Two Pairs of Boolean Functions in Computational Biology

Tamon Stephen

SIMON FRASER UNIVERSITY

IPAM Workshop on Optimization, Search and Graph-Theoretical Algorithms for Chemical Compound Space, April 14th, 2011
Outline

- Example 1: Metabolic Networks.
- Elementary Modes in Metabolic Networks.
- Monotone Boolean Functions.
- Minimal Cut Sets in Metabolic Networks.
- Joint Generation of Modes and Cut Sets.
- Example 2: Ancestral Genome Reconstruction.
- Conclusions.
Example 1: Metabolic Networks

- Set of co-dependent metabolic processes (i.e. chemical reactions) active in a cell.

- Reactions are characterized by their stoichiometry, the relationship between the metabolites (molecules) they produce and consume. For instance:

\[
\text{CH}_4 + 2\text{O}_2 \rightarrow \text{CO}_2 + 2\text{H}_2\text{O}
\]

  methane + oxygen → carbon dioxide + water

- In a steady state, the amounts of metabolites produced and consumed by the various reactions must balance.
Growth in *E. coli* bacteria is a complex process involving dozens of metabolites. It is modelled as a single reaction converting a fixed set of precursors into biomass.

To sustain growth, supplies of all these precursors are required.

We use a detailed model of the *E. coli* cell that has 89 metabolites with 110 possible reactions on them.

We would like to deduce which sets of reactions support biomass synthesis.

Similarly, we would like to know which sets of reactions are required for biomass synthesis.
Problem setup

A model of network of $k$ metabolites with $q$ reactions can be recorded as $k \times q$ stoichiometric matrix $N$.

Some reactions are reversible: it is possible to produce a given output from a given input, or vice-versa. However, most reactions are irreversible.

We can record in a vector $x$ the relative frequencies of active reactions in given steady-state. Inactive reactions will have coefficient 0, and reversible reactions may have negative coefficients.

Thus the possible modes representing steady states of the network are:

$$\{x \in \mathbb{R}^q | Nx = 0, x_i \geq 0 \ \forall i \in U\}$$

where $U$ is the set of irreversible reactions.
This table is a stoichiometric matrix that encodes 5 reactions involving 6 metabolites:

<table>
<thead>
<tr>
<th></th>
<th>$m_1$</th>
<th>$m_2$</th>
<th>$m_3$</th>
<th>$m_4$</th>
<th>$m_5$</th>
<th>$m_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1$</td>
<td>1</td>
<td>2</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$r_2$</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$r_3$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>$r_4$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-3</td>
</tr>
<tr>
<td>$r_5$</td>
<td>-2</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>-2</td>
</tr>
</tbody>
</table>

The reactions $r_1$, $r_2$, $r_3$, $r_5$ in this example form a mode, since there is a steady state in which they are all active. In that steady state, the reactions occur with relative frequencies $1 : 2 : 1 : 0 : 1$. 
Non-negative linear combinations of modes are again modes, so the system (1) usually has many solutions. These can be decomposed into sums of support minimal non-zero solutions, known as elementary modes (EMs).

These are the simplest functioning subsystems of the network, and as such are of interest to biologists.

The EMs are, up to a constant multiple, the extreme rays of the convex cone (1). They can be computed using variants of the double description method.

Double description may work well, but has a poor worst-case complexity.

Elementary modes are characterized up to a scalar multiple by the binary vector representing their support pattern.
Boolean Functions

- The relation of supporting a particular target reaction is an example of a boolean function: its inputs are binary (a reaction may or may not be included) and its output is binary (a set of reactions may or may not support the target reaction).

\[
\begin{array}{c|c}
 x_1 & 1 \\
 x_2 & 1 \\
 x_3 & 1 \\
 x_4 & 0 \\
 x_5 & 1 \\
\end{array}
\]

- This model is widely applicable: many complex systems are usefully characterized by their supporting subsystems.
Many useful boolean function are monotone in the sense that if the output is true, it remains true if more variables are set to true.

