PLOS | COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Memory functions reveal structural properties of gene regulatory networks

Edgar Herrera-Delgado[1,2], **Ruben Perez-Carrasco**[1,3], **James Briscoe**[1]*, **Peter Sollich**[2]*

1 The Francis Crick Institute, London, United Kingdom, **2** Department of Mathematics, King's College London, Strand, London, United Kingdom, **3** Department of Mathematics, University College London, London, United Kingdom

* james.briscoe@crick.ac.uk (JB); peter.sollich@kcl.ac.uk (PS)

Abstract

Gene regulatory networks (GRNs) control cellular function and decision making during tissue development and homeostasis. Mathematical tools based on dynamical systems theory are often used to model these networks, but the size and complexity of these models mean that their behaviour is not always intuitive and the underlying mechanisms can be difficult to decipher. For this reason, methods that simplify and aid exploration of complex networks are necessary. To this end we develop a broadly applicable form of the Zwanzig-Mori projection. By first converting a thermodynamic state ensemble model of gene regulation into mass action reactions we derive a general method that produces a set of time evolution equations for a subset of components of a network. The influence of the rest of the network, the bulk, is captured by memory functions that describe how the subnetwork reacts to its own past state via components in the bulk. These memory functions provide probes of near-steady state dynamics, revealing information not easily accessible otherwise. We illustrate the method on a simple cross-repressive transcriptional motif to show that memory functions not only simplify the analysis of the subnetwork but also have a natural interpretation. We then apply the approach to a GRN from the vertebrate neural tube, a well characterised developmental transcriptional network composed of four interacting transcription factors. The memory functions reveal the function of specific links within the neural tube network and identify features of the regulatory structure that specifically increase the robustness of the network to initial conditions. Taken together, the study provides evidence that Zwanzig-Mori projections offer powerful and effective tools for simplifying and exploring the behaviour of GRNs.

Author summary

Gene regulatory networks are essential for cell fate specification and function. But the recursive links that comprise these networks often make determining their properties and behaviour complicated. Computational models of these networks can also be difficult to decipher. To reduce the complexity of such models we employ a Zwanzig-Mori projection approach. This allows a system of ordinary differential equations, representing a network,

OPEN ACCESS

Citation: Herrera-Delgado E, Perez-Carrasco R, Briscoe J, Sollich P (2018) Memory functions reveal structural properties of gene regulatory networks. PLoS Comput Biol 14(2): e1006003. https://doi.org/10.1371/journal.pcbi.1006003

Editor: Paul Francois, McGill, CANADA

Received: August 30, 2017

Accepted: January 24, 2018

Published: February 22, 2018

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: EHD and PS acknowledge the stimulating research environment provided by the EPSRC Centre for Doctoral Training in Cross-Disciplinary Approaches to Non-Equilibrium Systems (CANES, EP/L015854/1) (https://www.epsrc.ac.uk/skills/students/centres/profiles/crossdisciplinaryapproachestononequilibriumsystems/). RPC was supported by the Wellcome Trust (WT098325MA) EHD and JB are supported by the Francis Crick Institute which receives its core funding from Cancer Research UK (FC001051) (https://www.cancerresearchuk.org/), UK Medical Research Council (FC001051) (https://www.mrc.ac.uk/), Wellcome Trust (FC001051 and WT098326MA) (https://wellcome.ac.uk/). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

to be reduced to an arbitrary subnetwork consisting of part of the initial network, with the rest of the network (bulk) captured by memory functions. These memory functions account for the bulk by describing signals that return to the subnetwork after some time, having passed through the bulk. We show how this approach can be used to simplify analysis and to probe the behaviour of a gene regulatory network. Applying the method to a transcriptional network in the vertebrate neural tube reveals previously unappreciated properties of the network. By taking advantage of the structure of the memory functions we identify interactions within the network that are unnecessary for sustaining correct patterning. Upon further investigation we find that these interactions are important for conferring robustness to variation in initial conditions. Taken together we demonstrate the validity and applicability of the Zwanzig-Mori projection approach to gene regulatory networks.

Introduction

Biological systems are complex, comprising multiple interacting components. In many cases this complexity makes it difficult to identify underlying mechanisms and to understand the function of a system. Gene regulatory networks (GRNs) are an example of this problem [1]. A GRN comprises the set of interacting genes responsible for the development, differentiation or homoeostasis of a tissue and provides a formal system-level, causative explanation for gene regulation. In physical terms, a GRN consists of modular DNA sequences—cis regulatory elements—that bind to specific sets of transcriptional activators and repressors, which control the expression of associated genes. Some of the regulated genes are themselves transcriptional regulators. Thus, at the core of a GRN is a recursive set of regulatory links that forms a transcriptional network, the dynamics of which is responsible for the spatial and temporal patterns of gene expression.

Attempts have been made to map large transcriptional networks, yet even for relatively small networks, the number of links and the feedback within the system make intuitive understanding difficult to obtain. Various computational models have been developed to address this. Logical models, which describe regulatory interactions qualitatively, provide a flexible and simplifying formalism to explore and understand the behaviour of a network [2]. However, these approaches are unable to capture subtler features of a network that depend on specific aspects of the timing or concentration of components of the network. For this, continuous models based on, for example, ordinary differential equations (ODEs) are often employed [3, 4]. These models describe gene regulation in much greater detail, and there are well-developed mathematical theories that provide powerful tools to distill the dynamical details of such systems. These have been used successfully to gain insight into the operation of GRNs and suggest explanations for otherwise difficult to understand behaviours, including emergent phenomena such as self-organisation, oscillations, spatial patterning, and scaling of pattern size [5–8]. A disadvantage of this approach, however, is that ODE models often require a large number of parameters, including binding affinities, degradation rates, production rates, etc. While it is often possible to numerically analyse large systems, the power of analytical tools is in many cases lost due to lack of knowledge of parameters or the complexity of the network.

Various techniques have been developed to reduce the complexity of models yet preserve specific features of the behaviour of the system. For Boolean networks it is possible to remove specific components while restructuring other parts of the network to conserve the logical structure, but this requires quite specific network topologies [9]. In the case of mass action reactions, one approach to simplification is to generate intermediate species in a systematic way such as to make different systems comparable. This then allows the identification of motifs that can be simplified. This approach comes with the natural limitation that the species involved cannot be chosen *a priori* and must conform to a particular network structure [10]. Other methods to reduce complexity involve removing nodes or perturbing the system and analysing *a posteriori* the effect that this has on the dynamics of the network. This can be helpful to develop intuition, but generally requires the investigation of a large number of combinations of perturbations [11, 12]. A further approach, although applicable only in rather specific cases, is to enforce a timescale separation between network nodes that is chosen not to perturb the dynamics significantly [13]. One can find "morphisms" that relate network structure to function in order to then simplify a set of nodes into potentially simpler motifs; this can reveal an interpretation of how certain motifs work but does not provide a direct conversion from the original network [14]. Less formal methods for replacing well understood network motifs with simpler motifs producing similar behaviour have also been extensively explored [15]. Along similar lines, entire parts of a network can be substituted with effective nodes with more complicated dynamics that produce similar output for a set of chosen species, although the feasiblity of this depends on network structure and the complexity of the original dynamics [16].

We focus in this paper on reducing model complexity by tracking a subnetwork that is embedded within the remainder—or "bulk"—of a larger system. Analysing the behaviour of the subnetwork and its interaction with the bulk can help reveal and rationalise the properties of the entire network. With any coarse graining approach, including those discussed above, a compromise must be reached between the precision of the method, i.e. how well it captures the dynamics of the full network, and the simplicity and interpretability of the resulting description. For example, the introduction of intermediate species can produce model reductions with high precision but these may not be interpretable [10], while morphism approaches provide interpretable insights into the structure but capture the full model dynamics only for specific initial conditions and system parameters [14]. We consider important for interpretability the ability to *choose* a subnetwork, guided by biological relevance or the availability of data; none of the methods described above allow this and instead identify a subnetwork based on their internal criteria.

One class of methods that offers the potential to balance accuracy and interpretability are Zwanzig-Mori projections [17, 18]. Originally developed to allow the extraction of macroscopic equations from a microscopic description of the dynamics—a brief overview can be found in *e.g.* [19]—these methods have since been used to separate a network into an arbitarily chosen subnetwork and bulk in ways that preserve substantial features of the original temporal dynamics [20, 21]. Used in this way, the approach describes the concentration of components in the subnetwork in detail, while the activities of the species in the bulk are replaced with so-called 'memory functions'. These memory functions are derived from the detailed kinetic description of the remainder of the network and summarise how, by acting via the bulk, the past behaviour of the subnetwork influences its current state (Fig 1). The resulting memory functions are functions of time difference, describing the amplitude of a signal returning from the bulk some specified time after the original signal left the subnetwork. Additionally, it is possible to separate each memory function in order to analyse through which bulk species the signal is flowing, thus providing a means to gain a more systematic understanding of a system.

Evolving from its original applications to critical dynamics and supercooled liquids near the glass transition [22], this approach has been applied to biochemical networks [20], where it has been used to analyse the dynamics of signaling in the EGFR (epidermal growth factor receptor) network. It has since been developed further to include also enzymatic Michaelis-

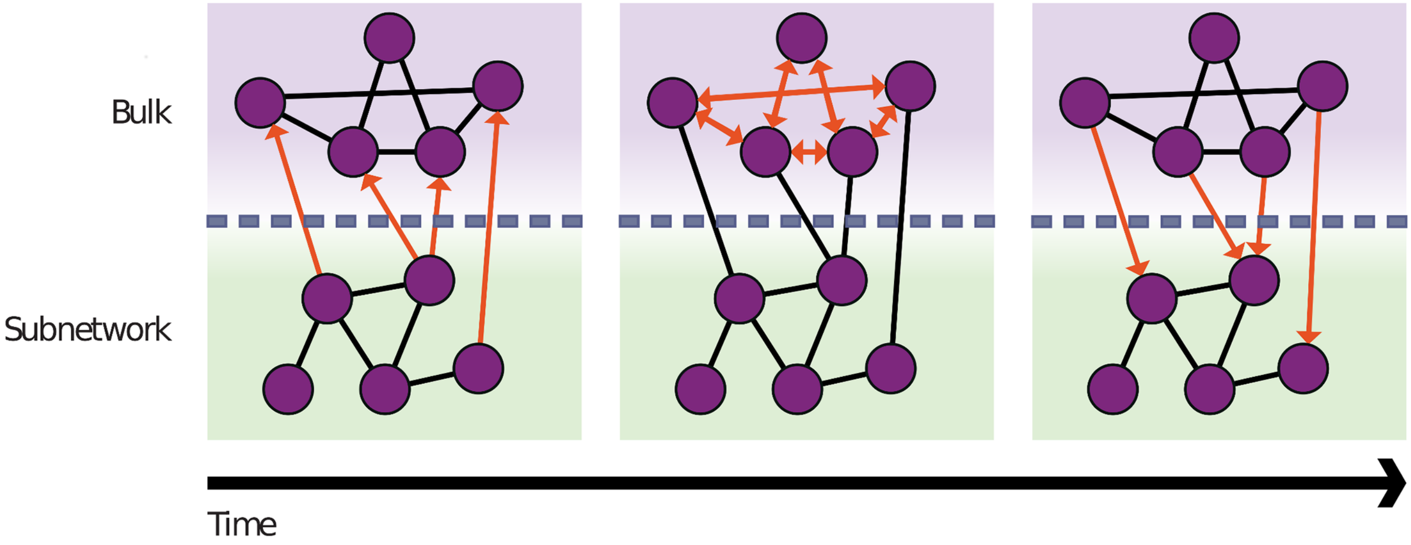


Fig 1. Sketch describing the concept of memory within a reaction network. A network is divided into bulk and subnetwork, *x* axis represents time. Concentration changes in the subnetwork act as signals that leave the subnetwork and travel into the bulk. There they interact with other bulk species and return at a later timepoint via bulk-to-subnetwork interactions. The net effect of such interactions is thus that the subnetwork reacts to its own past. The precise influence of past subnetwork states is governed by *memory functions* that depend on the time difference, *i.e.* on how long ago the relevant signal has left the subnetwork.

https://doi.org/10.1371/journal.pcbi.1006003.g001

Menten reactions [21]. Here we apply the Zwanzig-Mori approach to the analysis of GRNs. We derive a general method that is applicable to any thermodynamic state ensemble representation of protein dynamics including transcriptional networks. These thermodynamic equations mechanistically represent interactions between proteins via transcriptional regulation [23] and can be expanded into mass action reactions to which the Zwanzig-Mori projection formalism can be applied. This approach not only justifies a heuristic method where the thermodynamic equations are expanded to quadratic order around a steady state to obtain a set of effective unary and binary reactions, but also opens up the possibility of developing more advanced projection approximations. We test the method on a cross repressive motif, where we show that we are able to qualitatively reproduce the dynamics of the system and obtain a simple intuitive interpretation of the memory functions. Finally, we use the approach on a GRN from the vertebrate neural tube, a well characterised developmental GRN composed of four interacting transcription factors [24]. This reveals the importance of specific links within the network and identifies features that appear to be present primarily to increase the robustness of the network to initial conditions, rather than maintain the steady states. Taken together, the study provides evidence that Zwanzig-Mori projections are a powerful and efficient method to simplify and explore the behaviour of gene regulatory networks, with memory functions providing new tools for probing the dynamics near steady state.

Methods

Thermodynamic state ensemble models for gene regulation dynamics

Several classes of model have been developed to describe gene regulation. One such class consists of thermodynamic state ensemble models in which equations are constructed that relate interactions between key components of the regulatory mechanism to gene expression [23, 25, 26]. In these models, all possible states of a regulated gene are enumerated. Each state consists of a specific set of transcription factors bound to a DNA cis-regulatory element, weighted by the affinity of interactions. The rate at which a protein is produced is then represented as a ratio of the weight of the subset of states that promote gene expression, to the total weight of all states. We define the vector $\boldsymbol{x}$ such that it contains the protein concentrations for each gene. We can then generically define the time evolution for any protein concentration, indexed by j, as

$$\partial_t x_j = \sum_{\boldsymbol{n}} x_{(j,\boldsymbol{n})} \alpha_{(j,\boldsymbol{n})} - x_j \beta_j \tag{1}$$

$$x_{(j,\boldsymbol{n})} = \frac{w_{(j,n)} \prod_i x_i^{n_i}}{\sum_{n'} w_{(j,n')} \prod_i x_i^{n'_i}} \tag{2}$$

Here β_j is the decay rate of protein j. We use the summation over $\boldsymbol{n}$ to represent all possible binding conformations for the DNA that produces protein j. The term $\alpha_{(j,\boldsymbol{n})}$ represents the production rate of protein j from a particular DNA conformation $\boldsymbol{n}$. The factor $w_{(j,\boldsymbol{n})}$ is the specific affinity for individual proteins to the DNA that produces species x_j. Finally $x_{(j,\boldsymbol{n})}$ is the concentration of DNA producing protein j and in conformation $\boldsymbol{n}$. This is assigned in (2) as the weight of the conformation, dependent on the protein concentrations in a mass-action form, normalised by the total weight of all conformations. This ensures that $x_{(j,\boldsymbol{n})}$ lies in the range [0, 1], so it is not an absolute concentration; any overall DNA concentration scale is therefore to be understood as incorporated into the protein production rates $\alpha_{(j,\boldsymbol{n})}$. Accordingly one can also think of $x_{(j,\boldsymbol{n})}$ as the probability of finding a certain conformation $\boldsymbol{n}$ of the DNA producing protein j. As made explicit in (2), the DNA conformation label $\boldsymbol{n}$ is a collection of integers n_i counting how many copies of protein i are bound to the DNA.

Setup of projection method

The Zwanzig-Mori method in general starts from the choice of a set of observables for which dynamical equations are to be obtained. Initially we select, as the simplest possible observables, the deviations from steady state of the concentrations of the chosen subnetwork species. For these observables the Zwanzig-Mori projection leads to a set of dynamical equations of the form

$$\partial_t \delta x_i = \sum_{j=1}^{N^s} \delta x_j \Omega_{ji} + \int_0^t dt' \left(\sum_{j=1}^{N^s} \delta x_j(t') \underbrace{M_{ji}(t-t')}_{\text{Memory function}} \right) + r_i(t) \tag{3}$$

Here δx_i are the subnetwork concentration deviations as defined explicitly below and N^s indicates the number of subnetwork species. The Ω_{ji} define a *rate matrix* Ω that represents subnetwork interactions. The $r_i(t)$ are so-called *random forces* that arise from the uncertainty about the initial (time $t = 0$) state of the bulk concentrations. (For the linear dynamics they can be expressed in closed form [20] but this is not useful in our context as discussed in Sec. *Nonlinear projected equations* below.)

The key quantities in the eq (3) are the *memory functions* $M_{ji}(\Delta t)$, which describe how past concentration fluctuations $\delta x_j(t')$ influence the current time evolution. However, while the projected equations are formally exact, it is in general impossible to evaluate these memory functions explicitly. One scenario where this *can* be done is a network of unary reactions, where the full reaction equations are linear in the concentrations [20]. This suggests a heuristic approach to calculating memory functions in the current context, which is to linearize the thermodynamic equations. One can think of this as generating a set of effective unary reactions (at least loosely; see Sec. *Justifying the heuristics: Network expansion*). Explicitly, one linearly expands the time evolution eqs (1) with (2) inserted. The expansion is performed around a steady state with concentrations y_i, *i.e.* in terms of the deviations $\delta x_i = x_i - y_i$. The r.h.s. of (1) then contains only linear terms in the δx_i since constant terms cancel out because of the steady state condition. One can therefore write the time evolution equations in matrix form [20]

$$\partial_t \delta x_i = \sum_{j=1}^{N^{\rm s}} \delta x_j L_{ji} \tag{4}$$

for an appropriately defined matrix $\boldsymbol{L}$ with entries L_{ji}.

Projected equations for linearised dynamics

From the matrix $\boldsymbol{L}$ one can calculate the terms in the (linear) projected dynamics of the subnetwork concentrations from (3) as shown in [20]. We summarize the method here. One separates the set of all (linear in concentration) observables into subnetwork (s) and bulk (b), and assumes as above that these concentration deviations δx_i are numbered so that the first $N^{\rm s}$ are the subnetwork observables while the rest are the bulk observables. The matrix $\boldsymbol{L}$ for the linearised dynamics then separates into blocks according to the different observables:

$$\boldsymbol{L} = \left(\begin{array}{c|c} \boldsymbol{L}^{\rm S,S} & \boldsymbol{L}^{\rm S,B} \\ \hline \boldsymbol{L}^{\rm B,S} & \boldsymbol{L}^{\rm B,B} \end{array}\right) \tag{5}$$

where labels S and B represent respectively the set of species in the subnetwork and in the bulk.

