

Full Length Article

Neurons and neural networks to model proteins and protein networks

Sandhya Samarasinghe ^{a,*}, Tran Nguyen Minh-Thai ^b, Komal Sorthiya ^c, Don Kulasiri ^a^a Complex Systems, Big Data and Informatics Initiative (CSBD), Lincoln University, New Zealand^b Faculty of Information Systems, College of Information and Communication Technology, Can Tho University, Viet Nam^c Digital Natives Academy, Rotorua, New Zealand

ARTICLE INFO

Keywords:

Proteins
Neural networks
Neuron models of proteins
Protein interaction networks
Auto-associative neural networks
Hybrid models

ABSTRACT

This study demonstrates the success of Auto-associative neural networks (AANN) to represent protein networks, where each neuron maps to a protein and each neuron interaction to a specific protein interaction. Core mammalian cell cycle system with 12 proteins was used to train AANN with data generated from an ODE and Boolean models. When tested if AANN can find unknown system interactions, trained AANN with nonlinear (sigmoid) neurons captured accurate system dynamics but failed to capture the correct protein interactions. With correct protein interactions, AANN with linear neurons captured 50 % of protein behaviour and sigmoid AANN captured all protein dynamics correctly. This allowed hybrid-AANN with linear and nonlinear neurons. Self-learning ability of AANN was tested but it was not evident in the current model architecture. When tested for their ability to hold past memory by training AANN as a recurrent network, system dynamics revealed near perfect accuracy, with the network heavily relying on the past state to produce the current state. We also tested if neurons can be trained separately and assembled into AANN. Linear, nonlinear and binary (for representing Boolean) neurons were trained. Linear neurons modelled most proteins (70 %), and sigmoid neurons modelled all proteins correctly. Binary (perceptron) models successfully replicated Boolean rules of proteins. From these, a number of AANN models were assembled: sigmoid AANN accurately predicted the system; binary AANN revealed correct protein activation with temporal realism; two hybrid-AANN models, one with linear/sigmoid neuron models and another with binary/sigmoid neuron models, were successfully assembled to further simplify models.

1. Introduction

Proteins are ubiquitous in life, carrying out most functions in living organisms. They organise themselves into interaction networks in time and space inside a cell. Through these networks, proteins are produced, activated for a period to perform their assigned functions and deactivated or degraded when they are no longer needed. Protein networks are complex webs of interaction developed over evolutionary timescales and shared across species and adapted according to conditions making it challenging to understand how they organise and function. Understanding and deciphering meaning from these networks through computational modelling has become a major endeavour in systems biology (Kitano, 2007). Modelling protein networks can not only help understand the mechanisms underpinning biological function but also reveal potential causes of diseases and avenues to restore health.

In a molecular system such as a protein interaction network, an event or a signal can trigger a chain of events where one or more elements

influence one or more other elements as needed to perform the intended function, as shown in the example network of the core control system of mammalian cell cycle in Fig. 1. When triggered, these networks traverse through various states of protein activity over time as required to fulfil the intended function and return the network to the initial state. Much of the effort in studies on protein networks has gone into understanding their complexity and computationally modelling molecular interactions to explore emerging behaviour such as system stability and robustness or how they achieve specific functional outcomes, such as completing the required stages of cell cycle (Abroudi et al., 2017; Ling et al., 2010).

The current challenge in modelling protein networks is twofold: lack of precise knowledge of protein-protein interactions, and modelling approaches that are flexible and effective in revealing the full spectrum of systems dynamics representing the temporal unfoldment of behaviour of all proteins in system. A number of mathematical and computational modelling approaches have been used to study biological networks, each requiring different levels of information and data about the system.

* Corresponding author.

E-mail address: sandhya.samarasinghe@lincoln.ac.nz (S. Samarasinghe).

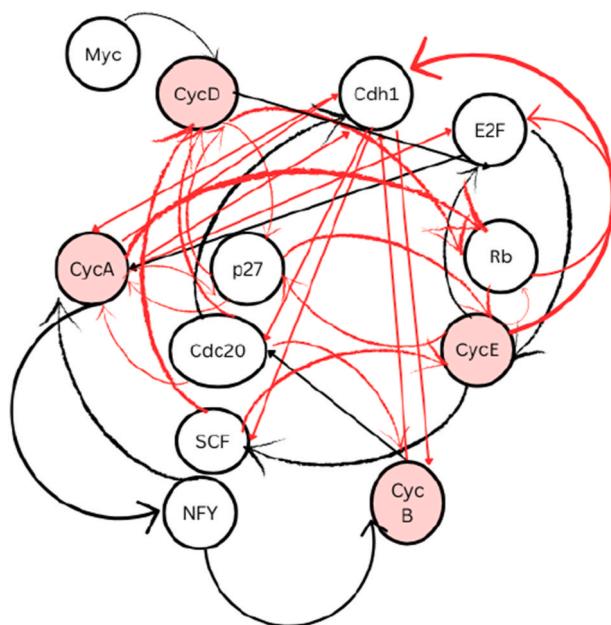


Fig. 1. Core control system of mammalian cell cycle featuring the four cyclins (CycD, CycE, CycA and CycB) and their regulators that collectively accomplish the timely production and degradation of cyclins (black arrows represent activation and red arrows inhibition).

These methods represent individual molecular interactions in varying levels of specificity and accuracy, mostly based on physics or logical reasoning. Ordinary Differential Equations (ODE) have by far being the most popular and most able to characterise continuous dynamics of a system (Tyson, 1991). They represent each reaction as a rate equation where parameters (kinetic constants) are assumed to be known, which is currently a bottleneck as most of these parameters are not available for many reactions in molecular systems. Further, these models are challenged by network size and complexity. Another popular modelling approach is Boolean modelling (Waidyaratne and Samarasinghe, 2018; Faure et al., 2006) that represents protein behaviour using IF-THEN rules in binary (on/off) form. The simplicity of binary rules helps Boolean models explore large scale system behaviour in qualitative (binary) form. However, they are unable to produce continuous system dynamics due to binary (on/off) protein behaviour and, due to the synchronous or asynchronous (random) firing of rules, they not only can miss real temporal behaviour of the protein system but also can send the system through spurious trajectories. These point to gaps or current limitations in modelling protein systems; specifically, models either require scarce information such as kinetic rates (ODE) or fail to capture continuous systems dynamics with temporal realism (Boolean). Therefore, addressing these limitations is a priority in this domain and we propose neural networks as a promising approach to bridge both these gaps simultaneously. This is because neural networks could potentially be designed to model individual proteins and protein networks that are able to capture continuous dynamics without relying on kinetic parameters. As more and more genomics and proteomics data are generated from high-throughput experiments, data driven models such as neural networks, that can accurately and efficiently represent protein networks and their continuous dynamics even with incomplete knowledge of reactions, can address aforementioned limitations and make significant contributions to systems biology.

Neural networks hold great promise to contribute to modelling molecular systems, but their full potential is yet to be explored. Neural Networks with different network configurations available for modelling different types of systems (feed-forward, feedback or recurrent etc.) can offer a rich platform for modelling continuous nonlinear system behaviour of protein systems. The biggest challenge for neural networks,

especially in contexts where model interpretation and gaining insights are crucial, has been the fact that most neural networks are considered black boxes due to their complex organisation of neurons in multiple layers that are not amenable to finding meaning or directly representing proteins as individual entities with specific interactions in the system (Samarasinghe, 2006). Therefore, neural representations where individual neurons represent proteins and the neural network represents the collection of interacting proteins can provide greater realism and enable meaningful insights into the system. These considerations led to the exploration in this study of neural network structures that exactly represent proteins and protein interaction systems and investigation of their potential to produce accurate individual protein behaviour as well as global system behaviour and their ability to shed some light on potential evolutionary characteristics of protein networks.

We, in fact, initiated this kind of one-to-one mapping of neurons to proteins in Ling et al. (2013). We demonstrated this idea by converting an ODE model of a small molecular network in the DNA damage repair pathway of the mammalian cell cycle to an equivalent Recurrent Neural Network (RNN) with specific feedback connections as in the ODE (Ling et al., 2013). The network was trained with data generated from the ODE model and the network was able to mimic the behaviour of the ODE system precisely and estimate the network parameters from data with high accuracy. This network maps neurons to proteins but in essence it transforms the ODE model into an algebraic form using neural modelling to implement a discretised version of ODE. As such, it still does not exploit the full capabilities of neural networks as a standalone modelling platform for protein networks. Therefore, in the proposed study, we aim to lift the neural representation of molecular networks to another level by developing and training neural networks that directly map neurons to proteins while preserving correct system interactions to model protein networks from data.

A neural network that fits the above description of protein networks is Auto Associative Neural Network (AANN). It is a type of a single layer neural network with neurons and their interactions as shown in Fig. 3b where the activity of all neurons collectively represents the state of the system at any point in time. In AANN, a neuron can influence specific other neurons through their unique interactions with them, similar to the way proteins interact in the example network in Fig. 1, and the current state of a neuron can be influenced through feedback connections between neurons. When triggered, AANN traverses through a series of states of neuron activity over time according to the interactions in the system until it reaches a final state (Hopfield, 1982). Unfolding behaviour of all neurons in the network in time as a system reveals systems dynamics. Considering the topological and functional similarity of protein networks to AANN, we hypothesised that protein networks, such as the core control system of mammalian cell cycle in Fig. 1 that was used as a case study here, can be represented by AANN, and unfolding of the AANN over time should capture the timely operation of proteins in the system and produce expected system behaviour. Further, owing to various options for continuous neuron activation functions (linear, nonlinear), we hypothesise that these ANNs can produce continuous models with accurate system dynamics similar to ODE models. A key feature here is that each protein is represented by an individual neuron in the system thereby making it possible to reason about the behaviour of individual proteins as well as their collective unfolding as a whole system. This avoids the bottleneck of an arbitrarily large number of neurons being involved in the representation of an individual protein or protein system as in a typical neural network model that would make it a black box.

Importantly, neural networks learn the behaviour of the system from data without the need for *a priori* knowledge of kinetic parameters for reactions in the system. This provides a great advantage for modelling protein systems where kinetic rates are not yet fully known. Further, as AANN is a replica of the protein system being modelled, AANN parameters represent the corresponding reaction kinetic rates; therefore, AANN can approximate kinetic rates from data which is of great benefit

to fully characterise protein systems and can promote their experimental validation. Most modelling approaches model the *known* behaviour of individual system elements, and the individual models are combined into a full system model that is then simulated. Neural networks are model-free systems that discover unknown model from data. They allow not only modelling protein networks as whole systems, such as AANN without needing to combine individual models, but they also allow modelling individual proteins separately and assembling them into AANN system models.

Additionally, with the possibility of binary neuron activation functions, AANN could produce discrete models with system dynamics that are more realistic than Boolean models. Importantly, AANN could represent discrete system dynamics quantitatively without rules and with correct temporal realism to overcome the limitation of Boolean models with qualitative discrete dynamics that lack temporal realism. In Boolean modelling, reactions are represented by logical (IF-THEN) rules and the state of the system elements by discrete on or off [0, 1] values that allow characterisation of large molecular systems simply and exploration of emerging behaviour qualitatively. These have proven valuable in gaining insights into system dynamics qualitatively (Naldi et al. (2011)). They are unable to capture system dynamics quantitatively due to their rule-based nature and unable to produce continuous systems dynamics due to lack of temporal realism in the reactions represented by Boolean rules [20, 21] which can also lead to spurious system trajectories in model simulation. These limitations can be overcome by AANN.

Further, reducing model complexity has been a concern for simplifying large molecular network models. Considering the previously stated challenges, there remains a significant concern as to how to develop simpler continuous models of complex biological systems that can still accurately predict the outcomes of biological experiments and help explore systems properties. In the past, model reduction or developing simpler models from scratch has been explored. AANN provides a convenient platform for model simplification and reduction. Some protein systems are simpler with quick activations and others maybe slow with linear or nonlinear behaviour. AANN could help develop simpler binary systems models or more complex linear or nonlinear models. Past examples in this endeavour includes exploiting rule-based modelling that employs a set of rules to approximate system behaviour by Danos et al. (2007) who transformed two ODE models for the EGF receptor signalling pathway using 70 rules (Brightman and Fell, 2000; Schoeberl et al., 2002). Faure et al. (2006) converted an ODE model of mammalian cell cycle into a simpler Boolean model. Alsharaiah et al. (2023) developed a fuzzy control system to represent the core control system of mammalian cell cycle using fuzzy IF-THEN rules that is a continuous analogue of Boolean models and therefore a compromise between Boolean and ODE models. In this fuzzy model, a single reaction could still involve a number of rules thus making model development still complex, especially for larger systems. Further, Naldi et al. (2011) explored a qualitative graph-based approach to reduce large logical networks to smaller ones. Petri nets have become popular for discrete and continuous dynamics modelling of molecular systems at various levels of abstraction; they seem to be promising but they can face the same issues as ODEs (Fujita et al., 2004). Therefore, AANN can provide a flexible data driven platform for model simplification.

Another advantage of the AANN is the possibility for greater model simplification with hybrid neuron models within the same (neural) modelling formalism. For example, it can incorporate discrete and continuous representations in the same model to simplify molecular systems. This can also help explicitly exploit time scale separation existing within some of these systems as some reactions can be very fast, such as protein activations, and some are slow, such as protein synthesis. Fast reactions can be represented by binary and slow reactions by continuous neurons within the same model. In the past, this required combining two modelling formalisms such as ODE and Boolean or ODE and petri net etc. For example, Fujita et al. (2004) developed a hybrid (discrete/continuous) model for fission yeast cell cycle by integrating

continuous and discrete representations. The challenge with this model is representing different levels of abstraction, making it difficult to understand. Singhania et al. (2011) developed a hybrid ODE/discrete representation for mammalian cell cycle with manual entering of discrete [0,1] states into the hybrid model. Herajy and Heiner (2012) presented a hybrid Petri Net (PN) model to simulate the eukaryotic cell cycle system to transform an ODE model for cyclin B, one of the main controllers of cell cycle. Abroudi et al. (2020) also developed a hybrid discrete/continuous PN for the core regulatory control system of mammalian cell cycle where logical rules and ODEs were integrated. Some of the already mentioned issues of ODE models apply to these hybrid models as well. AANN allows a flexible platform to combine binary, linear, nonlinear neuron functions within the same network thus greatly simplifying hybrid models and model development without having to integrate two or more different methods.

Additionally, the exploration of individual neuron models of proteins could also be promising for modular or bottom-up construction of large networks by assembling well trained individual protein models into AANN. These individual protein models can also help explore unique behaviours of individual proteins more closely. Finally, AANN is amenable to the exploration of the self-learning potential and memory of protein networks from an evolutionary perspective. It is possible that evolution uses self-adaptation strategies in evolving and fine-tuning protein networks which could be characterised as self-learning. In evolutionary settings, self-adaptation could also entail memory formation as in neural networks in the brain. In this study, we propose and explore all aforementioned capabilities of AANN.

2. Goal

This study aims to develop a novel approach to modelling individual proteins and protein-protein interaction networks based on neural information processing through data driven models. The approach makes significant theoretical and practical contributions to systems biology and protein modelling. From a theoretical perspective, it introduces a conceptual framework based on Auto-Associative Neural Networks that enables modelling sparsely and specifically connected systems such as protein interaction networks to unfold their continuous dynamics with correct temporal realism. From a practical perspective, proposed models are easy to develop, flexible, effective and meaningful while accurately reproducing the full spectrum of protein and network behaviour. Specifically, it proposes individual neurons to represent individual proteins in a protein network such as that in Fig. 1, and Auto-Associative Neural Networks (AANN) such as that in Fig. 3b with interaction mechanisms exactly as in a protein interaction network as in Fig. 1 to model a protein system to produce accurate continuous systems dynamics from data.