Monotone:

<table>
<thead>
<tr>
<th>AND</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OR</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Not Monotone:

<table>
<thead>
<tr>
<th>XOR</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The relation of supporting biomass synthesis defines a monotone boolean function.
Monotone boolean functions can be described in terms of either their minimal true or maximal false sets (clauses).

For instance, the **AND** function has a single minimal true clause, \( x_1 = 1, x_2 = 1 \).

The **OR** function has two minimal true clauses, \((1, 0)\) and \((0, 1)\).

The models of *e.coli* that we looked at had between 598 and 27,099 elementary modes, depending on which of four types of growth we were modelling.

**Question:** Given a monotone boolean \( f \) as a black box, can we efficiently generate its minimal true clauses?
A hypergraph consists of a base set of vertices and edges which are subsets of the vertices. It is Sperner if there are no nested edges.

Given a monotone boolean function \( f \), we can define a Sperner hypergraph whose vertices are the variables, and whose edges are the minimal true clauses.

Similarly, any Sperner hypergraph defines a monotone boolean function.
Closely related to the concept of elementary modes are **cut sets**. These are subsets of the reactions without which the network cannot operate in a steady state. For a cut set $C$, the system:

\[
\{ x \in \mathbb{R}^q | Nx = 0, x_i \geq 0 \ \forall i \in U, x_c = 0 \ \forall c \in C \}\tag{2}
\]

has only the trivial solution $x = 0$.

Note that the relation of being a cut set also defines a monotone boolean function.

As with modes, we focus on support **minimal cut sets** (MCSs). These are the smallest combinations of reactions that we need to shut down the system.

The list of such possible "knock out" strategies is particularly useful in the design of experiments.
A cut set must intersect each EM. Indeed the MCSs are exactly the minimal sets satisfying this condition. The hypergraphs defined by the EMs and MCSs are dual in the sense that one can be obtained from the other by taking the family of minimal sets intersecting all hyperedges. Such pairs of hypergraphs are called transversal.

We also call that associated monotone boolean functions dual.
Klamt and Gilles (2004) computed the MCSs by first computing the EMs and then finding the transversal hypergraph.

The EMs were computed using a custom double description code.

Their transversal hypergraph calculation used brute-force enumeration combined with clever pruning and took several hours for the cases they are interested in.

Their calculation also introduced some useful pre- and post-processing.

The transversal hypergraph problem is a well known combinatorial problem that arises in many applications.

A classical algorithm for it is effective in many cases, including this one, but has bad worst-case complexity.
Fredman and Khachiyan (1996) proposed a novel recursive algorithm for computing the dual of a monotone boolean function.

It can be modified to produce both a function and its dual from an oracle evaluating the function.

The algorithm works in worst-case time $m^{\text{poly}(\log m)}$, where $m$ is the combined size of the function and its dual.

Since biologists would like to compute both the EMs and the MCSs when analyzing a metabolic network, Fredman and Khachiyan’s algorithm is a natural fit.

For this example, the oracle is a linear program checking if the system (2) has a non-trivial solution.
The algorithm of Fredman and Khachiyan is poorly understood in practice. The only implementation we found of Fredman and Khachiyan’s algorithm is not public and uses hard-coded oracles.

We initially implemented a prototype of the simpler, but less efficient, version of the Fredman-Khachiyan algorithm in MATLAB, and called CPLEX as our linear programming oracle.

This prototype successfully generates the EMs and MCSs for biomass synthesis in *e.coli*, although it is slower than well-tuned existing techniques.

We now have a nice, open-source code to do this: [http://primaldual.de/cl-jointgen/](http://primaldual.de/cl-jointgen/)

It is written in Common Lisp to take advantage of the heavily recursive nature of the algorithm.
• The published computations used our original simple MATLAB prototype, and called CPLEX for the linear programming oracle. Results are recorded in the table below.

