Supplementary materials to: Inferring Orthologous Gene Regulatory Networks using Interspecies Data Fusion

Christopher A. Penfold, Jonathan B.A. Millar and David L. Wild

1 Gibbs updates in the OCSI frameworks

For the hierarchical OCSI we are interested in the following distribution:

$$\mathbb{P}(\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(d)},\mathcal{G}^{*}|\mathbf{X},\Phi,\beta) \quad \propto \quad \frac{\exp(-\beta\mathcal{E}(\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(d)},\mathcal{G}^{*}))}{\mathcal{Z}^{\mathrm{GK}}(\beta)}\prod_{j=1}^{d}\frac{\mathbb{P}(\mathbf{X}^{(j)}|\mathcal{G}^{(j)},\phi^{(j)})}{\mathcal{Z}^{\mathrm{GP}}(\phi^{(j)})}, \tag{1}$$

where the GRN in the kth species is represented as the union of the parental sets over the individual nodes: $\mathcal{G}^{(k)} = \bigcup_{i=1}^{g^{(k)}} \mathcal{P}a_i^{(k)}$. Supplementary Equation (1) can therefore be rewritten as:

$$\mathbb{P}(\mathcal{P}a_1^{(1)},\ldots,\mathcal{P}a_{g^{(1)}}^{(1)},\ldots,\mathcal{P}a_1^{(d)},\ldots,\mathcal{P}a_{g^{(d)}}^{(d)},\mathcal{P}a_1^{*},\ldots,\mathcal{P}a_{g^{*}}^{*}|\mathbf{X},\Phi,\beta) \quad \propto \quad \frac{\exp(-\beta\mathcal{E}(\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(d)},\mathcal{G}^{*}))}{\mathcal{Z}^{\mathrm{GK}}(\beta)}\prod_{j=1}^{d}\prod_{i=1}^{g}\frac{\mathbb{P}(\mathbf{X}_i^{(j)}|\mathcal{P}a_i^{(j)},\phi_i^{(j)})}{\mathcal{Z}^{\mathrm{GP}}(\phi_i^{(j)})}.$$

We may then Gibbs sample the parental set for each of the nodes in each of the species in turn:

$$\begin{aligned}
\mathcal{P}a_1^{(1)} &\sim \mathbb{P}(\mathcal{P}a_1^{(1)}|\mathcal{P}a_2^{(1)},\ldots,\mathcal{P}a_{g^{(1)}}^{(1)},\ldots,\mathcal{P}a_1^{(d)},\ldots,\mathcal{P}a_{g^{(d)}}^{(d)},\mathcal{P}a_1^{*},\ldots,\mathcal{P}a_{g^{*}}^{*},\mathbf{X},\Phi,\beta)\\
\mathcal{P}a_2^{(1)} &\sim \mathbb{P}(\mathcal{P}a_2^{(1)}|\mathcal{P}a_1^{(1)},\mathcal{P}a_3^{(1)},\ldots,\mathcal{P}a_{g^{(1)}}^{(1)},\ldots,\mathcal{P}a_1^{(d)},\ldots,\mathcal{P}a_{g^{(d)}}^{(d)},\mathcal{P}a_1^{*},\ldots,\mathcal{P}a_{g^{*}}^{*},\mathbf{X},\Phi,\beta),\\
&\vdots\\
\mathcal{P}a_1^{(d)} &\sim \mathbb{P}(\mathcal{P}a_1^{(d)}|\mathcal{P}a_1^{(1)},\ldots,\mathcal{P}a_{g^{(1)}}^{(1)},\ldots,\mathcal{P}a_{g^{(d-1)}}^{(d-1)},\mathcal{P}a_2^{(d)},\ldots,\mathcal{P}a_{g^{(d)}}^{(d)},\mathcal{P}a_1^{*},\ldots,\mathcal{P}a_{g^{*}}^{*},\mathbf{X},\Phi,\beta)\\
\mathcal{P}a_2^{(d)} &\sim \mathbb{P}(\mathcal{P}a_2^{(d)}|\mathcal{P}a_1^{(1)},\ldots,\mathcal{P}a_{g^{(1)}}^{(1)},\ldots,\mathcal{P}a_{g^{(d-1)}}^{(d-1)},\mathcal{P}a_1^{(d)},\ldots,\mathcal{P}a_{g^{(d)}}^{(d)},\mathcal{P}a_1^{*},\ldots,\mathcal{P}a_{g^{*}}^{*},\mathbf{X},\Phi,\beta),\\
&\vdots\\
\mathcal{P}a_1^{*} &\sim \mathbb{P}(\mathcal{P}a_1^{*}|\mathcal{P}a_2^{(1)},\ldots,\mathcal{P}a_{g^{(1)}}^{(1)},\ldots,\mathcal{P}a_1^{(d)},\ldots,\mathcal{P}a_{g^{(d)}}^{(d)},\mathcal{P}a_2^{*},\ldots,\mathcal{P}a_{g^{*}}^{*},\mathbf{X},\Phi,\beta)\\
&\vdots\\
\mathcal{P}a_{g^{*}}^{*} &\sim \mathbb{P}(\mathcal{P}a_{g^{*}}^{*}|\mathcal{P}a_1^{(1)},\ldots,\mathcal{P}a_{g^{(1)}}^{(1)},\ldots,\mathcal{P}a_1^{(d)},\ldots,\mathcal{P}a_{g^{(d)}}^{(d)},\mathcal{P}a_1^{*},\ldots,\mathcal{P}a_{g^{*}-1}^{*},\mathbf{X},\Phi,\beta).
\end{aligned}$$

For example the Gibbs update for node 1 in species 1 can be written as:

$$\mathbb{P}(\mathcal{P}a_1^{(1)}|\mathcal{P}a_2^{(1)},\ldots,\mathcal{P}a_{g^{(1)}}^{(1)},\ldots,\mathcal{P}a_1^{(d)},\ldots,\mathcal{P}a_{g^{(d)}}^{(d)},\mathcal{P}a_1^{*},\ldots,\mathcal{P}a_{g^{*}}^{*},\mathbf{X},\Phi,\beta) \quad = \quad \frac{\frac{\exp(-\beta\mathcal{E}(\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(d)},\mathcal{G}^{*}))}{\mathcal{Z}^{\mathrm{GK}}(\beta)}\frac{\mathbb{P}(\mathbf{X}_1^{(1)}|\mathcal{P}a_1^{(1)},\phi_1^{(1)})}{\mathcal{Z}^{\mathrm{GP}}(\phi_1^{(1)})}}{\sum_{pa}\frac{\exp(-\beta\mathcal{E}(\hat{\mathcal{G}}^{(1)},\ldots,\mathcal{G}^{(d)},\mathcal{G}^{*}))}{\mathcal{Z}^{\mathrm{GK}}(\beta)}\frac{\mathbb{P}(\mathbf{X}_1^{(1)}|pa,\phi_1^{(1)})}{\mathcal{Z}^{\mathrm{GP}}(\phi_1^{(1)})}},$$

where $\mathcal{P}a_1^{(1)} \in \mathcal{P}_c(\mathcal{T})$, $pa \in \mathcal{P}_c(\mathcal{T})$, $\mathcal{G}^{(1)} = \bigcup_{i=1}^{g^{(1)}} \mathcal{P}a_i^{(1)}$, and $\hat{\mathcal{G}}^{(1)} = pa \ \bigcup_{i=2}^{g^{(1)}} \mathcal{P}a_i^{(1)}$. Thus if we fix β and Φ, we may Gibbs sample the parental sets for each node in turn without the need to explicitly calculate these normalising constants. Similarly, the update for node 1 in the hypernetwork can be written:

$$\mathbb{P}(\mathcal{P}a_1^*|\mathcal{P}a_1^{(1)}, \ldots, \mathcal{P}a_{g^{(1)}}^{(1)}, \ldots, \mathcal{P}a_1^{(d)}, \ldots, \mathcal{P}a_{g^{(d)}}^{(d)}, \mathcal{P}a_2^*, \ldots, \mathcal{P}a_{g^*}^*, \mathbf{X}, \Phi, \beta) \quad = \quad \frac{\frac{\exp(-\beta\mathcal{E}(\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(d)},\mathcal{G}^*))}{\mathcal{Z}^{\mathrm{GK}}(\beta)}}{\sum_{pa^*} \frac{\exp(-\beta\mathcal{E}(\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(d)},\hat{\mathcal{G}}^*))}{\mathcal{Z}^{\mathrm{GK}}(\beta)}},$$

where $\mathcal{P}a_1^* \in \mathcal{P}_c(\mathcal{T})$, $\mathcal{G}^* = \bigcup_{i=1}^{g^*} \mathcal{P}a_i^*$, and $\hat{\mathcal{G}}^* = pa^* \ \bigcup_{i=2}^{g^*} \mathcal{P}a_i^*$.