At the core of neural network model development is a set of data representative of the system. In this study, the core protein network of mammalian cell cycle control system consisting of 12 proteins shown in Fig. 1 was used as the case study as explained in detail in Section 3. An ODE model developed for this cell cycle system by Abroudi et al. (2017) shows that the protein dynamics are nonlinear in time (Fig. 2). Individual protein models and AANN system models were trained with the data generated from this ODE model and a Boolean model of the same system developed by Alsharaiah et al. (2018). These data were used as benchmarks to show how well the proposed AANN performs against ODE and Boolean models. Data generated from these models were used to prove the modelling concept, as in reality experimental data on protein interaction networks are still limited. If successful, the model concept can be used with real data when they are available for model training or refining.

We explore models from two perspectives: one concerns the development of whole system protein models with AANN, and the other concerns development of individual protein models and assembly of these models into AANN. The reason for the latter is that, in biology, an in depth understanding of the behaviour of some crucial proteins can be

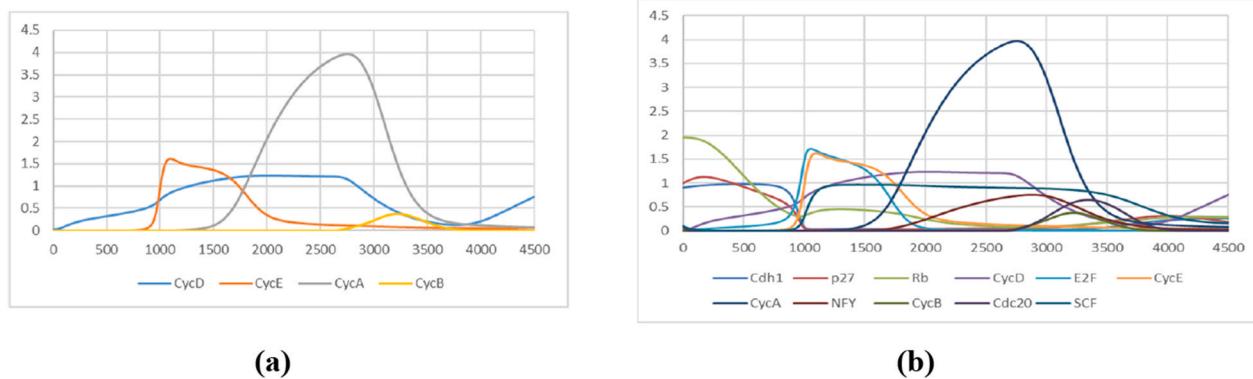


Fig. 2. Cell cycle cyclins and their regulators. a) Concentration of the four cell cycle cyclins (CycD, CycE, CycA and CycB) over cell cycle period (4500 time units representing 24 h), (b) Concentration of Cyclins along with their regulators.

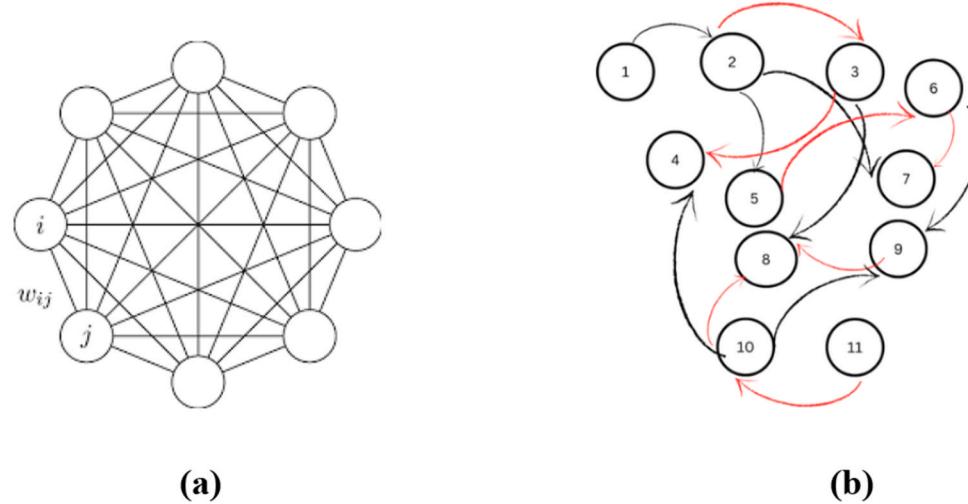


Fig. 3. Example Auto Associative Neural Networks (AANN) configurations: a) Fully connected bi-directional symmetrical ($w_{ij} = w_{ji}$) AANN (Hopfield network) and b) AANN with specific connections showing nodes and their interactions which can be positive (black) or negative (red).

useful. Further, data may not always be available for all proteins for whole systems modelling, and therefore, a modular approach such as developing individual protein models and their subsequent assembly into full models maybe practical in some cases. In both cases, we also explore the potential for hybrid AANN models for simplifying the models and to account for limited quantity and type (such as binary data from Western blot etc.) of data. Accordingly, a systematic approach was taken to develop the proposed AANN representations by asking and addressing a number of key questions directed at these two modelling perspectives as below. First four questions are aimed at whole system AANN models and the last three are aimed at individual models and their assembly into AANN. The questions are logically connected and aim to dive deep progressively into various model capabilities; for example, the first four questions explore systematically the ability of AANN models to - discover unknown protein interaction systems from data and model known protein interaction systems and their behaviour - and then explore other potential attributes of AANN such as self-learning and capacity for holding memory. These questions are about modelling protein networks as whole systems using AANN and what can be learned from AANN about potential evolutionary characteristics of protein networks and their ability to hold memory. The last three questions are about building individual protein models separately and their assembly into whole system AANN models to explore modular and hybrid construction of protein systems.

- 1) Can a neural network discover an *unknown* protein interaction system from data? More specifically, can a neural network in the form of AANN with individual neurons representing individual proteins and without *a priori* specified protein interactions be trained to discover the correct protein interaction patterns and produce accurate system dynamics (mimic the behaviour of all proteins over time) from data.
- 2) If the above model fails, we ask, can a neural network learn to produce accurate behaviour of a *known* protein system from data? Specifically, can AANN as above but with *a priori* specified connections as in the real system be trained to mimic continuous protein dynamics? Here, the pattern of protein interactions as they are in the real system is modelled. We aim to test few versions of AANN with linear neurons only, nonlinear neurons only and explore the potential of hybrid AANN (with both linear and nonlinear neurons within the system). This will reveal if the whole system is nonlinear or if some proteins behave linearly with respect to their activators. In particular, this will allow hybrid modelling in the same modelling formalism of neural networks without the need for integrating different modelling methods in hybrid modelling.
- 3) Can a neural network self-correct over time based on its own generated outputs to correctly mimic a protein system? Specifically, can a protein network represented by an AANN as in 2 above but trained with self-generated data mimic protein dynamics? Here, the system relies on reinforcement learning where it only uses its own outputs generated in the previous step as inputs in the next time step

- to train itself iteratively without relying on externally provided training input data. Target outputs are used only to guide training as reinforcement of the direction it should take. This question explores self-adaptation of protein networks.
- 4) Do protein networks have temporal memory? In neural networks with temporal memory, called recurrent networks, individual neurons self-feedback their outputs in previous step(s) as an additional input in the current step along with training data. In protein networks too, some proteins also self-produce and self-perpetuate in addition to being manipulated by their regulators. Therefore, we ask if a protein network can be a recurrent AANN with temporal memory (as in 2 above but trained with inputs from two sources: actual inputs and the output of each neuron/protein in the previous step self-feeding the same protein in the next step) to mimic protein dynamics? This helps explore the memory capacity of protein networks that have evolved over their developmental history.
 - 5) Turning the inquiry around from system level to entity level, we then ask – can individual neuron models be trained separately to mimic the behaviour of single proteins with linear, nonlinear and binary (on/off) neurons? These neurons can represent different rates of protein production, activation and degradation, leading to fast (binary type) and slow (linear and nonlinear) responses. These individual models will help understand and explore the behaviour of individual proteins in a system in greater detail and accuracy. They will be particularly meaningful as most protein data are still limited and they are collected on individual protein basis, not as a whole system.
 - 6) Can trained individual protein models be assembled into whole system AANN (binary, linear and nonlinear) to mimic accurate system dynamics? This allows individual models to be combined into an AANN systems model in modular or bottom-up approach to suit the available data (binary data or limited or ample continuous data). The binary AANN models also test the possibility of converting Boolean protein network models into neural network models.
 - 7) Can hybrid AANN be assembled from a combination of individual neuron models with a linear, nonlinear or binary function in the same model to properly represent the variety of protein behaviour patterns- fast (binary) and slow (linear or nonlinear). This hybrid mode allows models to exploit time scale separation (fast and slow reactions) existing in protein systems. Until complete datasets are available, this hybrid approach may be a powerful mechanism to understand protein systems and get the most benefit from available data.

3. Methods

The proposed approach can be more effectively presented in alignment with a specific example protein network. Therefore, to provide context for the study, here we describe the case study used for model development, that is the core control system of mammalian cell cycle that has been modelled, extended and advanced in the studies of [Faure et al. \(2006\)](#), [Singhania et al. \(2011\)](#) and [Alsharaiah \(2018\)](#) as Boolean models and [Abroudi et al. \(2017\)](#) as an ODE model. We first briefly describe this core-control system of mammalian cell cycle based on its ODE representation by [Abroudi et al. \(2017\)](#). We then describe the neural networks approach proposed to answer all the key questions raised in the Goal section.

3.1. Model development case study- core control system of mammalian cell cycle

Cell cycle is the process by which a cell implements cell division in biological organisms. Cell cycle is fundamental to development, regeneration and wound healing in organisms ([Bodenstein, 1913](#)). It is also the cause of many diseases including cancer due to uncontrolled cell division. Therefore, cell cycle progression is tightly controlled so that a

cell is divided only when and where it is necessary. Dysfunction in the cell cycle control system can lead to aberrant DNA that can cause diseases such as cancer ([Behl and Ziegler, 2014](#); [Farr et al., 2017](#)). Cyclin proteins are the drivers of cell cycle. The core cell cycle control system shown in [Fig. 1](#) is the network of essential cell cycle controllers (cyclins) and their regulators that control the progression of cell cycle. Much of cell cycle control is currently unknown. Therefore, most cell cycle modelling has focussed on representing known cyclin dynamics arising from the interaction with their regulators. In cell cycle, two daughter cells are produced from a single cell through DNA replication and cell division processes ([Behl and Ziegler, 2014](#)). To accomplish these tasks, cell cycle progresses through four phases - G1, S, G2, and M. In G1 and G2 phases, a cell grows in size, in S phase, DNA is replicated, and in M phase, DNA is segregated prior to final cell division. Cyclins are directly involved in these processes from initial preparation of the cell for division as well as the subsequent DNA replication and segregation. Therefore, as the controllers of cell cycle progression, Cyclin (Cyc) levels are carefully controlled during cell cycle through their timely synthesis and degradation by specific regulatory proteins ([Dong et al., 2014](#)). Additionally, to prevent accidental or unnecessary cell division, cells use cyclin inhibitors to keep cell cycle inhibited until new cells are needed.

In mammalian cell cycle, there are four main cyclins – CycD, CycE, CycA and CycB. CycD and CycE control the G1 phase by preparing the cell for division including growing the cell size to accommodate the contents of two daughter cells. CycE also helps assemble the DNA replication machinery in late G1 and CycE and CycA together control S phase by unwinding DNA for replication ([Limas and Cook, 2019](#)). CycA regulates further cell growth in G2 and CycB controls M phase by regulating DNA segregation.

According to [Fig. 1](#), cyclin synthesis and degradation are controlled by a number of regulatory proteins that are mainly transcription factors that produce them and ubiquitin ligases that help degrade them. For instance, Myc produces Cyclin D and SCF ubiquitinates and promotes its degradation ([Fig. 1](#)). Through these interactions, cyclins affect the timely progression of cell cycle through the four stages. [Fig. 2b](#) shows the behaviour of the activators and degraders that control cyclin levels in cell cycle based on the [Abroudi et al. \(2017\)](#) ODE model of cell cycle. [Fig. 2a](#) highlights the relative dynamics of the four cyclins and their aforementioned regions of activity in cell cycle.

We briefly describe how cyclins are controlled through cell cycle according to [Figs. 1 and 2](#) highlighting the role and timing of activation of the proteins in [Fig. 1](#). P27 is the main cell cycle inhibitor and thus its levels are high at the beginning of cell cycle. In the presence of growth factors, Myc activates cell cycle by producing Cyclin D. CycD is the first cell cycle cyclin leading the G1 phase and it is instrumental in ensuring cell's commitment to cell cycle by suppressing the inhibitory effect of p27. CycD also helps release the transcription factor E2F (bound to Rb) in order to facilitate in order the synthesis of CycE and CycA. CycD is degraded initially by SCF in G1 phase and CDC20 in M phase.

CycE is the second cyclin and gets produced in late G1 phase and one of its first tasks is to inhibit CDH1, the main degrader of CycB, so that CycB can perform its function when produced. A very important task of CycE in G1 is the assembly of the DNA replication machinery. These activities allow the cell to transition from G1 to S phase where DNA is replicated with the help of CycE and CycA. CycA is the third cyclin whose production is initiated in late G1 but it is greatly accelerated in S phase. Both CycE and CycA help unwind DNA to aid DNA replication in S phase ([Limas and Cook, 2019](#)). After supporting DNA replication, CycE is degraded by SCF in the S phase. CycA also promotes activation of the transcription factor NYF that further increase CycA. CycA promotes further cell growth in G2 phase. NYF also produces last cyclin, CycB, towards M phase. CycB helps segregate the newly copied DNA. CycB also activates CDC20 that degrades CycA, CycD and CycB partially, in M phase. This starts reactivation of CDH1, previously deactivated by CycE, to degrade CDC20 and CycB after DNA segregation at the end of M phase leading to the completion of cell division and

return of the system to its initial state.

3.2. Benchmark models and data for model development

First set of data was generated from [Abroudi et al. \(2017\)](#) ODE model of cell cycle with over 30 proteins, that contains the core cell cycle system in [Fig. 1](#), using the ODE equations given in Supplementary materials of [Abroudi et al. \(2017\)](#). Time-series of concentration levels of all 12 proteins shown in [Fig. 1](#) were collected resulting in 350 data points covering 4500 time units representing the span of cell cycle of 24 h. We used these continuous data to train linear and nonlinear whole system AANN as well as hybrid AANN (linear/nonlinear) to demonstrate that neural networks can be trained from data to produce continuous system dynamics similar to ODEs. With this data, we also trained individual protein models separately and assembled them to develop whole system AANN and hybrid (linear/nonlinear) models to test bottom-up modular approach to developing whole system models.

The second set of data was binary and generated from two sources. First is the [Alsharaiah \(2018\)](#) Boolean model of the core cell cycle network in [Fig. 1](#), from which binary activity of proteins was obtained. Due to the nature of Boolean models, these data cover the whole cell cycle in 12 time steps indicative of the points of change in the binary (on/off) state of protein(s) in the progression of cell cycle. To improve the resolution of the binary representation, we also developed a second binary data set by converting the continuous data from the above [Abroudi et al. \(2017\)](#) ODE model. These datasets were used to develop binary individual neuron models and binary AANN models. Linear and nonlinear models were coded in python and binary models were developed on Matlab ([Mathworks, USA](#)).