We use the labels S and B here to represent subnetwork and bulk species respectively, this will be useful once more subnetwork and bulk observables arise (see Sec. *Mathematical results*). The rate matrix Ω with entries Ω_{ji} in the projected eq (3) is then simply the subnetwork block of $\boldsymbol{L}$,

$$\boldsymbol{\Omega} = \boldsymbol{L}^{\rm S,S} \tag{6}$$

The memory functions similarly form the elements of a memory matrix, which can be expressed in terms of the blocks of $\boldsymbol{L}$ as [20]

$$\boldsymbol{M}(\Delta t) = \boldsymbol{L}^{\rm S,B} e^{\boldsymbol{L}^{\rm B,B}\Delta t} \boldsymbol{L}^{\rm B,S} \tag{7}$$

This form allows straightforward evaluation of the memory functions starting from any given matrix $\boldsymbol{L}$ describing the linearised dynamics.

Nonlinear projected equations

To capture the leading nonlinear corrections to the subnetwork dynamics, it is natural to enlarge the set of subnetwork observables by adding quadratic observables, i.e. products of concentration deviations. The projected equations then become

$$\begin{aligned}\partial_t \delta x_i = {} & \sum_{j=1}^{N^s} \delta x_j \Omega_{ji} + \sum_{1\le j\le k\le N^s}^{N^s} \delta x_j \delta x_k \Omega_{(jk)i} \\ & + \int_0^t dt' \left(\sum_{j=1}^{N^s} \delta x_j(t') \underbrace{M_{ji}(t-t')}_{\text{Linear memory}} + \sum_{1\le j\le k\le N^s}^{N^s} \delta x_j(t')\delta x_k(t') \underbrace{M_{(jk)i}(t-t')}_{\text{Nonlinear memory}} \right) \\ & + r_i(t)\end{aligned} \tag{8}$$

This form is analogous to (3) but now includes the quadratic observables that we have retained, both in the subnetwork interactions (rate matrix terms) and in the memory terms. Once again $r_i(t)$ represents the *random forces* but in this nonlinear case further characterization would require knowledge of the statistics of the initial bulk fluctuations, which is not generally available. We therefore disregard these terms, making (3) a *closed* system of equations for the time evolution of the subnetwork concentrations. Note that neglecting the random forces is equivalent to assuming that the bulk concentrations are at their steady state values at the initial time. Formally these initial bulk concentrations should be random rather than deterministic to make the Zwanzig-Mori projection well defined; we follow the strategy of [20] here and use initial Poisson distributions in the limit of vanishing variance. This limit has already been taken in the results above [20]. How large protein copy numbers have to be to allow stochastic effects to be safely neglected is something that could be studied in future work; the answer will depend, among other things, on the degree of nonlinearity of the GRN time evolution equations [27].

Moving on to the nonlinear memory functions, we are again faced with the difficulty that these cannot in general be evaluated explicitly. However, previous work [20] has developed a systematic approximation technique, which is applicable to the case of reaction networks with unary and binary reactions described by mass action kinetics. In such networks the time evolution equations for linear observables have linear and quadratic terms on the r.h.s. Inserting these into the equations for *quadratic* observables using the product rule gives *e.g.*

$$\partial_t(\delta x_1 \delta x_2) = (\partial_t x_1)\delta x_2 + \delta x_1(\partial_t \delta x_2) \tag{9}$$

and one sees that cubic terms arise. The approximation in [20] neglects these terms in the spirit of an expansion to second order in the changes from steady state δx. These changes are therefore implicitly assumed to be small. Defining a vector $\boldsymbol{z}$ that collects all δx variables and their products such as δx_1^2, $\delta x_1\ \delta x_2$, one can now again write a matrix form of the time evolution equations:

$$\partial_t z_\alpha = \sum_\beta z_\beta L_{\beta\alpha} \tag{10}$$

The rate matrix entries and memory functions can then be calculated [20] from the general formulae (6 and 7), applied to the expanded $\boldsymbol{L}$ matrix constructed as explained above. Because the subnetwork block S now contains linear {s} and quadratic observables {ss}, the matrices have the corresponding block structures. The coefficients $\boldsymbol{\Omega}_{ji}$ are collected in the block $\boldsymbol{\Omega}^{\mathrm{s,s}}$, for example, and the $\Omega_{(jk)i}$ in the block $\boldsymbol{\Omega}^{\mathrm{ss,s}}$ of the overall rate matrix Ω. The memory functions $M_{ji}(\Delta t)$ and $M_{(jk)i}(\Delta t)$ are similarly contained in blocks $\boldsymbol{M}^{\mathrm{s,s}}$ and $\boldsymbol{M}^{\mathrm{ss,s}}$ of the memory matrix $\boldsymbol{M}$.

To apply the above technique in order to derive nonlinear projected subnetwork equations for GRN dynamics, the obvious heuristic route is again to expand the dynamical eq (1), this time to second order in the δx_i, to obtain an effective set of unary and binary reactions. This set defines the matrix $\boldsymbol{L}$, from which the rate matrix and memory functions can then be found as explained above.

Justifying the heuristics: Network expansion

The heuristic method set out above for deriving linear or nonlinear subnetwork equations for GRN dynamics is a useful practical recipe, but its mathematical basis is not obvious. One difficulty is that the thermodynamic GRN eq (1) involve rational functions of the concentrations such as $1/(1 + x_1)$, and while these can be Taylor expanded they have only finite radius of convergence. Not only does the heuristic method then have to throw away an infinite number of higher order terms when truncating the Taylor expansion after linear or quadratic order, but it may implicitly be using the resulting approximation outside the regime where the expansion is even convergent. In addition, while we have loosely talked about the expanded time evolution equations as representing networks of effective reactions, such an interpretation is in no way assured as there is no underlying mechanistic model and hence no constraints that would ensure an appropriate stoichiometry or even positivity of rate constants.

Our mathematical contribution in this paper is to demonstrate that the above limitations can be resolved, and the heuristic approach justified, by initially *expanding* the thermodynamic eqs (1 and 2) into a network of well-defined unary and binary reactions. The way to achieve this is already suggested by (2): we represent each possible existing DNA conformation as an additional species whose concentration has its own dynamical evolution, such that in an appropriate limit of fast DNA binding and unbinding, it reaches the quasi-steady state (QSS) values of the DNA conformation concentrations (2). The prefix “quasi” refers to the fact that this steady state is for given protein concentrations, which themselves change slowly in time. We can then apply the Zwanzig-Mori method to this network to extract memory functions for the chosen subnetwork. In doing this we can justify removing higher order terms as in [20] as the system is composed of only first and second order reactions. Finally the limit of fast DNA binding and unbinding has to be taken. Our result is that this procedure results in exactly the same projected equations as the heuristic approach described above, thus putting the method on a firm mathematical footing. The conclusion applies not only in the context of GRNs but in all networks with the appropriate timescale separation, provided that all fast observables are treated as part of the bulk. More sophisticated projection methods can then be derived by retaining some fast observables (in our case, subnetwork DNA) in the subnetwork. For the case of Michaelis-Menten dynamics, previous work has shown that such an approach allows one to derive projected equations that retain all subnetwork nonlinearities [21]. More generally, by embedding GRN models based on thermodynamic state ensembles into the broad class of mass action reaction networks, our network expansion method opens up the possibility of incorporating stochastic effects and of applying a broad range of other approximation and model reduction techniques [28].

Results

In this section we first provide our mathematical results, where we use a network expansion technique to provide a justification for the heuristic approach to finding memory functions for GRNs that was described in Secs. *Projected equations for linearised dynamics* and *Nonlinear projected equations* above. We then demonstrate applications of the approach with case studies. We first illustrate the memory function method on a simple example system. Finally, the results from the biological application to neural tube patterning are given in Sec. *Application to neural tube network* onwards.

Mathematical results

Network expansion applied to a protein-DNA binding mechanism. Our network expansion approach necessitates defining new reaction rates for binding and unbinding of protein to/from DNA that are consistent with the affinities w from the thermodynamic description. To be explicit, we define $k^{p+}_{(j,\boldsymbol{n})}$ as the rate constant for protein p to bind to DNA coding for protein j and in binding conformation $\boldsymbol{n}$, with a similar meaning for $k^{p-}_{(j,\boldsymbol{n})}$. These rates will thus describe the interaction each DNA species has with the protein species that bind to it. With these rate constants we write down an equation for the time evolution of each DNA concentration $x_{(j,\boldsymbol{n})}$:

$$\partial_t x_{(j,\boldsymbol{n})} = \sum_p \left(k^{p+}_{(j,\boldsymbol{n}-\boldsymbol{e}_p)} x_{(j,\boldsymbol{n}-\boldsymbol{e}_p)} x_p - k^{p+}_{(j,\boldsymbol{n})} x_{(j,\boldsymbol{n})} x_p + k^{p-}_{(j,\boldsymbol{n}+\boldsymbol{e}_p)} x_{(j,\boldsymbol{n}+\boldsymbol{e}_p)} - k^{p-}_{(j,\boldsymbol{n})} x_{(j,\boldsymbol{n})}\right) \tag{11}$$

The first term on the r.h.s. of (11) represents binding of protein p to DNA conformation $(j, \boldsymbol{n} - \boldsymbol{e}_p)$ to make DNA conformation $(j, \boldsymbol{n})$. Here $\boldsymbol{e}_p$ is a unit vector whose i-th entry is δ_{ip}, *i.e.* a 1 at position p and zeros elsewhere. Similarly the third term on the right of (11) describes unbinding of protein p. The other two terms capture how DNA in conformation $(j, \boldsymbol{n})$ can be lost by these two processes.

We next establish the relationship between the rate constants k in the mass action representation (11) and the affinities w of the thermodynamic form. The rate constants need to be chosen to ensure the QSS (2) of the DNA concentrations. The simplest assumption is detailed balance, which requires that in this steady state the net change in DNA concentration from any *individual pair* of binding and unbinding reactions between two conformations vanishes. Detailed balance means, in particular, that where there are multiple reaction paths from one DNA conformation to another, which might be distinguished by the order in which the various proteins bind, each path carries zero reaction flux rather than having a preferred forward or backward reaction direction. Balancing in this way the net change from binding of protein p to DNA in conformation $(j, \boldsymbol{n})$ with the reverse unbinding transition, for the QSS concentrations (2), gives

$$k^{p+}_{(j,\boldsymbol{n})} x_p w_{(j,\boldsymbol{n})} \prod_i x_i^{n_i} = k^{p-}_{(j,\boldsymbol{n}+\boldsymbol{e}_p)} w_{(j,\boldsymbol{n}+\boldsymbol{e}_p)} \prod_i x_i^{n_i+\delta_{ip}} \tag{12}$$

The protein concentrations cancel from this as they should and we are left with a constraint on the rate constants:

$$\frac{k^{p+}_{(j,\boldsymbol{n})}}{k^{p-}_{(j,\boldsymbol{n}+\boldsymbol{e}_p)}} = \frac{w_{(j,\boldsymbol{n}+\boldsymbol{e}_p)}}{w_{(j,\boldsymbol{n})}} \tag{13}$$

To finish our construction, we need to ensure that the DNA concentrations reach the QSS values that we have imposed above with the detailed balance constraint (13). This requires that we make all DNA binding and unbinding reactions fast. Formally we introduce a fast rate factor γ and write all rate constants in the form $k^{p\pm}_{(j,\boldsymbol{n})} = \gamma \tilde{k}^{p\pm}_{(j,\boldsymbol{n})}$, with the understanding that we will take the limit $\gamma \to \infty$ at fixed $\tilde{k}$. This construction is analogous to the one previously used for Michaelis-Menten reactions [21]. As there, we also need to ensure that the changes to protein concentrations from binding to DNA are negligible, as such effects are not described by the original thermodynamic eq (1). (We do, however, include them in the finite-γ numerics of Fig 2B.) This is achieved by scaling all DNA concentrations as $x_{(j,\boldsymbol{n})} = \tilde{x}_{(j,\boldsymbol{n})}/\gamma'$. The intuition for this construction is that the amount of protein bound to DNA is, at most, of the order of the

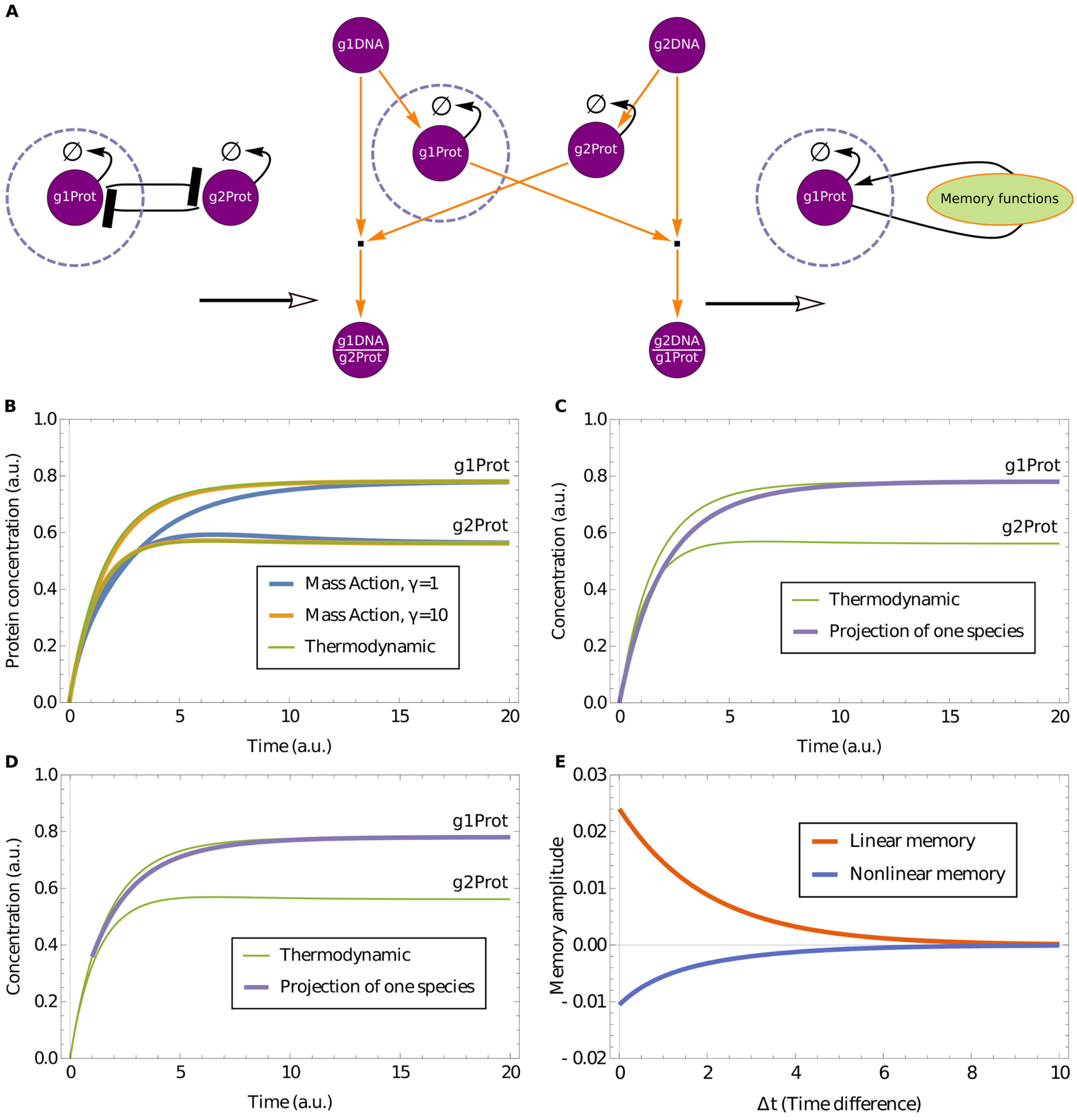


Fig 2. Example of application of Zwanzig-Mori projection. (**A**) Illustration of the methodology, for the example of a cross repressive motif. First the nonlinear thermodynamic reactions (left) are expanded into mass action reactions with an appropriate timescale separation (centre). This generates additional nodes that represent the possible DNA conformations for both proteins, with *e.g.* g1DNA/g2Prot indicating DNA for gene 1 with protein 2 bound to it. To the expanded network we can apply the projection approach, retaining only the concentration of protein 1 in the subnetwork. The effect of the rest of the network—the bulk—is captured via memory terms (right). (**B**) Comparison of the cross repressive motif described via the original thermodynamic equations and the expanded mass action equations with and without timescale separation. Already for a moderate fast rate factor of $\gamma = 10$ the mass action and thermodynamic time evolutions are visually indistinguishable. Two time courses are plotted in Fig. (**B**-**D**), for g1Prot & g2Prot. (**C**-**D**) Demonstration of the projection approach with g1Prot in subnetwork and g2Prot in bulk. The projected equations track the dynamics of the original thermodynamic equation with a reduced system that contains memory functions. However, one can observe in

(**C**) that accuracy can be lost in the transient if the system is initiated too far away from the fixed steady state. (**D**) Starting nearer to the steady state (at $t = 1$) substantially increases the accuracy of the projected description. (**E**) Example of memory functions of protein 1 to itself, decaying with time difference. The linear memory function is positive as expected from the network (positive feedback loop), while the nonlinear term is negative to correct for range-limiting nonlinearities that the linear terms cannot capture. Parameters used are $\alpha_1 = 1$, $\alpha_2 = 1$, $w_{\mathrm{p}1} = 1$, $w_{\mathrm{p}2} = 2$, $w_1 = 2$, $w_2 = 2$, $\beta_1 = 1/2$, $\beta_2 = 1/2$.

https://doi.org/10.1371/journal.pcbi.1006003.g002

total concentration of DNA, which vanishes for $\gamma' \to \infty$. Finally, to compensate for the DNA concentration scaling, protein production rates have to be scaled as $\alpha_{(j,\boldsymbol{n})} = \gamma'\tilde{\alpha}_{(j,\boldsymbol{n})}$.