For the non-hierarchical OCSI we are interested in sampling from the following distribution:

$$\mathbb{P}(\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(d)}|\mathbf{X}, \Phi, \beta) \quad \propto \quad \frac{\exp(-\beta\mathcal{E}(\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(d)}))}{\mathcal{Z}^{\mathrm{GK}}(\beta)} \prod_{j=1}^{d} \frac{\mathbb{P}(\mathbf{X}^{(j)}|\mathcal{G}^{(j)}, \phi^{(j)})}{\mathcal{Z}^{\mathrm{GP}}(\phi^{(j)})}, \tag{2}$$

which we may rewritten as:

$$\mathbb{P}(\mathcal{P}a_1^{(1)}, \ldots, \mathcal{P}a_{g^{(1)}}^{(1)}, \ldots, \mathcal{P}a_1^{(d)}, \ldots, \mathcal{P}a_{g^{(d)}}^{(d)}|\mathbf{X}, \Phi, \beta) \quad \propto \quad \frac{\exp(-\beta\mathcal{E}(\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(d)}))}{\mathcal{Z}^{\mathrm{GK}}(\beta)} \prod_{j=1}^{d} \prod_{i=1}^{g} \frac{\mathbb{P}(\mathbf{X}_i^{(j)}|\mathcal{P}a_i^{(j)}, \phi_i^{(j)})}{\mathcal{Z}^{\mathrm{GP}}(\phi_i^{(j)})}.$$

Again we may Gibbs sample the parental set for each of the nodes in each of the species in turn:

$$\begin{aligned}
\mathcal{P}a_1^{(1)} &\sim \mathbb{P}(\mathcal{P}a_1^{(1)}|\mathcal{P}a_2^{(1)}, \ldots, \mathcal{P}a_{g^{(1)}}^{(1)}, \ldots, \mathcal{P}a_1^{(d)}, \ldots, \mathcal{P}a_{g^{(d)}}^{(d)}, \mathbf{X}, \Phi, \beta) \\
\mathcal{P}a_2^{(1)} &\sim \mathbb{P}(\mathcal{P}a_2^{(1)}|\mathcal{P}a_1^{(1)}, \ldots, \mathcal{P}a_{g^{(1)}}^{(1)}, \ldots, \mathcal{P}a_1^{(d)}, \ldots, \mathcal{P}a_{g^{(d)}}^{(d)}, \mathbf{X}, \Phi, \beta), \\
&\vdots \\
\mathcal{P}a_1^{(d)} &\sim \mathbb{P}(\mathcal{P}a_1^{(d)}|\mathcal{P}a_2^{(1)}, \ldots, \mathcal{P}a_{g^{(1)}}^{(1)}, \ldots, \mathcal{P}a_2^{(d)}, \ldots, \mathcal{P}a_{g^{(d)}}^{(d)}, \mathbf{X}, \Phi, \beta), \\
&\vdots \\
\mathcal{P}a_{g^{(d)}}^{(d)} &\sim \mathbb{P}(\mathcal{P}a_2^{(d)}|\mathcal{P}a_1^{(1)}, \ldots, \mathcal{P}a_{g^{(1)}}^{(1)}, \ldots, \mathcal{P}a_1^{(d)}, \ldots, \mathcal{P}a_{g^{(d)}-1}^{(d)}, \mathbf{X}, \Phi, \beta).
\end{aligned}$$

Again, as an example, the Gibbs update for node 1 in species 1 can be written as:

$$\mathbb{P}(\mathcal{P}a_1^{(1)}|\mathcal{P}a_2^{(1)}, \ldots, \mathcal{P}a_{g^{(1)}}^{(1)}, \ldots, \mathcal{P}a_1^{(d)}, \ldots, \mathcal{P}a_{g^{(d)}}^{(d)}, \mathbf{X}, \Phi, \beta) \quad = \quad \frac{\frac{\exp(-\beta\mathcal{E}(\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(d)})}{\mathcal{Z}^{\mathrm{GK}}(\beta)} \frac{\mathbb{P}(\mathbf{X}_1^{(1)}|\mathcal{P}a_1^{(1)},\phi_1^{(1)})}{\mathcal{Z}^{\mathrm{GP}}(\phi_1^{(1)})}}{\sum_{pa} \frac{\exp(-\beta\mathcal{E}(\hat{\mathcal{G}}^{(1)},\ldots,\mathcal{G}^{(d)}))}{\mathcal{Z}^{\mathrm{GK}}(\beta)} \frac{\mathbb{P}(\mathbf{X}_1^{(1)}|pa,\phi_1^{(1)})}{\mathcal{Z}^{\mathrm{GP}}(\phi_1^{(1)})}},$$

where $\mathcal{P}a_1^{(1)} \in \mathcal{P}_c(\mathcal{T})$, $pa \in \mathcal{P}_c(\mathcal{T})$, $\mathcal{G}^{(1)} = \bigcup_{i=1}^{g^{(1)}} \mathcal{P}a_i^{(1)}$, and $\hat{\mathcal{G}}^{(1)} = pa \ \bigcup_{i=2}^{g^{(1)}} \mathcal{P}a_i^{(1)}$.

1.1 Direct Leveraging of Networks

Within the main text we have outlined the general OCSI models for joint inference when time series data is available for all species. Another important example of data fusion in multiple species is the case in which networks are known in one species and we are interested in leveraging this network into a more relevant or useful organism. This example is particularly important for transferring information from model organisms, for which networks may be available from yeast one-hybrid or ChIP-Seq, into more relevant/useful organism. Note that, in this case time series data may still be available for the more relevant organism, and we are thus required to leveraged data in the form of networks and in the form of time series gene expression into our organism of interest. For the hierarchical OCSI (Section 2.1 in the main text) we therefore have the following model:

$$\begin{aligned}\mathbb{P}(\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(i-1)},\mathcal{G}^{(i+1)},\ldots,\mathcal{G}^{(d)},\mathcal{G}^{*}|\mathbf{X},\mathcal{G}^{(i)},\Phi,\beta) &\propto \mathbb{P}(\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(i-1)},\mathcal{G}^{(i+1)},\ldots,\mathcal{G}^{(d)},\mathcal{G}^{*}|\mathcal{G}^{(i)},\beta)\\ &\times \prod_{j=\{1,2,\ldots,i-1,i+1,\ldots,d\}} \mathbb{P}(\mathbf{X}^{(j)}|\mathcal{G}^{(j)},\Phi^{(j)}). \qquad (3)\end{aligned}$$

where we have assumed species i is our model organism. Similarly, for the non-hierarchical model (Section 2.2. in the main text) we have the following model:

$$\begin{aligned}\mathbb{P}(\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(i-1)},\mathcal{G}^{(i+1)},\ldots,\mathcal{G}^{(d)}|\mathbf{X},\mathcal{G}^{(i)}\Phi,\beta) &\propto \mathbb{P}(\mathcal{G}^{(d)},\ldots,\mathcal{G}^{(i-1)},\mathcal{G}^{(i+1)},\ldots,\mathcal{G}^{(d)}|\mathcal{G}^{(i)},\beta)\\ &\times \prod_{j=\{1,2,\ldots,i-1,i+1,\ldots,d\}} \mathbb{P}(\mathbf{X}^{(j)}|\mathcal{G}^{(j)},\phi^{(j)}), \qquad (4)\end{aligned}$$

where all other terms are as described in the main text. Even for the two species case, in which we are inferring a network only for a single species, we are nonetheless leveraging data (in the form of a network) from a second species, as well as time series for the species of interest, and are therefore leveraging data from multiple species. Inference in the above models remains similar to that of the full models i.e., via a series of Gibbs updates of parental sets for each node in each species in turn, except for the network in the species we have fixed (the model organism). Indeed, in this way we may choose to fix only sections of the networks of our model organisms, with the resultant Gibbs updates then applied to all nodes in all species except the nodes in the model organism whose parental sets have been experimentally verified and this fixed.