3.3. Neural network model development

A systematic and incremental approach was taken to develop neural network models to answer the series of questions raised earlier to help achieve the goal of the study. Accordingly, this section first provides a brief theoretical coverage of AANN and its training to provide the background. Then it focuses on how the method was applied to answer these questions that refer to the possibility of developing three categories of neural network model: i) Developing whole system AANN models. These are also used to test their capacity for self-learning and retaining temporal memory, (ii) Developing individual neuron models separately and their subsequent assembly into whole system AANN models and (iii) Developing hybrid AANN.

3.3.1. Auto-Associative neural networks (AANN) to represent protein networks

Although black box type large neural networks with multiple layers can be devised for molecular systems, we chose to avoid this approach in the interest of developing meaningful models that could also be simpler. To this end, we gained inspiration from the nature of molecular systems themselves for an appropriate network formalism. Accordingly, we sought a neural network formalism with 2 main features: (i) one-to-one mapping of its neurons to individual proteins and (ii) unique and specific interactions among neurons mapping to unique and specific molecular interactions. On this basis, we proposed Auto-Associative neural networks (AANN) to represent the system.

AANN in its generic form is known as Hopfield networks. In Hopfield networks, all neurons in the system interact with each other (full connectivity) with symmetric bidirectional connections (equal feedforward and feedback connections between neurons) and without self-feedback (entities do not self-regulate themselves), and the system as a whole unfolds in time ([Hopfield, 1982](#)). [Fig. 3a](#) shows a generic fully connected Hopfield network consisting of elements and their interactions. These networks traverse through a series of whole system states and converge to a steady stable state (fixed attractor) in asynchronous mode (only one neuron is activated at a time) ([Hopfield, 1982](#)) as this topology allows

minimisation of a Lyapunov type energy function ([Morales and Froese, 2019](#)). Molecular systems, however, display more complex and specific interaction patterns than Hopfield networks. Accordingly, they could display potentially different and even richer behaviours including fixed and limit cycle attractors. Fixed attractor is when the system reaches a fixed steady final state and limit cycle is when the system reaches a point where it cycles through few states in a loop continuously. Cell cycle itself can be considered a limit cycle where the system reaches its original state after each cell division.

Studies ([Zarco and Froese, 2018a&b](#)) have shown that fully connected networks even with nonsymmetric weights and self-feedback connections converge to stable states even in synchronous mode (all neurons are activated simultaneously). Using *C. elegans* connectome, they further demonstrated that networks could learn to self-optimise connections (weights) over time to find better final states (attractors). This may have evolutionary significance. Further, recent studies have revealed that networks even without full connectivity and non-symmetric bidirectional connections (as in [Fig. 3b](#)) and with or without self-feedback can produce realistic system dynamics and display a rich combination of fixed and limit cycle attractors ([Watson et al., 2011; Samarasinghe, 2023](#)). [Fig. 3b](#) is a modified version of Hopfield network with specified connections that can be activating or inhibitory. Interestingly, [Watson et al. \(2011\)](#) stated that self-interested agents spontaneously form distributed networks according to organisational principles familiar in neural networks. Thus, AANN appears able to model networked systems with unique feedforward and feedback connections and display system trajectories and stable states. Therefore, there is great potential for moulding AANN as in [Fig. 3b](#) to represent molecular networks that are also networked systems with specific feedforward and feedback connections, and in some cases with self-feedback where proteins self-regulate themselves. Importantly, AANN could represent agents (proteins) forming distributed networks that self-optimize their structure and connections for better efficiency and utility.

The above observations motivated us to propose the theoretical framework of neural networks in the form of AANN as in [Fig. 3b](#) to model molecular systems as it resembles generic protein interaction networks with specific positive and inhibitory interactions. We hypothesise that this neural network configurations and its attributes can represent the core cell cycle system in [Fig. 1](#). The form of the whole system AANN for our study then is exactly the same as [Fig. 1](#). Importantly, this format also allows us to exploit one to one mapping of neurons to proteins so that AANN can be a direct analogue of the molecular system. This realistic system representation can offer incredible advantages allowing us to gain realistic insights into the behaviour of the system and its emergent properties, study the effect of mutations/faults leading to diseases and find ways to intervene in the case of aberrations to restore the system to its natural state. In this study, we focus on systems dynamics (collective behaviour of all proteins in the system over time) only and we aim to study other emergent properties such as stability and resilience in a future study. However, we explore concepts of potential evolutionary learning and memory in these networks.

3.3.1.1. Learning in AANN.

AANN traverses multiple states through time until a steady state is reached. Correctly trained AANN of a molecular system therefore could capture the dynamics of its system states correctly in terms of concentration levels and timing of operation of all proteins in the system. Two main forms of learning in neural networks are supervised and unsupervised learning. Supervised learning involves target data that help determine the error of prediction for each neuron in the system at each model iteration and guide learning towards minimisation of overall system error. Unsupervised learning finds internal patterns in data without specific target data. AANN employ supervised learning. However, the training data they need are the whole system state (e.g., concentration of all proteins) at each time step such that

AANN uses the current system state as input to produce the next system state at the next time as the output. Inputs and outputs are the response of same entities over time. Therefore, for an AANN protein system, data for concentration of all proteins at various time points for the duration of the process being modelled are needed.

Specific supervised learning used in AANN is Hebbian learning that allows to capture simultaneous activity of connected neurons. For example, in the AANN shown in Fig. 3b that we will use to represent our protein system, according to Hebbian learning, if a neuron activates another, the connection between them will change according to the strength of the signal being communicated. If neuron j activates neuron i , the strength of the signal is the weighted input I_{ij} received by the activated neuron i . For multiple neurons feeding neuron i , the strength of the signal at time $t+1$ is represented by the weighted sum of inputs I_{ij} (u_i in Eq. 1)) coming from j connected neurons into neuron i at time t . The weighted sum is further processed by the chosen neuron activation function σ (sigmoid etc.) to produce the output v_i (Eq. (2)) representing the activation of neuron i at time $t+1$. In AANN, all neurons do this processing simultaneously to compute the output of all neurons at time $t+1$ as below.

$$u_i = \sum w_{ij} * I_{ij} \quad (1)$$

$$v_i = \sigma(u_i) \quad (2)$$

where w_{ij} represents the strength of connection or weight between neurons i and j . The error of prediction for neuron i , E_i , is calculated from the target output t_i and predicted output v_i

$$E_i = t_i - v_i \quad (3)$$

In order to use Hebbian idea in model training, original Hebbian ideas is slightly modified to assume that weight change w_{ij} between two neurons is proportional to the activating signal I_{ij} received by neuron i from neuron j and corresponding error E_i produced for the signal of the activated neuron i as

$$\Delta w_{ij} = \beta E_i I_{ij} \quad (4)$$

where β is the learning rate assumed to be constant between 0 and 1.

The input data I consists of a vector of concentrations of all proteins in the system at time t . This represents the level of regulators of each protein at time t . Target data t are the concentrations of the same proteins at the next time step revealing how they have changed as a result of interactions. Starting with the initial state of the system (e.g., initial protein concentrations at time $t = 0$), the network repeatedly processes neural activity (Eqs. (1)–(3)) and adjusts connection weights between proteins (Eq. (4)) over training iterations until the system reaches the goal of predicting the next state accurately to match the data or there is no change in the predictions with further iterations. Mean Square Error (MSE) is used as the loss or error function in training and R^2 (coefficient determination) is used as the metric for prediction accuracy. Test data is used for an independent measure of model performance based on MSE and R^2 .

Using this AANN foundation, we elaborate below how it was applied to address our original questions. As discussed in the Goal section, we first explore whole system AANN models in terms of their ability to discover unknown protein systems (section 3.3.1.1), model known protein systems (section 3.3.1.2), their ability to self-evolve (section 3.3.1.3) and hold memory (section 3.3.1.4). Then we turn our attention to individual neuron models and their assembly into AANN in section 3.3.2. Here, continuous (linear and sigmoid) individual neurons are trained in Section 3.3.2.1. Binary individual models are trained in Section 3.3.2.2 with binary data from two sources – the Boolean model and ODE data converted to binary data as discussed earlier. Then section 3.3.3 is devoted to developing whole system AANN by assembling individual neuron models. Specifically, Section 3.3.3.1 presents the

assembly of linear, nonlinear and hybrid linear/nonlinear continuous AANN and section 3.3.3.2 presents the assembly of binary neuron models into binary AANN. Finally, hybrid binary/nonlinear (discrete/continuous) AANN system for time scale separation is presented in section 3.3.3.3.

3.3.1.2. Can a neural network discover an unknown protein interaction system from data and represent its system dynamics? The generic question being addressed here is whether a neural network can discover a protein interaction system from data meaningfully. In neural network terms, we ask if an AANN with *a priori* unspecified (arbitrary) connection pattern be trained with data to capture the correct protein interactions in a real system and correctly represent its system dynamics. In practical terms, this tests if the AANN can learn the pattern of connections existing in the real system from data alone and predict accurate protein dynamics. If it does, then AANN is capable of finding unknown protein interactions by making strong internal representations from data as it learns to represent the system dynamics. On the other hand, if it does produce correct system behaviour but incorrect connection pattern, then it shows that there is more than one connection configuration that produces the expected system behaviour. This itself would be an interesting observation and raise interesting new questions about evolution of these networks on long evolutionary time scales. It will also caution against expecting machine learning models to be true representations of reality and provide true insights into an unknown system. In contrast, if both connection pattern and predicted system behaviour are incorrect, AANN is unable to capture both from data.

In this case, the model has 12 neurons representing 12 proteins (as in Fig. 1) but connections between neurons are arbitrary and unspecified, i.e., all proteins can interact with each other as in Fig. 3a. Weights (w_{ij}) represent the strength of each of these connections. Nonlinear (sigmoid neurons) are used in this first test case as they can represent linear or nonlinear responses. The data used are continuous time-series protein concentration levels from the Abroudi et al. (2017) ODE model. Data set is split into 90 % training and 10 % test data. Starting with random initial weight values, AANN uses Hebbian learning (Eqs. (1)–(4)) to modify connection weights during training until the next predicted system state (concentration of all proteins at the next time step) is as close to the real system state provided by the data as possible. The inputs are the state of all proteins at time t and outputs are the state of all proteins at time $t+1$. For each protein, these inputs represent the concentration of its regulators (activators and degraders), and output is its concentration due to regulation by the inputs. There were 350 equal time steps representing 24 h to complete cell cycle.

3.3.1.3. Can AANN learn to discover a known protein system and produce accurate behaviour from data? If an AANN with arbitrary connections cannot discover the protein interaction system from data, we train an AANN with connections as in the real system where specific proteins interact with each other as in Fig. 1. Here we explore two types of neuron function-linear and sigmoid. Linear is used to assess the possibility of model simplification. This is especially useful in data limited situations. We also test hybrid linear and sigmoid functions in the AANN if linear function turns out to be satisfactory for some but not all. This would also constitute model simplification from all nonlinear neuron AANN model and capture differences in protein interactions where some element reactions are simpler (linear) and some are nonlinear. The same data and training as above are used. Results would reveal if a trained AANN with connections as in the real system can reveal correct network protein dynamics. If it can produce realistic behaviour, AANN can make strong internal representations from training data. It will also show if the hybrid (linear/nonlinear) formalism benefit accurate representation of the system, in particular, as experimental data are still scarce for characterising the whole spectrum of behaviour of proteins.

3.3.1.4. Can a neural network self-evolve over time on its own to mimic a protein system? Self-evolved networks may optimise themselves through trial and error guided by self-correction based on their own system generated data/outputs. We tested if the design of molecular systems potentially allows or uses these concepts of evolutionary learning. This is a stringent test on an AANN with *a priori* specified molecular interactions (as in Section 3.3.1.2) asking if it could generate the next system state solely from the current system state generated by the network without relying on any external input data at each time step. It uses externally provided target data only to compute error that guides the direction of training in the form of reinforcement learning. If it can produce realistic behaviour, AANN can make strong internal representations from its own behaviour and adapt as it learns to represent the system, resembling what may happen in the evolutionary formation of molecular networks through natural selection. We used the same target data used in the previous two experiments to train the AANN with molecular interactions of the core-cell cycle system.

3.3.1.5. Do protein networks have memory? Biological systems tend to form temporal memory. Recurrent neural networks with self-feedback can retain past memory through the integration of past outputs as additional inputs in deciding current state. To test if the design of molecular systems could support memory, our AANN with the specific molecular interactions of cell cycle was trained with system generated output for each protein feeding back as input to the same protein at the next time step along with the externally provided standard input from its activators and degraders. Same data as in previous cases were used here. If self-feedback improves prediction of systems behaviour, then protein interaction systems can retain memory of their past.

3.3.2. Can individual neurons model individual proteins? Linear, nonlinear and binary neuron models

As highlighted before, another beneficial strand of inquiry is if individual neurons can be trained to capture individual protein dynamics. The value of this is multiple fold. Not all proteins are equal, and some are critical for the functioning of a system and disease development or mitigation. Therefore, characterising the behaviour of individual proteins can help gain clarity on their specific activity and support any decision making associated with them. Further, our knowledge of molecular systems is evolving rapidly and new proteins acting in various molecular systems are continually being discovered; therefore, effective models trained to accurately represent individual proteins can be valuable to assimilate the new knowledge being generated into existing models. Further, these individual models themselves can be assembled into systems models (AANNs) to study whole system behaviour. As such, individual models can provide a modular bottom-up approach to develop models of large molecular networks. With this aim in mind, in this section, we explore the potential of individual neurons to model individual proteins separately and then assemble the neuron models into AANN systems models. Particular models explored are linear, nonlinear (sigmoid) and binary (perceptron) neurons. If binary models are successful, it serves two purposes: it allows simplification of continuous neural models to binary models and also provides a quantitative binary neuron model alternative to qualitative (IF-THEN rules based) Boolean models and importantly allows simulation of accurate system dynamics by eliminating the bottleneck of spurious trajectories and attractors experienced by Boolean models due to lack of temporal realism.

3.3.2.1. Continuous individual neuron models representing individual proteins. Here we train individual neurons preserving their connections found in the whole system in Fig. 1. For convenience, Table S1 in supplementary materials shows each protein and its regulators for all proteins in the system. Positive regulators are denoted by (+) and negative regulators by (-) in Table S1. In developing individual neuron models, these are the inputs into the neuron. Fig. 4 shows one neuron with

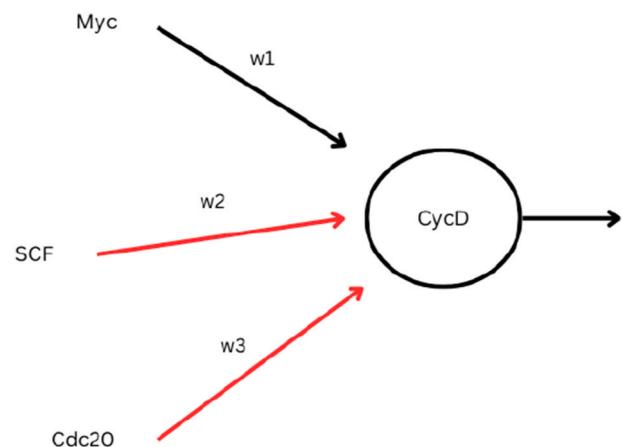


Fig. 4. Schematic of an individual neuron model of a protein. The neuron representing CycD protein is shown with producer Myc with a positive interaction (black arrow) and degraders SCF and CDC20 with inhibitory interactions (red arrows).

connections to its regulators (activation– black arrow and inhibition–red arrows). Others are developed in the same way. The aim here is model simplicity and accuracy; therefore, for continuous modelling, linear neurons are first attempted followed by nonlinear (sigmoid) neuron models to represent continuous protein dynamics. We train linear and nonlinear neurons with training data collected from Abroudi et al. (2017) ODE model. Each model was trained separately with gradient descent learning (Eqs. (5)–(8)). Model predicted behaviour is tested against data using MSE and R^2 as accuracy metrics.