With the above rescalings, the limit $\gamma \to \infty$ ensures that the "fast" species, *i.e.* the DNA concentrations are always in QSS with respect to the "slow" protein concentrations. The limit $\gamma' \to \infty$, on the other hand, means that the effects of binding to, and unbinding from, DNA can be neglected in the equations for the protein concentrations. These statements can be shown mathematically along the lines of the arguments in [21]. After dropping all tildes on rates and concentrations again—this is the notational convention we adopt for the rest of the paper—the time evolution of the concentrations in our expanded network is given by (1) for the protein equations, while for the DNA species one has

$$\partial_t x_{(j,\boldsymbol{n})} = \gamma \sum_p (k^{p+}_{(j,\boldsymbol{n}-\boldsymbol{e}_p)} x_{(j,\boldsymbol{n}-\boldsymbol{e}_p)} x_p - k^{p+}_{(j,\boldsymbol{n})} x_{(j,\boldsymbol{n})} x_p + k^{p-}_{(j,\boldsymbol{n}+\boldsymbol{e}_p)} x_{(j,\boldsymbol{n}+\boldsymbol{e}_p)} - k^{p-}_{(j,\boldsymbol{n})} x_{(j,\boldsymbol{n})}) \tag{14}$$

Note that in the above argument we do not require a specific relation between γ and γ'. In reality the rate for producing a single protein from the appropriate DNA, which is of order γ', is lower than the DNA binding/unbinding rates, which are $O(\gamma)$. However, quantifying this would be non-trivial as the model described here simplifies many biological steps into each of the two elementary processes of protein production and binding/unbinding. In any event, for our argument to hold we only need both γ and γ' to be large, and for simplicity we take then $\gamma = \gamma'$ as was done in [21].

Obtaining linearised projected equations. We now proceed to use the expanded network to derive linearised projected equations. After the expansion, this requires us to keep track of deviations from steady state of all protein and DNA concentrations. We label these observables by "s" and "b" for subnetwork and bulk proteins as before, and "a" for DNA. Below we will refer to the DNA species in the system as "fast", as expressed mathematically by the factor γ on the r.h.s. of (14), while protein species are "slow". Note that the *rate constants* for protein production are also fast, of order γ', but that the resulting *changes* to protein concentrations are slow because of the low concentration of (protein-producing) DNA.

We construct our $\boldsymbol{L}$ matrix by placing the DNA in the bulk, such that S = {s} and B = {b, a}, giving the block form

$$\boldsymbol{L} = \left(\begin{array}{c|c} \boldsymbol{L}^{\mathrm{S,S}} & \boldsymbol{L}^{\mathrm{S,B}} \\ \hline \boldsymbol{L}^{\mathrm{B,S}} & \boldsymbol{L}^{\mathrm{B,B}} \end{array}\right) = \left(\begin{array}{c|cc} \boldsymbol{L}^{\mathrm{s,s}} & \boldsymbol{L}^{\mathrm{s,b}} & \boldsymbol{L}^{\mathrm{s,a}} \\ \hline \boldsymbol{L}^{\mathrm{b,s}} & \boldsymbol{L}^{\mathrm{b,b}} & \boldsymbol{L}^{\mathrm{b,a}} \\ \boldsymbol{L}^{\mathrm{a,s}} & \boldsymbol{L}^{\mathrm{a,b}} & \boldsymbol{L}^{\mathrm{a,a}} \end{array}\right) \tag{15}$$

The expression (7) for the resulting memory function now explicitly involves the fast rate factor γ and the rate constants $\tilde{k}^{p\pm}_{(j,\boldsymbol{n})}$ used in the construction of our expanded network. These parameters appear in the third column in the second expression for $\boldsymbol{L}$ in (15), which encodes the time evolution equations for the DNA species. Our remaining task is to take the limit $\gamma \to \infty$, with the aim of obtaining an expression for the memory function that only involves parameters of the original thermodynamic eqs (1 and 2). To take this limit, one notes that all blocks in the third column of (15) are proportional to γ. As the memory function contains an exponential of $\boldsymbol{L}$, it will then have contributions decaying for time differences of order $1/\gamma$, arising from the DNA dynamics, as well as slow contributions decaying on timescales of order unity.

For $\gamma \to \infty$ the fast memory function contributions become effectively instantaneous and add to the rate matrix, and only the slow contributions remain in the memory function. This separation into fast and slow pieces of the memory can be performed using the method in [21] because $\boldsymbol{L}$ has exactly the same division into fast and slow blocks as there. It leads to the simple result that the final rate matrix and (slow) memory function for $\gamma \to \infty$ can be calculated from an effective $\boldsymbol{L}$-matrix no longer involving the fast degrees of freedom. To state the expression for the effective $\boldsymbol{L}$, we first isolate the slow piece of our original $\boldsymbol{L}$:

$$\boldsymbol{L}_{\backslash a} = \left(\begin{array}{c|c} \boldsymbol{L}^{\mathrm{s,s}} & 0 \\ \hline 0 & \boldsymbol{L}^{\mathrm{b,b}} \end{array}\right) \tag{16}$$

(The zero blocks here arise from the fact that subnetwork and bulk proteins do not interact directly in our GRN setting, but only via the DNA species.) The effective $\boldsymbol{L}$-matrix that gives us the rate matrix and memory function for $\gamma \to \infty$ is then of the form $\boldsymbol{L}_{\mathrm{eff}} = \boldsymbol{L}_{\backslash a} + \Delta\boldsymbol{L}_{\backslash a}$, which written out reads:

$$\boldsymbol{L}_{\mathrm{eff}} = \left(\begin{array}{c|c} \boldsymbol{L}^{\mathrm{s,s}} & 0 \\ \hline 0 & \boldsymbol{L}^{\mathrm{b,b}} \end{array}\right) - \left(\begin{array}{c|c} \boldsymbol{L}^{\mathrm{s,a}}(\boldsymbol{L}^{\mathrm{a,a}})^{-1}\boldsymbol{L}^{\mathrm{a,s}} & \boldsymbol{L}^{\mathrm{s,a}}(\boldsymbol{L}^{\mathrm{a,a}})^{-1}\boldsymbol{L}^{\mathrm{a,b}} \\ \hline \boldsymbol{L}^{\mathrm{b,a}}(\boldsymbol{L}^{\mathrm{a,a}})^{-1}\boldsymbol{L}^{\mathrm{a,s}} & \boldsymbol{L}^{\mathrm{b,a}}(\boldsymbol{L}^{\mathrm{a,a}})^{-1}\boldsymbol{L}^{\mathrm{a,b}} \end{array}\right) \tag{17}$$

Here the second term including the minus is the additional contribution $\Delta\boldsymbol{L}_{\backslash a}$ from the fast degrees of freedom. The matrix $\boldsymbol{L}_{\mathrm{eff}}$ can be inserted directly into (6 and 7) to obtain the projected equations for linearised dynamics, with the fast rate limit already taken.

Fast binding limit as quasi-steady state elimination. As in [21], the result (17) has a simple interpretation: $\boldsymbol{L}_{\mathrm{eff}}$ can be obtained by eliminating the fast degrees of freedom from the linearised time evolution equations using a QSS condition. Using the block form (5), these equations can be written as

$$\partial_t \boldsymbol{\delta x}^{\mathrm{sT}} = \boldsymbol{\delta x}^{\mathrm{sT}}\boldsymbol{L}^{\mathrm{s,s}} + \boldsymbol{\delta x}^{\mathrm{aT}}\boldsymbol{L}^{\mathrm{a,s}} \tag{18a}$$

$$\partial_t \boldsymbol{\delta x}^{\mathrm{bT}} = \boldsymbol{\delta x}^{\mathrm{bT}}\boldsymbol{L}^{\mathrm{b,b}} + \boldsymbol{\delta x}^{\mathrm{aT}}\boldsymbol{L}^{\mathrm{a,b}} \tag{18b}$$

$$\partial_t \boldsymbol{\delta x}^{\mathrm{aT}} = \boldsymbol{\delta x}^{\mathrm{sT}}\boldsymbol{L}^{\mathrm{s,a}} + \boldsymbol{\delta x}^{\mathrm{bT}}\boldsymbol{L}^{\mathrm{b,a}} + \boldsymbol{\delta x}^{\mathrm{aT}}\boldsymbol{L}^{\mathrm{a,a}} \tag{18c}$$

where $\boldsymbol{\delta x}^{\mathrm{sT}}$, $\boldsymbol{\delta x}^{\mathrm{bT}}$ and $\boldsymbol{\delta x}^{\mathrm{aT}}$ are vectors collecting the concentration deviations of subnetwork proteins, bulk proteins and DNA species, respectively. All blocks of $\boldsymbol{L}$ appearing in (18c) are proportional to γ. This justifies elimination of the fast variables using a QSS condition for these fast (DNA) degrees of freedom, effectively setting the r.h.s. of (18c) to zero. The fast species concentrations are then expressed as:

$$\boldsymbol{\delta x}^{\mathrm{aT}} = -(\boldsymbol{\delta x}^{\mathrm{sT}}\boldsymbol{L}^{\mathrm{s,a}} + \boldsymbol{\delta x}^{\mathrm{bT}}\boldsymbol{L}^{\mathrm{b,a}})(\boldsymbol{L}^{\mathrm{a,a}})^{-1} \tag{19}$$

If we now substitute this back into the time evolution eqs (18a and 18b) we obtain:

$$\begin{aligned} \partial_t \boldsymbol{\delta x}^{\mathrm{sT}} &= \boldsymbol{\delta x}^{\mathrm{sT}}(\boldsymbol{L}^{\mathrm{s,s}} - \underbrace{\boldsymbol{L}^{\mathrm{s,a}}(\boldsymbol{L}^{\mathrm{a,a}})^{-1}\boldsymbol{L}^{\mathrm{a,s}}}_{\Delta \boldsymbol{L}^{\mathrm{s,s}}}) + \boldsymbol{\delta x}^{\mathrm{bT}}(\underbrace{-\boldsymbol{L}^{\mathrm{b,a}}(\boldsymbol{L}^{\mathrm{a,a}})^{-1}\boldsymbol{L}^{\mathrm{a,s}}}_{\Delta \boldsymbol{L}^{\mathrm{b,s}}}) \\ \partial_t \boldsymbol{\delta x}^{\mathrm{bT}} &= \boldsymbol{\delta x}^{\mathrm{sT}}(-\boldsymbol{L}^{\mathrm{s,a}}(\boldsymbol{L}^{\mathrm{a,a}})^{-1}\boldsymbol{L}^{\mathrm{a,b}}) + \boldsymbol{\delta x}^{\mathrm{bT}}(\boldsymbol{L}^{\mathrm{b,b}} - \boldsymbol{L}^{\mathrm{b,a}}(\boldsymbol{L}^{\mathrm{a,a}})^{-1}\boldsymbol{L}^{\mathrm{a,b}}) \end{aligned} \tag{20}$$

Writing these time evolution equations in matrix form gives exactly the effective $\boldsymbol{L}$-matrix

$\boldsymbol{L}_{\text{eff}}$ defined in (17), as claimed. Two terms have been highlighted here by brackets for later comparison.

Equivalence to heuristic linearisation. So far we have shown that the rate matrix and memory functions for the linearised dynamics can be found by applying standard projection results to a set of time evolution equations for only the slow degrees of freedom. This reduced set of equations is obtained by first expanding the thermodynamic equations into a set of mass-action equations, linearising, and finally eliminating the fast variables using a QSS assumption. We next show that this procedure is equivalent to directly linearising the original thermodynamic equations, for any case where there is a timescale separation. This generalises previous work [21] that only considered separate (non-interacting) fast species. To verify the equivalence, it is useful to write the expanded mass-action equations in the generic form

$$\partial_t x_s = R_s(\boldsymbol{x}^{\text{s}}, \boldsymbol{x}^{\text{b}}, \boldsymbol{x}^{\text{a}}), \quad \partial_t x_b = R_b(\boldsymbol{x}^{\text{s}}, \boldsymbol{x}^{\text{b}}, \boldsymbol{x}^{\text{a}}), \quad \gamma^{-1}\partial_t x_a = R_a(\boldsymbol{x}^{\text{s}}, \boldsymbol{x}^{\text{b}}, \boldsymbol{x}^{\text{a}}) \tag{21}$$

where x_s, x_b and x_a are generic components of $\boldsymbol{x}^{\text{s}}$, $\boldsymbol{x}^{\text{b}}$ and $\boldsymbol{x}^{\text{a}}$, respectively. This expanded description is constructed so that for $\gamma \to \infty$, when the x_a can be replaced by their QSS values x_a^*, one recovers the original thermodynamic equations. These can therefore be written as

$$\partial_t x_s = R_s(\boldsymbol{x}^{\text{s}}, \boldsymbol{x}^{\text{b}}, \boldsymbol{x}^{\text{a}*}) \tag{22}$$

$$\partial_t x_b = R_b(\boldsymbol{x}^{\text{s}}, \boldsymbol{x}^{\text{b}}, \boldsymbol{x}^{\text{a}*}) \tag{23}$$

where the dependence of the x_a^* on $\boldsymbol{x}^{\text{s}}$ and $\boldsymbol{x}^{\text{b}}$ is defined implicitly by

$$R_a(\boldsymbol{x}^{\text{s}}, \boldsymbol{x}^{\text{b}}, \boldsymbol{x}^{\text{a}*}) = 0 \tag{24}$$

Expanding now the thermodynamic equations to linear order around a steady state one has

$$\begin{aligned} \partial_t \delta x_s = & \sum_{s'} \frac{\partial R_s}{\partial x_{s'}} \delta x_{s'} + \sum_b \frac{\partial R_s}{\partial x_b} \delta x_b \\ & + \sum_{s'} \underbrace{\sum_a \frac{\partial x_a^*}{\partial x_{s'}} \frac{\partial R_s}{\partial x_a}}_{\Delta \boldsymbol{L}^{\text{s,s}}} \delta x_{s'} + \sum_b \underbrace{\sum_a \frac{\partial x_a^*}{\partial x_b} \frac{\partial R_s}{\partial x_a}}_{\Delta \boldsymbol{L}^{\text{b,s}}} \delta x_b \end{aligned} \tag{25}$$

Here all derivatives are evaluated at the steady state, and the terms in the second line arise from the variation of the QSS values of the fast variables. The required coefficients of the type $\frac{\partial x_a^*}{\partial x_s}$ can be found by differentiating (24) with respect to the relevant slow variable, here x_s:

$$0 = \frac{\partial R_a}{\partial x_s} + \sum_{a'} \frac{\partial R_a}{\partial x_{a'}} \frac{\partial x_{a'}^*}{\partial x_s} \tag{26}$$

We can now show that (25) is identical to (20) derived above. To see this, we note that by linearising the expanded mass action eq (21) one can identify the entries of $\boldsymbol{L}$: for example, $\boldsymbol{L}^{\text{b,s}}$ has entries $\partial R_s/\partial x_b$, $\boldsymbol{L}^{\text{s,a}}$ has entries $\gamma\, \partial R_a/\partial x_s$ etc. The QSS coefficients arising from (26) can then be written in matrix form as $\partial x_a^*/\partial x_s = -(\boldsymbol{L}^{\text{s,a}}(\boldsymbol{L}^{\text{a,a}})^{-1})_{sa}$, with a similar expression for $\partial x_a^*/\partial x_b$. Inserting these expressions into (25) does indeed lead directly to (20), with matching terms indicated by the brackets. An exactly analogous argument demonstrates the equivalence for $\partial_t\, \delta x_b$.

With the above arguments we have justified the heuristic method of Sec. *Projected equations for linearised dynamics* for obtaining linearised projected equations for the dynamics of a subnetwork within a larger GRN written in thermodynamic form: expand the GRN equations to linear order in protein concentrations around a steady state. Then construct from this the matrix $\boldsymbol{L}_{\text{eff}}$ and partition this into "s" and "b" blocks according to the chosen subnetwork–bulk split. Finally determine the rate matrix and memory function matrix from (6 and 7). We invoked an expanded network of binary reactions involving DNA conformations to *derive* that this is the correct method, but note that its *application* only requires the original thermodynamic equations as input. Therefore the resulting projected equations are, as they should be, independent of the details of the rates $\tilde{k}^{p\pm}_{(j,\boldsymbol{n})}$ used in the construction of the expanded network.

Derivation of nonlinear projected equations. To capture more of the nonlinearity inherent in GRNs one can extend the approach described in the previous sections, to include terms that are quadratic in the deviations from a steady state. The memory function then contains fast and slow contributions, with the former being transformed into additional rate matrix terms when $\gamma \to \infty$. The resulting rate matrix and slow memory functions can be obtained from an effective $\boldsymbol{L}$-matrix involving only slow (linear and quadratic) observables. As before, this $\boldsymbol{L}_{\text{eff}}$ can be viewed as arising from a QSS elimination of the fast observables. Finally, one can show that this explicit elimination of the fast observables is equivalent to directly expanding the original thermodynamic equations to *quadratic* order in concentration deviations from the steady state: this justifies the heuristic method given in the Sec. *Nonlinear projected equations*.

Except for the last step, the above chain of argument is analogous to the linearised dynamics case, being based entirely on the identical structure of the partition—compare (5) and (S1 Text, Eq. S.1)—of $\boldsymbol{L}$ into fast and slow blocks. The final reduction to a quadratic expansion of the thermodynamic equations is more subtle because the projection approach treats quadratic observables as distinct and not directly tied to products of linear observables. We defer the details to S1 Text and note only that our derivation there both simplifies and generalises that in [21], from enzyme species that do not interact with each other to arbitrary interactions of the fast degrees of freedom.

Case studies

Zwangzig-Mori projection of a cross-repression motif. We applied the Zwanzig-Mori projection method to a cross repressive motif to illustrate our approach. Fig 2A shows the expansion procedure for this very simple regulation network: each of two genes is repressed through one binding site for the other factor. This system can be expanded to represent all possible DNA conformations, bound and unbound, as shown graphically by the four outer nodes in the network in the centre of Fig 2A. As the arrows indicate, in the expanded network unbound DNA can bind protein to form bound DNA; the reverse process is also possible. In this way one creates a set of reaction equations that has been extended and brought into mass action form by including the concentration of DNA conformations. The equations used for Fig 2 are of the form of (1) & (2):

$$\partial_t[\text{g1Prot}] = \alpha_1(w_1 + w_{\text{p2}}[\text{g2Prot}])^{-1} - \beta_1[\text{g1Prot}] \tag{27}$$

and conversely for g2Prot. In numerical simulations of the expanded network we first checked the fast rate limit, which we control by setting the ratio, γ, of typical DNA reaction rates to protein production rates. Fig 2B shows that already with a moderate value of $\gamma = 10$ the predicted time courses of protein concentrations are almost identical to those predicted by the thermodynamic equations. As $\gamma \to \infty$ the trajectories from the mass action equations and the thermodynamic equations become identical.

