Supplementary Material for “Information Processing by Biochemical Networks: A Dynamic Approach”

Clive G. Bowsher
Centre for Mathematical Sciences, University of Cambridge, U.K.
C.Bowsher@statslab.cam.ac.uk

Table of Contents

1. Mathematical Methods (SM1)
2. The NFκB Signalling Network (SM2)
3. The MIDIA Algorithm (SM3)
4. Modularisation of the NFκB Signalling Network (SM4)
5. Core Species of the TLR Signalling Network (SM5)

1 Mathematical Methods (SM1)

1.1 Stochastic Kinetic Models

A Stochastic Kinetic Model (SKM) is a continuous-time pure jump process describing the dynamic evolution of the state of the molecular network, $x(t)$. Any change in $x(t)$ is the result of the occurrence of some biochemical reaction, $m$. The stoichiometric matrix is given by $S := [S_1, S_2, ..., S_M]$, where the column vector $S_m$ consists of the changes in the number of molecules of each species caused by reaction $m$. The stochastic dynamic evolution of $x(t)$ is governed by the conditional reaction intensities, $[\lambda_m(t); m = 1, ..., M]$, which can be thought of as the instantaneous reaction rates at time $t$ conditional on the trajectory (or ‘history’), $X(t) = (x(s); 0 \leq s \leq t)$. Conditional intensities, $\lambda_A(t)$, for the levels of any group of species $A$ are defined and constructed by Bowsher (2010).

The term SKM implies that each reaction intensity, $\lambda_m(t)$, ‘depends only on’ (is measurable with respect to) the history of the reactants of reaction $m$, $X_{R[m]}(t) = (x_{R[m]}(s); 0 \leq s \leq t)$. Stochastic chemical kinetic theory (Gillespie, 1992) implies that these reaction intensities take the form

$$\lambda_m(t) = c_m g_m\{x_{R[m]}(t^-)\},$$

(1)

where $c_m > 0$ is a deterministic rate constant and $g_m\{\cdot\} \geq 0$ is a continuous function depending only on the levels of $R[m]$, the reactants of reaction $m$, ‘immediately before’ time $t$.

Equation (1) has a firm physico-chemical basis under the assumptions that the system is spatially homogeneous, confined to a fixed volume and held at constant temperature. The SKM is then a Markov chain with state vector $x(t)$ and obeys the chemical master equation (Gillespie, 1992). The class of SKMs thus includes, but is not limited to, all stochastic dynamic models that may be simulated using a Gillespie (1977) algorithm or variant thereof. SKMs in general are considerably less restrictive in that they allow dependence of each reaction intensity on the entire past trajectory of its reactants (e.g., on the time elapsed since the last jump in some reactant levels) and also on the time evolution of variables taken to be deterministic such as temperature, light level and time itself. SKMs can thus accommodate, for example, general memory effects (due perhaps to inhomogeneity arising from imperfect mixing), non-exponential waiting times, and deterministic circadian or environmental effects.

The methods of this paper apply to standard SKMs. For the straightforward case where the levels of all reactants of any reaction in the SKM are changed by that reaction, this means simply that zero’th
order reactions (those with no reactants) are specified to each have only 1 product. The TLR network is such a case and is a standard SKM. (A slightly more general definition of a standard SKM is given in Bowsher 2010). Use is also made of the following regularity condition.

**Condition 3** A subset of reactions $\Gamma$ is said to be identified by consumption of reactants if and only if: 
1. For all $m \in \Gamma$, $S_{im} \leq 0$ for all $i \in R[m]$ and $S_{km} < 0$ for some $k \in R[m]$ (provided that $R[m]$ is not empty), and $S_{im} \geq 0$ for all products $i$ of reaction $m$; and 
2. there does not exist a pair of distinct reactions $m, \tilde{m} \in \Gamma$ such that $S_m = S_{\tilde{m}}$, where $S_m$ denotes the vector formed by setting all positive elements of $S_m$ to zero.