Individual neural computations are similar to those in the AANN. Linear or sigmoid neuron computes the weighted sum u of inputs (I_i) it receives via connection w_i (Eq. (5)) and produces neuron output (v) (Eq. (6)). σ is sigmoid or linear function. Error E is the difference between the predicted and target (Eq. (7))

$$u = \sum w_i * I_i \quad (5)$$

$$v = \sigma(u) \quad (6)$$

$$E = t - v \quad (7)$$

Weight w_i undergoes change according to gradient of error E with respect to w_i which simplifies to

$$\Delta w_i = \beta E I_i \quad (8)$$

Learning rate β (0,1) is adjusted until the best outcome is attained for training and testing datasets.

3.3.2.2. Binary neuron models representing individual neurons. Aim with these is to investigate if neural equivalent of Boolean like discrete models can be developed for proteins from data. These binary neuron models can do away with IF-THEN rules altogether and additionally provide quantitative discrete models instead of qualitative discrete models of proteins as in Boolean models. For the same reason of model simplicity without rules, binary neuron models could help develop models faster. Importantly, this helps explore the potential of transforming Boolean systems models into binary neural network systems such as AANN through the assembly of individual binary models as explored later in this paper. We propose binary neurons (perceptrons) for this purpose. Individual perceptron models are developed for all proteins in the system using two sets of binary data as discussed below.

3.3.2.2.1. Individual binary (perceptron) neuron models from boolean model generated data. Here we develop individual perceptron models to represent individual binary protein dynamics. Perceptrons are binary neurons producing binary (0 or 1) output from real or binary input data

using a binary (threshold) function. These models test the ability of perceptrons to capture the conditions governing on/off states of neurons. Initially, models are trained with the Alsharaiah (2018) Boolean model derived data and model performance is compared against data. Alsharaiah (2018) Boolean model depicts core cell cycle shown in Fig. 1 using Boolean rules for individual proteins according to their relations shown in Table S1. Neuron input variables for these models were the same as in the corresponding continuous cases above, but their values are binary indicating whether the inputs are active or not; similarly, the output is binary indicating whether the inputs have collectively activated the neuron/protein or not. An individual perceptron training follows Hebbian rule as in Eqs. (1)–(4) with a binary neuron transfer function and binary inputs and output. Neuron responses are compared against data to assess individual protein behaviour.

3.3.2.2.2. Perceptron models from binary data transformed from continuous ODE model data. As the Boolean model data have only 12 time steps, we refine the binary neuron models with more data. Here we develop individual perceptron models from the larger binary dataset for 350 time steps obtained from binarising the ODE model data. To prepare the dataset, we use the idea that proteins are considered active when they reach half their maximum concentration. Accordingly, concentration values below the half-way mark were assigned a value of 0 and those above that were assigned a value of 1. Individual model configurations and training are similar to the binary models trained with Boolean model data in the previous section. The only difference is that there is more data providing granular view of the protein operation. Neuron responses are compared against data to assess individual protein behaviour.

3.3.3. Developing whole system neural networks by assembling individual neuron models

Individual neurons developed above are assembled into linear, nonlinear, binary and hybrid AANNs. The reason for testing this variety is to discover the simplest possible systems model that captures the correct system behaviour while depicting unique behaviour of individual proteins. In the binary case, simplification is achieved by focusing on just the active or inactive states of proteins; this format could suit the most commonly available binary experimental data for proteins. With hybrid binary/continuous case, further simplification can be achieved through timescale (slow/fast) separation of reactions. This section explores these possibilities.

3.3.3.1. Bottom-up construction of continuous whole system models – assembly of linear, nonlinear and hybrid continuous AANN. Depending on the success of the individual linear and nonlinear neuron models developed in Sections 3.3.2.1, they are assembled into continuous AANN systems models. Further, hybrid (linear/sigmoid) AANN are developed by combining well performing linear neuron models with sigmoid neuron models. Since connections in the system are exactly preserved in the individual neurons, when combined, they should encompass the whole system interactions. Therefore, individual models are simply assembled and tested for whole system behaviour. When assembled, properly trained individual neurons are expected to produce correct whole system protein dynamics without additional training.

3.3.3.2. Bottom-up construction of binary whole systems- assembly of binary neuron models into AANN. In the binary case, individual perceptron models developed in Section 3.3.2.2 are assembled into perceptron AANN system models for both cases of binary data. Correctly tuned individual models are expected produce correct activation (on/off) of proteins in the system similar to the individual models and thus show their ability to transform qualitative Boolean models into quantitative perceptron (discrete) AANN systems models.

3.3.3.3. Hybrid discrete/continuous AANN system for time scale separation. Another point investigated is the potential of binary and continuous neuron models to help exploit time scale separation in molecular systems. With respect to cell cycle, processes such as synthesis of the four cyclins can take hours, whereas their activations and inhibitions can be very fast in nano to milli second time scale. We test this idea using hybrid AANN, where perceptrons represent fast processes and nonlinear neurons represent slow protein synthesis processes. This was developed by assembling nonlinear neuron models of the four cyclins and perceptron models of the rest of the proteins of cell cycle into a single hybrid AANN system as described below.

In reality, hybrid can even include linear neurons to form hybrid models of binary/linear/nonlinear forms if some slow activations can be better represented linearly, but for the sake of brevity, we explore only binary/nonlinear option. Important point here is that hybrid models are developed within the same (neural) modelling paradigm. In developing the proposed hybrid model, an issue arises from the fact that perceptron output is binary and thus all fast-activating proteins produce binary outputs. These proteins (as binary inputs) participate in regulating slow activating proteins represented by nonlinear neurons in the hybrid AANN. However, our previously developed individual nonlinear neuron models have real-valued (non-binary) inputs. Further, our perceptrons have been trained with binary inputs. In the proposed hybrid model, some of these inputs come from the nonlinear neurons that produce real-valued outputs. Therefore, our originally trained perceptrons and nonlinear neurons cannot be assembled directly. For this reason, we retrained the nonlinear neurons to receive binary [0,1] inputs when they are from fast activators along with any real-valued inputs coming from slow activating proteins according to Fig. 1. Similarly, we retrained the perceptrons to receive real-valued inputs coming from nonlinear neurons along with any binary inputs coming from fast activating proteins. Thus, both perceptrons and nonlinear neurons had binary and real valued inputs according to the type of protein they were connected to, whether fast acting or slow acting. Trained individual models are then assembled into a hybrid (perceptron/sigmoid) AANN model. This may not produce the smooth continuous dynamics for the slow activating proteins as in the continuous ODE model, or as in our nonlinear neuron AANN models with real-valued inputs developed above, due to the presence of binary inputs but the assumption is that they would approximate these patterns well. Capturing correct trends and timing of protein behaviour have received greater emphasis than predicting precise protein concentrations in current models of molecular networks due to various uncertainties including imprecise knowledge of molecular systems, limited availability of protein data to cover the whole spectrum of system behaviour and noise in data. Therefore, in situations where these conditions are prominent, the proposed hybrid AANN may be attractive to study system dynamics. If the hybrid model is correct, it is expected that all perceptrons will reveal fast activations in discrete (binary) form in correct temporal order and all sigmoid neurons will reveal slow activations approximating the continuous concentration patterns of the four cyclins.

4. Results

This section addresses the key questions of this study in the order they were raised in the Goal section. It is worth emphasising again that the main aim of modelling is first and foremost to capture the correct dynamics of the core cell cycle system (i.e., protein behaviour over time) in terms of trends and timing of activation/deactivation of proteins. It was expected that models will approximate the actual protein concentration levels as well. These aspects are assessed by comparing the performance of relevant models against target data. Efficacy of AANN learning was measured by MSE and R^2 achieved by models on training data and independent test data. Low MSE close to 0 and high R^2 close to 1.0 means that the models have captured the continuous protein dynamics to a very high level of accuracy. This means that the models

accurately predict concentration of each protein at each time step of iteration until the end of the simulation period. From a biological perspective, high accuracy means that models have correctly captured the pattern and effect of interaction among proteins as well as the activation time of individual proteins and their duration of activation until the completion of cell cycle. Learning rate of 0.001 was found to be the best for all models and training time for an AANN model was less than a minute on a laptop with 12th Gen Intel (R) Core™ i7-12700K 3.6 GHz processor with 128 Gb RAM. The fast training time was partly due to the sparse connection matrix for this system with only 33 protein interactions (which is only 25 % of a fully connected system with all proteins interacting with each other). Another reason is the small training dataset size representing 350 time points. We expect that training time for larger models will scale well. This is because connection matrix can be expected to be sparse even for larger systems and dataset size could not be any larger than 350 time points.

4.1. Whole system neural network (AANN) predicts highly accurate systems dynamics but fails to discover unknown protein interaction system from data

The whole system AANN model with sigmoid neuron functions was trained *without a priori* specified interactions among proteins and the aim was to test if the model can correctly identify the correct protein interaction patterns and produce accurate protein behaviour over time. Models were trained successfully. The actual and predicted protein dynamics of the trained AANN are shown for training data in Fig. 5a and b for the two early (G1) cyclins (CycD and CycE) and two later cyclins (CycA and CycB), respectively, along with their activators and inhibitors. The predicted patterns are shown with dashed lines and target patterns with solid lines of the same colour. Mean square Error (MSE) quickly drops during training as shown in Fig. 1c. These results show that the network has captured the nonlinear trends and timing of regulation of cyclins as well as their regulator proteins remarkably well. It has also predicted individual cyclin and regulator protein concentrations with high accuracy as indicated by the closeness between actual and predicted values in Fig. 5a and b, Mean Square Error (MSE) training plots in Fig. 5c and final MSE for the system of 0.0048. Analogous cyclin and regulator dynamics for the test data are shown in Fig. 6 which shows that the model generalises well to unseen data and thus has captured the generic pattern of cyclin regulation as expressed in the data. This indicates that whole system AANN models that accurately capture protein dynamics (trends, timing of activation and concentration of proteins) can be developed from protein data. Very low error (MSE) means that the Hebbian Learning used to train AANN has been able to capture the trends in the complex interaction patterns among proteins incrementally

at each time step of simulation.

Table S2 in Supplementary Materials shows the final weights from the trained whole system AANN. The table reveals that the AANN has exploited a rich array of all possible connections in traversing the data to capture the trends in them. The actual connections that AANN was expected to find are bolded in the table. This shows that the model has not captured the expected connection pattern. How has it then produced correct protein dynamics? This reveals a very important finding that there can exist a number of potential connection patterns that can produce real system behaviour correctly. One of these would be the correct connection configuration but, in most cases, we would not be able to confidently identify when the correct configuration shows up without considerable domain knowledge and interpretation. It could also point to the possibility that evolution fine-tunes these potential patterns to reach the most desirable protein interactions (connections) pattern. Thus, when connections are not specified *a priori* and network is allowed to discover the connections freely, model results can be highly accurate, but the connections may not display reality. Therefore, AANN cannot identify specific interactions from data, and they must be specified before training. This also highlights the danger in using machine learning to explore patterns in data hoping to gain insights into the system from models, which in this example case is highly misleading. From a modelling perspective, this reveals the importance of prior knowledge of the system in the development of models of protein dynamics. This is a new finding that could benefit future modelling. The findings from this model compelled us to design AANN with the specific connection pattern found in the real system, which is explored in the next section.

4.2. Whole system neural network (AANN) models with *a priori* specified connections capture systems dynamics

4.2.1. AANN with linear neuron functions and *a priori* specified connections correctly captures the dynamics of half the proteins in the system

As whole system AANN with *a priori* unspecified connections could not find the correct protein interaction patterns from data, we developed AANN with *a priori* specified connections as in the real system (Fig. 1) and trained first with linear neurons to assess if the simplicity of linear neurons can approximate protein behaviour in the system. Since both training and testing results were remarkably similar, for clarity, results for all data are presented here and for subsequent models. Results in Fig. 7 show model performance against data for the four cyclins along with their regulators. The training MSE for the whole system was 0.122 and Table 1 shows the prediction R² for all the proteins in the system. These results show that the model can capture the dynamics (trend, timing and concentration) of nearly half the proteins including 3 of the 4

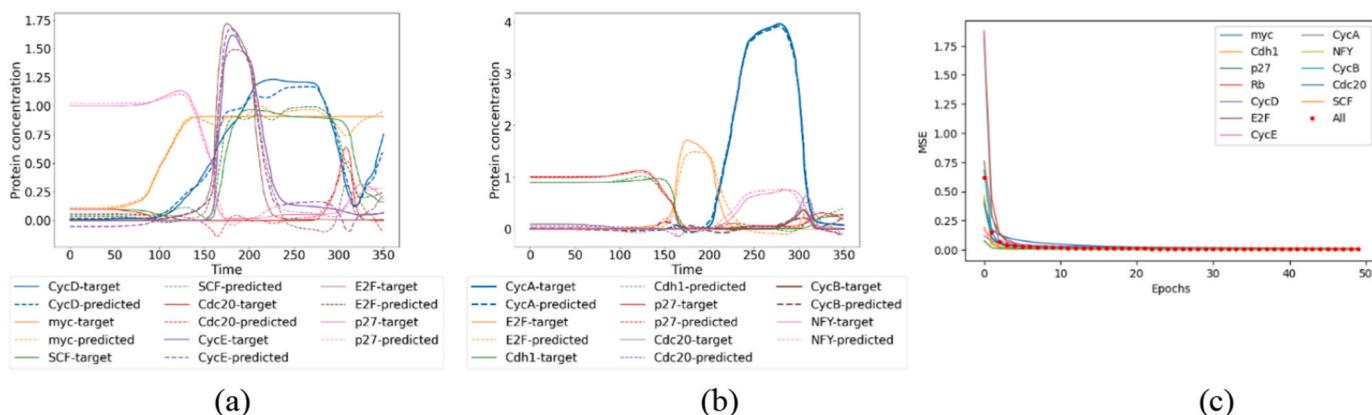


Fig. 5. Performance of whole system AANN with *a priori* unspecified connections between neurons with sigmoid activation function, on training data. a) Predicted and actual concentration of the two early (G1) cell cycle cyclins (CycD and CycE) and their regulators over the cell cycle period. b) Predicted and actual concentration of the two later cyclins (CycA and CycB) and their regulators over the cell cycle period. c) Mean Square Error during training.

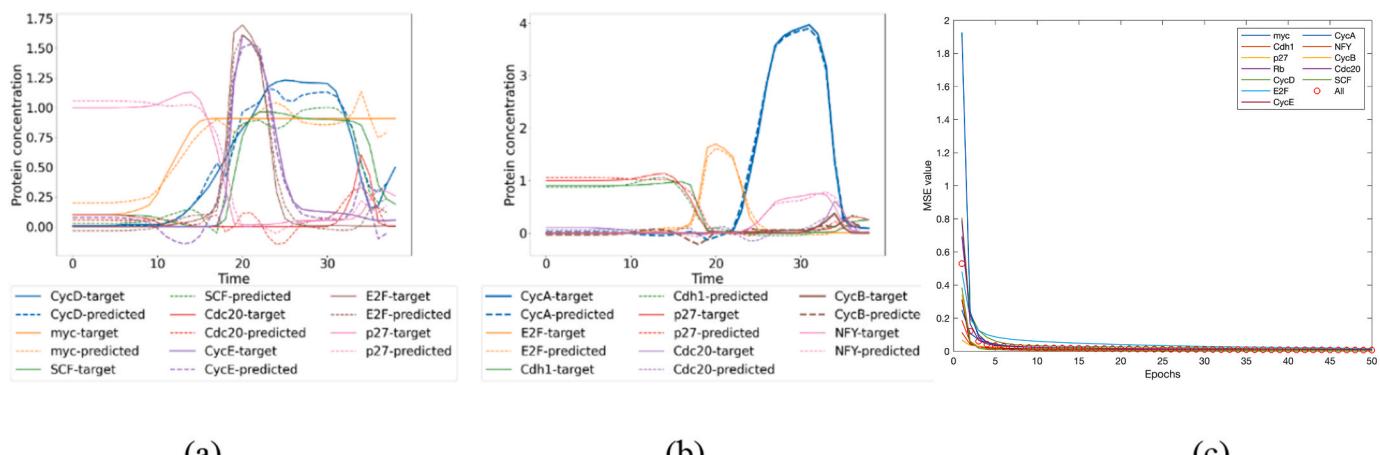


Fig. 6. Performance of whole system AANN with *a priori* unspecified connections between neurons with sigmoid activation function, on test data. a) Predicted and actual concentration of the two early (G1) cell cycle cyclins (CycD and CycE) and their regulators over the cell cycle period. b) Predicted and actual concentration of the two later cyclins (CycA and CycB) and their regulators over the cell cycle period. c) Mean Square Error during training.

