enzyme complexes.

Then, by going through the same process as before to solve for the concentrations will result in

$$\nu = \frac{k_2[\mathrm{S}]_0[\mathrm{E}]^t}{[\mathrm{S}]_0 + \frac{k_2 + k_{-1}}{k_1}\left(1 + \frac{k_3[\mathrm{I}]}{k_{-3}}\right)} = \frac{\nu_{\max}[\mathrm{S}]_0}{[\mathrm{S}]_0 + K_M\left(1 + \frac{[\mathrm{I}]}{K_I}\right)}$$

wherein we introduce an inhibition constant K_I. Put into words, the competitive inhibitor reduces the apparent substrate affinity.

4.2.2 Non-Competitive Inhibition

If we instead assume the following reaction scheme

$$\begin{array}{ccccc} \mathrm{E} + \mathrm{S} & \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} & \mathrm{E} \cdot \mathrm{S} & \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} & \mathrm{E} + \mathrm{P} \\ \Big\downarrow{\scriptstyle k_3} \Big\uparrow{\scriptstyle k_{-3}} & & \Big\downarrow{\scriptstyle k_5} \Big\uparrow{\scriptstyle k_{-5}} & & \\ \mathrm{E} \cdot \mathrm{I} & \underset{k_{-4}}{\overset{k_4}{\rightleftharpoons}} & \mathrm{E} \cdot \mathrm{I} \cdot \mathrm{S} & & \end{array}$$

then the resulting velocity (assuming again quasi steady-state for complexes and inhibitor conservation) will be

$$\nu = \frac{\nu_{\max}[\mathrm{S}]_0}{[\mathrm{S}]_0 + K_M}\left(1 + \frac{[\mathrm{I}]}{K_I}\right)^{-1},$$

for a new constant K_I that depends on $k_1, k_2, \ldots, k_{-5}$. Hence, non-competitive inhibition will reduce the maximal reaction velocity.

4.2.3 Cooperativity

The converse of the previous section is cooperativity, whereby usually a large assembly of enzymes are spatially close together and acts as an integrated metabolic factory. In this case instead of considering each enzyme in the chain we simplify the model by creating an artificial reaction that requires n substrate molecules to produce n products

$$\mathrm{E} + n\mathrm{S} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \mathrm{E} \cdot n\mathrm{S} \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} \mathrm{E} + n\mathrm{P}$$

This results in the rate law

$$\nu = \frac{\nu_{\max}[\mathrm{S}]_0^n}{[\mathrm{S}]_0^n + K_M^n}$$

and we call n the Hill coefficient. Increasing the Hill coefficient will change the signal response characteristic from hyperbolic ('graded') to sigmoidal ('ultrasensitive').

4.3 Caveats / Challenges

5 System Identification

We consider an ODE model ($x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$) containing a parameter vector $p \in \mathbb{R}^{n_p}$ given by

$$\frac{dx}{dt} = f(x(t), u(t), p), \tag{1}$$

which has a (generally unknown) solution

$$x(t, p) = x(0) + \int_0^t f(x(s, p), u(s), p)\ ds \tag{2}$$

We denote by $p^\star$ the optimal set of parameters, that is, those that bring the model closest to reality. The goal of this section is to find $p^\star$.

5.1 Measurements

Hereinafter we suppose we have N field measurements $\{x_i\}_{i=1}^N$ taken at discrete time intervals $\{t_i\}_{i=1}^N$. We assume that the measured values are given by the true value corrupted by i.i.d zero-mean Gaussian noise with covariance Σ_ϵ, i.e.

$$x_i = x(t_i, p^\star) + \epsilon_i, \quad i = 1, \dots, N \tag{3}$$

where $\epsilon_i \sim \mathcal{N}(0, \Sigma_\epsilon)$. Then we define the identification error $e_i = x(t_i, p) - x_i$ for our estimate of p and functional that is to be minimized

$$\phi(p) = \frac{1}{2} \sum_{i=1}^N e_i^\mathsf{T} Q e_i$$

with a (positive semidefinite) weighting matrix Q. For instance if $Q = I$ all errors are weighted equally. In reality since our measurements have a certain uncertainty (stemming from the tools we are using) it is more common to set Q to be the inverse of the covariance matrix of the measurements $Q = \Sigma_\epsilon^{-1}$.