The regularity condition implies that no two reactions in $\Gamma$ change reactants identically and will be satisfied for all reactions in the network by most SKMs, possibly after explicit inclusion of enzymes etc. in reaction mechanisms. For a set of species $A$, let $\Delta(A)$ denote the set of reactions that change the level of some species in $A$. Then, for the TLR network, Condition 3 holds for the subset of reactions $\Delta(V_{de} \setminus S_{de}) \cap \Delta(V_{ed} \setminus S_{de})$, for all adjacent pairs $(d, e)$ in $T_M$ (as required by Theorems 1 and 2 below).

### 1.2 Kinetic Independence Graphs

Kinetic independence graphs or KIGs (Bowsher, 2010) are directed, cyclic graphs with node set equal to the set of biochemical species, $V$. The graphs are distinct from and have a different purpose to the diagrams of the systems biology graphical notation (Le Novère, N., et al., 2009). The parents of each species $k$ are denoted $pa(k) := \{i \in V \mid i \rightarrow k\}$. For a given reaction network, the KIG is constructed so that, for each species $k$, all species other than $k$ and its parents are known to be irrelevant for the conditional intensity of $k$ and hence irrelevant for its instantaneous kinetics. The parents of $k$ are determined as follows. First, determine $\Delta(k)$, the set of reactions that change the level of $k$. Second, determine $R[\Delta(k)]$, the collection of all species that participate as a reactant in at least one of the reactions in $\Delta(k)$. Note that $R[\Delta(k)] = \bigcup_{m \in \Delta(k)} R[m]$. Then

$$pa(k) = R[\Delta(k)] \setminus \{k\}, \quad \text{for all } k \in V. \quad (2)$$

For any set of species $A$, the kinetics of $A$ are *locally independent* of all species other than its parents $pa(A)$ and $A$ itself, i.e., $\lambda^A(t)$ is a function only of $x_{A \cup pa(A)}(t)$ and does not depend on the levels of any other species.

In general, the following information is needed for all reactions $m$ in order to construct the KIG of a network – the identity of the reactants $R[m]$, together with that of the species (reactants and products) whose levels are changed by the reaction, that is $\{i \in V \mid S_{im} \neq 0\}$. No knowledge of the rate parameters $c_m$ is required.

### 1.3 Dynamic conditional independences and junction trees

We state here new theorems and proofs required for the validity of our methods.

**Theorem 1:** Let $T_M$ be the junction tree returned by the algorithm MIDIA. Denote the modules of $T_M$ by $\{M_d\}$, and associate each edge in the tree between adjacent modules with the modules’ overlap or ‘separator’, $S_{de} := M_d \cap M_e$. Define $V_{de}$ ($V_{ed}$) as the union of the species in the modules in $T_M^d$ ($T_M^e$) where the $T_M^*$ are the 2 sub-trees obtained by cutting the edge $M_d \sim M_e$ in $T_M$, and $M_d \subset V_{de}$ ($M_e \subset V_{ed}$).

Then $S_{de} = V_{de} \cap V_{ed}$ and the separation $V_{de} \perp_G V_{ed}|S_{de}$ holds in the undirected KIG, $G$. Furthermore, provided that Condition 3 holds for the set of reactions $\Delta(V_{de} \setminus S_{de}) \cap \Delta(V_{ed} \setminus S_{de})$, the dynamic independence $V_{de \setminus t} \perp_{S_{de \setminus t}} V_{ed \setminus t}$ holds for all pairs of adjacent modules in $T_M$.

**Proof:** Let $T_{M,1}$ be the junction tree returned by stage 3 of the MIDIA algorithm. Its sub-tree species must be separated in $G$ (see Bowsher 2010, Proposition 5.2). A single updating of the tree by Species
Copying (as in step 7 of Algorithm 1, SM3) – and hence any finite number of updatings – leaves the following properties unchanged (for any adjacent pair of modules in the tree): $\bar{S}_d := M_d \cap M_e = V_{de} \cap V_{ed}$, the separation $V_{de} \perp_G V_{ed} \mid \bar{S}_d$ (where bars denote objects in the current form of the tree), and the junction tree property. These properties therefore also hold for the final tree, $T_M$. The result of Section 2.2 of the main text may now be applied to the partition $[V_{de} \setminus S_{de}, V_{ed} \setminus S_{de}, S_{de}]$ since (i) the graphical separation $(V_{de} \setminus S_{de}) \perp_G (V_{ed} \setminus S_{de}) \mid S_{de}$ holds; and (ii) the Species Copying stage ensures that any 2 reactions that change $S_{de}$ identically also have the same membership both of $\Delta(V_{de} \setminus S_{de})$ and $\Delta(V_{ed} \setminus S_{de})$.