We chose to separate the system so that one protein species, g1Prot, is in the subnetwork and the other, g2Prot, is in the bulk. After expanding into protein and DNA species we proceeded to place all DNA species along with g2Prot in the bulk. Because we have performed the expansion around a steady state, the method is more precise when close to the steady state as expected (Fig 2D). Note that the cross-repressive motif with one binding site is not bistable. It could be made bistable by adding binding sites; with the current approach the behaviour around each steady state would then have to be analysed separately, as we also do in the neural tube application in the Sec. *Application to neural tube network*. More general projection methods can overcome this limitation as detailed in the Discussion, although it is clear that there are limits of principle here. For example, where the dynamical choice of steady state depends crucially on the initial concentration of a bulk species, this effect cannot be captured by a subnetwork description that by definition does not track this species.

Next we analyse the memory functions in our cross-repressive example, which can be thought of as describing the strength of a signal returning from the bulk a specific time after the original signal entered the bulk from the subnetwork (Fig 2E). The amplitude of the leading *linear* memory function—where the signal is $\delta x_1(t')$—is positive. This makes sense as, in the cross repressive motif, a protein is inhibiting its inhibitor, thus promoting its own production. The second order memory function governs the effects of the signal $\delta x_1^2(t')$. It is, in this case, negative and acts as a correction that captures nonlinearities beyond the linear memory. We observe that the decay in time of the memory functions is determined by the decay rate of the bulk protein. This is consistent with the memory function describing a protein repressing its own repressor, hence the effect only lasts as long as the repressor protein is present. Overall, the memory functions provide compact and readable descriptions of how the bulk modifies and influences the activity of the subnetwork.

Application to neural tube network. Reducing part of a network into memory functions offers a way to simplify and analyse the effect of the factors in this bulk part on the remaining subnetwork. To illustrate this approach we chose to analyse a four gene network involved in the embryonic patterning of the vertebrate neural tube [29, 30]. The developing neural tube is a well characterised example of developmental pattern formation. In this tissue, the secreted molecule, Sonic-Hedgehog (Shh) forms a ventral to dorsal gradient [31]. This in turn generates discrete domains of gene expression that define the progenitors of the distinct neuronal subtypes that comprise spinal cord circuitry (Fig 3A and 3B). A transcriptional network has been identified that is controlled by graded Shh signaling and is responsible for specifying ventral progenitor domains [30, 32]. Genetic and molecular experiments have determined the activity of the components of this network (Fig 3C) and documented the temporal dynamics of pattern formation in vitro and in vivo [24, 30, 33]. For the three most ventral domains of progenitors, four transcription factors are essential, Nkx2.2, Olig2, Pax6 and Irx3. The most ventral domain (p3 domain) is characterised by high Nkx2.2 and low levels of the other three proteins; the domain adjacent to this (pMN domain) has high levels of Olig2, medium levels of Pax6 and low levels of the other two proteins. Dorsal to this, the p2 domain expresses high levels of Pax6 and Irx3 and low levels of the other two proteins.

Mathematical models have been formulated based on the experimental data and these are able to replicate key aspects of cell patterning [24, 33] (Fig 3D). We take advantage of these to develop a Zwanzig-Mori projection of the system (using equations as described in [24] & S3 Text). To this end, we chose Nkx2.2 and Olig2 to be the subnetwork species, given that they are receiving direct input from Shh (Fig 3C), and replaced Irx3 and Pax6 with memory functions.

Linear memory analysis. We first examined the properties of the linear memory functions. These are the most substantial contributions, close to the specific steady states, of the species in the bulk (Pax6 and Irx3) on the temporal changes in the activity of Nkx2.2 and Olig2. We note that the system is multistable (Fig 3E) and different combinations of steady

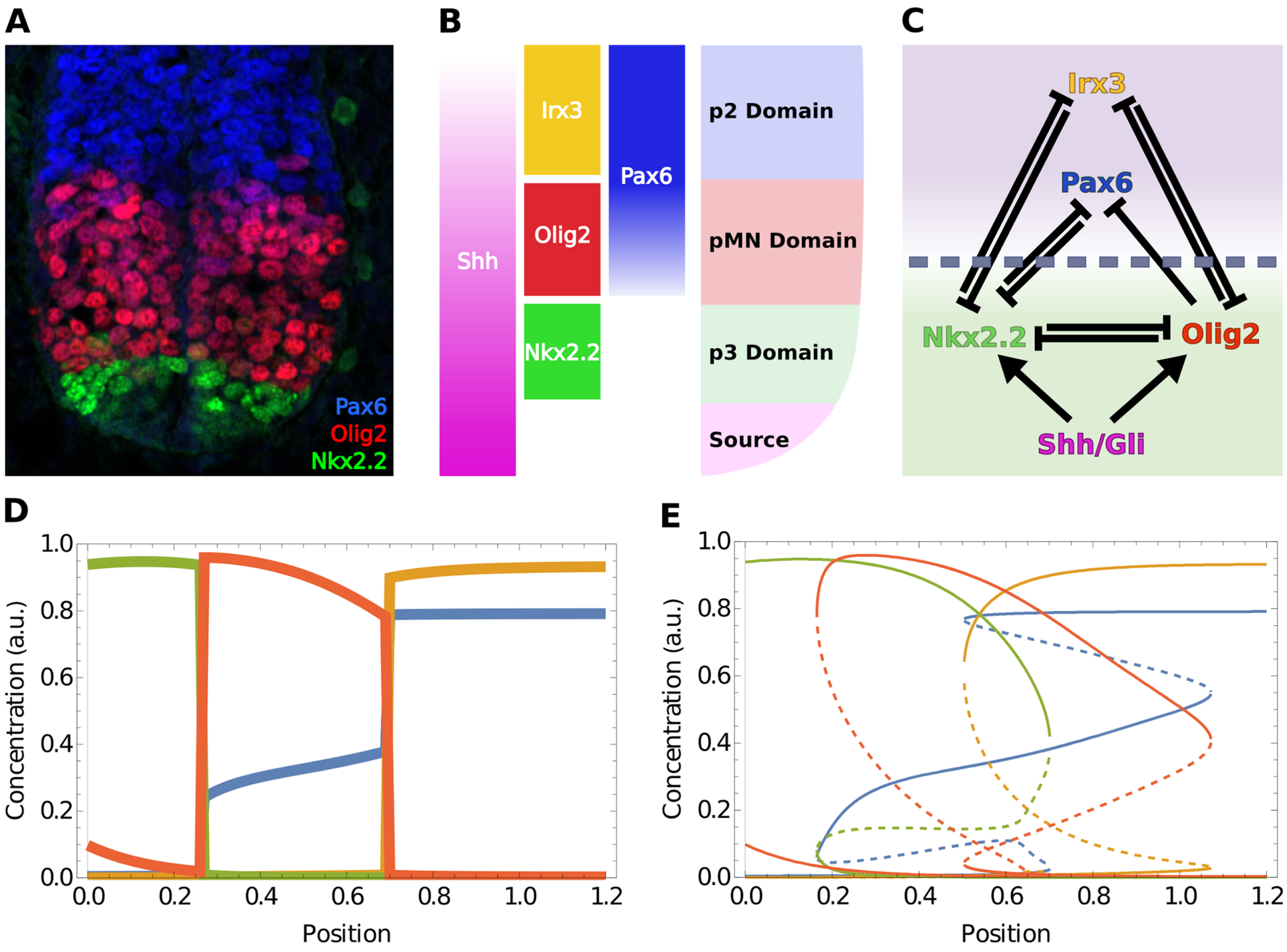


Fig 3. Patterning of the vertebrate neural tube. (**A**) Antibody staining of Wild type (WT) mouse stained for three of the main bands in dorso-ventral patterning. (Image provided by Katherine Exelby). (**B**) Illustration of neural tube patterning: ventral Shh secreted from the notocord and floor plate (termed "Source") generates patterned domains along the dorso-ventral axis. Each domain is defined by the expression of a characteristic set of genes. (**C**) GRN that patterns the three most ventral domains of the neural tube. The chosen separation of bulk (purple) and subnetwork (green) in the application of the Zwanzig-Mori projection is also shown. (**D**) Simulations of steady state pattern along the dorsoventral axis using a set of thermodynamic equations of the form of eqs (1 and 2). These equations were taken—along with appropriate initial conditions of 0 for all species—from [24]. For all plots where *x* axis represents neural tube position, zero corresponds to the most ventral point. (**E**) Full bifurcation diagram illustrating the multistable nature of the network. Shown are steady state concentrations of the four molecular species against neural tube position, with unstable steady states marked dashed. Colours in (**D**, **E**) identify genes/proteins in the same way as in the labelling of the illustration (**B**) and of the network nodes in (**C**). This colour code is used throughout the paper unless otherwise noted.

https://doi.org/10.1371/journal.pcbi.1006003.g003

states are available at different positions along the dorsal ventral axis (Shh gradient). We therefore analysed each possible steady state along the neural tube. In this case, as the bulk species are mutual repressors with the subnetwork species, they form a positive feedback loop with the factors in the subnetwork. As a consequence, each of the linear memory functions is positive and exponentially decaying on the time-scale of protein degradation of the bulk species. Thus the relative contribution of each of the memory functions can be assessed by comparing their amplitude, defined as the memory function value at zero time difference.

We calculated the amplitudes of the memory functions and determined the memory effects that Nkx2.2 and Olig2 receive over time. Furthermore, we used the method developed in [20] to *decompose* the memory into contributions from memory signals passing through the two

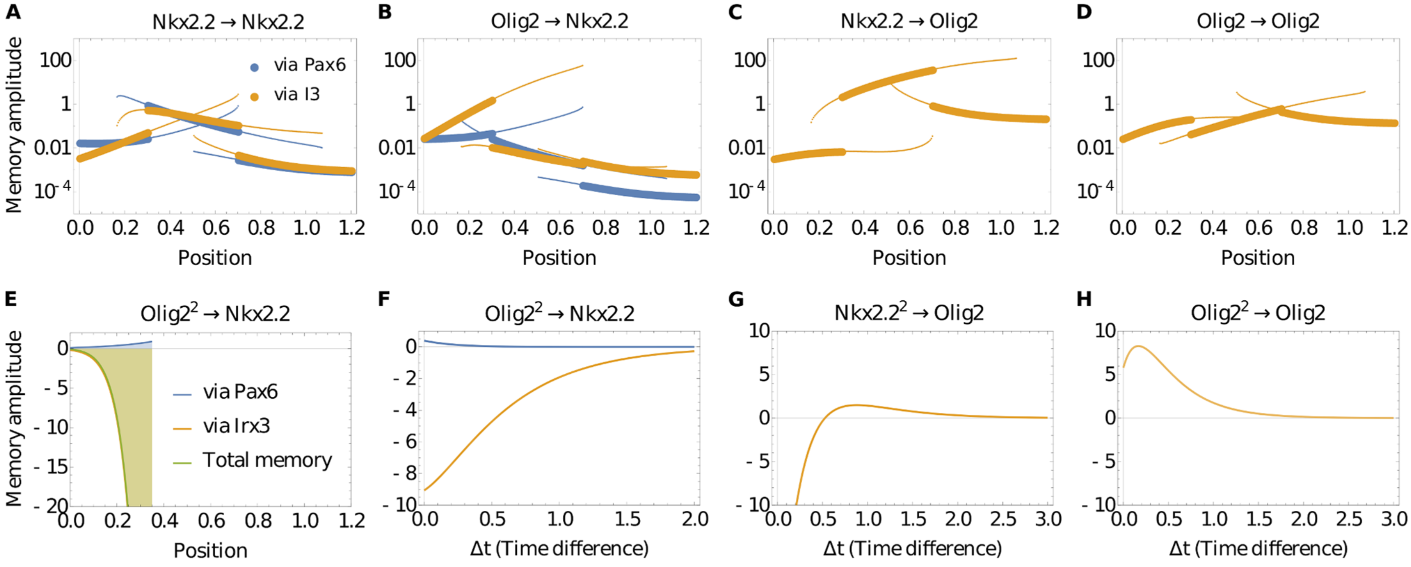


Fig 4. Memory amplitude and temporal dynamics. (**A**) Amplitude of memory (memory function at $\Delta t = 0$) of Nkx2.2 to itself along the neural tube. There are multiple lines as the analysis was performed at all possible stable steady states. The vertical axis is logarithmic to make the range of amplitudes easier to appreciate. Colours identify the memory amplitude contribution from each of the two possible bulk channels, via Irx3 and Pax6, respectively. Thick lines indicate physiological states, while thin lines indicate states that are not usually observed in vivo. (**B**) Linear memory amplitude of (past) Olig2 on Nkx2.2 along the neural tube. The memory via Pax6 is for the most part below the memory via Irx3 in each pair of corresponding curves. (**C,D**) Memory amplitudes of Olig2 to Nkx2.2 (**C**) and to itself (**D**). No channel decomposition is performed as Olig2 receives memory only via the Irx3 channel. (**E**) Nonlinear memory of (past) Olig2 squared on Nkx2.2 in the p3 domain, where dynamics are dominated by Irx3; memory via Pax6 is negligible by comparison. (**F**) Nonlinear memory function of (past) Olig2 squared on Nkx2.2 from position 0.2 in (**E**), plotted to show that the relative contribution of the Pax6 channel is small also for all time differences. (**G,H**) Nonlinear memory functions of Olig2 to Nkx2.2 (**G**) and itself (**H**) in the p2 domain at position 0.7, exemplifying the potential for nontrivial time dependences (including non-monotonicity and sign changes) in the nonlinear memory functions.

https://doi.org/10.1371/journal.pcbi.1006003.g004

different bulk species (Fig 4A–4D). This allowed us to determine the importance of specific regulatory interactions between transcription factors (TFs) at every position of the neural tube model, and assess their respective contributions. The results indicate that, for the most part, there is only a small memory amplitude of Nkx2.2 to the past of Olig2 passing through Pax6 (Fig 4B; note the logarithmic *y*-axis): the memory of (past) Olig2 on Nkx2.2 is largely dominated by memory through Irx3. The only exception is for steady states that are not reached during normal neural tube patterning (lower pair of curves in the neural tube position range ≈ [0.2 . . . 0.45] in Fig 4B). These steady states exist because the system is multistable, but the initial conditions that lead to them are incompatible with physiological conditions. We additionally observe in Fig 4A that the memory of Nkx2.2 to itself is for the most part shared between Irx3 and Pax6, with no particular TF being dominant. There are again exceptions from this, but only in steady states that are biologically unreasonable. In the case of the memory of Olig2, it can only be influenced by Irx3 as is clear from the structure of the network in Fig 3C. Channel decomposition is therefore unnecessary. We observe that the memory effects of Olig2 to itself are strongest in the p3 domain, and the opposite is true (memory of Nkx2.2 on Olig2 is stronger) in the pMN domain.

We observe in Fig 3 that the order of magnitude of the memory amplitudes changes substantially with neural tube position. This is a consequence of the system becoming more or less sensitive to concentration fluctuations of a given species within the network. An example of this is Nkx2.2 memory to itself via Pax6 in the pMN domain (Fig 4A), where dorsally the lower levels of Shh mean that the binding sites of Nkx2.2 are less occupied by the active form of Gli (Gli is the transcriptional effector downstream of Shh that binds to Nkx2.2 & Olig2, see [24] for details on how this is implemented). Active Gli sets the rate of production of Nkx2.2 and thus has a direct effect on the amplitude of the memory functions. In addition to this, as one moves dorsally in the pMN domain, the steady state concentration of Pax6 is increasing (Fig 3E, also observed experimentally). This increase causes the system to become insensitive to fluctuations of Pax6 proteins as Pax6 binding sites are typically already saturated. This phenomenon is at work also in the other memory function amplitudes that change across neural tube position as seen in Fig 4A–4D.

Nonlinear memory analysis. We next examined the second order memory functions. These provide corrections to the linear memory terms and encode information about how the bulk nodes drive the system towards a steady state. Quantitative accuracy in capturing the dynamics of the full nonlinear network will also be improved to the extent possible within an expansion around a steady state, but such accuracy is not our main focus—it is the additional insights gained from the memory functions that we are after.

It is apparent that once memory terms are included, different trajectories in the subnetwork plane *cross*. This is visually most obvious with the full nonlinear memory terms (S1C and S1G Fig) and consistent with the full dynamics (S1D and S1H Fig). The crossings arise from the fact that, in the presence of memory, systems in the same state can follow different trajectories towards a steady state depending on their past. In the full dynamics, the additional information as to which path a system will take is contained in the bulk species, which effectively hold the information on the past of the system. In the case of the memory functions, the memory is not contained in network nodes outside of the subnetwork, but is represented explicitly via the memory terms, with the time difference dependence of the memory functions indicating, for example, the timescale of the memory. This offers a useful abstraction: while it is not easy to visualize the four-dimensional concentration space for the full dynamics, the projection approach reduces the system to two components with memory of their own past.

In the nonlinear memory functions, we can assess the importance of the link from Olig2 to Pax6 by considering the squared deviations of Olig2: the Pax6 channel contribution to this memory necessarily involves signals being propagated via this Olig2→Pax6 link. We find that the Pax6 channel makes a very small contribution to the nonlinear memory amplitude in comparison to the Irx3 channel. This is shown in Fig 4E for the p3 domain but is true also for all other states. We additionally checked the time-dependence of the memory function: from Fig 4F one sees that the relative contribution of the Pax6 channel to the nonlinear memory of (past) Olig2 squared on Nkx2.2 remains small at all time-differences. Together with our results above for linear memory functions, we thus conclude that the repression of Pax6 by Olig2 is dispensable for maintaining an established dorsal-ventral pattern. We plot an example of nonlinear memory functions (Fig 4G and 4H) to illustrate that their dependence on time difference can include non-monotonicities and sign changes that mean memory amplitudes do not tell the full story of the effect of nonlinear memory. The effects of including the memory terms can be confirmed and further visualised by considering the effective drift vector, which contains the rates of change of the concentrations of Nkx2.2 and Olig2. We describe this approach in S2 Text, where we also discuss further how specific contributions of the nonlinear memory affect the dynamics, for example in the approach to the state in the p2 domain.

Exploration of network properties. The analysis of both linear and nonlinear memory functions in the previous section suggested that the repression of Pax6 by Olig2 is not critical for the dynamics of the steady states observed in neural tube patterning. We based this on the observation that the amplitude of memory that is transmitted through this link is relatively small compared to the other memory functions, for all biologically relevant steady states.