1.2 Estimation of $\mathcal{Z}^{\text{GK}}(\beta)$

For the models described in Supplementary Equations (1) and (2) (Equations (1) and (5) in the main text) we may readily sample network structure when β and Φ are fixed via a series of Gibbs updates. In order to compare models at different temperature values, however, we must estimate $\mathcal{Z}^{\text{GK}}(\beta)$ in the joint prior distribution over network structures. For the hierarchical model we have the following prior over network structures:

$$\mathbb{P}(\mathcal{G}^{(1)},\ldots\mathcal{G}^{(d)},\mathcal{G}^{*}|\beta) = \frac{\exp\Big(-\beta\mathcal{E}(\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(d)},\mathcal{G}^{*})\Big)}{\mathcal{Z}^{\text{GK}}(\beta)},$$

where $\mathcal{E}(\cdot)$ is calculated from Equation (4) in the main text, and the normalising constant is given by:

$$\mathcal{Z}^{\text{GK}}(\beta) = \sum_{g^{(1)}\in G^{(1)}}\sum_{g^{(2)}\in G^{(2)}}\cdots\sum_{g^{(d)}\in G^{(d)}}\sum_{g^{*}\in G^{*}}\exp\Big(-\beta\mathcal{E}(g^{(1)},\ldots g^{(d)},g^{*})\Big).$$

For the non-hierarchical model we have the following joint prior:

$$\mathbb{P}(\mathcal{G}^{(1)},\ldots\mathcal{G}^{(d)}|\beta) = \frac{\exp\Big(-\beta\mathcal{E}(\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(d)})\Big)}{\mathcal{Z}^{\text{GK}}(\beta)},$$

where $\mathcal{E}(\cdot)$ is given by Equation (6) in the main text, and the normalising constant can be written as:

$$\mathcal{Z}^{\mathrm{GK}}(\beta) \quad = \quad \sum_{g^{(1)}\in G^{(1)}} \sum_{g^{(2)}\in G^{(2)}} \cdots \sum_{g^{(d)}\in G^{(d)}} \exp\left(-\beta\mathcal{E}(g^{(1)}, \ldots g^{(d)})\right).$$

Whilst we cannot calculate $\mathcal{Z}^{\mathrm{GK}}(\beta)$ via explicit enumeration, we may sample network structures within a population Markov Chain Monte Carlo algorithm and estimate $\mathcal{Z}^{\mathrm{GK}}(\beta)$ from the samples via Thermodynamic Integration (Calderhead and Girolami, 2009). Since Φ remains fixed $\mathcal{Z}^{\mathrm{GP}}(\beta)$ cancels out when comparing the marginal likelihoods for models at different temperatures β.

2 Illustration of Graph Kernels

In **Figure 1** and **Figure 2** we illustrate Graphlet kernels and the Weisfeiler-Lehman (WL) graph kernels respectively. A graphlet kernel compares the similarity of two networks by decomposing each graph into subgraphs of size k and comparing the counts or frequency of each of the subgraphs. An extension to this is outlined in Shervashidze *et al.* (2011), which counts the common labeled connected subgraphs. In particular there are 4 types of graphlet with 3 nodes, and for each graphlet the number of possible labellings of the 3 nodes can be computed given the node label alphabet. The WL sequentially relabels nodes according to subtree patterns, and compares the updated relabelled graphs.

3 Benchmarking OCSI on the DREAM4 Datasets

The OCSI algorithm was first benchmarked on the DREAM4 datasets (Prill *et al.*, 2010; Greenfield *et al.*, 2010; Schaffter *et al.*, 2011; Marbach *et al.*, 2009). These datasets were chosen so that the results can be directly compared to a related approach that uses an identical Gaussian process-based model for the dynamics of gene expression (Penfold and Wild, 2011; Penfold *et al.*, 2012). This datasets consist of 5 time series that were generated from a network with identical topology albeit with perturbed dynamics. Since the dynamic model underpinning each time series has been perturbed, this test case may be considered equivalent to the case in which there are 5 species that are closely related enough to conserved network topology (retaining a 1:1 mapping between orthologues). Initial benchmarking was split into three cases:

1. The 5 time series were separated out, representing 5 species with identical networks, in which orthologues map 1:1 across species.
2. The 5 datasets was separated out as above, except that one node in each species was mislabelled. This was achieved by selecting a different node in each species and assigning it a new unique label not present in any of the other 4 networks. This corresponds to the case in which 10% of orthologues were incorrectly labelled, and highlights how the algorithm might perform in more realistic conditions in which orthologues are not know with 100% accuracy.
3. Only the first two time series were used, representing a two species system.

In all instances three separate MCMC chains were run, consisting of a series of Gibbs updates applied to the parental sets of each of the 10 nodes in the network in each of the species in turn. Additionally, for

Framework 1, a Gibbs update was applied to each of the 10 or 15 genes in the hypernetwork[1]. For the two dataset case this corresponds to 25 or 20 Gibbs updates per single MCMC step for Frameworks 1 and Framework 2 respectively, whilst for the 5 dataset case this consisted of 55 and 50 Gibbs sweeps per MCMC step for Framework 1 and 2 respectively. In total 3000 complete Gibbs updates were applied, with the first 500 steps discarded for burn-in. Performance was gauged against the gold standard networks using the average area under the receiver operating characteristic curve (AUC) evaluated. To evaluate the AUC the marginal distributions for the networks were first calculated, and a plot of the false positive rate versus the true positive rate numerically integrated in MATLAB (see e.g., Penfold and Wild (2011) for more details on how these properties were calculated).

The AUC score as a function of increasing β is shown for Framework 1 in Figure 3(a) for the 5 species case. Here OCSI was run for a range between $\beta = 0.1$ up to $\beta = 10$. In general, at very low values of β, performance of OCSI (solid line) was consistent with the performance of the standard CSI approach (dashed line), and was therefore significantly better than random. The AUC increased compared to the standard CSI as β increased for both the WL and graphlet kernels, peaking at $\beta = 5$ and $\beta = 8$ respectively, with the highest AUC scores observed with the WL kernel. It should be noted, however, that the general trend for AUC in the graphlet kernel appeared to still be increasing beyond $\beta = 10$ and more accurate networks might be expected if β was incremented beyond this. Performance of the shortest path kernel did not appear to increase with β. For all Graph kernels, if β was increased too much, exploration in the MCMC chains began to fail, as characterised by the increased variance in estimates across the 3 chains. Inspection of individual chains for higher values of β show decreased exploration of the parameter space, suggesting that chains have become trapped in local modes. Indeed for high values of β there will be many suboptimal whenever networks between species are similar.

Similar trends are observed for Framework 2, Figure 3(b), with the WL and graphlet kernels offering better performance than independent inference (with CSI), particularly for values of β that were intermediate over the range we tested. Here the shortest path kernel also appears to perform moderately well at intermediate values of β of the range tested, with optimal values found at approximately $\beta = 2$ for the WL and graphlet kernels and 1 for the shortest path kernel. As with Framework 1, the WL kernel appeared to perform better than the other two graph kernels. For the 5 datasets/species case, however, Framework 2 appeared to be less stable than Framework 1 i.e., the value of β at which exploration in the MCMC chain began to fail was lower than for Framework 1. This is likely due to the fact that for Framework 1 there are d terms involved in calculating the energy associated with the similarity (see Equation (4) in the main text), whilst for Framework 2 there are $(d^2 - d)/2$ (see Equation (6) in the main text), suggesting that direct leveraging of datasets may have more modes and those modes may be deeper than when leveraging information via a hypernetwork, apart from when the number of species $d \leq 3$. This is supported by the observation that in the two dataset case Framework 2 is more stable than Framework 1 (compare Supplementary Figure 3 with Supplementary Figure 4). This suggests that when leveraging relatively few datasets (< 3) Framework 2 is preferable, whilst for situations where there are datasets from many different species Framework 1 may be more suitable.

The OCSI framework also shows a general increases in AUC as a function of β when a small fraction (10%) of nodes were mislabelled (Supplementary Figures 5), with the WL kernel performing best. This suggests that these methods are relatively robust to incorrect labelling i.e., where one node in each of the individual datasets/species was assigned its own unique label rather than the true label corresponding to its correct set of orthologues.

[1] For cases were 1 node was mislabelled in each of the 5 datasets, this resulted in a new unique node in the hypernetwork. Thus, whilst there were only 10 nodes in the hypernetwork when all were correctly labelled, there were 15 when mislabelled.