**Theorem 2:** Let $T_M$ be the junction tree returned by the algorithm MIDIA and suppose that Condition 3 holds for the set of reactions $\Delta(V_{de} \setminus S_{de}) \cap \Delta(V_{ed} \setminus S_{de})$, for all pairs of adjacent modules in $T_M$. Then, for each module $M_d$ in $T_M$, it follows that $M_{de,t} \leftrightarrow S_{de,t} [\bigcup_{e \neq d} M_e]$. Note that each module interface $S_d = M_d \cap \{\bigcup_{e \in ne(d)} M_e\}$, where $ne(d)$ are the indices of the modules adjacent to $M_d$ in the junction tree.

**Proof:** The junction tree property of $T_M$ implies that $S_d = M_d \cap \{\bigcup_{e \in ne(d)} M_e\}$. Let the modules adjacent to $M_d$ be indexed by $e = 1, ..., E$. By Theorem 1, $V_{de,t} \leftrightarrow S_{de,t} V_{ed,t}$ for $e = 1, ..., E$, since $S_d = S_{de} \cup \{\bigcup_{f \neq e} S_d\}$ and $S_d \subset V_{de}$ for all $f \neq e$. Consider the statement $M_d \leftrightarrow \bigcup S_d \cup \bigcup_{e \neq d} M_e$ (where subscripts $t$ indicating trajectories are omitted). The statement holds for $i = 1$ since $M_d \subset V_{de}$. Suppose it holds for $1 < i < E$; then it will hold for $(i + 1)$ provided that $M_d \leftrightarrow \bigcup_{e \neq i} \bigcup S_d \cup \bigcup_{e \neq d} M_e$. This is the case since $V_{de+1} \leftrightarrow S_d \bigcup_{e \neq i} V_{de} \cup V_{de+1}$, whence $V_{de,i} \perp \bigcup S_d \cup \bigcup_{e \neq d} M_e$. It follows by induction that $M_d \leftrightarrow \bigcup S_d \cup \bigcup_{e \neq d} M_e$, which completes the proof since $V = \bigcup_{e \neq d} M_e$.

## 2 The NFκB Signalling Network (SM2)

The network is a state-of-the-art, stochastic dynamic model of the core mechanisms of NFκB signalling recently developed using extensive experimental data by Ashall et al. (2009). The component reactions of the network used are shown below and are based on the ‘3-feedback’ model preferred by Ashall et al. (2009). IKK has 3 forms (ne neutral, a active, and i inactive); $O$ denotes a generic output protein; the prefix $g_e$ denotes gene, $t$ transcript, $p$ phosphorylation, and $n$ nuclear location; and $R$ is the Tumor Necrosis Factor (TNF)-Receptor. For other species names, the reader is referred to Ashall et al. (2009).

$$
\begin{align*}
\emptyset & \rightarrow \text{TNF} \\
\text{TNF} + R & \leftrightarrow \text{TNF}.R \\
\text{TNF}.R + \text{IKK} ne & \rightarrow \text{TNF}.R + \text{IKK} a \\
\text{IKK} i & \rightarrow \text{IKK} ne \\
\text{IKK} a + \text{IKB} a & \rightarrow \text{IKK} a + \text{pIKB} a \\
p\text{IKB} a.\text{NFkB} & \rightarrow \text{NFkB} \\
\text{IKK} a + \text{IKB} e & \rightarrow \text{IKK} a + \text{pIKB} e \\
p\text{IKB} e.\text{NFkB} & \rightarrow \text{NFkB} \\
\text{IKB} e + \text{NFkB} & \rightarrow \text{IKB} e.\text{NFkB} \\
n\text{IKB} a + n\text{NFkB} & \rightarrow n\text{IKB} a.n\text{NFkB} \\
n\text{IKB} e + n\text{NFkB} & \rightarrow n\text{IKB} e.n\text{NFkB} \\
n\text{NFkB} & \rightarrow n\text{NFkB} \\
\text{IKB} a & \leftrightarrow n\text{IKB} a \\
n\text{IKB} a.n\text{NFkB} & \rightarrow \text{IKB} a.n\text{NFkB} \\
\text{IKB} e & \leftrightarrow n\text{IKB} e \\
n\text{IKB} e.n\text{NFkB} & \rightarrow \text{IKB} e.n\text{NFkB} \\
\text{IKB} a + n\text{NFkB} + g_P & \rightarrow n\text{NFkB}.g_P + t_P \\
t_P & \rightarrow \emptyset \\
P & \rightarrow \emptyset
\end{align*}
$$