5.2 Sensitivity

Because of the dynamics, the influence of each parameter on the model may change over time and the *sensitivity function* quantifies this exactly. Formally if we consider an initial set of parameters $\bar{p}$ the sensitivity over time is defined[2] to be

$$S(t) = \left.\frac{\partial x(t, p)}{\partial p}\right|_{\bar{p}}, \quad S : \mathbb{R} \to \mathbb{R}^{n_p \times n_x}.$$

However since we don't know $x(t, p)$, to compute $S(t)$ we use the differential sensitivity equation. To derive it we proceed by inserting (2) in $S(t)$ which gives

$$\begin{aligned} S(t) &= \frac{\partial}{\partial p} \int_0^t f(x(s, p), u(s), p)\ ds \\ &= \int_0^t \frac{\partial f}{\partial x}\frac{\partial x}{\partial p} + \frac{\partial f}{\partial p}\ ds = \int_0^t \frac{\partial f}{\partial x} S(s) + \frac{\partial f}{\partial p}\ ds \end{aligned}$$

then we remove the integral by differentiating with respect to time to obtain to obtain the differential equation

$$\frac{dS}{dt} = \left.\frac{\partial f}{\partial x}\right|_{\bar{p}} S(t) + \left.\frac{\partial f}{\partial p}\right|_{\bar{p}}, \quad S(0) = 0.$$

This equation can be (numerically) solved for $S(t)$. Since the ODE model is usually also solved numerically, it is common to directly solve the augmented system

$$\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{S} \end{bmatrix} = \underbrace{\begin{bmatrix} f(x, u, p) \\ \left.\frac{\partial f}{\partial x}\right|_{\bar{p}} S + \left.\frac{\partial f}{\partial t}\right|_{\bar{p}} \end{bmatrix}}_{\tilde{f}(\xi, u, p)}, \quad \xi(0) = \begin{bmatrix} x(0) \\ 0 \end{bmatrix}.$$

5.3 Gradient-based Methods

To find $p^\star$ for (1) we want to minimize the error $\phi(p)$ that is generally non-linear in p, and to do so we use Newton's method. To start we use an initial guess for the parameters p_0, then we iteratively apply steps $s_0, s_1, \dots$ as $p_{k+1} = p_k + s_k$ to reach (get close to) $p^\star$. At each iteration we use a second order approximation

$$\phi(p_{k+1}) = \phi(p_k + s_k) \approx \phi(p_k) + \left.\frac{\partial \phi}{\partial p}\right|_{p_k} s_k + \frac{1}{2} s_k^\mathsf{T} \frac{\partial^2 \phi}{\partial p^2} s_k$$

and then we choose s_k such that the above is minimized by solving

$$\left.\frac{\partial^2 \phi}{\partial p^2}\right|_{p_k} s_k = -\left.\frac{\partial \phi}{\partial p}\right|_{p_k}.$$

The iteration is halted when the quantity $\|\phi(p_{k+1}) - \phi(p_k)\|$ becomes small enough.

Newton's method as just described does not make use of the statistical information we know about our measurements. To incorporate statistical information we compute how of the structure of the Hessian of ϕ is related to $S(t)$:

$$\begin{aligned} \frac{\partial^2 \phi}{\partial p^2} &= \frac{\partial^2}{\partial p^2} \left\{ \frac{1}{2} \sum_{i=1}^N [x(t_i, p) - x_i]^\mathsf{T} Q [x(t_i, p) - x_i] \right\} \\ &= \sum_{i=1}^N \frac{\partial}{\partial p} \left\{ [x(t_i, p) - x_i]^\mathsf{T} Q \frac{\partial x}{\partial p} \right\} \\ &= \sum_{i=1}^N \frac{\partial x}{\partial p}^\mathsf{T} Q \frac{\partial x}{\partial p} + x(t_i, p) Q \frac{\partial^2 p}{\partial p^2} - x_i Q \frac{\partial^2 x}{\partial p^2} \\ &= \sum_{i=1}^N S(t_i)^\mathsf{T} Q S(t_i) + e_i Q \frac{\partial^2 x}{\partial p^2}. \end{aligned}$$