4 Generating *in silico* datasets with duplicate genes

To test the OCSI approach for case in which orhologues did not map 1:1 between species, datasets were generated according to ODE models (see Figure 2 in the main text). Here the model in species A is based upon the repressillator (Elowitz and Leibler, 2000), a systems containing 3 genes, which interact via mRNA translation/degradation and proteins translation/degradation. A MATLAB implementation of the repressilator is available from the SYSMIC course (http://sysmic.ac.uk/home.html) and can be written as:

$$\begin{aligned}
\dot{m_1} &= -m_1 + \frac{\alpha}{1+p_3^n} + \alpha_0, \\
\dot{p_1} &= -\beta(p_1 - m_1), \\
\dot{m_2} &= -m_2 + \frac{\alpha}{1+p_1^n} + \alpha_0, \\
\dot{p_2} &= -\beta(p_2 - m_2) \\
\dot{m_3} &= -m_3 + \frac{\alpha}{1+p_2^n} + \alpha_0, \\
\dot{dp_3} &= -\beta(p_3 - m_3)
\end{aligned}$$

where m_i and p_i represents the expression level of the ith mRNA and protein respectively, and where $\alpha, \beta, n, \alpha_0$ represent model parameters. For species A these parameters were fixed to $[1, 1.75, 5, 1000] + \epsilon$ where epsilon represents Gaussian additive noise. A second species, B, was considered to have evolved from species A by perturbation of the dynamics i.e., it has the same model structure as above, with parameters $[1, 1.75, 5, 5] + \epsilon$. An intermediate species, I, was assumed to have evolved from species A by duplication of the represillator system. Subsequent loss of links, loss of duplicate node $3'$, and perturbation of dynamics led to species D, which is described by the following ODE system:

$$\begin{aligned}
\dot{m_1} &= -m_1 + \frac{\alpha}{1+p_3^n} + \alpha_0, \\
\dot{p_1} &= -\beta(p_1 - m_1), \\
\dot{m_2} &= -m_2 + \frac{\alpha}{1+p_1^n} + \alpha_0, \\
\dot{p_2} &= -\beta(p_2 - m_2) \\
\dot{m_3} &= -m_3 + \frac{\alpha}{1+p_2^n} + \alpha_0, \\
\dot{p_3} &= -\beta(p_3 - m_3) \\
\dot{m_4} &= -m_4 + \frac{\alpha}{1+p_2^n} + \alpha_0, \\
\dot{p_4} &= -\beta(p_4 - m_4) \\
\dot{m_4} &= -m_5 + \frac{\alpha}{1+p_3^n} + \alpha_0, \\
\dot{dp_4} &= -\beta(p_4 - m_4) \\
\dot{m_5} &= -m_6 + \frac{\alpha}{1+p_4^n} + \alpha_0, \\
\dot{dp_5} &= -\beta(p_5 - m_5)
\end{aligned}$$

where the parameters $[1, 1.75, 5, 5] + \epsilon$. Species E was assumed to evolved from species I via loss of several links in the network, and is described by the following ODE system:

$$
\begin{aligned}
\dot{m_1} &= -m_1 + \frac{\alpha}{1 + p_3^n} + \alpha_0, \\
\dot{p_1} &= -\beta(p_1 - m_1), \\
\dot{m_2} &= -m_2 + \frac{\alpha}{1 + p_1^n} + \alpha_0, \\
\dot{p_2} &= -\beta(p_2 - m_2) \\
\dot{m_3} &= -m_3 + \frac{\alpha}{1 + p_2^n} + \frac{\alpha}{1 + p_5^n} + \alpha_0, \\
\dot{dp_3} &= -\beta(p_3 - m_3) \\
\dot{m_4} &= -m_4 + \frac{\alpha}{1 + p_3^n} + \frac{\alpha}{1 + p_6^n} + \alpha_0, \\
\dot{dp_4} &= -\beta(p_4 - m_4) \\
\dot{m_5} &= -m_5 + \frac{\alpha}{1 + p_4^n} + \alpha_0, \\
\dot{dp_5} &= -\beta(p_5 - m_5) \\
\dot{m_6} &= -m_6 + \frac{\alpha}{1 + p_5^n} + \alpha_0, \\
\dot{dp_6} &= -\beta(p_6 - m_6)
\end{aligned}
$$

where the parameters $[1, 1.75, 5, 1000]$. Finally, species F evolved from species E via perturbation of the dynamics, with parameters $[1, 1.75, 5, 5] + \epsilon$. For all species time series were generated from the ODE system using MATLAB `ode45`.

5 Results on *in silico* dataset with duplicate nodes

The following supplementary Figures and files exist for benchmarking on *in silico* ODE models.

Figure 7 represents the AUC for the 2 species inference using the WL kernel on the ODE *in silico* datasets outlined in Supplementary Section 4.

Figure 8 represents the AUC for the situation in which the network in species A is known and fixed, and leverage into species E alongside time series for species E using the WL kernel on the ODE *in silico* datasets outlined in Supplementary Section 4.

Figure 9 represents the AUC for the situation in which the network in species C is known and fixed, and leverage into species E alongside time series for species E using the WL kernel on the ODE *in silico* datasets outlined in Supplementary Section 4.

6 Applications in Yeast

The following supplementary Figures and files exist for the benchmarking in yeast:

Figure 10 represents the leveraging of networks from *S.cerevisiae* into *S.pombe* additionally using time series data from *S.pombe.*

Table 1 indicates the number of cell cycle literature links identified when propagating information from *S.cerevisiae* into *S.pombe* using OCSI. Here two sources of literature are used Stark *et al.* (2006) and Pancaldi *et al.* (2012).

Supplementary File 1 contains *S.pombe* genes used for network inference (Supplementary_Table_1.csv)

Supplementary File 2 contains *S.cerevisiae* genes used for constructing the network (Supplementary_Table_2.csv)

Supplementary File 4 contains a Cytoscape file (Shannon *et al.*, 2003; Cline *et al.*, 2007; Smoot *et al.*, 2011; Saito *et al.*, 2012) the with the Inferred *S.pombe* networks containing the top 100 links (All_Networks_beta_15_Top100.cys)

Supplementary File 5 contains a Cytoscape file with the inferred *S.pombe* networks containing the top 200 links (All_Networks_beta_15_Top200.cys)

Supplementary File 6 contains a Cytoscape file with the inferred *S.pombe* networks containing top 300 links (All_Networks_beta_15_Top300.cys)

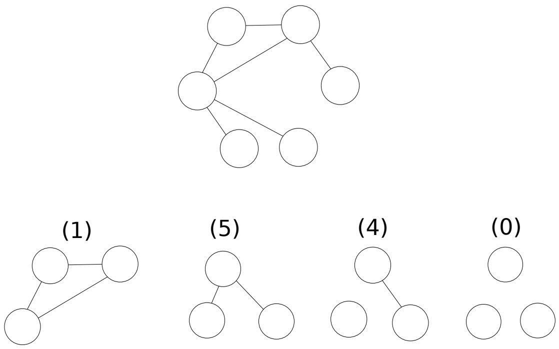

Figure 1: Informally, a graphlet kernel compares the similarity of two networks by decomposing each graph into subgraphs of size k and comparing the counts or frequency of each of the subgraphs. In this paper we exclusively use 3-node connected graphlets and hence $k = 3$. The example graph, above, can be decomposed into 4 different 3-node graphlets, with the number of instances of each graphlet indicated in brackets. For this example the k-spectrum $f_{\mathcal{G}} = (1, 5, 4, 0)$. Given another graph $\mathcal{G}'$, we can calculate the kernel: $K_{\mathrm{g}}(\mathcal{G}, \mathcal{G}') = f_{\mathcal{G}}^{\top} f_{\mathcal{G}'}$. Where the graphs are significantly different in size the counts $f_{\mathcal{G}}$ and $f'_{\mathcal{G}}$ may be heavily skewed. Here the k-spectrum may be normalised to frequency: $D_{\mathcal{G}} = f_{\mathcal{G}} / N_{\mathcal{G}}$, where $N_{\mathcal{G}}$ is the total number of graphlets in $\mathcal{G}$. The normalised kernel is then: $K_{\mathrm{g}}(\mathcal{G}, \mathcal{G}') = D_{\mathcal{G}}^{\top} D_{\mathcal{G}'}$.