<table>
<thead>
<tr>
<th>Problem</th>
<th>acet</th>
<th>succ</th>
<th>glyc</th>
<th>gluc</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM’s</td>
<td>289</td>
<td>2722</td>
<td>7472</td>
<td>18481</td>
</tr>
<tr>
<td>MCS’s</td>
<td>245</td>
<td>1255</td>
<td>2970</td>
<td>4225</td>
</tr>
<tr>
<td>Time to generate</td>
<td>194.8</td>
<td>10672.2</td>
<td>103511.2</td>
<td>677599.3</td>
</tr>
</tbody>
</table>

• The prototype successfully generates the EMs and MCSs for biomass synthesis in \textit{e.coli} on the substrates acetate, succinate, glycerol and glucose respectively.

• The problem size was reduced by keeping the preprocessing used by the old brute-force algorithm.

• The \texttt{cl-jointgen} code is faster than the MATLAB prototype. Currently it includes only the simplest version of the Fredman-Khachiyan algorithm, a more advanced version may be substantially faster.
Example 2: Ancestral Genome Reconstruction

We are interested in deducing information about the genomes of ancestral species based on the genomes of species descended from them.
Consider the situation where we have identified several genetic markers that appear uniquely in each of the extant genomes.

Assume that these are present in ancestral genome, but that mutations may have changed the order of some of the markers.

We would like to find the order of the markers in the ancestral genome.

We know which markers are adjacent in the extant genomes, and expect that the order of the markers in the ancestral animal will remain the same except for a few mutations.
• An **ancestral synteny** is a set of markers that are thought to have been consecutive in the ancestor.

• The set of all ancestral syntenies can be encoded in a binary matrix $M$. The following matrix represents 4 markers and 5 syntenies:

$$
\begin{array}{c|cccc}
 & m_1 & m_2 & m_3 & m_4 \\
\hline
s_1 & 1 & 1 & 1 & 0 \\
s_2 & 1 & 0 & 1 & 1 \\
s_3 & 1 & 0 & 0 & 1 \\
s_4 & 0 & 1 & 1 & 1 \\
s_5 & 0 & 0 & 1 & 1 \\
\end{array}
$$

• Note that in this case, no ordering of the markers is consistent with all the syntenies.
The Consecutive Ones Property

- If all syntenies are correct, then the columns of $M$ can be rearranged so that the ones on each row are consecutive. If this is possible, $M$ is said to have the consecutive ones property (C1P).

- There is an excluded minor characterization on C1P matrices and a non-trivial linear time algorithm to test if a matrix is C1P.

- If a matrix is not C1P, then at least some of the rows must represent “false positive” syntenies.

- We would like to identify these false positives.

- One possible approach is to look for the largest C1P subset of the rows. However, this may not be unique, and in any case is NP-complete.
Conflicting Sets

- If a subset of the rows defines a matrix that is not C1P, we call it a **conflicting set**.

- A conflicting set of rows is called a **minimal conflicting set** if any proper subset of the row is C1P.

- One approach to reconstructing the ancestral genome is to identify rows that are likely “false positives” by looking at the **conflicting index**, the number of minimal conflicting sets containing the row.

- Computing the conflicting index is \#P hard.

- The relation of being a conflicting sets defines a monotone boolean function on the set of rows of the matrix.
Maximal C1P Submatrices

- The natural monotone boolean function structure of conflicting sets draws our attention to the dual function of the minimal conflicting sets.

- This turns out to be the function that identifies whether the complement of a set of rows form a C1P matrix.

- Thus its minimal clauses are the complements of the maximal C1P submatrices.

- Indeed, the maximal C1P submatrices are candidate reconstructions of the ancestral genome.

- These candidate reconstructions are a useful byproduct of the conflicting index calculation.
Conclusions

- Monotone boolean functions are a natural structure emerging in large systems, such as those in biology. And chemistry?
- We found monotone boolean functions in two quite different applications: metabolic networks and ancestral genome reconstruction.
- It is desirable and possible to generate them for useful large-scale applications, using algorithms based on those of Fredman and Khachiyan.
- We have an open-source code to do this: http://primaldual.de/cl-jointgen/
- This algorithm should be more widely known, and better understood both in theory and in practice.
- When a monotone boolean function appears, it is worth looking at its dual.
Thank you!