Now, if we take the expectation, since the error is zero-mean (assume unbiased estimator p of $p^\star$)

$$\mathbb{E}\left\{ \frac{\partial^2 \phi}{\partial p^2} \right\} = \sum_{i=1}^N S(t_i)^\mathsf{T} Q S(t_i),$$

and in particular if we replace Q with inverse of the measurement error covariance matrix Σ_ϵ^{-1} we obtain the *Fischer information matrix*

$$F(p) = \sum_{i=1}^N S(t_i)^\mathsf{T} \Sigma_\epsilon^{-1} S(t_i),$$

[2] We can also consider the sensitivity to be a function of p, so $S(t, p)$ is a function that shows how the parameters deviate from an initial value p after some time t.

which combines the uncertainty of the measurements and the effect of the ODE dynamics. Therefore, by using $F(p)$ instead of the Hessian in Newton's method will improve the performance on the identification of $p^\star$. For $F(p)$ there is a notorious given by the Cramér-Rao inequality on the covariance matrix of the parameter estimates Σ_p

$$\Sigma_p \succeq F^{-1}(p) \quad \text{or} \quad \sigma_{p_j}^2 \geq \frac{1}{F_{jj}(p)}$$

if we assume that the parameters are statistically independent from each other ($F(p)$ and Σ_p are diagonal). The inequality states that the variance of the parameter estimate cannot be better than the inverse of the information (which in this case is a combination of measurements precision and sensitivity).

5.4 Evolutionary Methods

5.5 Goodness-of-fit

After finding an estimate of $p^\star$ by minimizing $\phi(p)$ we need to confirms the statistical validity of the result. Because we assume that the error is zero-mean Gaussian if $Q = \Sigma_\epsilon^{-1}$ then $\phi(p)$ is χ^2-distributed and we therefore use a χ^2-test. The degrees of freedom of χ_k^2 is $k = N - n_p$. By choosing an α-value (confidence interval) we can define a threshold Δ_α from a cumulative χ_k^2 distribution and conclude our estimate p is statistically significant if it lies in the set

$$\{p \in \mathbb{R}^{n_p} : \phi(p) - \phi(p^\star) \leq \Delta_\alpha\}$$

5.6 Optimal Experiment Design

6 Simplified Dynamic Models

We suppose that there is an ODE model defined with $x \in \Omega \subseteq \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$

$$\frac{dx}{dt} = f(x(t), u(t)),$$

where Ω is our region of the phase space that is of interest. Then recall that nullclines are hypersurfaces $\{x \in \Omega : f(x, u) = 0\}$ (u is known). To simplify the ODE model, observe that nullclines indicate where f changes sign, and consequently the behaviour of the dynamics. Therefore, we can partition Ω depending on the sign into regions $\mathcal{R} = \{R_1, \ldots, R_m\}$, formally this could be written by defining a sign pattern function

$$\pi : \mathcal{R} \to \{-, 0, +\}^n$$
$$R \mapsto \text{sign}(x) \text{ for any } x \in R.$$

Having split Ω into the regions $\mathcal{R}$ we say that there is an *transition* $R_i \to R_j$ if there is a solution $x(t)$ of the ODE system such that $x(0) \in R_i$ and $x(T) \in R_j$ in finite time ($T < \infty$) without ever leaving the domain $R_i \cup R_j$. Or in other words, starting in R_i the system will eventually directly go to R_j.