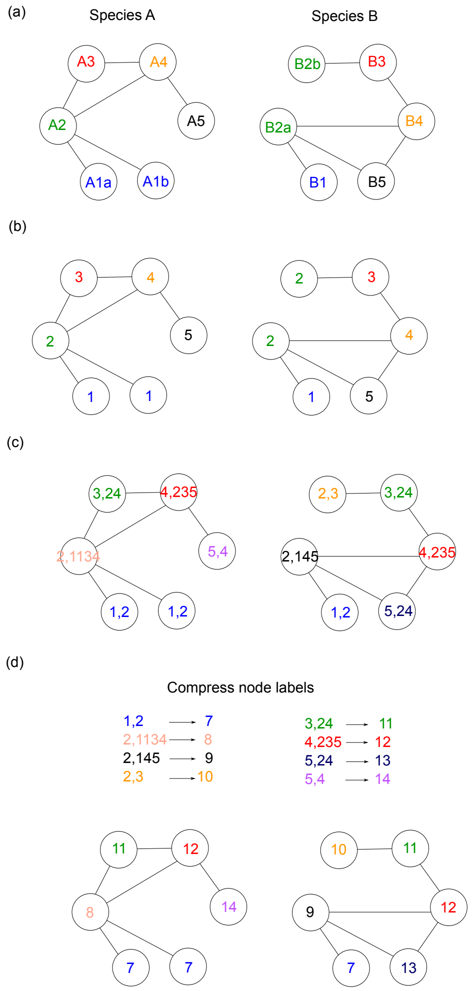

Figure 2: Illustration of Weisfeiler-Lehman graph kernels adapted from Shervashidze *et al.* (2011). (a) A pair of graphs indicating the GRNs for two species A and B. In this example species A contains two copies of gene 1 denoted $A1a$ and $A1b$, whilst species B has one orthologue of gene 1, denoted $B1$. Conversely, species B has two copies of gene 2 ($B2a$ and $B2b$) with species A only containing one orthologue ($A2$). The remainder of the genes map 1:1 across the species. Each node may be assigned a label according to the orthologues it has (b). From the base network a feature vector can be calculated according the node labels in the two graphs: $\phi^{(0)}_{\mathrm{WLsubtree}}(\mathcal{G}) = (2,1,1,1,1)$ and $\phi^{(0)}_{\mathrm{WLsubtree}}(\mathcal{G}') = (1,2,1,1,1)$. Here we can calculate: $K^{(0)}_{\mathrm{WLsubtree}}(\mathcal{G},\mathcal{G}') =< \phi^{(0)}_{\mathrm{WLsubtree}}(\mathcal{G}), \phi^{(0)}_{\mathrm{WLsubtree}}(\mathcal{G}') >= 7$. Within the WL kernel networks undergo a series of relabelling. The first step (c) involves appending each node label with the ordered list of connected nodes. Updated node labels can then be compressed (d) to yield an updated graph (note the topology remains the same, but the labels change). Updated feature vectors can again be calculated: $\phi^{(1)}_{\mathrm{WLsubtree}}(\mathcal{G}) = (2,1,1,1,1,2,1,0,0,1,1,0,1)$, $\phi^{(1)}_{\mathrm{WLsubtree}} and(\mathcal{G}') = (1,2,1,1,1,1,0,1,1,1,1,1,0)$. Note these updated feature vectors contain the original uncompressed feature vectors. The updated WL kernel is then $K^{(1)}_{\mathrm{WLsubtree}}(\mathcal{G},\mathcal{G}') = 16$. Each subsequent round of relabelling will capture more information about the networks and their similarity. In this paper we choose height $h = 2$.

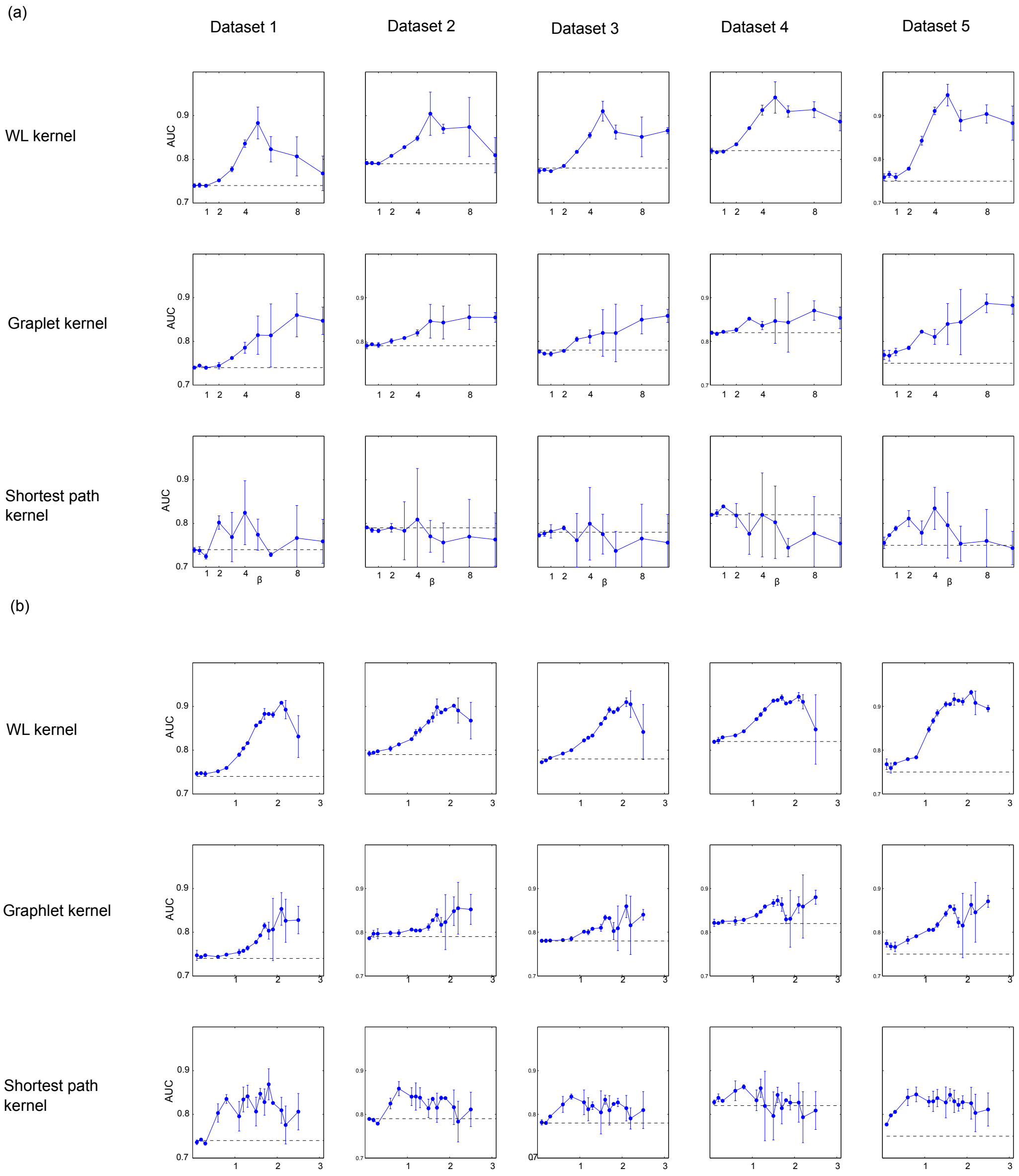

Figure 3: A five species joint inference benchmark using the DREAM4 datasets. Area under the Receiver Operating Characteristic (AUC) curve (y-axis) as a function of β (x-axis) using time series for network 5 in the10-gene DREAM4 challenge. This dataset consist of 5 perturbed time series (generated from an identical network) that are represented by the different columns. Each time series was thus assumed to correspond to a separate species, with orthologues mapping 1:1 between species. (a) The AUC for OCSI Framework 1 is shown for the three graph kernels: WL kernel (top row), a 3-node connected graphlet (middle row) and a shortest path kernel (bottom row). Performance of the standard CSI algorithm is indicated by a black dashed line. In general increasing β results in better performance for the WL and graphlet kernels. (b) The AUC as a function of β for OCSI Framework 2 on the same DREAM4 dataset as used in (a), with different time series represented by the different columns. Again, increasing β results in significantly better performance compared to the standard CSI approach (dashed lines).