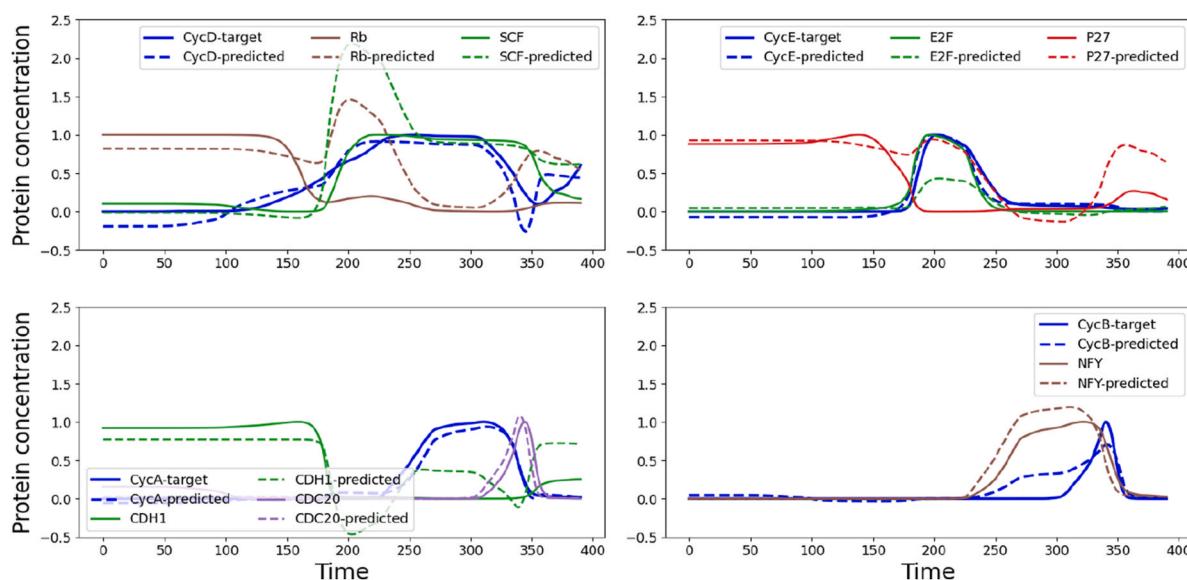


Fig. 7. Whole system AANN with *a priori* specified connections as in the real system and linear neuron activation functions.

Table 1

R^2 for prediction of behaviour of all proteins in the system modelled by AANN with linear activation functions.

Cdh1	p27	Rb	CycD	E2F	CycE	CycA	NFY	CycB	Cdc20	SCF
0.610	0.104	-0.303	0.846	0.577	0.939	0.954	0.878	0.573	0.812	-0.205

cyclins in the system (CycD, CycE, CycA, NFY, Cdc20) very well (R^2 from 0.8 to 0.95) indicating that good proportion of proteins in this protein-protein interaction (PPI) network is linear in interaction with their activators/inhibitors and this could hold true for other PPI networks as well. This is also a new finding that could benefit future modelling. Those proteins that were not modelled well with a linear function indicate that some protein dynamics are nonlinear: these proteins were Cdh1, p27, Rb, E2F, CycB and SCF and some of these proteins have the most complex interactions in the system. For example, Rb keeps suppressing E2F which is the transcription factor for CycE and A, and the release of E2F from Rb is a two-step process involving two phosphorylation events involving CycD. As another example, p27 is the inhibitor of Cyclins E, A and B but p27 energises CycD when complexing with it that leads to suppression of inhibitory activity of p27 on these other cyclins.

For these poorly modelled proteins, only three models (p27, Rb and SCF) show very poor results and others (E2F, Cdh1 and CycB) seem to show a slight vertical shift from the expected behaviour or inadequate peaks while maintaining the general trend. These results indicate the AANN needs to have enough capacity to model protein systems. If nonlinearity is present in the system, but AANN is only equipped with linear neurons, it fails to represent those proteins with nonlinear behaviour. This required us to develop AANN model with nonlinear neuron processing as explored in the next section.

4.2.2. AANN with sigmoid functions and *a priori* specified connections captures all protein dynamics in the system

Since linear neurons could not capture the dynamics of some proteins, the whole system AANN was trained with all sigmoid neuron

functions. Prediction plots are shown in Fig. 8 and training MSE plots for the system is shown in Fig. S1 in Supplementary Materials. MSE for the whole system has been reduced to 0.0047 compared to all linear neuron case above with MSE of 0.122. The results in Fig. 8 showing the model performance on data for the four cyclins and their regulators reveal that all elements can be modelled using sigmoid function. The three proteins (p27, Rb and SCF) that were very poorly modelled by linear function have responded extremely well to sigmoid with R^2 values exceeding 0.96 (Table 2). The other three proteins ((E2F, Cdh1 and CycB) that were predicted moderately well by linear function have also received significant improvement with sigmoid with new R^2 values ranging from 0.85 to 0.98. Those elements that modelled well with linear neurons have received a marginal improvement with sigmoid function. Thus, AANN with one-to-one mapping of proteins to neurons with sigmoid functions can well represent protein networks. This is a main outcome from the whole system AANN modelling of protein interaction systems. From a modelling perspective, AANN with capacity to model nonlinearity through nonlinear neurons can incrementally learn to find the correct nonlinear protein interaction patterns. Further, results indicate that AANN with nonlinear neurons can also well represent, or even slightly improve, the protein behaviour already well captured by linear neurons. Comparison of the two AANN models, one with linear and the other with sigmoid functions, reveals that some protein behaviour can be captured by linear neurons and others require nonlinear neurons. This allowed us to explore the development of hybrid linear/nonlinear models for model simplification as demonstrated in the next section.

4.3. Simplified hybrid continuous AANN with a mixture of linear and sigmoid functions demonstrate accurate system behaviour

Model simplification is desirable in data limiting situations. Hybrid AANN with a combination of linear and nonlinear activation functions can help achieve this simply within the same model. Combining those proteins that did well with linear functions (CycD, CycE, CycA, NFY, Cdc20) and those that did well with sigmoid functions (Cdh1, p27, Rb, E2F, CycB and SCF) in an AANN produced similar behaviour to original results in the respective systems shown in Figs. 7 and 8. Further training of the hybrid system even further improved results for the four cyclins and their regulators, in particular for Cdc20 and NFY, as shown in Fig. 9 and Table 3. Thus, simplified hybrid AANN with one-to-one mapping of neurons to proteins and linear and nonlinear neurons to represent linear

and nonlinear interactions between proteins can be developed for representing protein interaction networks. This is another main outcome of the whole system AANN modelling of protein interaction systems. It also indicates that AANN allows model simplification by choosing the neuron functions that best represent the level of complexity or characteristic of specific protein interactions. Our results so far from the AANN model experiments with different neuron activation functions revealed that, with specified connections, AANN can represent the systems dynamics accurately. Therefore, in the next section, we explore other attributes such as self-learning and memory capacity of AANN.

4.4. Whole system AANN cannot self-correct on its own to mimic a protein system

The question addressed here is whether AANN can self-correct over time based on its own generated outputs without relying on external data. When the sigmoid AANN was trained solely on the data generated from the AANN itself in the previous step as input in the current step, and using externally provided target protein data only as a guide in reinforcement learning (for calculating prediction error for pointing in the right direction), the AANN could not capture protein dynamics in the system, as shown by the poor behaviour of all proteins in the system in Fig. S2 in Supplementary Materials. This is because the prediction error in each time step accrues over time rendering system generated data unreliable as input after some time. Training a network with self-generated data is an extreme test of neural works; therefore, self-learning without proper input data is an extreme challenge for neural networks in general, and our AANN has also experienced this challenge. This indicates that if evolution uses similar ideas as AANN and reinforcement learning in constructing and evolving protein networks, it cannot reach the desired outcome by only using its internally generated outputs but needs reliable information about intermediate protein states. This could be an interesting avenue to explore in future. One option would be to incorporate temporal nature of data, for example, by using moving averages of system generated inputs that can embed past states in the current input, or explicitly integrate specific recurrent structures that can hold memory of intermediate states into the network architecture. This also connects to our next exploration of AANN's capacity to hold past memory.

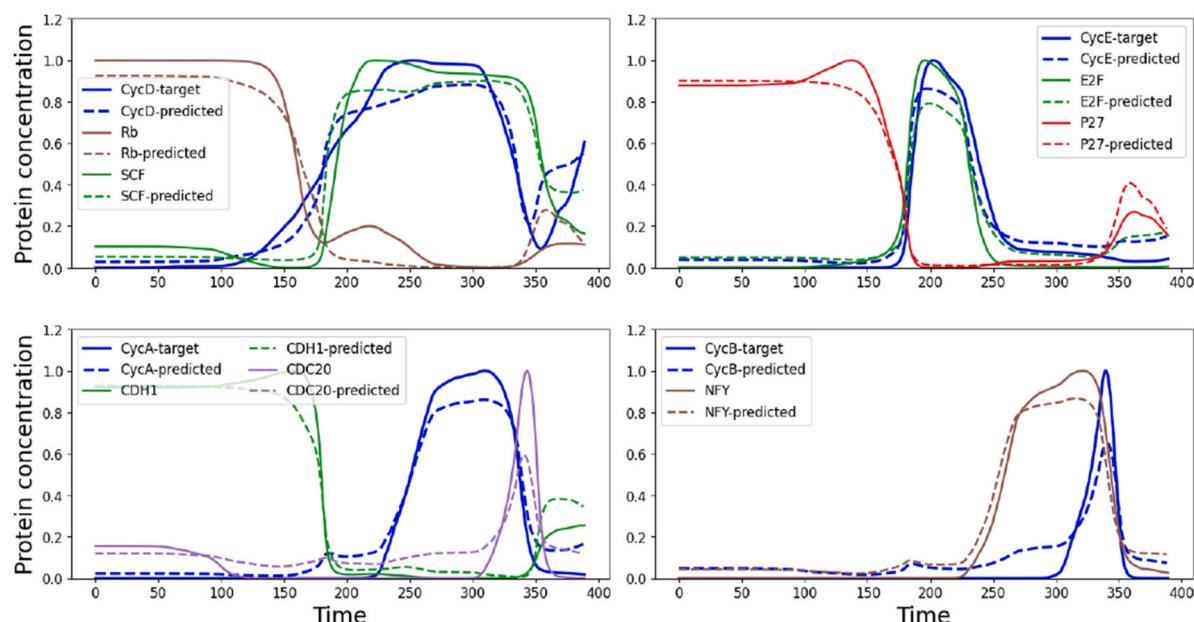


Fig. 8. Whole system AANN with *a priori* specified connections as in the real system and nonlinear (sigmoid) neuron activation functions.

Table 2

R^2 for prediction of behaviour of all proteins in the system modelled by AANN with nonlinear (sigmoid) neuron activation functions.

Cdh1	p27	Rb	CycD	E2F	CycE	CycA	NFY	CycB	Cdc20	SCF
0.983	0.989	0.965	0.936	0.934	0.966	0.976	0.981	0.853	0.862	0.97

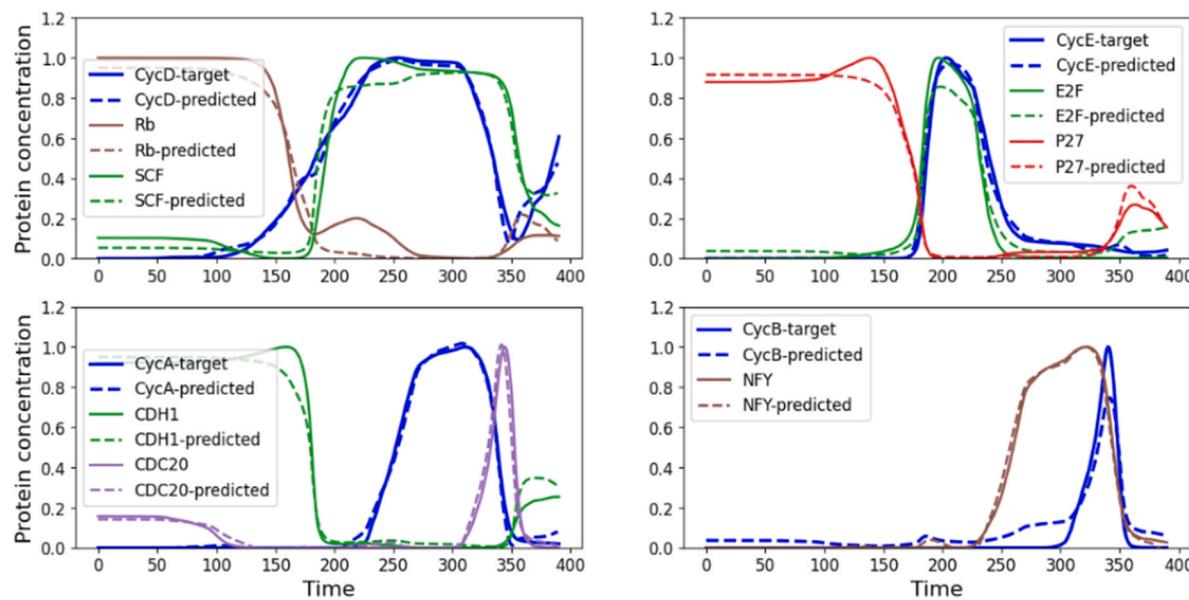


Fig. 9. Hybrid whole system AANN with *a priori* specified connections as in the real system and a combination of linear and nonlinear (sigmoid) neuron activation functions to represent corresponding proteins behaviour.

Table 3

R^2 for the prediction of behaviour of all proteins in the system modelled by hybrid AANN with a combination of linear and nonlinear (sigmoid) neuron activation functions.

Cdh1	p27	Rb	CycD	E2F	CycE	CycA	NFY	CycB	Cdc20	SCF
0.983	0.989	0.965	0.965	0.987	0.934	0.995	0.997	0.853	0.976	0.97

4.5. Recurrent AANN shows past memory of system state in capturing whole system dynamics

In this experiment, we explored if protein systems can contain memory of their unfoldment. In this case, each protein in the AANN receives their regular inputs from the external dataset and an additional input, which is its AANN generated output in the previous step (thus increasing the total inputs in the system to 24). Weight matrix of AANN is accordingly larger to depict the weights associated with 24 inputs. Interestingly, recurrent AANN model with just linear functions for all neurons can capture the whole system dynamics with near perfect agreement as shown for the individual proteins in the system in Fig. 10. These predictions are superior to those from all the previous AANN models (Figs. 7–9) (R^2 above 0.98 for all proteins). This is very interesting in terms of accurately predicting the next outcome. Closer inspection of the AANN weight matrix shown in Table S2 in Supplementary Materials revealed that the model had stronger weights on recurrent connections (those along the diagonal of the second half of the weight matrix). This indicates that protein time-series data display strong lag-1 linear correlations where previous time step can strongly influence the outcome at the next step. This could also be an interesting avenue to explore further in future. With this investigation, we have completed the whole system AANN models, and now turn our attention to the exploration of whether individual proteins can be modelled by individual neurons and whether these models can be assembled into a system model representing the protein network.