For $P = A20, \text{IKB} a, \text{IKB} e, O :$

$$
\begin{align*}
n\text{NFkB}.g_P + t_P & \rightarrow n\text{NFkB}.g_P + t_P \\
t_P & \rightarrow t_P + P
\end{align*}
$$
The reaction network used is a parsimonious description capturing the essential aspects – for example, recruitment of adaptor and other proteins to the TNF-receptor is omitted, as is the direct inhibition by IκBε of promoter binding by nNFκB (e.g., nNFκB.g.P + nIκBα → nNFκB.nIκBα + g.P). The role of A20 in inhibiting conversion of IKK from its inactive to its neutral form is emphasised by Ashall et al. (2009), although the exact mechanism remains unclear – to capture this effect the simple sequestration mechanism IKKi + A20 ↔ IKKi.A20 is included in the reaction network. Delayed transcription of IκBε, also emphasised by Ashall et al. (2009), is accommodated in our framework by a time-dependent reaction intensity for the reaction nNFκB.g.IκBε → nNFκB.g.IκBε + t.IκBε.

3 The MIDIA Algorithm (SM3)

The Modularisation Identification by Dynamic Independence Algorithm (MIDIA) consists of the following stages:

- **Clique Decomposition and Aggregation** – Stages 1, 2 and 3 of Figure 3 (for an algorithmic description of these stages, see Bowsher 2010, Algorithm 5.1; for an accessible and succinct textbook treatment of the underlying graphical concepts see Cowell et al., 2007). As described in the main text, the procedure is applied to \(T^T\), a minimal triangulation of the undirected KIG of the reaction network.\(^1\) The clique decomposition of \(T^T\) is organised as a (rooted) junction tree in which the modules are numbered and the unique parent of each module carries a lower number than its child. When modularising a large network it is necessary to automate the pairwise aggregation scheme in order to obtain a coarser-grained modularisation than the clique decomposition itself (which is usually too detailed). The procedure used here (with the exception of Input-Output path analysis, see Section 2.4) works ‘backwards’ through the junction tree in an iterative manner until all module residuals are of at least the specified minimum size – each iteration identifies the highest numbered module failing the residual size criterion and aggregates that module with its parent by deleting the relevant edge and merging the 2 modules. The resultant tree, \(T_{M,I}\), is a junction tree.

- **Species Copying** – Stages 4 and 5 of Figure 3. The need for the species copying procedure is explained in the main text of the paper. Its name refers to the process of enlarging the intersection of 2 neighbouring modules, \(S_{de}\), by ‘copying’ certain species from a module containing those species, \(M_g\) say, both to \(M_d\) (which lies on the unique path between \(g\) and \(c\)) and \(M_e\). The species are also copied to all other modules on that path (when not already present in a module), in order to preserve the junction tree property. The aim is to ensure that when the reactions \(\Delta(S_{de})\) are grouped (i.e., allocated to equivalence classes) according to the change in the levels of the species \(S_{de}\) caused by the reaction, subsequent sub-grouping of reactions that are equivalent in terms of their membership of both sets \([\Delta(V_{de}\setminus S_{de}), \Delta(V_{cd}\setminus S_{de})]\) does not in fact alter the partition of the reactions \(\Delta(S_{de})\) obtained. When this condition is satisfied we say the partition, \(\mathcal{P}_{S_{de}}(\Delta(S_{de}))\), cannot be refined. When the condition fails, the groups that admit such sub-grouping are termed the failing reaction groups in \(\mathcal{P}_{S_{de}}(\Delta(S_{de}))\), whilst the remainder are termed the passing reaction groups. The subscript \(S_{de}\) in the notation \(\mathcal{P}_{S_{de}}(\cdot)\) refers to the set determining the sub-groupings, whilst the operator \(\mathcal{P}_{S_{de}}\) acts on a group of reactions (here, \(\Delta(S_{de})\)).