To test this prediction of the memory function analysis, we removed the repressive link from Olig2 to Pax6 in the full model. Consistent with our prediction of the relative insignificance of

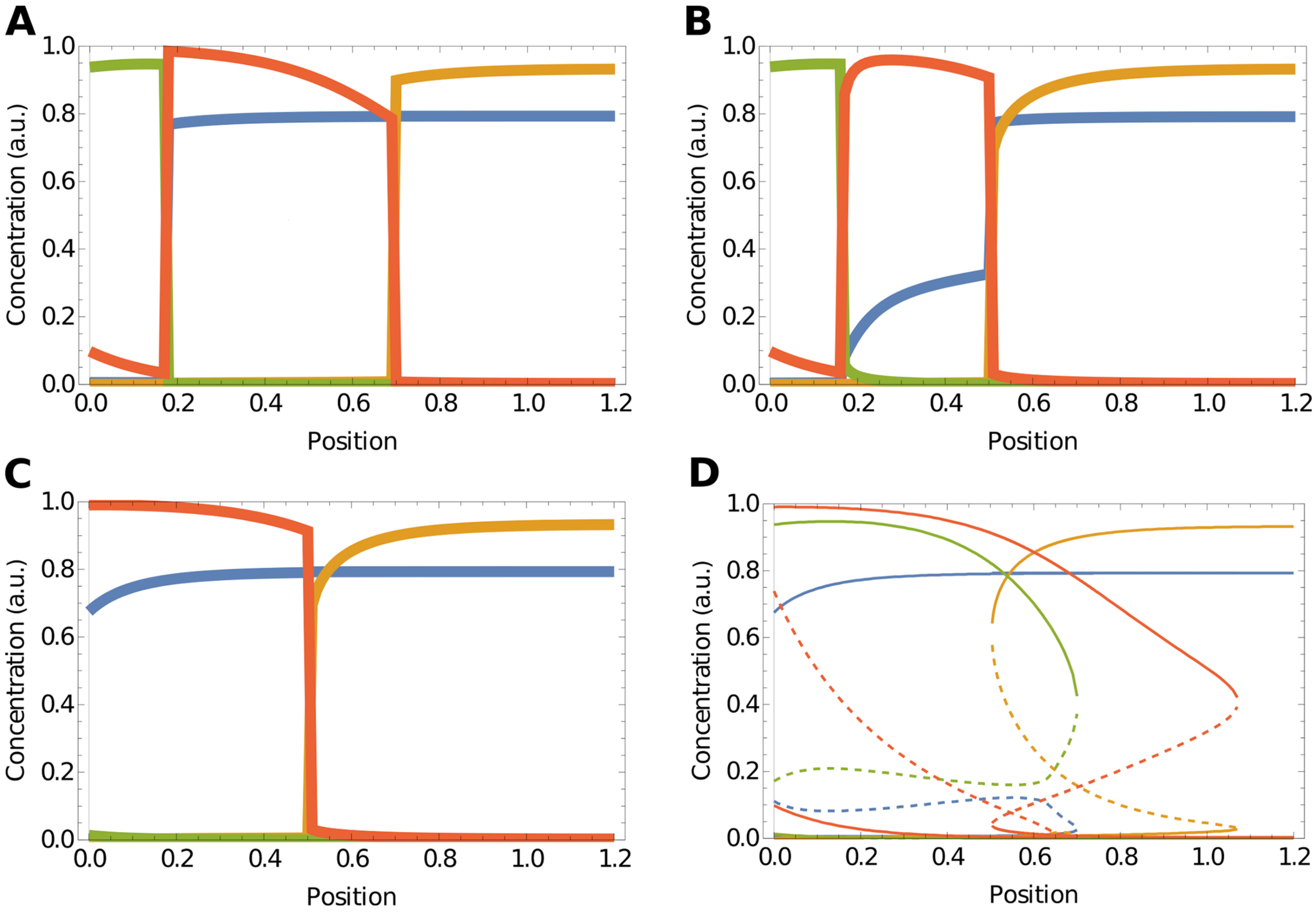


Fig 5. Olig2 repression of Pax6 increases robustness to initial conditions. (**A**) Patterning without the repressive link from Olig2 to Pax6, showing steady states reached from standard initial conditions. Compared to the full network, the qualitative domain structure is conserved. (**B**) Patterning for the full network for initial conditions with high levels of Pax6 and Irx3 is qualitatively identical to low initial levels of Pax6 and Irx3. Initial conditions [Pax6] = [Irx3] = 1. (**C**) Patterning without the repressive link from Olig2 to Pax6 for initial conditions with high levels of Pax6 and Irx3. The p3 domain is lost as a consequence of the different initial conditions. (**D**) Bifurcation diagram of the network without Olig2-Pax6 repression. The region in which the pMN state (Olig2 high) is stable has expanded ventrally, thus making the ventralmost region bistable.

https://doi.org/10.1371/journal.pcbi.1006003.g005

the link from Olig2 to Pax6, the resulting steady state concentrations of the simulations are similar to those in the original system (Fig 5A). The only qualitative change is the increased level of Pax6 between neural tube position 0.2 and 0.7, which results from the loss of the repressive link. But the spatial aspects, such as the positional sequence of genes and qualitative concentrations, of the domains are otherwise unaltered.

Since the repressive link between Olig2 and Pax6 has been experimentally documented [33–35] but from our analysis is not required for the dynamics around the steady state, we sought to understand what purpose it might serve during development. We performed simulations from varying initial conditions for both WT (Fig 5B) and the system lacking the link (Fig 5C). These show that the p3 domain is reached from only a small range of initial conditions following the removal of the repressive link between Olig2 and Pax6. To rationalise this, we performed a bifurcation analysis (Fig 5D). This indicated that the removal of the Olig2 inhibition of Pax6 markedly increased the range of positions at which the Olig2 steady state is present, thus facilitating its invasion into the Nkx2.2 domain. In particular, under initial conditions of high Pax6 and Irx3, the pMN domain is induced ventrally. In the unperturbed network it is not possible to reach this steady state in the ventralmost part because the system is monostable, allowing only the p3 domain to arise. On the other hand, in the absence of Olig2 inhibition of Pax6, the ventral region is bistable. Hence, depending on the initial conditions a pMN fate can be reached (Fig 5D). Taken together, this analysis suggests that the repressive link from Olig2 to Pax6 contributes to the robustness of the system for patterning the neural tube.

A potential explanation for the seemingly dispensable regulatory interaction between Olig2 and Pax6 arises from considering the normal development of the neural tube. During embryonic neural tube development, neural progenitors are generated by the process of neural induction and this initiates the expression of neural genes including Pax6 and Irx3. At the same time, cells of the notochord (which underlies the ventral midline of the neural tube) begin to secrete Shh. Shh spreads into the neural tube and neural progenitors respond to the signal. The consequence is that in the ventral neural tube neural progenitors begin responding to Shh at intermediate levels of Pax6 and Irx3. A more appropriate initial condition, representative of the in vivo situation, is for Pax6 and Irx3 to be somewhere between 0 and maximum. Moreover, heterogeneity between cells, as well as variation in the exact timing of developmental events along the rostral-caudal axis of the embryo, means that the precise initial conditions —the levels of Pax6 and Irx3 when cells respond to Shh signalling—will vary. Hence the robustness provided by the Olig2-Pax6 link may play an important role in ensuring reliable pattern formation in the developing neural tube.

Discussion

We have developed a generally applicable method for applying Zwanzig-Mori projections to transcriptional networks that allows the analysis of subnetworks extracted from a larger network. We demonstrated the approach on a simple genetic cross repressive motif, to illustrate the minimal example of a memory function, and on a more complicated, but well-characterised, transcriptional network operating in the ventral part of the vertebrate neural tube. This showed how the method allows the function and importance of specific links within a network to be defined in an intuitive manner. We used these insights to identify structural features of the neural tube network that appear primarily to increase the robustness of the network to initial conditions, rather than maintain its steady state.

The projection technique is straightforward to implement using the methodology described, and can be applied to any GRN following the thermodynamic formalism (including activation, repression, competitive and cooperative binding). Further protein-protein interactions could readily be incorporated as these would be represented by simple first order (unary) or second order (binary) interactions. Thus, in addition to gene regulation, nonlinear protein-protein interactions mediated by enzymes such as those found in signal transduction pathways could be included in the analysis [21]. It would also be possible to include Hill functions as long as these possess integer exponents [27]. Note that our treatment relies on mass action kinetics, which implicitly assumes that the different molecular species are well mixed by diffusion processes that are faster than any of the reaction kinetics [36]. The size of the matrices involved in constructing the memory functions scales as the square of the number of bulk species, which in our experience means that networks with up to 200 nodes can be investigated without difficulty. Thus, the generality of the method and the ability to implement it algorithmically provide a comprehensive mathematical toolkit to simplify and analyse dynamical systems describing a range of cellular and molecular processes.

In our mathematical treatment we took advantage of an argument of timescale separation for molecular mechanisms operating in transcriptional networks, in which fast processes (DNA-protein interactions) are assumed to be in QSS compared to the slow processes (changes in protein concentration). We also assume the effective concentration of DNA species to be small in comparison to that of protein species. We consider these to be reasonable assumptions because DNA-protein interactions occur at a much faster rate than the changes in protein concentrations produced by transcription and translation [37]. Moreover, there are usually at most four DNA copies of each gene per cell, whereas the number of individual protein molecules is normally considerably higher. The production of individual proteins per DNA copy number, *i.e.* the relevant rate constant, can then be large while the resulting relative changes to protein concentration are slow. These assumptions allowed us to construct an extended reaction network involving fast molecular (DNA) species and at most binary reactions. This construction is sufficiently general for it to be applicable to other systems with a timescale separation, as long as they satisfy the above conditions for the relevant concentrations and rate constants.

An important component of this Zwanzig-Mori projection method is that the memory functions describe all the behaviours of a system in the vicinity of a steady state. Even though this comes at the cost of predictions becoming increasingly inaccurate the further the system is from the fixed point, the information around the steady state is sufficient to distil relevant properties of the network. We therefore employ the method as a tool for probing the dynamics near a chosen steady state, rather than for tracking dynamics. The benefit is that the resulting memory functions are substantially simpler than the original system. There is no need to perform simulations to analyse memory functions or to test different positions in phase space as a steady state is approached. Moreover, the method can also be applied to an arbitrarily chosen subnetwork of a generic network whereas other model reduction approaches—as discussed in the Introduction—do not allow this, and are also more restrictive as they require specific network motifs or functional forms of the dynamics.

The memory functions produced by the Zwanzig-Mori approach allow exploration of the amplitudes and timescales of the interactions between the subnetwork and bulk. This can reveal information that is not otherwise easy to discern from the parameters of the full dynamics or from the functional form of the equations. Each memory function represents the total of the contributions from a given subnetwork component that feeds back to the subnetwork over time after passing through the bulk. This total memory can be decomposed into channels that describe the flow of signals through specific components in the bulk. In this way, the channels that dominate a memory effect can be identified, providing insight into the dynamical mechanisms responsible for achieving and maintaining a steady state. The memory functions thus help to extract features of the dynamics that would not be readily detectable in direct simulations of the time evolution equations.

An example of the type of insight provided by the projection method came from our analysis of the neural tube network. We focused on a subnetwork comprising Nkx2.2 and Olig2. Decomposition of the memory functions suggested that an experimentally documented repression of Pax6 by Olig2 [34] plays only a minor role in sustaining the steady state pattern. Consistent with this prediction, simulations in which this regulatory link had been eliminated in the full model showed qualitatively unchanged steady states. We note that removal of the repressive link between Olig2 and Pax6 is distinct from the removal of Olig2 itself [24, 30, 33]. In the Olig2 mutant, the repressive effect of Olig2 on both Nkx2.2 and Pax6 are eliminated whereas the removal of the Pax6-Olig2 link leaves the repressive effect of Olig2 on Nkx2.2 intact. The memory function analysis thus raised the question of what purpose the Olig2-Pax6 regulation might serve and prompted us to explore the transient system dynamics, specifically the effect of the initial conditions, which are unknown in the *in vivo* system. Simulations in the absence of the Olig2-Pax6 repression, compared to the full model, revealed a marked increase in sensitivity to these initial conditions. Comparison of the bifurcation diagrams of the systems, with and without the Pax6-Olig2 link, indicated that the well-defined Nkx2.2 monostable region (p3 domain)—which appears at high levels of Shh signalling in the full system—was replaced by a region of bistability for Nkx2.2 and Olig2 in the absence of the Olig2-Pax6 regulation. In this latter case the steady state induced by high levels of signal therefore depends on the initial concentrations of Pax6 and Irx3. This suggests that, while not necessary to maintain steady states, the Olig2-Pax6 regulation ensures access to the appropriate steady states irrespective of the initial levels of Pax6 and Irx3. As these may differ between cells and at different locations along the rostral-caudal axis of the neural tube, the Olig2-Pax6 regulatory link might make the system less sensitive to these variations. This implies that the primary purpose of Olig2 repression of Pax6 is to increase the robustness of pattern formation.

Alternative implementations of the Zwanzig-Mori projection [38–40] have been developed that do not rely on expanding around a fixed steady state and we are currently adapting these for transcriptional networks. While these alternative methods may provide a solution to the specific limitations of our current Zwanzig-Mori projection method, the nonlinear memory functions they produce are much richer. Further work will be needed to fully understand the more complex information they encode. Projection methods might also provide a way to analyse systems for which only partial information is available. Given a known subnetwork that is not able to fully reproduce experimental observations, the incorporation of memory functions provides a means to explore the possible effects of unknown factors. This could then be used to identify plausible network structures that generate these memory functions. Taken together therefore, the projection approach provides tools to simplify, visualise and explore the behaviour of large networks that would otherwise be difficult to analyse in their entirety.

Supporting information

S1 Text. Justification of heuristic method for quadratic observables. In this Appendix we give the details of the projection method with linear and quadratic observables, applied to an expanded network as outlined in Sec. *Nonlinear projected equations*.
(PDF)

S2 Text. Further analysis of memory effects. In this Appendix we give additional detail of the wide variety of effects that memory functions contribute to the behaviour of a system. Specifically, we explore the effect of the contributions from nonlinear memory terms and provide an alternative visualisation using a first order time expansion of the effective drift of the system.
(PDF)

S3 Text. Model details. Equations and parameters used in model of neural tube patterning network.
(PDF)

S1 Fig. Trajectories approaching a steady state, with and without memory. Trajectories approaching a high Olig2 state in the pMN domain from a range of initial conditions, with no memory (**A**), with only linear memory (**B**), with linear and nonlinear memory (**C**), and the full dynamics (**D**). Trajectories approaching a low Olig2, low Nkx2.2 state in the p2 domain from different initial conditions, with no memory (**E**), with only linear memory (**F**), with linear and nonlinear memory (**G**), and the full dynamics (**H**). All figures are parametric plots, showing Nkx2.2 and Olig2 concentrations on the *x*- and *y*-axis, respectively, with time as curve parameter. The trajectories are coloured to represent the norm of the drift vector of the system as

indicated by the colour scale; high values indicate the system is evolving quickly while the opposite is true for low values. Scalebars on the right apply to the corresponding row. Path crossing can be seen in (**C**–**D**) & (**G**–**H**), illustrating the importance of nonlinear memory terms to reproduce qualitative features of the full thermodynamic equations.
(TIF)

S2 Fig. Nonlinear memory amplitudes. (**A**-**D**) Nonlinear memory amplitudes across neural tube positions within the pMN domain. The plot titles indicate the type of memory, *e.g.* (**C**) shows memory of (past) Nkx2.2 squared fluctuation on Olig2. Memory effects of past Olig2-fluctuations in the pMN domain are negligible. (**E**-**H**) Nonlinear memory amplitudes across neural tube positions within the p2 domain. Nkx2.2 receives very little quadratic memory influence in this domain. The *x*-axis represents neural tube position in all plots. Blue and yellow lines indicate the decomposition into Pax6 and Irx3 channels, while green lines indicate the total memory.
(TIF)

S3 Fig. Effective drift changes as a consequence of memory terms. Colour map and contour plots of the norm of the effective drift, in conditions identical to those in S1A–S1C Fig for the first row (pMN domain) and S1E–S1G Fig (p2 domain) for the second row. (**A**,**D**) Memoryless drift, (**B**,**E**) drift with linear memory, (**C**,**F**) drift with linear and nonlinear memory. The scalebar on the right applies to each entire row. Effective drift norms have been calculated for time $t = 0.8$.
(TIF)

Acknowledgments

We are grateful to Katherine Exelby for providing microscopy images.

Author Contributions

Conceptualization: Edgar Herrera-Delgado, James Briscoe, Peter Sollich.

Formal analysis: Edgar Herrera-Delgado, Peter Sollich.

Funding acquisition: James Briscoe, Peter Sollich.

Investigation: Edgar Herrera-Delgado.

Methodology: Edgar Herrera-Delgado, Peter Sollich.

Project administration: James Briscoe, Peter Sollich.

Software: Edgar Herrera-Delgado.

Supervision: James Briscoe, Peter Sollich.

Validation: Edgar Herrera-Delgado, Ruben Perez-Carrasco, James Briscoe, Peter Sollich.

Visualization: Edgar Herrera-Delgado, Peter Sollich.

Writing – original draft: Edgar Herrera-Delgado.

Writing – review & editing: Edgar Herrera-Delgado, Ruben Perez-Carrasco, James Briscoe, Peter Sollich.