4.6. Individual neuron models can capture single protein dynamics from data

4.6.1. Linear neurons modelling individual proteins can represent a substantial proportion of proteins in the network

Individual proteins models can be useful in limited data contexts and when understanding the dynamics of individual proteins is important, such as crucial proteins in a system that can disrupt the system or cause diseases. Further, they can afford modular design of protein networks through assembly of individual protein models. Results for training individual neurons with linear activation function to represent a single protein are shown in Fig. 11 and R^2 for the prediction of individual protein dynamics are shown in Table 4. Corresponding MSE graphs showing low errors are presented in Fig. S3 in Supplementary Materials. Fig. 11 and Table 4 show that over 70 % of proteins are fairly accurately represented and it is only 3 proteins (Cdh1, p27, Rb) that linear neurons had difficulty modelling, with Cdh1 still showing relatively good R^2 value of 0.77. These results indicate that the performance of individually trained linear neurons is better than that of whole system linear AANN where six proteins (50 % of the system) including the three above were poorly modelled. Therefore, training proteins separately can produce better results than training all proteins together as a whole system. The results highlight again the possibility that the dynamics of a substantial number of individual proteins in a PPI system could be linear with respect to their regulators and could be represented by individual neurons with linear activation function. This ability of a neuron to represent

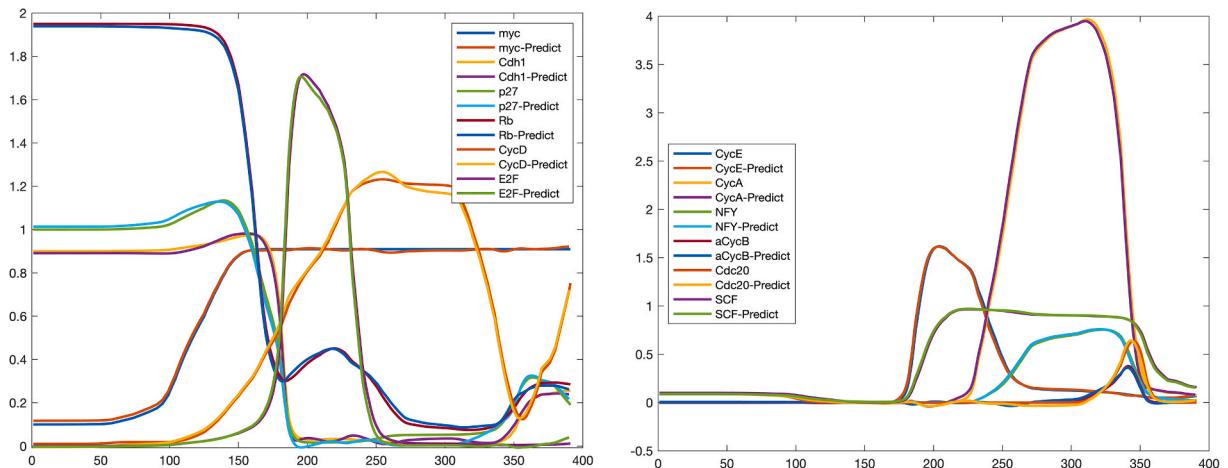


Fig. 10. Output from whole system Recurrent AANN with actual inputs and previous output from each protein feeding back to itself as input at the current time step superimposed on target data.

Table 4

R^2 for the prediction from individual neuron models with linear activation function to represent individual proteins.

Cdh1	p27	Rb	CycD	E2F	CycE	CycA	NFY	CycB	Cdc20	SCF
0.77	-0.22	-0.667	0.936	0.896	0.964	0.972	0.956	0.814	0.856	0.845

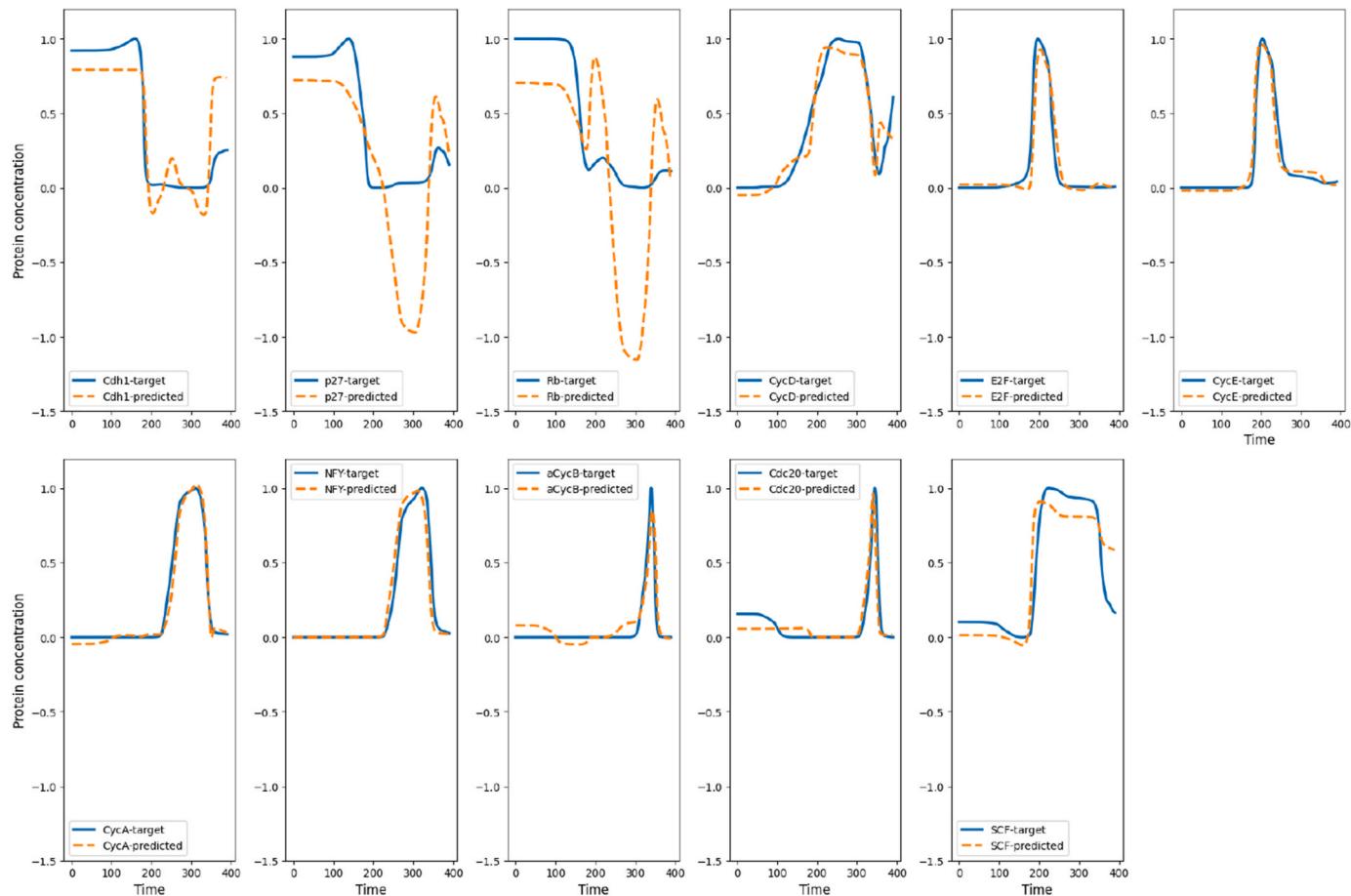


Fig. 11. Performance of individual protein neuron models with linear activation function. Most proteins are well represented by linear neuron models.

an individual protein is quite promising because it strengthens the case of proteins as computing units similar to neurons. From a modelling perspective, results indicate that individual neurons can learn to better represent specific neuron behaviour without the constraint of having to optimise a number of neurons simultaneously as in the AANN. Since linear neurons could not model all proteins, in the next section, we develop nonlinear individual neuron models of proteins.

4.6.2. Individual proteins modelled with sigmoid neurons can represent all proteins in the network

Results for training individual neurons with sigmoid activation function to represent single protein dynamics are shown in Fig. 12 and R^2 for the prediction of individual protein dynamics are shown in Table 5. Corresponding MSE graphs showing even lower errors than those for above individual linear neuron models are presented in Fig. S4 in Supplementary Material. Fig. 12 and Table 5 show that all proteins have been well modelled by sigmoid neurons while the three that were

challenging for the linear neuron have been remarkably improved. Some discrepancy is seen at the tail end of data which was evident in the sigmoid results for the whole system AANN as well (see Fig. 8). These are anomalies in the data due to some discrepancy between ODE model and real cell cycle at the end regions. Results show that here too protein behaviour from individual sigmoid models is better than when the sigmoid activation functions were used in the whole system AANN model (Fig. 8) where prediction accuracy is slightly smaller for all proteins. Therefore, training proteins separately with sigmoid function can produce better results than training all proteins together as a whole system. Now that we have success with individual linear neuron models for some proteins and sigmoid neurons for all proteins, this offers the opportunity to assemble individual sigmoid neuron models into AANN system models as well as develop hybrid linear/nonlinear AANN models. We present these results in the next section.

Table 5

R^2 for the prediction from individual neuron models with sigmoid activation function to represent individual proteins.

Cdh1	p27	Rb	CycD	E2F	CycE	CycA	NFY	CycB	Cdc20	SCF
0.984	0.989	0.966	0.937	0.934	0.966	0.977	0.981	0.853	0.860	0.969

4.7. Individual protein neural models can be assembled into whole system AANN – sigmoid and hybrid linear/sigmoid AANN can capture whole protein system dynamics

We assembled individual neuron models into whole system AANN to represent whole system dynamics. One way to develop system AANN is to assemble sigmoid neuron models into one whole AANN. Another way is to develop hybrid AANN. As discussed previously, AANN development can be simplified if individual protein models are simpler. Therefore, hybrid AANN can be developed by combining well performing linear neuron models in Fig. 11 with well-performing sigmoidal neuron models in Fig. 12. Both these models can be assembled without further training. Here individual neurons with trained weights directly copy into the AANN, so when in operation as a whole, the system dynamics produced by the AANN can be expected to be similar to the protein dynamics from the individual models shown in Fig. 11 or 12.

AANN results for all sigmoid neurons were exactly as shown in Fig. 12. Therefore, they are not plotted here to avoid redundancy. Fig. 13 shows the assembled hybrid AANN model results for the four cyclins and their regulators. It contains the five best performing linear neuron models (CycD, CycE, CycA, NFY, Cdc20) and other proteins models are with sigmoid neurons. Table 6 shows very high prediction R^2 for this model. These results reveals that the hybrid model can well represent the protein system. Therefore, where possible, hybrid neuron models can simplify whole system models. Further, individual neuron models can serve to explore further other properties of single proteins and the hybrid AANN system can be used to explore other systems dynamics properties such as robustness, stability and disease development. Results also point out that AANN is amenable to modular assembly of individual neuron models. These results encouraged us to proceed with the

exploration of the possibility of binary neurons to represent binary regulation of proteins that could replace Boolean protein models and whole system binary AANN assembly from these individual models.

4.8. Binary neurons (perceptrons) can represent activation (on/off) state of individual proteins and they can be assembled into AANN system models

Current protein measurements are still largely made from Western blot experiments that show if a protein is active or not at a given time. This indicates that some protein network models can benefit from binary modelling. Since proteins in our system responded well to linear and nonlinear modelling, it was attempted to ascertain if binary functions could represent activation state of some or all of these proteins. Even if continuous protein data are available, as proteins display different time scales of operation from rapid to slow, binary representation of fast acting proteins in a system could still benefit protein network modelling. If individual proteins can be modelled by binary neurons or perceptrons, then they can capture rapid protein interactions like activations and inhibitions. Here, we first present the results from the exploration of whether the Boolean model (Alsharaiah, 2018) developed for the core cell cycle system in Fig. 1 could be transformed into a perceptron model that captures the essence of the corresponding Boolean rules. Then we present the refined perceptron models based on binarised data obtained from ODE model data. We first develop individual binary protein models and then test their assembly into perceptron AANN whole system models.

4.8.1. Individual perceptron models can represent boolean rules of individual proteins

The Boolean model of Alsharaiah (2018) for the core cell cycle

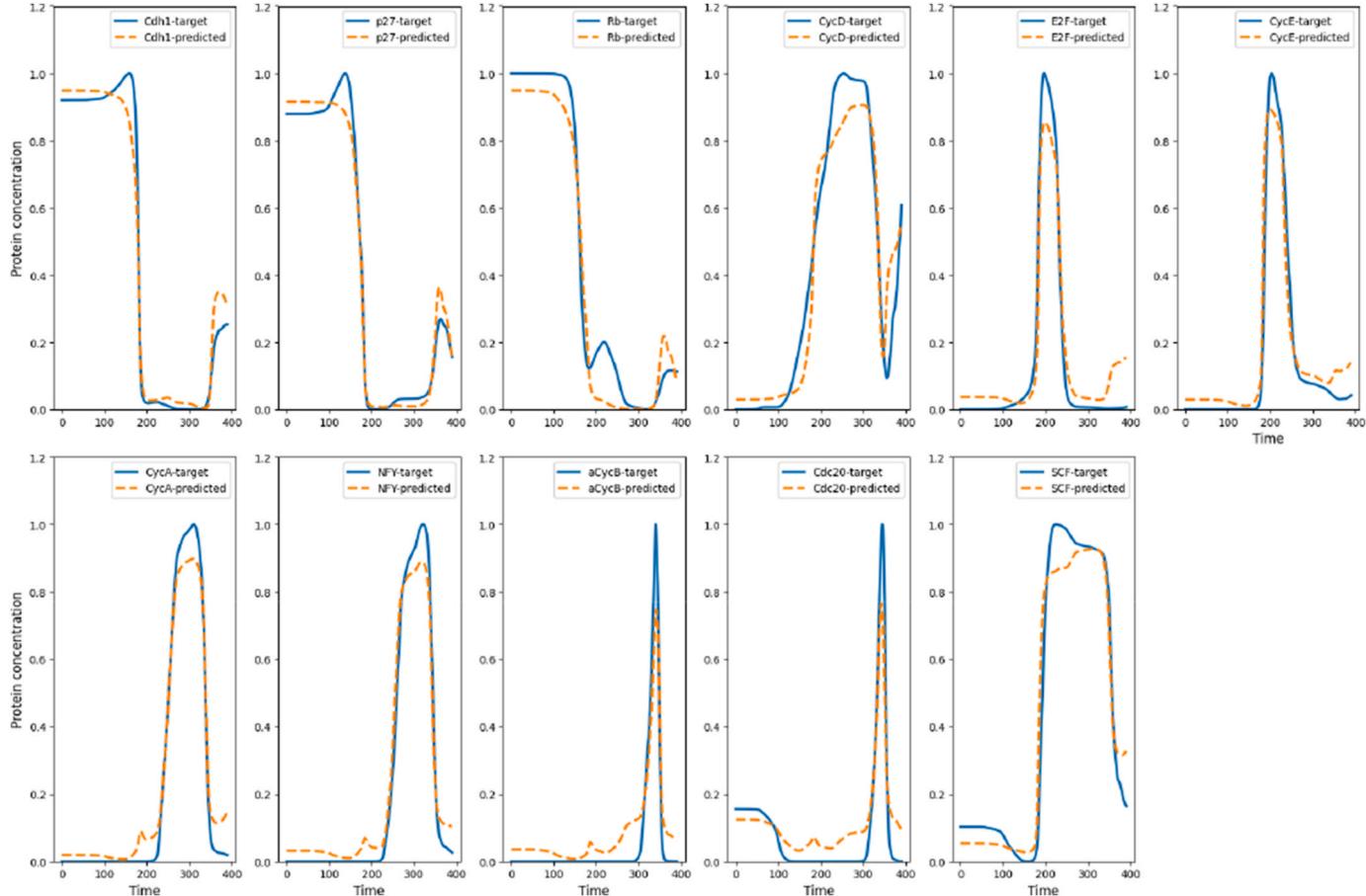


Fig. 12. Individual neuron models with sigmoid activation functions modelling single protein dynamics.