---

\(^1\)The term triangulation refers to the operation of adding edges to the undirected version of the KIG, \(G\), so that it becomes a triangulated graph. A minimal triangulation is one for which removal of any one of the edges added results again in an untriangulated graph. Since \(T^T\) is triangulated, it can be decomposed recursively (Cowell et al., 2007) until all the resulting subgraphs are complete (completeness meaning that any pair of nodes in the subgraph are connected). Such a recursive decomposition produces a collection of subgraphs containing the cliques of \(T^T\), i.e. the complete subgraphs of \(T^T\) that become incomplete if an additional node is included. Efficient algorithms have been developed in the graphical literature for both minimal triangulation and clique decomposition (Olesen and Madsen, 2002; Cowell et al., 2007) which may be used when implementing the MIDIA algorithm.
The structure of the Species Copying algorithm for a given edge is detailed in Algorithm 1 below. Although at first sight the algorithm may appear somewhat complicated, Algorithm 1 is relatively straightforward amongst potential approaches to solving this problem. The procedure is applied iteratively working ‘backwards’ through the junction tree, beginning with the edge involving the highest numbered module (and with all modules included in the donor list), then treating the edge involving the next highest numbered module, etc. At the start of each iteration, the ‘child module’ in the edge just treated is excluded from the list of donor modules – this is enough to ensure that for all edges, $S_{ab}$ say, treated in previous iterations, the triple $[V_{ab}, S_{ab}, V_{ba}]$ remains unchanged by this and subsequent iterations. The final output of the entire algorithm, $T_M$, is a junction tree.

**Algorithm 1 (Species Copying for an Edge)** Let $C$ denote the set of candidate species for addition to the edge, $S_{de}$. Set the variable $\text{stage}$ to 1, the set of accepted species $A = \emptyset, R = \emptyset, S = S_{de}$ and $T$ equal to the current junction tree.

1. Set $C = \{(\text{all species contained in the current list of donor modules}) \setminus S_{de}\} \cap \{\text{all species changed by the reactions in the failing reaction groups in } P_{S_{de}}(\Delta(S_{de}))\};$

2. If $(P_{S \cup A}(\Delta(S \cup A))$ cannot be refined), return($T);$

3. If (stage $= 2$ and $P_{S \cup A}(\Delta(S))$ cannot be refined), set $S = S \cup A, A = \emptyset, R = \emptyset, \text{stage} = 1$ and $C = \{(\text{all species contained in the current list of donor modules}) \setminus S\} \cap \{\text{all species changed by the reactions in the failing reaction groups in } P_{S}(\Delta(S))\};$

4. Set $c$ to a species in $C$ which results in the largest reduction in the number of failing reaction groups when $P_{A \cup S}(\Delta(S))$ is compared with $P_{A \cup S}(\Delta(S))$, or if this reduction is zero, the largest reduction in the number of reactions contained in failing reaction groups. If the reduction is zero, set stage to 2, $C = C \cup R$ and go to step 2;

5. If (there exist 2 reactions in a passing reaction group of $P_{S}(\Delta(S))$ that change $c$ identically), then reject $c$, setting $C = C \setminus c$ and go to step 2;

6. If (stage $= 1$ and $P_{S \cup A \cup c}(\Delta(S \cup A \cup c) \setminus (\Delta(S)))$ can be refined) then reject $c$, setting $C = C \setminus c, R = R \cup c$, and go to step 2;