References

1. Levine M, Davidson EH. Gene regulatory networks for development. Proceedings of the National Academy of Sciences. 2005; 102(14):4936–4942. https://doi.org/10.1073/pnas.0408031102
2. Glass L, Kauffman SA. The logical analysis of continuous, non-linear biochemical control networks. Journal of Theoretical Biology. 1973; 39(1):103–129. https://doi.org/10.1016/0022-5193(73)90208-7 PMID: 4741704
3. Mogilner A, Wollman R, Marshall WF. Quantitative Modeling in Cell Biology: What Is It Good for? Developmental Cell. 2006; 11(3):279–287.
4. Craciun G, Tang Y, Feinberg M. Understanding bistability in complex enzyme-driven reaction networks. Proceedings of the National Academy of Sciences. 2006; 103(23):8697–8702. https://doi.org/10.1073/pnas.0602767103
5. Kondo S, Miura T. Reaction-Diffusion Model as a Framework for Understanding Biological Pattern Formation. Science. 2010; 329(5999):1616–1620. https://doi.org/10.1126/science.1179047 PMID: 20929839
6. Novák B, Tyson JJ. Design principles of biochemical oscillators. Nature Reviews Molecular Cell Biology. 2008; 9(12):981–991. https://doi.org/10.1038/nrm2530 PMID: 18971947
7. Perkins TJ, Jaeger J, Reinitz J, Glass L. Reverse engineering the gap gene network of Drosophila melanogaster. PLoS Computational Biology. 2006; 2(5):417–428. https://doi.org/10.1371/journal.pcbi.0020051
8. Umulis DM, Shimmi O, O'Connor MB, Othmer HG. Organism-Scale Modeling of Early Drosophila Patterning via Bone Morphogenetic Proteins. Developmental Cell. 2010; 18(2):260–274. https://doi.org/10.1016/j.devcel.2010.01.006 PMID: 20159596
9. Naldi A, Remy E, Thieffry D, Chaouiya C. Dynamically consistent reduction of logical regulatory graphs. Theoretical Computer Science. 2011; 412(21):2207–2218. https://doi.org/10.1016/j.tcs.2010.10.021
10. Feliu E, Wiuf C. Simplifying biochemical models with intermediate species. Journal of The Royal Society Interface. 2013; 10(87):20130484–20130484. https://doi.org/10.1098/rsif.2013.0484
11. Okino MS, Mavrovouniotis ML. Simplification of Mathematical Models of Chemical Reaction Systems. Chemical Reviews. 1998; 98(2):391–408. https://doi.org/10.1021/cr950223l PMID: 11848905
12. Gay S, Soliman S, Fages F. A graphical method for reducing and relating models in systems biology. Bioinformatics. 2010; 26(18). https://doi.org/10.1093/bioinformatics/btq388 PMID: 20823324
13. Sunnaker M, Cedersund G, Jirstrand M. A method for zooming of nonlinear models of biochemical systems. BMC systems biology. 2011; 5(1):140. https://doi.org/10.1186/1752-0509-5-140 PMID: 21899762
14. Cardelli L. Morphisms of reaction networks that couple structure to function. BMC Systems Biology. 2014; 8(1):84. https://doi.org/10.1186/1752-0509-8-84 PMID: 25128194
15. Alon U. Network motifs: Theory and experimental approaches. Nature Reviews Genetics. 2007; 8(6):450–461. https://doi.org/10.1038/nrg2102 PMID: 17510665
16. Apri M, de Gee M, Molenaar J. Complexity reduction preserving dynamical behavior of biochemical networks. Journal of Theoretical Biology. 2012; 304:16–26. https://doi.org/10.1016/j.jtbi.2012.03.019 PMID: 22465110
17. Mori H. Transport, Collective Motion, and Brownian Motion. Progress of Theoretical Physics. 1965; 33(3):423–455. https://doi.org/10.1143/PTP.33.423
18. Zwanzig R. Memory effects in irreversible thermodynamics. Physical Review. 1961; 124(4):983–992. https://doi.org/10.1103/PhysRev.124.983
19. Ritort F, Sollich P. Glassy dynamics of kinetically constrained models. Advances in Physics. 2003; 52(4):219–342. https://doi.org/10.1080/0001873031000093582
20. Rubin KJ, Lawler K, Sollich P, Ng T. Memory effects in biochemical networks as the natural counterpart of extrinsic noise. Journal of Theoretical Biology. 2014; 357:245–267. https://doi.org/10.1016/j.jtbi.2014.06.002 PMID: 24928151
21. Rubin KJ, Sollich P. Michaelis-Menten dynamics in protein subnetworks. Journal of Chemical Physics. 2016; 144(17):174114. https://doi.org/10.1063/1.4947478 PMID: 27155632
22. Gotze W, Sjogren L. Relaxation processes in supercooled liquids. Reports on Progress in Physics. 1992; 55(3):241–376. https://doi.org/10.1088/0034-4885/55/3/001
23. Bintu L, Buchler NE, Garcia HG, Gerland U, Hwa T, Kondev J, et al. Transcriptional regulation by the numbers: Models. Current Opinion in Genetics and Development. 2005; 15(2):116–124. https://doi.org/10.1016/j.gde.2005.02.007 PMID: 15797194

24. Cohen M, Page KM, Perez-Carrasco R, Barnes CP, Briscoe J. A theoretical framework for the regulation of Shh morphogen-controlled gene expression. Development. 2014; 141(20):3868–3878. https://doi.org/10.1242/dev.112573 PMID: 25294939

25. Shea MA, Ackers GK. The OR control system of bacteriophage lambda. Journal of Molecular Biology. 1985; 181(2):211–230. https://doi.org/10.1016/0022-2836(85)90086-5 PMID: 3157005

26. Sherman MS, Cohen BA. Thermodynamic state ensemble models of cis-regulation. PLoS Computational Biology. 2012; 8(3):e1002407. https://doi.org/10.1371/journal.pcbi.1002407 PMID: 22479169

27. Thomas P, Straube AV, Grima R. The slow-scale linear noise approximation: an accurate, reduced stochastic description of biochemical networks under timescale separation conditions. BMC systems biology. 2012; 6(ii):39. https://doi.org/10.1186/1752-0509-6-39 PMID: 22583770

28. Schnoerr D, Sanguinetti G, Grima R. Approximation and inference methods for stochastic biochemical kinetics—A tutorial review. Journal of Physics A: Mathematical and Theoretical. 2017; 50(9):093001. https://doi.org/10.1088/1751-8121/aa54d9

29. Dessaud E, McMahon AP, Briscoe J. Pattern formation in the vertebrate neural tube: a sonic hedgehog morphogen-regulated transcriptional network. Development. 2008; 135(15):2489–2503. https://doi.org/10.1242/dev.009324 PMID: 18621990

30. Cohen M, Briscoe J, Blassberg R. Morphogen Interpretation: The Transcriptional Logic Of Neural Tube Patterning. Current Opinion in Genetics and Development. 2013; 23(4):423–428. https://doi.org/10.1016/j.gde.2013.04.003 PMID: 23725799

31. Roelink H, Porter JA, Chiang C, Tanabe Y, Chang DT, Beachy PA, et al. Floor plate and motor neuron induction by different concentrations of the amino-terminal cleavage product of sonic hedgehog autoproteolysis. Cell. 1995; 81(3):445–455. https://doi.org/10.1016/0092-8674(95)90397-6 PMID: 7736596

32. Briscoe J, Pierani A, Jessell TM, Ericson J. A homeodomain protein code specifies progenitor cell identity and neuronal fate in the ventral neural tube. Cell. 2000; 101(4):435–445. https://doi.org/10.1016/S0092-8674(00)80853-3 PMID: 10830170

33. Balaskas N, Ribeiro A, Panovska J, Dessaud E, Sasai N, Page KM, et al. Gene regulatory logic for reading the sonic hedgehog signaling gradient in the vertebrate neural tube. Cell. 2012; 148(1-2):273–284. https://doi.org/10.1016/j.cell.2011.10.047 PMID: 22265416

34. Zhou Q, Anderson DJ. The bHLH transcription factors OLIG2 and OLIG1 couple neuronal and glial subtype specification. Cell. 2002; 109(1):61–73. https://doi.org/10.1016/S0092-8674(02)00677-3 PMID: 11955447

35. Kutejova E, Sasai N, Shah A, Gouti M, Briscoe J. Neural Progenitors Adopt Specific Identities by Directly Repressing All Alternative Progenitor Transcriptional Programs. Developmental Cell. 2016; 36 (6):639–653. https://doi.org/10.1016/j.devcel.2016.02.013 PMID: 26972603

36. Smith S, Grima R. Breakdown of the reaction-diffusion master equation with nonelementary rates. Physical Review E. 2016; 93(5). https://doi.org/10.1103/PhysRevE.93.052135

37. Garcia HG, Sanchez A, Kuhlman T, Kondev J, Phillips R. Transcription by the numbers redux: Experiments and calculations that surprise. Trends in Cell Biology. 2010; 20(12):723–733. https://doi.org/10.1016/j.tcb.2010.07.002 PMID: 20801657

38. Zwanzig, R. Systems Far from Equilibrium. vol. 132 of Lecture Notes in Physics. Garrido L, editor. Berlin, Heidelberg: Springer; 1980. Available from: http://link.springer.com/10.1007/BFb0025609.

39. Chorin AJ, Hald OH, Kupferman R. Optimal prediction and the Mori-Zwanzig representation of irreversible processes. Proceedings of the National Academy of Sciences. 2000; 97(7):2968–2973. https://doi.org/10.1073/pnas.97.7.2968

40. Zwanzig R. Nonequilibrium Statistical Mechanics. Oxford University Press; 2001. Available from: https://books.google.co.uk/books?id=4cI5136OdoMC.

S1 Appendix: Justification of heuristic method for quadratic observables

In this Appendix we give the details of the projection method with linear and quadratic observables, applied to an expanded network as outlined in Sec. *Nonlinear projected equations.* The expanded network is one of unary and binary reactions involving the concentrations of subnetwork proteins, $\boldsymbol{x}^{\mathrm{s}}$, of bulk proteins, $\boldsymbol{x}^{\mathrm{b}}$, and of DNA species, $\boldsymbol{x}^{\mathrm{a}}$. The mass action kinetics for this network can be put into the form of an $\boldsymbol{L}$-matrix, defined for linear and quadratic observables as in (10). We partition this into blocks according to

$$\boldsymbol{L} = \left(\begin{array}{c|c} \boldsymbol{L}^{\mathrm{S,S}} & \boldsymbol{L}^{\mathrm{S,B}} \\ \hline \boldsymbol{L}^{\mathrm{B,S}} & \boldsymbol{L}^{\mathrm{B,B}} \end{array}\right) = \left(\begin{array}{c|cc} \boldsymbol{L}^{\tilde{\mathrm{s}},\tilde{\mathrm{s}}} & \boldsymbol{L}^{\tilde{\mathrm{s}},\tilde{\mathrm{b}}} & \boldsymbol{L}^{\tilde{\mathrm{s}},\tilde{\mathrm{a}}} \\ \hline \boldsymbol{L}^{\tilde{\mathrm{b}},\tilde{\mathrm{s}}} & \boldsymbol{L}^{\tilde{\mathrm{b}},\tilde{\mathrm{b}}} & \boldsymbol{L}^{\tilde{\mathrm{b}},\tilde{\mathrm{a}}} \\ \boldsymbol{L}^{\tilde{\mathrm{a}},\tilde{\mathrm{s}}} & \boldsymbol{L}^{\tilde{\mathrm{a}},\tilde{\mathrm{b}}} & \boldsymbol{L}^{\tilde{\mathrm{a}},\tilde{\mathrm{a}}} \end{array}\right) \tag{S1.1}$$

Here $\tilde{\mathrm{s}}$ contains the "subnetwork only" observables {s} (linear) and {ss} (quadratic, like $\delta x_s \delta x_{s'}$), while $\{\tilde{\mathrm{b}}\}$ collects the slow bulk observables {b, sb, bb}. The fast bulk observables are gathered in $\{\tilde{\mathrm{a}}\}$, which contains {a, sa, ba, aa}. Note that with this partitioning of observables we have allocated all fast (DNA) species to the bulk. This is different from the approach in [21] where some fast (enzyme) species were retained in the subnetwork in order to retain more of the nonlinearities. In our case one could similarly keep in the subnetwork those DNA species that produce subnetwork proteins, but it turns out that this makes the final elimination of fast variables rather intricate and so we leave this as an option to pursue in future work.

For our GRN equations, subnetwork and bulk protein species do not interact, so the blocks $\boldsymbol{L}^{\tilde{\mathrm{s}},\tilde{\mathrm{b}}}$ and $\boldsymbol{L}^{\tilde{\mathrm{b}},\tilde{\mathrm{s}}}$ are in fact zero. This restriction is not required for our treatment, however, and direct protein-protein interactions could be included in the formalism without modification. As in the case of the linearised dynamics (Sec. *Equivalence to heuristic linearisation*), only the third column of (S1.1) is fast, *i.e.* has entries proportional to γ (plus subleading terms of order unity arising from the time derivatives of slow-fast product observables such as ba).

From (S1.1) one can obtain the rate matrix and memory functions for the projected subnetwork equations, for any finite γ. The limiting values of these quantities for $\gamma \to \infty$ can then be found from a matrix $\boldsymbol{L}_{\mathrm{eff}}$ for only the slow (protein) observables. Our aim in this Appendix is to show that this $\boldsymbol{L}_{\mathrm{eff}}$ is identical to the analogous matrix that one obtains by directly expanding the original slow (GRN) equations to second order in the protein concentrations. This then justifies the heuristic method of constructing the nonlinear rate matrix and memory functions described in Sec. *Nonlinear projected equations.*

S1.1 Generic form of notation

It will be useful to write the full time evolution equations for the expanded network in the generic form

$$\partial_t \boldsymbol{x}^{\mathrm{l}} = \boldsymbol{R}^{\mathrm{l}}(\boldsymbol{x}^{\mathrm{l}}, \boldsymbol{x}^{\mathrm{a}}) \tag{S1.2}$$

$$\gamma^{-1}\partial_t \boldsymbol{x}^{\mathrm{a}} = \boldsymbol{R}^{\mathrm{a}}(\boldsymbol{x}^{\mathrm{l}}, \boldsymbol{x}^{\mathrm{a}}) \tag{S1.3}$$

These are equations (1, 14) from the main text, or more generically (21), but we have grouped together the subnetwork and bulk concentration vectors $\boldsymbol{x}^{\mathrm{s}}$ and $\boldsymbol{x}^{\mathrm{b}}$ into a single vector of slow variables $\boldsymbol{x}^{\mathrm{l}}$ to keep the notation for the following discussion compact. The vector $\boldsymbol{x}^{\mathrm{a}}$ contains the fast variables, which in the GRN context are concentrations of DNA conformations, while γ is a fast rate parameter as before. In the limit of large γ, the fast variables are always in QSS with the slow ones so that the expanded network reduces to the thermodynamic state ensemble (in the GRN case) dynamics

$$\partial_t \boldsymbol{x}^{\mathrm{l}} = \boldsymbol{R}^{\mathrm{l}}(\boldsymbol{x}^{\mathrm{l}}, \boldsymbol{x}^{\mathrm{a}}) \tag{S1.4}$$

$$0 = \boldsymbol{R}^{\mathrm{a}}(\boldsymbol{x}^{\mathrm{l}}, \boldsymbol{x}^{\mathrm{a}}) \tag{S1.5}$$

where the second equation implicitly determines $\boldsymbol{x}^{\mathrm{a}}$ as a function of $\boldsymbol{x}^{\mathrm{l}}$. (In the main text we marked this QSS value by an asterisk; we omit this here for notational simplicity.) This is the generic form of equations (1,2) in the main text.

The matrix $\boldsymbol{L}$ is obtained in the above generic setting by expanding around a fixed point to second order to write the equations of motion as

$$\begin{aligned}\partial_t \boldsymbol{x}^{\mathrm{l}\mathrm{T}} &= \boldsymbol{x}^{\mathrm{l}\mathrm{T}}\boldsymbol{L}^{\mathrm{ll}} + \boldsymbol{x}^{\mathrm{a}\mathrm{T}}\boldsymbol{L}^{\mathrm{al}} + \boldsymbol{x}^{\mathrm{ll}\mathrm{T}}\boldsymbol{L}^{\mathrm{ll,l}} + \boldsymbol{x}^{\mathrm{la}\mathrm{T}}\boldsymbol{L}^{\mathrm{la,l}} + \boldsymbol{x}^{\mathrm{aa}\mathrm{T}}\boldsymbol{L}^{\mathrm{aa,l}} && \text{(S1.6)}\\ \gamma^{-1}\partial_t \boldsymbol{x}^{\mathrm{a}\mathrm{T}} &= \boldsymbol{x}^{\mathrm{l}\mathrm{T}}\boldsymbol{L}^{\mathrm{la}} + \boldsymbol{x}^{\mathrm{a}\mathrm{T}}\boldsymbol{L}^{\mathrm{aa}} + \boldsymbol{x}^{\mathrm{ll}\mathrm{T}}\boldsymbol{L}^{\mathrm{ll,a}} + \boldsymbol{x}^{\mathrm{la}\mathrm{T}}\boldsymbol{L}^{\mathrm{la,a}} + \boldsymbol{x}^{\mathrm{aa}\mathrm{T}}\boldsymbol{L}^{\mathrm{aa,a}} && \text{(S1.7)}\end{aligned}$$

All $\boldsymbol{x}$ appearing here and below are deviations $\delta\boldsymbol{x}$ from steady state; we drop the δ to lighten the notation. The $\boldsymbol{x}^{\mathrm{ll}}$ etc are product variables – we assume the indices are ordered to avoid duplicate observables – and the $\boldsymbol{L}$ matrices contain the appropriate derivatives of the "drift" functions R at the fixed point, *e.g.* $\boldsymbol{L}^{\mathrm{ll,a}}$ has elements $L_{ll',a} = \partial_{x_l}\partial_{x_{l'}}R_a$ for $l < l'$ and $L_{ll,a} = (1/2)\partial_{x_l}\partial_{x_l}R_a$ when the two indices are equal. From the above equations then follow the evolution equations for the slow product variables $\boldsymbol{x}^{\mathrm{ll}}$ and the fast products $\boldsymbol{x}^{\mathrm{la}}$ and $\boldsymbol{x}^{\mathrm{aa}}$; see (S1.13, S1.18) below. From the product rule these equations only involve product variables on the r.h.s.; third order terms are in principle present but discarded within the second order expansion. Collecting all variables into a vector $\boldsymbol{z}$ that concatenates $\boldsymbol{x}^{\mathrm{l}}$, $\boldsymbol{x}^{\mathrm{ll}}$, $\boldsymbol{x}^{\mathrm{a}}$, $\boldsymbol{x}^{\mathrm{la}}$, $\boldsymbol{x}^{\mathrm{aa}}$ gives the time evolution equation in the form $\partial_t \boldsymbol{z}^{\mathrm{T}} = \boldsymbol{z}^{\mathrm{T}}\boldsymbol{L}$, where the matrix $\boldsymbol{L}$ has the block form (S1.1) if one restores the split of slow observables into subnetwork and bulk.