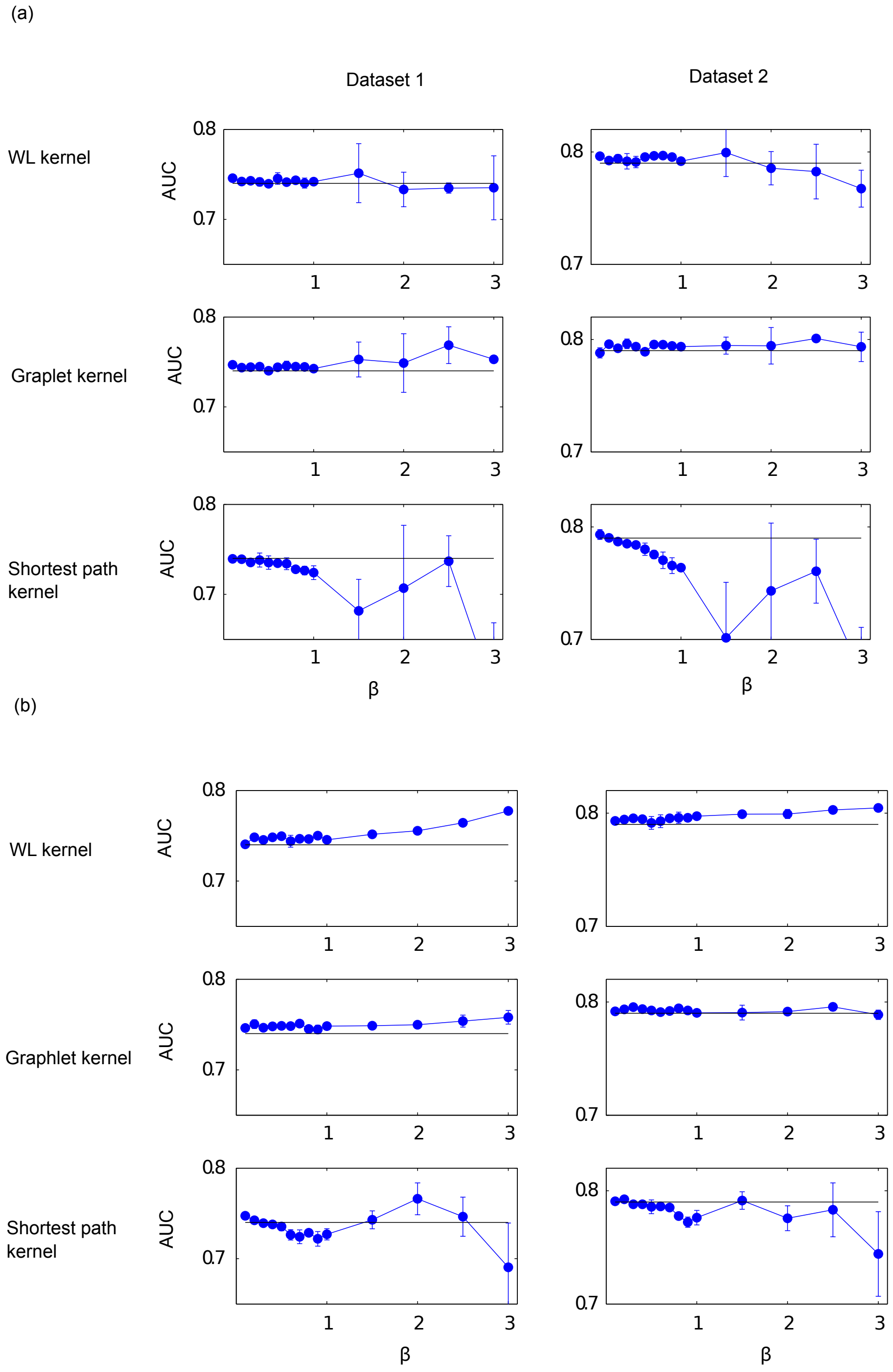

Figure 4: A two species joint inference benchmark using the DREAM4 datasets. Area under Receiver Operating Characteristic (AUC) curve (y-axis) as a function of β (x-axis) for network 5 of the DREAM4 challenge using OCSI. Here only time series 1 and 2 were used, and were assumed to correspond to two species with orthologues mapping 1:1 between them. (a) The AUC for OCSI Framework 1 for the three graph kernels: WL kernel (top row), a 3-node connected graphlet (middle row) and a shortest path kernel (bottom row). Here only the first 2 time series from the DREAM4 dataset were used (represented by the two columns). Performance of the standard CSI algorithm is indicated by a black dashed line. In general increasing β results in better performance for the WL and graphlets kernels. (b) AUC as a function of β for OCSI Framework 2 on the same DREAM4 dataset. Again, increasing β results in significantly better performance compared to the standard CSI approach.

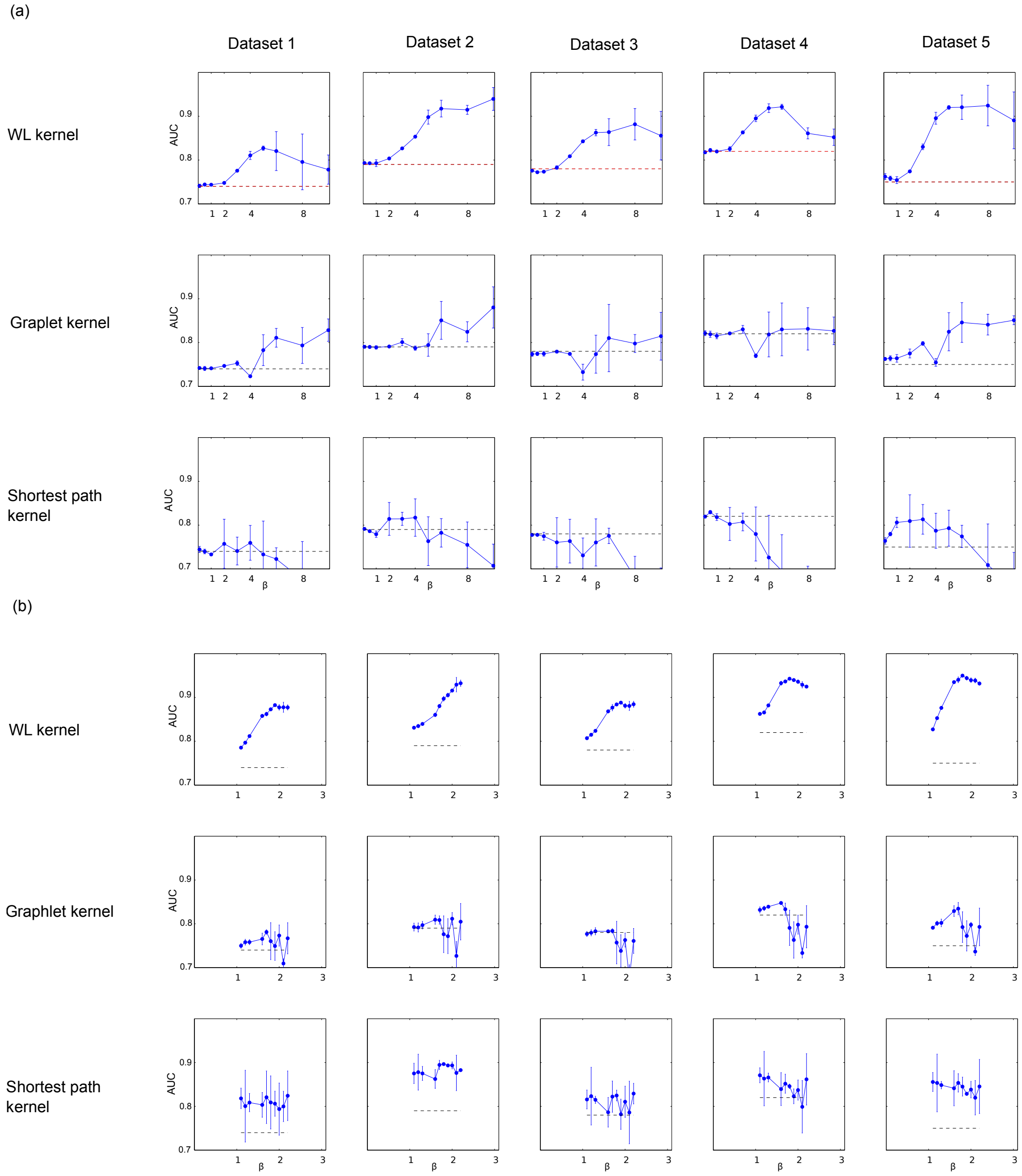

Figure 5: A five species joint inference benchmark with mislabelled nodes using the DREAM4 datasets. Area under Receiver Operating Characteristic (AUC) curve (y-axis) as a function of β (x-axis) for network 5 of the DREAM4 challenge. In this case all 5 time series were again used for the inference, with 10% of nodes are mislabelled. (a) AUC for OCSI Framework 1 for the WL kernel (top row), a 3-node connected graphlet (middle row) and a shortest path kernel (bottom row). Performance of the standard CSI algorithm is indicated by a black dashed line. In general increasing β results in better performance for the WL and graphlets kernels as in previous examples. (b) AUC as a function of β for OCSI Framework 2 on the same DREAM4 dataset when 10% of nodes are mislabelled. Again, increasing β results in significantly better performance compared to the standard CSI approach.