Given the above, in principle we can obtain a transition graph $G = (V, E)$ by letting $V = \mathcal{R}$ be the nodes and $E \subseteq \mathcal{R} \times \mathcal{R}$ be the transitions $R_i \to R_j$. The transition graph is a qualitative simplification of the ODE model, and it is a conservative one, which means that all behaviours of the ODE model are captured by the graph (but not necessarily the converse). This is useful as it can be used to reject hypotheses, but in reality we usually do not (cannot) start with an ODE model, so we cannot directly extract such models from real experiments. However, the idea of partitioning the state space Ω in qualitatively homogeneous regions can be saved.

6.1 Piecewise Linear Models

Since we usually do not know how the state space looks like (or often even the ODE model), we simplify the dynamics. We consider only production and degradation of a n-gene network with a linear model

$$\frac{dx_i}{dt} = f_i(x) - g_i(x)x_i \tag{4}$$

where $1 \leq i \leq n$ and

$$f_i(x) = \sum_{l \in L} \kappa_{il} b_{il}(x), \qquad g_i(x) = \sum_{l \in L} \gamma_{il} b_{il}(x)$$

are given in terms of kinetic constants κ_{il}, γ_{il} and regulator functions $b_{il} : \mathbb{R}^n_{\geq 0} \to 0, 1$. The regulator functions describe the logical conditions (in terms of concentrations) required such that a protein encoded by gene i is synthesized (or degraded) at the rate k_{il} (or $\gamma_{il} x_i$).

To further simplify the regulator functions, which until now can be arbitrarily complex (e.g. Hill-like), are replaced with step functions

$$s^+(x, \theta) = \begin{cases} 1 & \text{if } x > \theta \\ 0 & \text{if } x < \theta \end{cases}, \quad s^-(x, \theta) = \begin{cases} 0 & \text{if } x > \theta \\ 1 & \text{if } x < \theta \end{cases},$$

that reduce the dynamics to active (1) and inactive (0) if the concentration is above or below a threshold value θ. For example if a gene i is expressed at rate κ_i only in the presence of proteins a and b, that is the concentrations $x_a > \theta_a$ and $x_b > \theta_b$, then the regulator function for gene i would be

$$f_i(x) = \kappa_i s^+(x_a, \theta_a) s^+(x_b, \theta_b).$$

Now, because all functions are step functions the state space Ω of $\frac{d}{dt}x = f(x) - g(x)x$ (where $g(x) = \text{diag}(g_1, \ldots, g_n)$) will be partitioned into rectangular regions $R_1, \ldots, R_m$ divided by the threshold values θ_{il}. Furthermore, since the right hand side is piecewise constant, in each partition R_i the function is linear, making it easier to find if there is a transitions $R_i \to R_j$ to the adjacent regions R_j.

6.2 Boolean Networks

7 Stochastic Systems

When there are a very low number of molecules (copy numbers), then we cannot use *ensemble* models from the previous sections (ODE models). This is because

- Relative fluctuations depend on molecule numbers. For n molecules the fluctuation is $\Delta n \approx 1/\sqrt{n}$ (from thermodynamics) so $\Delta n/n \approx 1/\sqrt{n^3}$.
- ...

`TODO: separation between intrinsic and extrinsic noise`

Extrinsic noise Variability of (assumed) parameters.

Intrinsic noise Effect of small molecule numbers.

We assume again spatial homogeneity ("well-stirred") and for the derivation that the reaction volume Ω is constant.

Consider a set of N distinct chemical species $\{S_1, S_2, \ldots, S_N\}$, and each species has a number of molecules n_i for S_i. Then the state of the system can be represented by a vector $n(t) \in \mathbb{R}^N$. The molecules react with each other via M possible reaction channels $\{R_1, \ldots, R_M\}$. We will assume that reactions happen instantaneously (are "fired" when a reaction channel is activated).

For each reaction R_j we define a state-change vector v_j such that $n + v_j$ corresponds to the state after the reaction R_j has been fired. Then to define "when" reactions occur, we define (again for each reaction R_j) the propensity function $a_j(n)$ so that $a_j(n)dt$ describes the average probability that in the system in state $n(t)$ one reaction of type R_j occurs in the time interval $[t, t + dt]$.