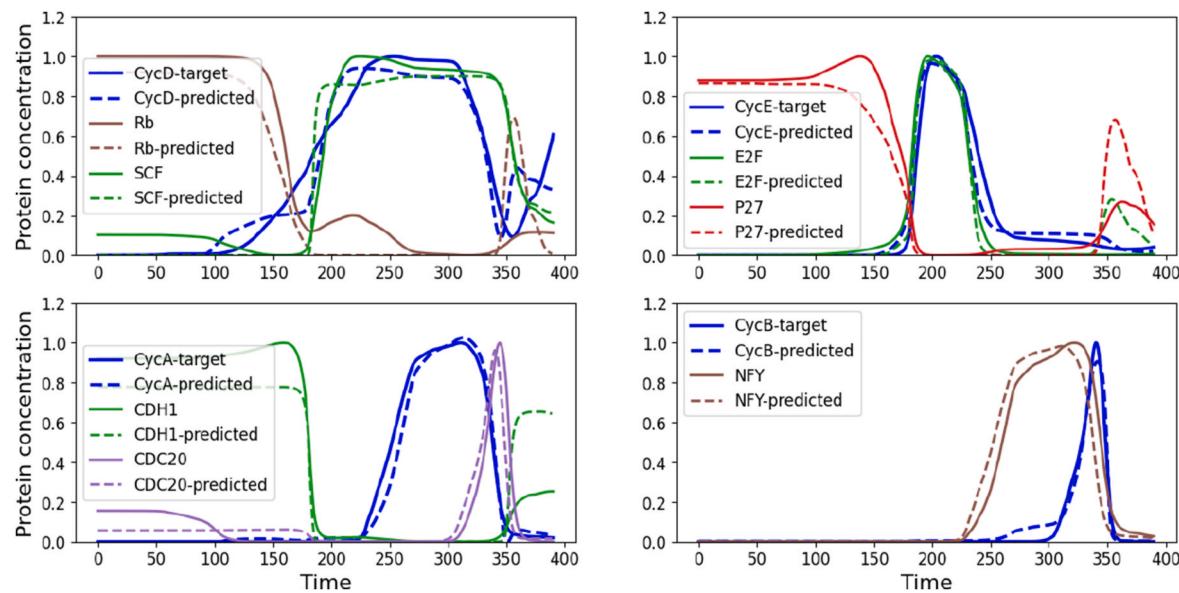


Fig. 13. Performance of the hybrid AANN developed from the assembly of the best performing linear and sigmoid individual protein neuron models producing accurate whole system dynamics of the protein network.

Table 6

Prediction R^2 for the hybrid AANN model with linear and sigmoid activation functions.

Cdh1	p27	Rb	CycD	E2F	CycE	CycA	NFY	CycB	Cdc20	SCF
0.984	0.989	0.966	0.936	0.934	0.964	0.972	0.956	0.853	0.856	0.969

system containing a Boolean rule for the activation of each protein was run to obtain data for training perceptron models. Boolean model provided 12 distinct binary temporal states spanning cell cycle for the 12 proteins. Individual neuron models were trained with inputs from their corresponding regulators using Hebbian learning as explained in the methods section. Fig. 14 shows the response of individual perceptron neurons that modelled single protein dynamics superimposed on target data generated from the Boolean model for that protein. Results for all proteins are in good agreement with the Boolean data indicating that an individual perceptron neuron can represent a Boolean rule and thus able to replace a qualitative binary Boolean rule with a quantitative binary function. This allowed us to assemble these into a whole system binary AANN.

4.8.2. Individual perceptron models assembled into whole system AANN correctly predict systems states

We assembled the individual perceptron models into a binary AANN model to represent the whole system. In concept, this is equivalent to the original (Alsharaiah et al., 2018) Boolean model. The main difference is that now we have a quantitative binary system with perceptrons instead of a qualitative binary system with IF-THEN rules. Since individual models have been trained with all relevant inputs existing in the system, when assembled, they completely characterise the system. The collective systems behaviour is shown in Fig. 15, and it has captured the real system correctly in terms of correct trend, timing and order of activation/deactivation of proteins. For example, at the beginning of cell cycle in G1 phase, CycD is produced (purple line in Fig. 15) which then deactivates Rb (red line). P27 complexes with CycD to support this process and thus p27 levels drop (green line). Rb activates the transcription factor E2F (cyan line). E2F synthesises CycE and A (brown and dark blue lines). CycE deactivates CDH1 (first blue line) to prevent it from degrading CycA. After these G1 processes, cell reaches S phase for DNA replication supported by CycE and CycA. Cell then moves to G2 where additional cell growth happens and CycA activates the transcription

factor NFY (magenta line). This synthesises CycB (dark green line) that segregates DNA in M phase. CycB reactivates CDH1 (first blue line) that degrades CycA. CDH1 also activates Cdc20 (grey line) that degrades CycB thus ending cell cycle. This indicates that the perceptron AANN systems model can correctly capture the binary representation of the real system and thus replace Boolean model of the system. Further, this model can be simulated to ascertain other emergent system properties including stability and mutational effects. In the next section, we extend binary modelling of proteins using the larger binary dataset.

4.8.3. Individual perceptron models from continuous data from ODE converted to binary format

Since the previous Boolean model produced only 12 time steps characterising the system, we sought to make our perceptron models more comprehensive. Individual perceptron models similar to those in the above section were developed for the individual proteins in the system using the larger set of binary data converted from ODE data. Fig. 16 shows the results from trained individual perceptron models (blue) superimposed on binary data (red); here, the original continuous data from the ODE model (green) are also shown for greater clarity and comparison. The figure indicates that the perceptron models (blue) have well captured the timing and trend of the protein behaviour from binary data (red) while following the trend of ODE continuous data (green) thus providing a highly reliable binary approximation of continuous protein dynamics captured by the ODE model. With these improved results, we develop a more comprehensive whole system binary AANN model in the next section.

4.8.4. Assembly of individual perceptron models trained with binary data from ODE into a whole system perceptron AANN

When individual perceptrons were assembled and simulated as a whole system, the produced system behaviour was identical to the behaviour of the collection of protein in Fig. 16 as shown in Fig. S5 in Supplementary Materials. This is because, individual models are directly

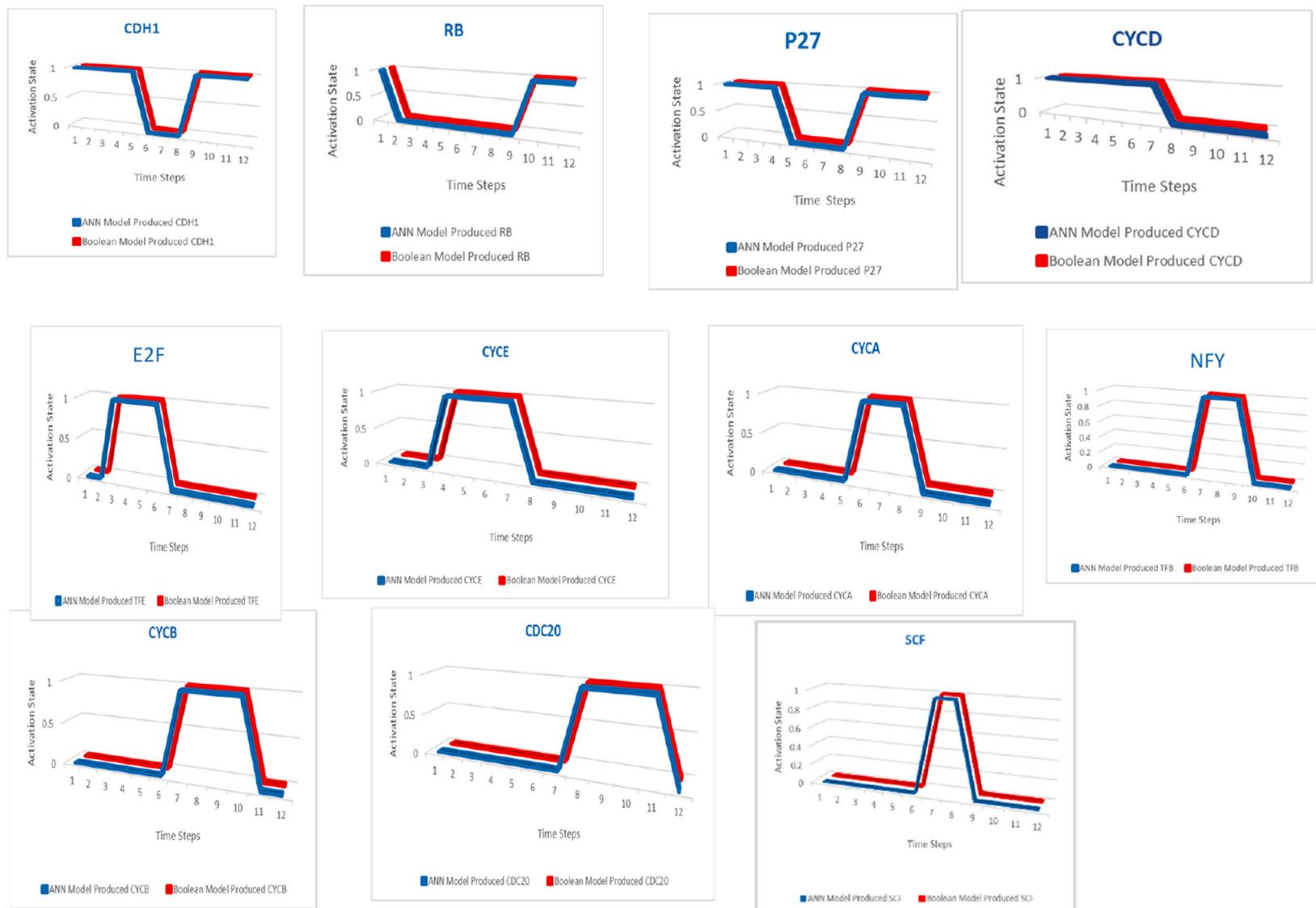


Fig. 14. Response of each individual perceptron neuron representing single protein dynamics superimposed on Boolean model data for that protein (blue-neuron model prediction and red- Boolean data).

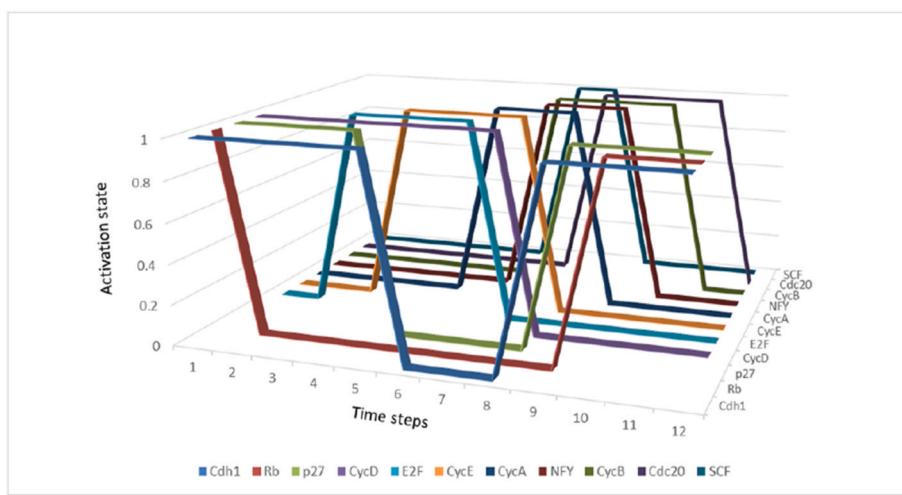


Fig. 15. Whole system dynamics from the Binary (perceptron) AANN model assembled from individual perceptron protein models.

assembled with their weights and interactions into the AANN system model and thus assembled individual models preserve the system interactions and correctly depicts the whole system dynamics. Fig. S5 (with individual components as shown in Fig. 16) thus is the binarised version of ODE results shown in Fig. 2. Now that we have accurate binary protein models, as well as accurate sigmoid protein models from Section 4.6.2, in the final section below, we explore the possibility of exploiting time scale separation existing in protein system using these two types of neuron models.

4.8.5. Hybrid binary/continuous AANN can exploit time scale separation and simplify depiction of protein dynamics

Since perceptrons were able to capture binary response of proteins well, they can be used for representing fast acting proteins in a hybrid AANN where slow activating proteins (e.g., Cyclins) are represented by continuous functions (linear or nonlinear) which in this case was chosen to be sigmoid for convenience. Here, in the system, the binary outputs from perceptrons are used as inputs for neurons with continuous functions and, vice versa, nonbinary data produced by continuous neurons are used by perceptrons for generating binary outputs according to the type of interactions. Since we have not had this training situation in the previous training considered above, we retrained the individual perceptrons and sigmoid neurons with the required mixture of binary and continuous data prepared from the ODE model data. Here, the data for the four cyclins (D, E, A and B) were kept as real values and data for their regulators were kept as binary. Hebbian learning was used to train the perceptrons and gradient descent learning was used to train cyclin models. Fig. 17 shows the dynamics of the four slow activating cyclins (CycD, CycE, CycA and CycB) and the seven fast acting regulators of the cyclins. Due to the presence of binary inputs, the four sigmoidal neurons representing cyclins approach continuous dynamics in a stepped fashion as expected (green lines) but they still reach the required concentration levels (red asterisks) at the right times as shown in the four cyclin graphs. The rest of the graphs show that fast activating proteins follow the binary target data with high accuracy.

When the above individual binary and continuous models are assembled into an AANN, weights/interactions are directly translated from the individual neurons and, therefore, system dynamics generated by it are expected to be similar to the collection of graphs in Fig. 17. The behaviour of proteins in the binary/sigmoid AANN is shown Fig. 18 that demonstrates that the collective behaves similar to the real system in terms of timing and level of activation of proteins. Therefore, hybrid perceptron/sigmoid AANN models can be used to exploit timescale separation in molecular systems and obtain realistic system dynamics. This opens a beneficial avenue to explore systems dynamics in situations

where data are limited and still obtain a realistic understanding of, and gain insights into, whole system dynamics.