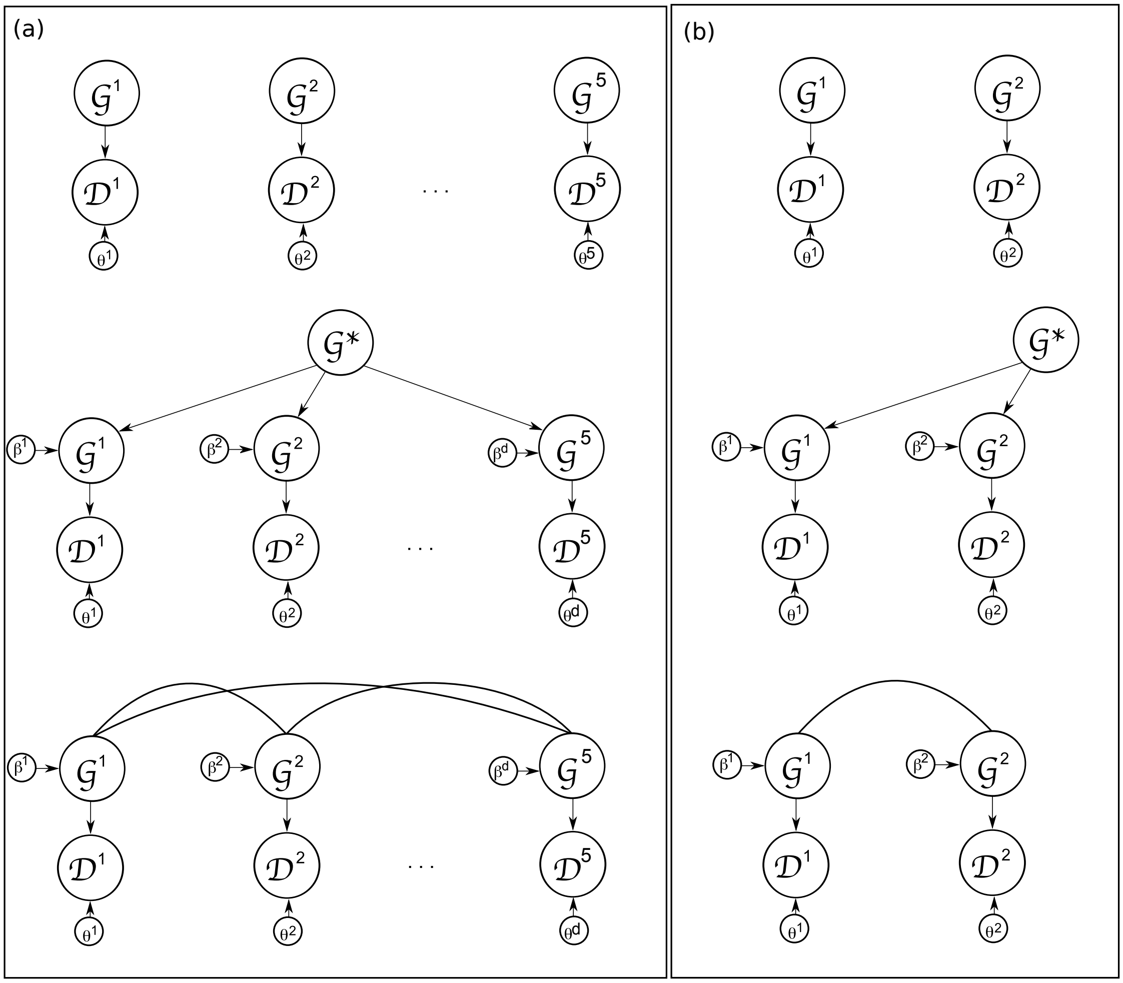

Figure 6: Graphical model representations of CSI and OCSI frameworks. (a) Inference with 5 species using the independent CSI model (top), OCSI Framework 1 (middle) and OCSI Framework 2 (bottom); (b) Inference with two species with the independent CSI framework (top), OCSI Framework 1 (middle) and OCSI Framework 2 (bottom).

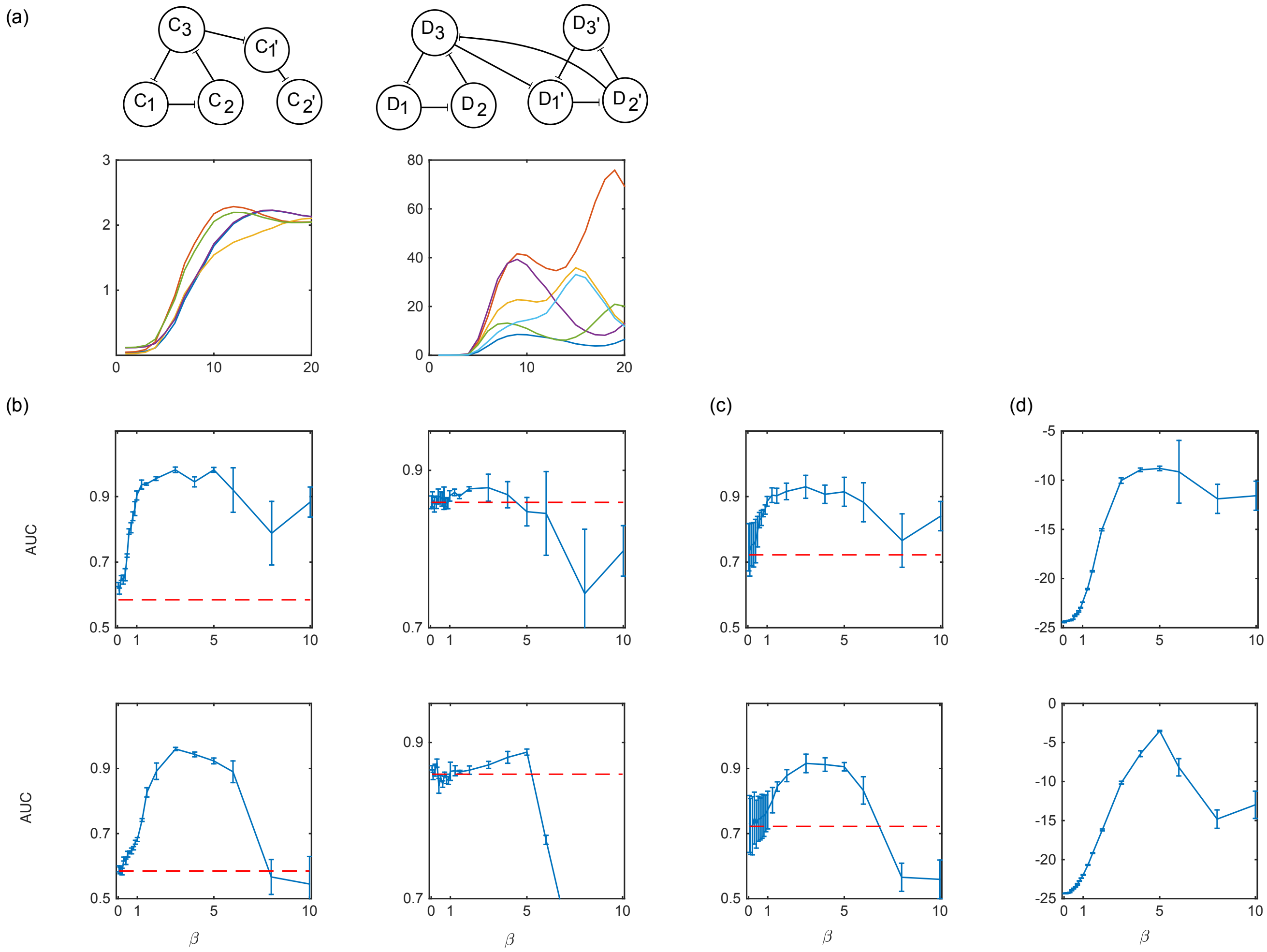

Figure 7: A two species joint inference benchmark using ODE datasets. (a) Joint learning using two datasets, one from species C and another from species D. (b) AUC for the two species for non-hierachical/Framework 2 (top) and hierarchical/Framework 1 (bottom). (c) AUC averaged over both species. (d) Estimated marginal likelihood.

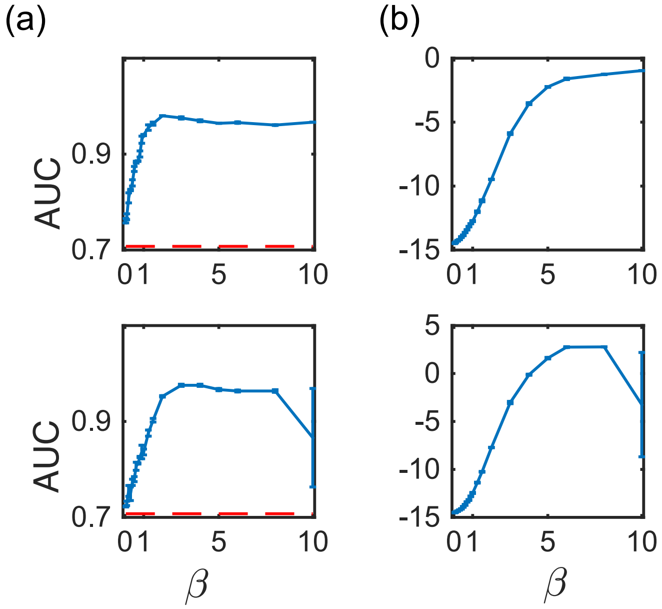

Figure 8: Leveraging of the network in species A, alongside time series of species E to infer the network in E. (a) AUC for the inferred network in species E for Framework 1 (bottom) and Framework 2 (top) as a function of β. (b) Estimated marginal likelihood for the models as a function of β.