S1.2 Heuristic method

The heuristic method involves a direct expansion of the slow equations. To obtain the general form of this, one writes the dynamical equations as

$$\begin{aligned}\partial_t \boldsymbol{x}^{\mathrm{l}\mathrm{T}} &= \boldsymbol{x}^{\mathrm{l}\mathrm{T}}\boldsymbol{L}^{\mathrm{ll}} + \boldsymbol{x}^{\mathrm{a}\mathrm{T}}\boldsymbol{L}^{\mathrm{al}} + (\boldsymbol{x}^{\mathrm{l}}\circ\boldsymbol{x}^{\mathrm{l}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{ll,l}} + (\boldsymbol{x}^{\mathrm{l}}\circ\boldsymbol{x}^{\mathrm{a}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{la,l}} + (\boldsymbol{x}^{\mathrm{a}}\circ\boldsymbol{x}^{\mathrm{a}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{aa,l}} && \text{(S1.8)}\\ \gamma^{-1}\partial_t \boldsymbol{x}^{\mathrm{a}\mathrm{T}} &= \boldsymbol{x}^{\mathrm{l}\mathrm{T}}\boldsymbol{L}^{\mathrm{la}} + \boldsymbol{x}^{\mathrm{a}\mathrm{T}}\boldsymbol{L}^{\mathrm{aa}} + (\boldsymbol{x}^{\mathrm{l}}\circ\boldsymbol{x}^{\mathrm{l}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{ll,a}} + (\boldsymbol{x}^{\mathrm{l}}\circ\boldsymbol{x}^{\mathrm{a}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{la,a}} + (\boldsymbol{x}^{\mathrm{a}}\circ\boldsymbol{x}^{\mathrm{a}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{aa,a}} && \text{(S1.9)}\end{aligned}$$

where the "circle" product indicates the actual products of the regular linear observables, with the same index ordering as in the projection approach (so that all the $\boldsymbol{L}$-matrices are as before). One now needs to determine $\boldsymbol{x}^{\mathrm{a}}$ by setting the r.h.s. of (S1.9) to zero, and substitute the result into (S1.8). As we are only expanding to second order in $\boldsymbol{x}^{\mathrm{l}}$, it is enough also to obtain $\boldsymbol{x}^{\mathrm{a}}$ to this order. Starting with the first order of (S1.9) one obtains

$$\boldsymbol{x}^{\mathrm{a}\mathrm{T}} = -\boldsymbol{x}^{\mathrm{l}\mathrm{T}}\boldsymbol{L}^{\mathrm{la}}(\boldsymbol{L}^{\mathrm{aa}})^{-1} \equiv \boldsymbol{x}_0^{\mathrm{a}\mathrm{T}} \qquad \text{(S1.10)}$$

where $\boldsymbol{x}_0^{\mathrm{a}}$ will be a convenient shorthand. (Note that the coefficient matrix $-\boldsymbol{L}^{\mathrm{la}}(\boldsymbol{L}^{\mathrm{aa}})^{-1}$ is the one we worked out in the main text below (26), in a slightly more pedestrian fashion.) All *second* order terms in (S1.9) can now be evaluated to the required accuracy by replacing $\boldsymbol{x}^{\mathrm{a}}$ by $\boldsymbol{x}_0^{\mathrm{a}}$. Solving for $\boldsymbol{x}^{\mathrm{a}}$ then gives

$$\boldsymbol{x}^{\mathrm{a}\mathrm{T}} = \boldsymbol{x}_0^{\mathrm{a}\mathrm{T}} - \left[(\boldsymbol{x}^{\mathrm{l}}\circ\boldsymbol{x}^{\mathrm{l}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{ll,a}} + (\boldsymbol{x}^{\mathrm{l}}\circ\boldsymbol{x}_0^{\mathrm{a}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{la,a}} + (\boldsymbol{x}_0^{\mathrm{a}}\circ\boldsymbol{x}_0^{\mathrm{a}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{aa,a}}\right](\boldsymbol{L}^{\mathrm{aa}})^{-1} \qquad \text{(S1.11)}$$

Inserting into (S1.8) and dropping terms of higher than quadratic order gives the required expansion of the slow equations,

$$\begin{aligned}\partial_t \boldsymbol{x}^{\mathrm{l}\mathrm{T}} = {} & \boldsymbol{x}^{\mathrm{l}\mathrm{T}}\boldsymbol{L}^{\mathrm{ll}} + \boldsymbol{x}_0^{\mathrm{a}\mathrm{T}}\boldsymbol{L}^{\mathrm{al}} + (\boldsymbol{x}^{\mathrm{l}}\circ\boldsymbol{x}^{\mathrm{l}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{ll,l}} + (\boldsymbol{x}^{\mathrm{l}}\circ\boldsymbol{x}_0^{\mathrm{a}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{la,l}} + (\boldsymbol{x}_0^{\mathrm{a}}\circ\boldsymbol{x}_0^{\mathrm{a}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{aa,l}}\\ & - \left[(\boldsymbol{x}^{\mathrm{l}}\circ\boldsymbol{x}^{\mathrm{l}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{ll,a}} + (\boldsymbol{x}^{\mathrm{l}}\circ\boldsymbol{x}_0^{\mathrm{a}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{la,a}} + (\boldsymbol{x}_0^{\mathrm{a}}\circ\boldsymbol{x}_0^{\mathrm{a}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{aa,a}}\right](\boldsymbol{L}^{\mathrm{aa}})^{-1}\boldsymbol{L}^{\mathrm{al}} \qquad \text{(S1.12)}\end{aligned}$$

This then determines the $\boldsymbol{L}$-matrix for the slow variables from which rate matrix and memory functions are determined in the heuristic approach.

S1.3 Expanded network approach

As explained in the main text based on the reasoning in [21], if one writes down expressions for the rate matrix and memory functions from the full matrix $\boldsymbol{L}$ and then takes the fast rate limit $\gamma \to \infty$, the resulting rate matrix and (slow) memory function can be found from a matrix $\boldsymbol{L}_{\mathrm{eff}}$ describing only the dynamics of the slow variables $\boldsymbol{x}^{\mathrm{l}}$ and $\boldsymbol{x}^{\mathrm{ll}}$. This $\boldsymbol{L}_{\mathrm{eff}}$ is obtained by eliminating the fast variables $\boldsymbol{x}^{\mathrm{a}}$, $\boldsymbol{x}^{\mathrm{la}}$ and $\boldsymbol{x}^{\mathrm{aa}}$ using the condition that they are in QSS. These conditions are, within the projection approach, *linear* equations because product observables are treated as independent from linear observables.

What we then need to demonstrate is that this distinct elimination assigns to $\boldsymbol{x}^{\mathrm{la}}$ and $\boldsymbol{x}^{\mathrm{aa}}$ the same values as the direct expansion approach, namely $(\boldsymbol{x}^{\mathrm{l}}\circ\boldsymbol{x}_0^{\mathrm{a}})$ and $(\boldsymbol{x}_0^{\mathrm{a}}\circ\boldsymbol{x}_0^{\mathrm{a}})$. Once this is shown, it follows that the linear fast variables $\boldsymbol{x}^{\mathrm{a}}$ are eliminated in the same way in the two approaches, because the same quadratic fast variables are substituted into the relevant equations (S1.7,S1.9). Thus *all* fast variables are eliminated in the same way from the time evolution equation for the slow variables, to the quadratic order we consider here. As the slow $\boldsymbol{L}$-matrix in the heuristic approach and the $\boldsymbol{L}_{\mathrm{eff}}$ in the expanded networm method are both obtained from this time evolution equation for the slow variables, they are therefore equal as we want to show.

S1.3.1 Elimination of $\boldsymbol{x}^{\mathrm{la}}$

Using the product rule of differentiation, the equations of motion for the $\boldsymbol{x}^{\mathrm{la}}$ are

$$\partial_t \boldsymbol{x}^{\mathrm{laT}} = \gamma((\boldsymbol{x}^{\mathrm{lT}} \circ \boldsymbol{x}^{\mathrm{lT}} \boldsymbol{L}^{\mathrm{la}})) + \gamma((\boldsymbol{x}^{\mathrm{lT}} \circ \boldsymbol{x}^{\mathrm{aT}} \boldsymbol{L}^{\mathrm{aa}})) + ((\boldsymbol{x}^{\mathrm{lT}} \boldsymbol{L}^{\mathrm{ll}} \circ \boldsymbol{x}^{\mathrm{aT}})) + ((\boldsymbol{x}^{\mathrm{aT}} \boldsymbol{L}^{\mathrm{al}} \circ \boldsymbol{x}^{\mathrm{aT}})) \tag{S1.13}$$

Here the double brackets on the right indicate that after the circle products are taken the real products have to be replaced by product variables, to remain within the projection framework.

We want to eliminate the $\boldsymbol{x}^{\mathrm{la}}$ from the condition that the r.h.s. vanishes. Fortunately for large γ the first two terms, which stem from the time evolution of $\boldsymbol{x}^{\mathrm{a}}$, dominate; the last two become negligible in comparison. Thus one has to solve

$$((\boldsymbol{x}^{\mathrm{lT}} \circ \boldsymbol{x}^{\mathrm{lT}} \boldsymbol{L}^{\mathrm{la}})) + ((\boldsymbol{x}^{\mathrm{lT}} \circ \boldsymbol{x}^{\mathrm{aT}} \boldsymbol{L}^{\mathrm{aa}})) = 0 \tag{S1.14}$$

From the structure of this one sees that the $\boldsymbol{x}^{\mathrm{l}}$ only act as "spectators", while considering the second factors one has to solve the same equation as at linear order. The solution is therefore expected to be $\boldsymbol{x}^{\mathrm{laT}} = ((\boldsymbol{x}^{\mathrm{lT}} \circ \boldsymbol{x}_0^{\mathrm{aT}}))$ as we want to show.

To see this in more detail we write out (S1.14) in components:

$$\sum_{l'} x_{(ll')} L_{l'a} + \sum_{a'} x_{la'} L_{a'a} = 0 \tag{S1.15}$$

Here we have written $x_{(ll')}$ to indicate that the indices are to be taken as ordered, *i.e.* $x_{(ll')} = x_{ll'}$ if $l \leq l'$ and $x_{(ll')} = x_{l'l}$ otherwise. The proposed solution is

$$x_{la} = ((\boldsymbol{x}^{\mathrm{lT}} \circ \boldsymbol{x}_0^{\mathrm{aT}}))_{la} = -\sum_{l'a'} x_{(ll')} L_{l'a'} (\boldsymbol{L}^{\mathrm{aa}})^{-1}_{a'a} \tag{S1.16}$$

This does solve (S1.15) because

$$\sum_{a''} x_{la''} L_{a''a} = -\sum_{l'a'a''} x_{(ll')} L_{l'a'} (\boldsymbol{L}^{\mathrm{aa}})^{-1}_{a'a''} L_{a''a} = -\sum_{l'} x_{(ll')} L_{l'a} \tag{S1.17}$$

S1.4 Elimination of $\boldsymbol{x}^{\mathrm{aa}}$

We proceed again using the product rule of differentiation, which gives as the equations of motion for the $\boldsymbol{x}^{\mathrm{aa}}$

$$\partial_t \boldsymbol{x}^{\mathrm{aaT}} = \gamma((\boldsymbol{x}^{\mathrm{lT}} \boldsymbol{L}^{\mathrm{la}} \circ \boldsymbol{x}^{\mathrm{aT}})) + \gamma((\boldsymbol{x}^{\mathrm{aT}} \boldsymbol{L}^{\mathrm{aa}} \circ \boldsymbol{x}^{\mathrm{aT}})) + \gamma((\boldsymbol{x}^{\mathrm{aT}} \circ \boldsymbol{x}^{\mathrm{lT}} \boldsymbol{L}^{\mathrm{la}})) + \gamma((\boldsymbol{x}^{\mathrm{aT}} \circ \boldsymbol{x}^{\mathrm{aT}} \boldsymbol{L}^{\mathrm{aa}})) \tag{S1.18}$$

Here all terms contribute for large γ, but one sees that in the first two the right factor of $\boldsymbol{x}^{\mathrm{a}}$ is again a "spectator" and similarly with the left factor for the last two terms. Accordingly one can show that the proposed solution, which is $\boldsymbol{x}^{\mathrm{aaT}} = ((\boldsymbol{x}_0^{\mathrm{aT}} \circ \boldsymbol{x}_0^{\mathrm{aT}}))$, ensures that each pair of terms vanishes separately.

For explicit calculation it is again useful to write out components. The aa' component of the first two terms of (S1.18), without the overall factor of γ, reads

$$\sum_{l'} x_{l'a'} L_{l'a} + \sum_{a''} x_{(a'a'')} L_{a''a} \tag{S1.19}$$

The proposed solution is

$$x_{aa'} = ((\boldsymbol{x}_0^{\mathrm{aT}} \circ \boldsymbol{x}_0^{\mathrm{aT}}))_{aa'} = \sum_{l_1 l_2 a_1 a_2} x_{(l_1 l_2)} L_{l_1 a_1} (\boldsymbol{L}^{\mathrm{aa}})^{-1}_{a_1 a} L_{l_2 a_2} (\boldsymbol{L}^{\mathrm{aa}})^{-1}_{a_2 a'} \tag{S1.20}$$

Substituting this and the solution (S1.16) for x_{la} turns (S1.19) into

$$-\sum_{l'l''a''} x_{(l'l'')} L_{l''a''} (\boldsymbol{L}^{\mathrm{aa}})^{-1}_{a''a'} L_{l'a} + \sum_{a'' l_1 l_2 a_1 a_2} x_{(l_1 l_2)} L_{l_1 a_1} (\boldsymbol{L}^{\mathrm{aa}})^{-1}_{a_1 a'} L_{l_2 a_2} (\boldsymbol{L}^{\mathrm{aa}})^{-1}_{a_2 a''} L_{a''a} \tag{S1.21}$$

The last factors in the second term again cancel, reducing it to

$$\sum_{l_1 l_2 a_1} x_{(l_1 l_2)} L_{l_1 a_1} (\boldsymbol{L}^{\mathrm{aa}})^{-1}_{a_1 a'} L_{l_2 a} \tag{S1.22}$$

After a relabelling of summation indices this is identical to the first term. This means that (S1.19) vanishes, *i.e.* the first two terms on the r.h.s. of (S1.18) cancel. Similarly the last two terms vanish, showing that (S1.20) is the correct QSS assignment of the $\boldsymbol{x}^{\mathrm{aa}}$.

S1.4.1 Equations of motion for $\boldsymbol{x}^{\mathrm{ll}}$

Above we have shown that the direct and the projection elimination procedures give the same time evolution equation for $\boldsymbol{x}^{\mathrm{l}}$. The same can then also be checked straightforwardly for the product variables $(\boldsymbol{x}^{\mathrm{l}} \circ \boldsymbol{x}^{\mathrm{l}})$ and their projection analogues $\boldsymbol{x}^{\mathrm{ll}}$. By analogy with (S1.13), the latter evolve in time according to

$$\partial_t \boldsymbol{x}^{\mathrm{llT}} = ((\boldsymbol{x}^{\mathrm{lT}} \circ \boldsymbol{x}^{\mathrm{lT}} \boldsymbol{L}^{\mathrm{ll}})) + ((\boldsymbol{x}^{\mathrm{lT}} \circ \boldsymbol{x}^{\mathrm{aT}} \boldsymbol{L}^{\mathrm{al}})) + ((\boldsymbol{x}^{\mathrm{lT}} \boldsymbol{L}^{\mathrm{ll}} \circ \boldsymbol{x}^{\mathrm{lT}})) + ((\boldsymbol{x}^{\mathrm{aT}} \boldsymbol{L}^{\mathrm{al}} \circ \boldsymbol{x}^{\mathrm{lT}})) \tag{S1.23}$$

The real products obey the same equation, just written differently:

$$\partial_t (\boldsymbol{x}^{\mathrm{l}} \circ \boldsymbol{x}^{\mathrm{l}})^{\mathrm{T}} = (\boldsymbol{x}^{\mathrm{lT}} \circ \boldsymbol{x}^{\mathrm{lT}} \boldsymbol{L}^{\mathrm{ll}}) + (\boldsymbol{x}^{\mathrm{lT}} \circ \boldsymbol{x}^{\mathrm{aT}} \boldsymbol{L}^{\mathrm{al}}) + (\boldsymbol{x}^{\mathrm{lT}} \boldsymbol{L}^{\mathrm{ll}} \circ \boldsymbol{x}^{\mathrm{lT}}) + (\boldsymbol{x}^{\mathrm{aT}} \boldsymbol{L}^{\mathrm{al}} \circ \boldsymbol{x}^{\mathrm{lT}}) \tag{S1.24}$$

From both, the fast products $(\boldsymbol{x}^{\mathrm{l}} \circ \boldsymbol{x}^{\mathrm{a}})$ (respectively $\boldsymbol{x}^{\mathrm{la}}$) and $(\boldsymbol{x}^{\mathrm{a}} \circ \boldsymbol{x}^{\mathrm{a}})$ (respectively $\boldsymbol{x}^{\mathrm{aa}}$) then need to be eliminated. As we have already established that these eliminations are identical, also the resulting equations for the ll-product variables must be identical.

S2 Appendix: Further analysis of memory effects

In this Appendix we give additional detail of the wide variety of effects that memory functions contribute to the behaviour of a system. Specifically, we explore the effect of the contributions from nonlinear memory terms and provide an alternative visualisation using a first order time expansion of the effective drift of the system.

S2.1 Nonlinear memory effects

In Fig. S1 we show trajectory plots comparing the predictions without memory (left column) to those with linear and linear+quadratic memory (middle columns) and to the dynamics of the full network. Focussing directly onto the description with all memory terms included, we can gain insight into what drives the distinct dynamics of Fig. S1C,G by analysing the respective nonlinear memory functions, specifically their amplitude (value at $\Delta t = 0$) as shown in Fig. S2A-D. We note that in most cases the memory contributions arise via Irx3. (In the case of memory of Olig2 this is the only possibility in any case.) A clear exception to this is the case of nonlinear Nkx2.2 memory to itself (Fig. S2A) where the memory is dominated ventrally by the Pax6 channel and dorsally by the Irx3 channel. These memory functions also change with time difference, but it turns out that they decay on similar timescales so that the memory amplitude gives a reasonable characterization of their strength. We observe that Olig2 does not produce memory effects that are substantial enough to change the dynamics (Fig. S2B,D): the dominant nonlinear memory effects come from Nkx2.2 (Fig. S2A,C). We see further that Nkx2.2 represses both itself (negative nonlinear memory amplitude, Fig. S2A) and, even more strongly, Olig2 (Fig. S2C). These two terms combined mean that at high levels of Nkx2.2, the memory functions will impede Olig2 from increasing as quickly as it would in cases without nonlinear memory, while simultaneously decreasing the concentration of Nkx2.2. Both effects are consistent with what we observe in Fig. S1C, D.