7. Accept $c$, setting $A = A \cup c$ and $C = C \setminus c$. Now either $c \in V_{de}$ or $c \in V_{ed}$, but not both (where sub-tree species sets refer to the current state of the tree). Update $T$ as follows: if $c \in V_{de}$ (resp. $V_{ed}$) then ‘copy’ $c$ to all modules on the shortest path between a module containing $c$ and $M_e$ (resp. $M_d$) – this path inevitably passes through $M_d$ (resp. $M_e$);

8. Go to step 2;

The motivation for the central aspects of Algorithm 1 are as follows. The algorithm accepts candidate species, adding them to $A$, so as to reduce the number of failing reaction groups in $P_{S_{de} \cup A}(\Delta(S_{de}))$. It does this by concentrating on the partition $(P_{S_{de} \cup A}(\cdot))$ of the reactions in the failing groups of $P_{S_{de}}(\Delta(S_{de}))$. Notice that with an unrestricted donor list, the set of candidates $C$ in Step 1, if added to $S_{de}$, is guaranteed to ensure that every reaction in those failing reaction groups changes $(S_{de} \cup C)$ uniquely (and hence the associated partition cannot be refined). Step 5 ensures that the passing reaction groups of $P_{S_{de}}(\Delta(S_{de}))$ either appear unchanged in $P_{S_{de} \cup A}(\Delta(S_{de}))$ or are split into singletons (since the reactions involved then all change every $c$ accepted differently and hence change $S_{de} \cup A$ differently). Again, singletons cannot be refined. If stage $= 1$, then Step 6 rejects candidates whose acceptance would result in the ability to refine the partition of reactions that change the accepted species but do not change $S_{de}$. Stage 1 terminates and stage 2 commences the first time a zero reduction is encountered in Step 4. Step 3 then results in a further ‘iteration’ of the procedure (with $S$ re-initialised as $S \cup A$) when stage $= 2$ and $P_{S \cup A}(\Delta(S))$ cannot be refined (even though the partition of ultimate interest $P_{S \cup A}(\Delta(S \cup A))$ can still be refined).
4 Modularisation of the NFκB Signalling Network (SM4)

The junction tree representation, $T_M$, of the modularisation returned by the MIDIA algorithm is shown in Figure 1 below.

Figure 1. Modularisation $T_M$ for the NFκB Signalling Network. The modularisation is the one returned after stage 5 of the MIDIA algorithm [using a minimum residual size of 4 (except for the ‘root’ module)], and is based on dynamic conditional independence (see main text). Each module $M_d$ is labelled with the corresponding residual ($M_d \setminus S_d$); each edge is labelled with the intersection of the 2 modules it connects.

5 Core Species of the TLR Signalling Network (SM5)

For the large group of 12 inputs in the TLR network, $TLRI_{L1,7,9}$, we identified a ‘core’ of 48 species involved in the three sequences of encoders ending at each of the three output modules (and beginning in each case with edge 29 in $T_M$) – see main text. These core species are listed below (species names as in Li et al. 2009, Table S1):

MYD88-D(c), TRAF6-D(c), TIR_MyD(c), TIR(c), TIRAP(c), TIR_TIRAP(c), TICAM1(c), TICAM1P(c), IRAK1C_TOLLIP(c), TIFA(c), TRD3A(c), TLR2(c), TLR7/L-D(v), TLR8/L-D(v), TLR9/L-D(v), TLR9(v), TLR9(c), TLR9_SIGIRR(c), TLR10(c) ILLR1(c), IRF7(c), IRF7-2P(c), CASP3(c), IKK_RIP2_NOD1P(c), IKK_RIP2_NOD2P(c), IKK_RIP2_TRIP6_TRAF2(c), IKK_PKR(c), IKK_PKR_TRAF6(c), IKK_RIP1_TICAM1P(c), IKK_SRC(c), IKK(c), IKK-2P(c), AKT-2P(c), PI3K1A(c), PI3K1A-P(c), NFKB(p50/p65)(c), NFKB(p50/p65)-P3(c),
NFKB(p50/p65)-P5(c), NFKB_P3K(c), NFKB_P3K-4P(c), NFKB_P5K(c), NFKB_P5K-4P(c), adp(c), atp(c), adp(n), atp(n), h(c), h(l), UBIQ(c).

References