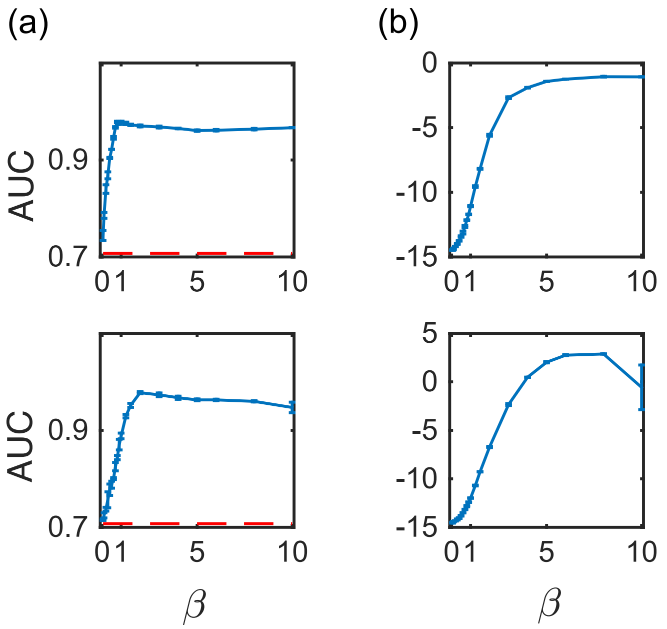

Figure 9: A two species network leveraging benchmark using ODE datasets. (a) Leveraging of the network in species C, alongside time series of species E to infer the network in E. (a) AUC for the inferred network in species E for Framework 1 (bottom) and Framework 2 (top) as a function of β. (b) Estimated marginal likelihood for the models as a function of β.

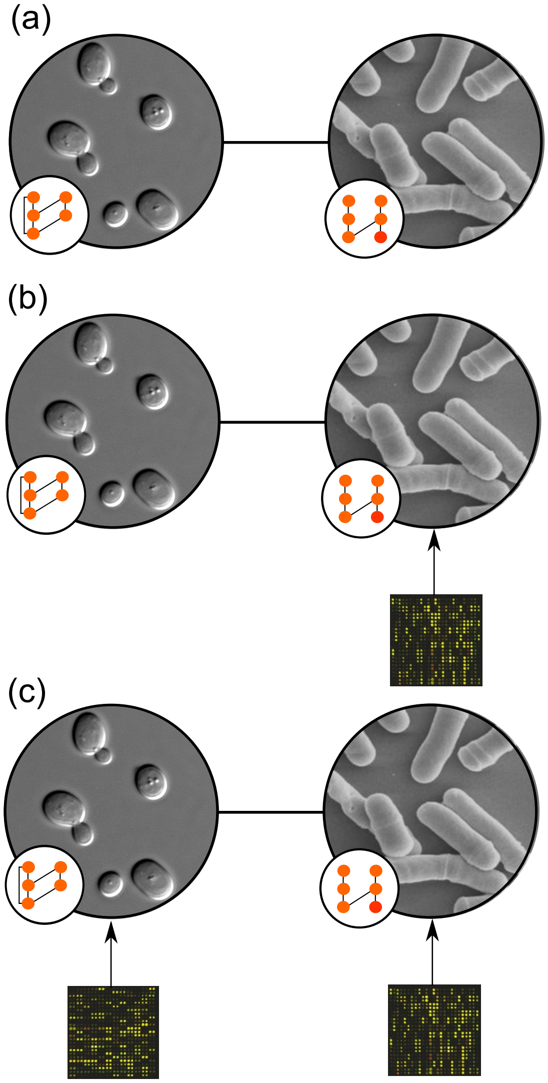

Figure 10: (a) An *S.pombe* cell-cycle network is inferred by directly leveraging an *S.cerevisiae* network (Li *et al.*, 2004) alongside *S.pombe* cell-cycle time series dataset (Rustici *et al.*, 2004). (b) Graphical representation of the joint inference of networks in *S.cerevisiae* and *S.pombe* using time series data from both species, as well as prior information about the *S.cerevisiae* network. In both examples networks were inferred using Framework OCSI framework 2 due to the small number of species.

Table 1: Number of links in agreement with the literature. Here we indicate how many of the inferred *S.pombe* network connections are found in the literature Stark *et al.* (2006) and Pancaldi *et al.* (2012) when we propagate the *S.cerevisiae* network without *S.pombe* time series expression data (Prior only) versus using network leveraging (NL; combining the *S.cerevisiae* network with *S.pombe* data to infer the *S.pombe* network) and joint inference (JI; inferring networks in both *S.cerevisiae* and *S.pombe* using time series data from both). We also indicate the expected number of connections for a random network based statistics of the hypergeometric distribution. It should be noted that these numbers are not necessarily indicative of overall performance as connections in Stark *et al.* (2006) are limited, whilst connections in Pancaldi *et al.* (2012) represent inferences and protein binding rather than transcriptional regulation.

		Connections in Stark *et al.* (2006)			Connections in Pancaldi *et al.* (2012)		
	Network	WL	Graphlet	SP	WL	Graphlet	SP
Top 200	Prior only	4	3	3	25	33	26
	OCSI (L)	4	2	1	31	36	33
	OCSI (JI)	4	2	2	37	30	33
	Random $\mathbb{E}(n)$	0.73			21		
Top 300	Prior only	4	5	4	35	47	36
	OCSI (L)	8	4	5	48	53	46
	OCSI (JI)	7	3	2	56	44	44
	Random $\mathbb{E}(n)$	1.5			31		
Top 500	Prior only	7	7	8	47	68	58
	OCSI (L)	12	6	10	82	75	67
	OCSI (JI)	10	4	5	79	66	69
	Random $\mathbb{E}(n)$	3.7			52		
Top 1000	Prior only	17	14	14	93	124	113
	OCSI (L)	21	15	15	150	126	139
	OCSI (JI)	14	8	11	156	120	125
	Random $\mathbb{E}(n)$	7.3			105		

References

Calderhead, B. and Girolami, M. A. (2009). Estimating Bayes factors via thermodynamic integration and population MCMC. Comput Stat Data An, **53**(12), 4028–4045.

Cline, M. S., Smoot, M., Cerami, E., Kuchinsky, A., Landys, N., Workman, C., Christmas, R., Avila-Campilo, I., Creech, M., Gross, B., *et al.* (2007). Integration of biological networks and gene expression data using cytoscape. *Nature protocols*, **2**(10), 2366–2382.

Elowitz, M. B. and Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, **403**(6767), 335–338.

Greenfield, A., Madar, A., Ostrer, H., and Bonneau, R. (2010). DREAM4: Combining genetic and dynamic information to identify biological networks and dynamical models. PLoS One, **5**(10), e13397.

Li, F., Long, T., Lu, Y., Ouyang, Q., and Tang, C. (2004). The yeast cell-cycle network is robustly designed. Proc. Natl. Acad. Sci., **101**(14), 4781–4786.

Marbach, D., Schaffter, T., Mattiussi, C., and Floreano, D. (2009). Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of computational biology*, **16**(2), 229–239.

Pancaldi, V., Saraç, Ö. S., Rallis, C., McLean, J. R., Převorovskỳ, M., Gould, K., Beyer, A., and Bähler, J. (2012). Predicting the fission yeast protein interaction network. *G3: Genes— Genomes— Genetics*, **2**(4), 453–467.

Penfold, C., Buchanan-Wollaston, V., Denby, K., and Wild, D. (2012). Nonparametric bayesian inference for perturbed and orthologous gene regulatory networks. Bioinformatics, **28**(12), i233–241.

Penfold, C. A. and Wild, D. L. (2011). How to infer gene networks from expression profiles, revisited. J R Soc Interface Focus, **6**(1), 857–870.

Prill, R. J., Marbach, D., Saez-Rodriguez, J., Sorger, P. K., Alexopoulos, L. G., Xue, X., Clarke, N. D., Altan-Bonnet, G., and Stolovitzky, G. (2010). Towards a rigorous assessment of systems biology models: The DREAM3 challenges. PLoS One, **5**(2), e9202.

Rustici, G., Mata, J., Kivinen, K., Lió, P., Penkett, C., Burns, G., Hayles, J., Brazma, A., Nurse, P., and Bähler, J. (2004). Periodic gene expression program of the fission yeast cell cycle. Nat Genet, **36**(8), 809–17.

Saito, R., Smoot, M. E., Ono, K., Ruscheinski, J., Wang, P.-L., Lotia, S., Pico, A. R., Bader, G. D., and Ideker, T. (2012). A travel guide to cytoscape plugins. *Nature methods*, **9**(11), 1069–1076.

Schaffter, T., Marbach, D., and Floreano, D. (2011). Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, **27**(16), 2263–2270.

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, **13**(11), 2498–2504.

Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. (2011). Weisfeiler-lehman graph kernels. J Mach Learn Res., **12**, 2539–2561.

Smoot, M. E., Ono, K., Ruscheinski, J., Wang, P.-L., and Ideker, T. (2011). Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*, **27**(3), 431–432.

Stark, C., Breitkreutz, B., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. (2006). Biogrid: A general repository for interaction datasets. Nucleic Acids Res, **Database issue**, D535–9.