In the p2 domain the memoryless and linear memory predictions for trajectories are quite close, with levels of Nkx2.2 and Olig2 decreasing on similar timescales towards the steady state (Fig. S1E–H). Nonetheless, plots showing the norm of the effective drift indicate the trend that memory pushes the system more quickly towards low Nkx2.2 (Fig. S3D,E, see below). Including the nonlinear memory then further enhances this tendency: the levels of Nkx2.2 drop more rapidly than Olig2 on the approach to the steady state (Fig. S1G,H & Fig. S3F). The behaviour is not identical to but mimics that of the full thermodynamic model: the nonlinear memory effects contribute qualitative information about the behaviour of the system further away from the steady state. This points to memory terms being important for first reducing levels of Nkx2.2 before reducing Olig2 levels.

Understanding why this is the case requires a more detailed analysis of the memory effects. The memory amplitudes show that Nkx2.2 is not subject to substantial memory effects (Fig. S2E,F) and the largest memory effects act on Olig2 (Fig. S2G,H). At first glance, the latter memory amplitudes suggests that Nkx2.2 substantially represses Olig2 while Olig2 activates itself to a lesser degree. However, on closer inspection of the temporal dependence of the memory functions, the repression by Nkx2.2 is a short pulse that for larger Δt turns into an activation of Olig2 (Fig. S1F). Meanwhile, the Olig2 self-activation is a sustained signal that lasts much longer than the repression by Nkx2.2. When taken together, these two memory terms (combined with cross terms, not shown) promote activation of Olig2 from high levels of Nkx2.2 even after a substantial time difference has elapsed, of the order of the decay time of the linear memory functions. This is consistent with what we observe in Fig. S1G,H: the initial high levels of Nkx2.2 lead to Olig2 levels being sustained via the memory, while Nkx2.2 decays in a way almost unaffected by memory. Eventually Olig2 also decays, once enough time has passed for the nonlinear memory effects to fade away.

S2.2 Calculation of effective drift in the presence of memory

It is useful to be able to visualize the effects of memory terms in the projected equations in terms of an effective drift. This is possible at least perturbatively for short times, as we now show. We illustrate the method for the linearized dynamics, where the projected equation (3) without the random force read in vector form:

$$\partial_t \boldsymbol{x}^{\mathrm{T}} = \boldsymbol{x}(t)^{\mathrm{T}} \boldsymbol{\Omega} + \int_0^t dt' \boldsymbol{x}(t)^{\mathrm{T}} \boldsymbol{M}(t - t') \tag{S2.1}$$

For small times t, the memory function can be treated as approximately constant so that one obtains in an expansion to first order in t:

$$\int_0^t dt' \boldsymbol{x}(t)^{\mathrm{T}} \boldsymbol{M}(t-t') \simeq t\, \boldsymbol{x}(t)^{\mathrm{T}} \boldsymbol{M}(0) \tag{S2.2}$$

Combining with the rate matrix term gives the effective drift

$$\partial_t \boldsymbol{x}^{\mathrm{T}} = \boldsymbol{x}(t)^{\mathrm{T}} [\boldsymbol{\Omega} + t\boldsymbol{M}(0)] \tag{S2.3}$$

On the right hand side one has a function of the current concentrations only, as announced. Given that this is a linear expansion in t we refrain from exploring the system too far away from $t = 0$. Higher order expansions in t could be performed yet the first order is enough to demonstrate the qualitative contributions of the memory terms.

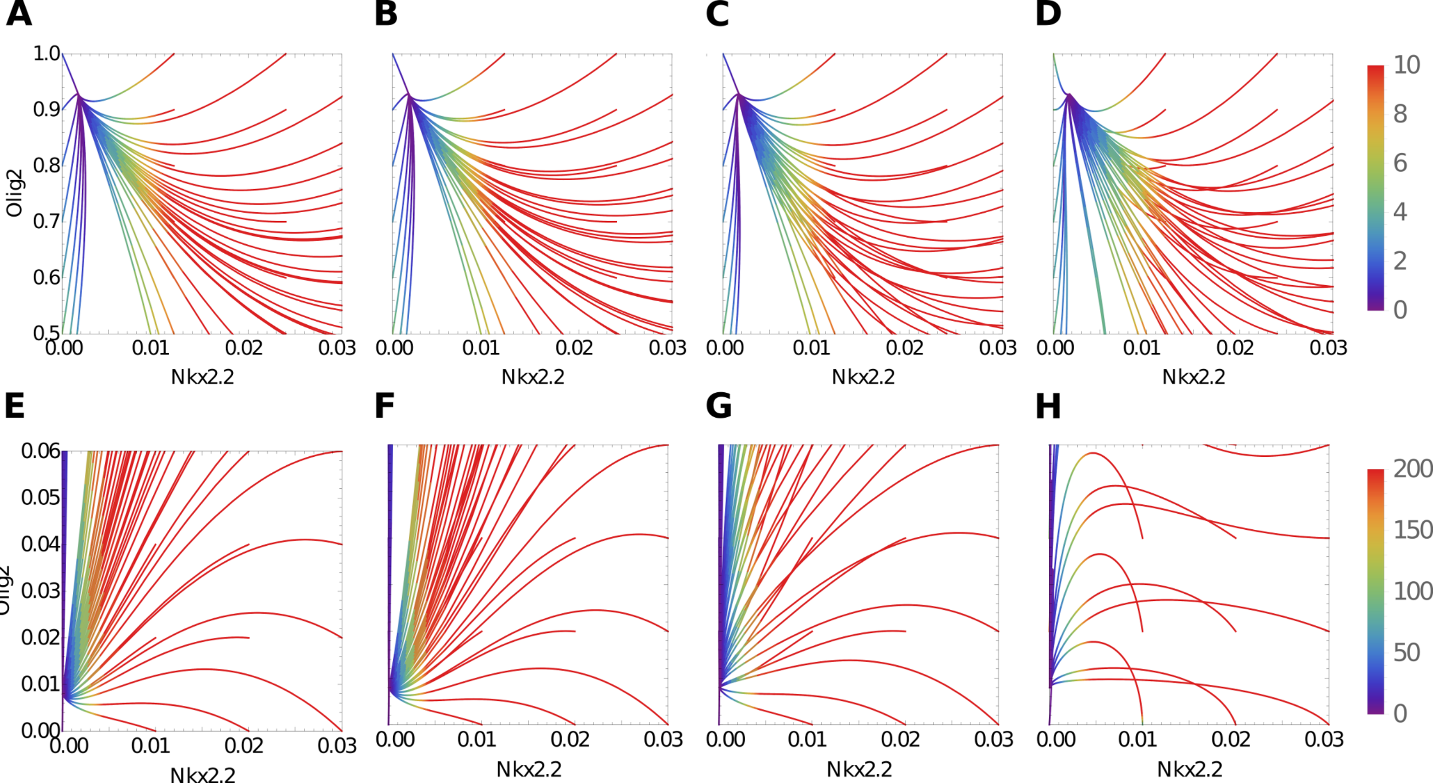
We illustrate the approach in Fig. S3A-F with contour plots of the norm of the effective drift vector on the r.h.s. of (S2.3), for time $t = 0.8$. Bearing in mind that the system will pass quickly through regions where the drift is high and spend most of its time in regions where it is low, these plots confirm the effects seen in the trajectory plots: inclusion of the memory terms causes the low-drift region to shift to lower Nkx2.2 concentrations, *i.e.* the system will more rapidly reduce Nkx2.2 and then spend more time increasing Olig2 at small Nkx2.2.

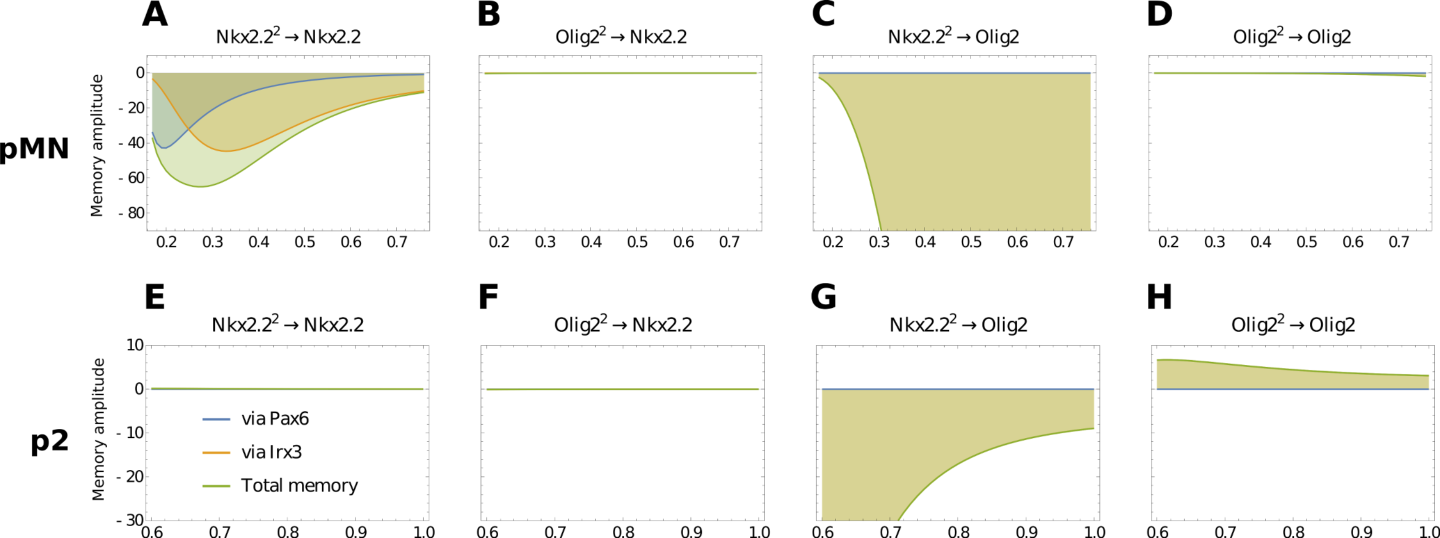
S3 Appendix: Model details

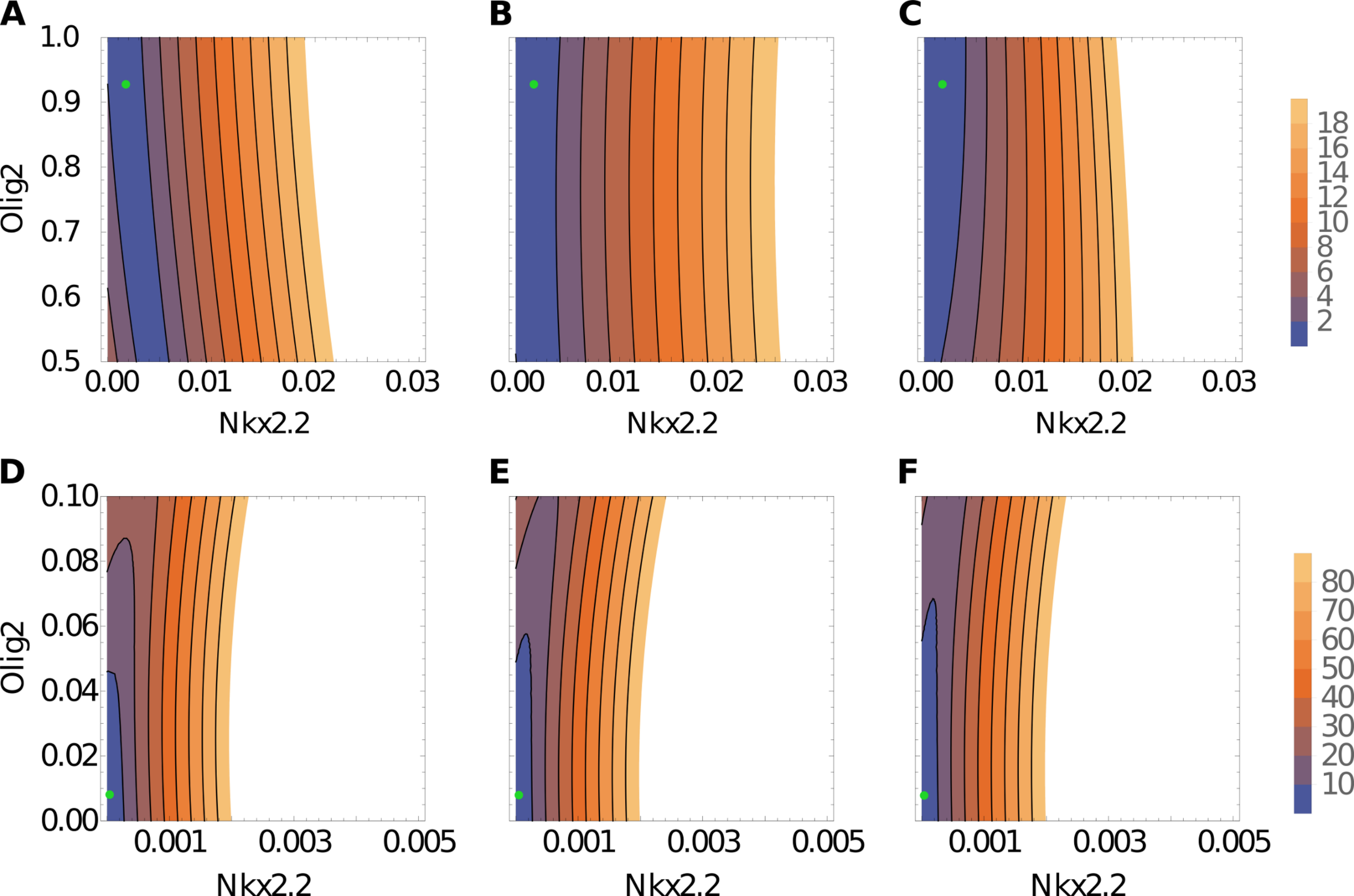
The full network equations for the neural tube application in Sec. *Application to neural tube network* are of the form (1, 2) and taken from [24]:

$$
\begin{aligned}
\partial_t[\mathrm{P}] &= \alpha_{\mathrm{P}} \frac{w_{\mathrm{P}}}{w_{\mathrm{P}} + (1 + w_{\mathrm{PO}}[\mathrm{O}])^2 (1 + w_{\mathrm{PN}}[\mathrm{N}])^2} - \beta_{\mathrm{P}}[\mathrm{P}] \\
\partial_t[\mathrm{O}] &= \alpha_{\mathrm{O}} \frac{w_{\mathrm{O}}(1 + w_{\mathrm{OS}}[\mathrm{S}])}{w_{\mathrm{O}}(1 + w_{\mathrm{OS}}[\mathrm{S}]) + (1 + w_{\mathrm{ON}}[\mathrm{N}])^2 (1 + w_{\mathrm{OI}}[\mathrm{I}])^2} - \beta_{\mathrm{O}}[\mathrm{O}] \\
\partial_t[\mathrm{N}] &= \alpha_{\mathrm{N}} \frac{w_{\mathrm{N}}(1 + w_{\mathrm{NS}}[\mathrm{S}])}{w_{\mathrm{N}}(1 + w_{\mathrm{NS}}[\mathrm{S}]) + (1 + w_{\mathrm{NP}}[\mathrm{P}])^2 (1 + w_{\mathrm{NO}}[\mathrm{O}])^2 (1 + w_{\mathrm{NI}}[\mathrm{I}])^2} - \beta_{\mathrm{N}}[\mathrm{N}] \\
\partial_t[\mathrm{I}] &= \alpha_{\mathrm{I}} \frac{w_{\mathrm{I}}}{w_{\mathrm{I}} + (1 + w_{\mathrm{IO}}[\mathrm{O}])^2 (1 + \mathrm{w}_{\mathrm{IN}}[\mathrm{N}])^2} - \beta_{\mathrm{I}}[\mathrm{I}]
\end{aligned}
$$

Here [S] is the level of Sonic Hedgehog signaling - the net amount of Gli activity, see [24] for details. This is taken to be an exponential gradient, $[\mathrm{S}] = e^{-s/0.15}$ where $s \in [0, 1.2]$ labels the neural tube position. The remaining symbols in brackets indicate the concentration of the TFs. Parameters used are $\alpha_{\mathrm{P}} = 2$, $\alpha_{\mathrm{O}} = 2$, $\alpha_{\mathrm{N}} = 2$, $\alpha_{\mathrm{I}} = 2$, $\beta_{\mathrm{P}} = 2$, $\beta_{\mathrm{O}} = 2$, $\beta_{\mathrm{N}} = 2$, $\beta_{\mathrm{I}} = 2$, $w_{\mathrm{PO}} = 1.9$, $w_{\mathrm{PN}} = 26.7$, $w_{\mathrm{P}} = 3.84$, $w_{\mathrm{ON}} = 60.6$, $w_{\mathrm{OI}} = 28.4$, $w_{\mathrm{OS}} = 180$, $w_{\mathrm{O}} = 38.24$, $w_{\mathrm{NP}} = 4.8$, $w_{\mathrm{NO}} = 27.1$, $w_{\mathrm{NI}} = 47.1$, $w_{\mathrm{NS}} = 373$, $w_{\mathrm{N}} = 21.92$, $w_{\mathrm{IO}} = 58.8$, $w_{\mathrm{IN}} = 76.2$, $w_{\mathrm{I}} = 18.72$.


A
B
C
D
E
F
G
H
Olig2
Nkx2.2
1.0
0.9
0.8
0.7
0.6
0.5
0.00
0.01
0.02
0.03
0.06
0.05
0.04
0.03
0.02
0.01
0.00
10
8
6
4
2
0
200
150
100
50
0


pMN
p2
A
Nkx2.2² → Nkx2.2
B
Olig2² → Nkx2.2
C
Nkx2.2² → Olig2
D
Olig2² → Olig2
E
Nkx2.2² → Nkx2.2
F
Olig2² → Nkx2.2
G
Nkx2.2² → Olig2
H
Olig2² → Olig2
Memory amplitude
0
- 20
- 40
- 60
- 80
0.2
0.3
0.4
0.5
0.6
0.7
10
0
- 10
- 20
- 30
0.6
0.7
0.8
0.9
1.0
via Pax6
via Irx3
Total memory


A
B
C
D
E
F
Olig2
Nkx2.2
1.0
0.9
0.8
0.7
0.6
0.5
0.00
0.01
0.02
0.03
18
16
14
12
10
8
6
4
2
0.10
0.08
0.06
0.04
0.02
0.00
0.001
0.003
0.005
80
70
60
50
40
30
20
10