5. Discussion, conclusions and future directions

This study proposed and explored the potential of neural networks to learn to capture systems dynamics of molecular interaction networks from data. A strong focus of the investigation was on the possibility of development of simpler and transparent neural networks as opposed to traditional neural networks that are complex and difficult to understand black boxes. If successful, it was expected that these neural networks could address the limitations of popular modelling approaches, such as the requirement of kinetic parameters and inherent complexity of continuous ODE models and limitation of representation of continuous behaviour and temporal reality by Boolean models. To address model simplicity and transparency, this study proposed the format of Auto-Associative Neural Networks (AANN) to represent molecular interaction networks with one-to-one mapping of neurons to proteins and demonstrated its validity using the core control system of the mammalian cell cycle as a case study. We proposed AANN to represent molecular networks considering the similarity between the two systems, i.e., elements interact with each other in specific patterns in the temporal unfolding of system. A systematic study was conducted to answer a series questions raised in the Goal section that explored in depth the potential AANN to represent molecular systems and their behaviour.

Results revealed that the proposed AANN can successfully model the dynamics of protein networks. One-to-one mapping of proteins to neurons in the whole system demonstrated the possibility of developing transparent and realistic neural network models without the bottleneck of models being black boxes. We also demonstrated that individual proteins can be represented by specifically trained individual neuron models, and we successfully demonstrated a bottom-up approach to developing AANN by assembling individual neuron models into AANN systems models. This expands the range of possibility of neuronal models in terms of having individual protein models that can help explore individual protein dynamics and system models that can help explore system dynamics as a whole. The individual protein models turned out to produce slightly superior behaviour to their counterpart in AANN trained as a whole system. Further, we also showed that AANN can be developed as binary models revealing that AANN can correctly represent Boolean models within a quantitative framework as opposed to qualitative IF-THEN rule-based framework. We also showed that hybrid binary and nonlinear AANN systems models can be successfully developed to exploit timescale separation existing in protein networks thus greatly aiding realistic and simpler model development to suit the

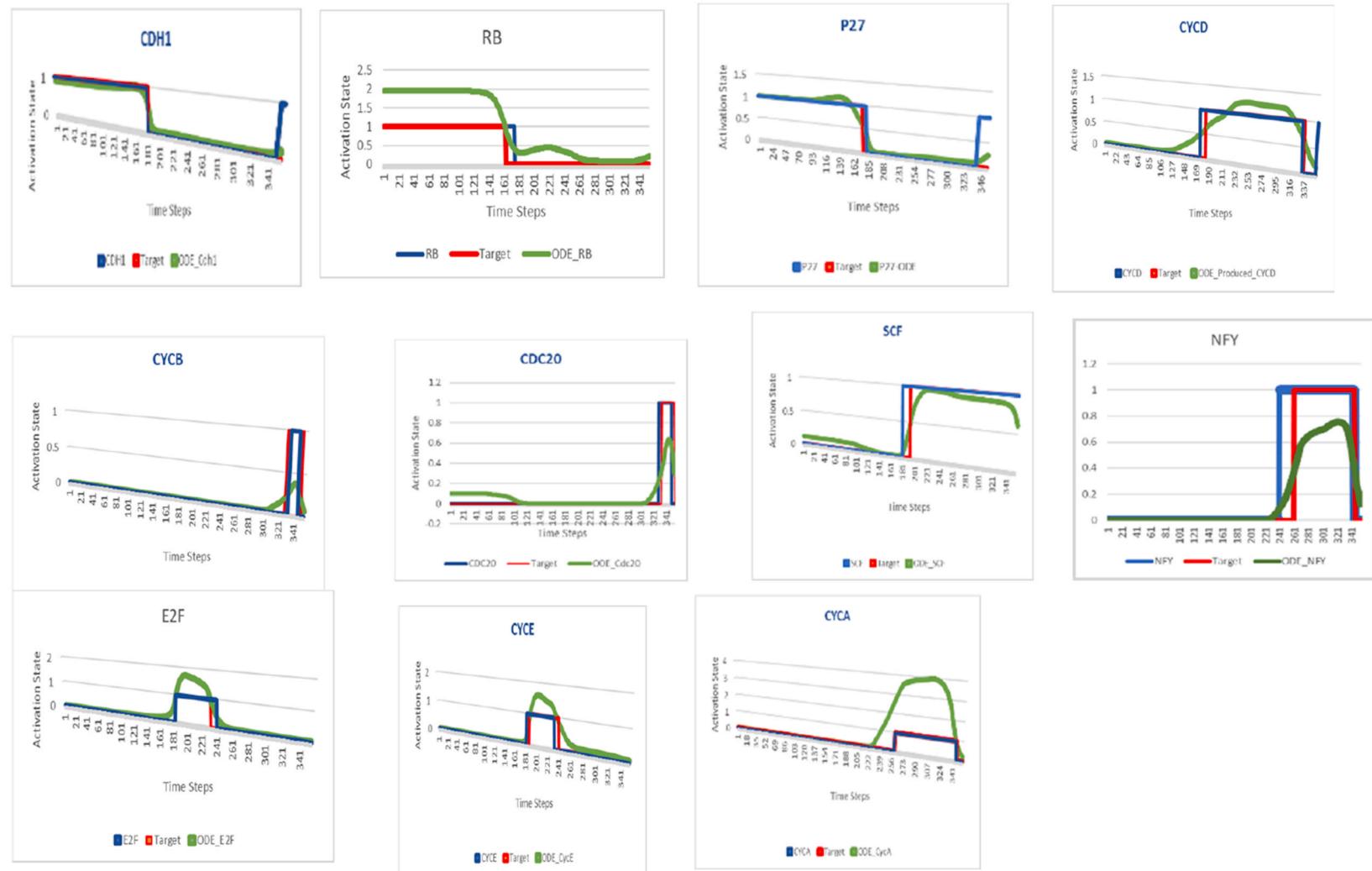


Fig. 16. Response of individual perceptron neurons modelling single protein dynamics, trained with binary data converted from continuous data generated from ODE model, superimposed on binary data and continuous ODE model data.

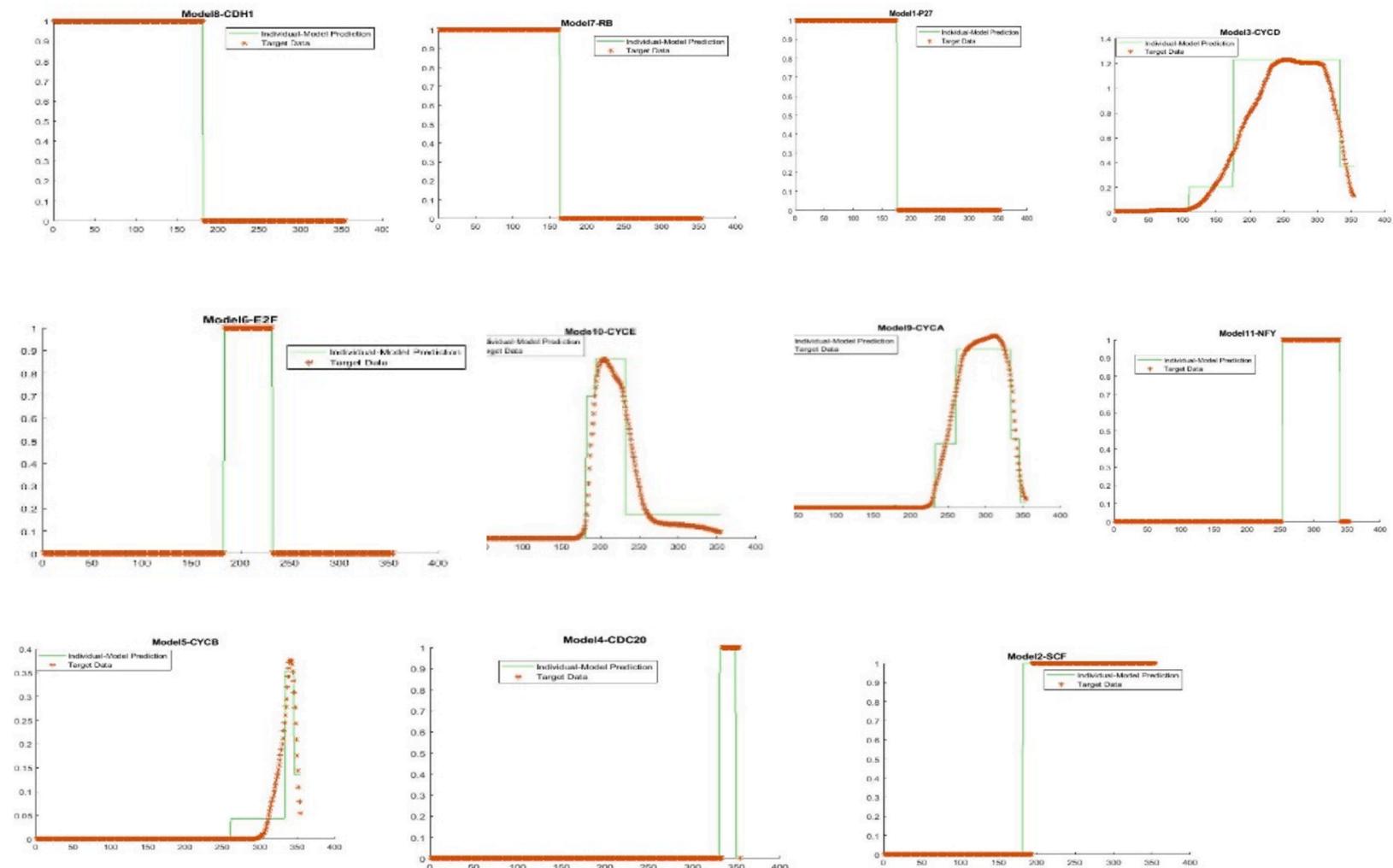


Fig. 17. Individual protein models retrained for the binary/continuous AANN exploiting timescale separation between slow and fast reactions. The four slow acting proteins (cyclins) are represented by sigmoid neurons and the seven fast acting proteins by perceptrons trained from ODE generated continuous data and binarised ODE data, respectively. Shown from left to right and top to bottom are Cdh1, Rb, p27, CycD, E2F, CycE, CycA, NFY, CycB, Cdc20 and SCF.

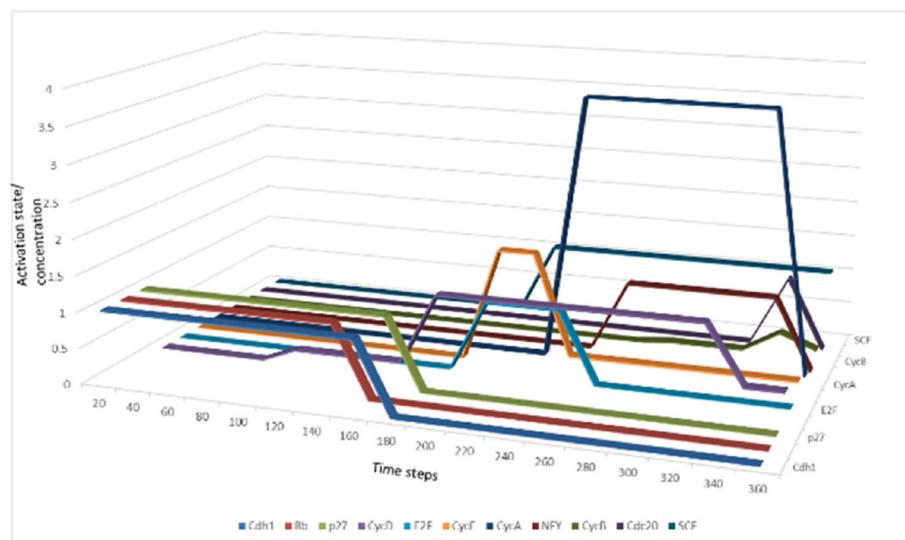


Fig. 18. Whole system dynamics from the hybrid binary/sigmoid AANN with the four slow acting proteins (cyclins) represented by sigmoid neurons and the 7 fast acting proteins depicted by perceptrons.

level of data availability, knowledge of the system and the scope of investigation. The specific questions asked in this paper revealed many insights as highlighted below.

The first question asked was whether a fully connected AANN trained with continuous data could find the correct connectivity pattern as in the actual molecular network and produce accurate system dynamics. The results revealed that the AANN model (sigmoid activation functions) can replicate the system dynamics very accurately. However, the model relies on a fully connected configuration and thus it could not extract the proper network interactions. This interesting result cautions against using data to generate models randomly as they could produce highly accurate results without capturing the correct interaction patterns or correct internal representation of the system. It also indicates that there could be other interaction patterns that produce near identical results as the original system with its specific interactions. This, in particular, emphasises the need of prior knowledge of molecular interactions in producing accurate and biologically realistic models from data. This new finding could greatly benefit future modelling.

Next, we asked whether an AANN with specific connection pattern as in the original system could capture correct system dynamics. We also asked if simpler AANN models could be developed with simpler activation functions. An all linear neuron AANN system model was attempted which resulted in close to half the neurons demonstrating correct protein dynamics. This new finding that some protein behaviour is linear could also benefit future modelling. Based on this observation, in the next modelling attempt, an all sigmoid AANN system model was trained, and it modelled all proteins accurately indicating a simple sigmoid can capture linear and nonlinear interactions in the system. This is also a new insight that could help future modelling. Thus, AANN are quite suitable for modelling protein-protein interactions with one-to-one mapping of neurons to proteins with *a priori* knowledge of interactions. This is a main outcome from the whole system AANN modelling of the protein interaction system. In order to simplify the AANN model, poorly performing linear neurons in the all-linear AANN model were replaced with well performing nonlinear (sigmoid) neurons and a hybrid linear/nonlinear AANN model was developed that produced correct system dynamics. These results revealed that model simplification is possible with one-to-one mapping of neurons to proteins and a mixture of neuron functions to suit the observed protein dynamics or the level of availability of data and, importantly, this allows the same (neuron) modelling paradigm for developing hybrid models without requiring two different modelling formalisms. This is another main outcome of the whole

system AANN modelling of the protein interaction system.

The next question raised was whether AANN as protein network models can learn to self-correct over time using internally generated responses. Idea behind this exploration was that evolution may employ self-correction mechanisms in assembling and fine-tuning protein networks. However, our results could not establish this as model performance was poor being unable to capture correct systems dynamics. Still, this could be an interesting issue to explore with a better AANN model configuration that could allow self-learning from its own responses. Following this, we asked whether AANN can shed light on the potential to capture memory in protein networks. Idea behind this was that evolution of protein networks may involve some form of memory. Recurrent AANN trained to capture system memory produced near identical results to target data. Closer inspection revealed a strong lag-1 correlation of neuron output between two consecutive time steps indicating that system can capture past memory. This could be an interesting avenue to further explore in future studies.

In the questions that followed, we turned the perspective around and explored individual protein models and a bottom-up approach to modelling molecular networks. We first explored if individual neuron models can be trained to represent individual proteins with a view to proposing it as an approach to study individual protein dynamics as well as an approach develop whole system AANN models in a modular fashion by assembling individual models. We first asked if individual linear neurons could model individual proteins. It was found that individual linear models could represent 70 % of proteins, more than the whole system linear AANN could (close to 50 %). Then, we trained the proteins with sigmoid functions and all protein dynamics were correctly captured by sigmoid neurons. This reveals that individual sigmoid neurons can be used to represent individual proteins and study their behaviour. Therefore, in the case where protein behaviour is needed to be studied separately for various purposes, single neuron models (linear or sigmoid) may be used simply, effectively and reliably. This ability of a neuron to represent an individual protein is quite promising because it strengthens the case of proteins as computing units similar to neurons. Further individual protein dynamics from these models were even slightly better than the individual protein dynamics from whole system AANN. Thus, training proteins separately with sigmoid function can produce better results than training all proteins together as a whole system. Accordingly, when individual neuron models were assembled into AANN, slightly superior performance to AANN trained as a full system resulted. These results reveal that individual neuron models

could be assembled into a whole system AANN model, and it could provide some advantage over AANN trained as a whole system. This is not surprising as individual model training does not have some of the constraints of the whole system training where system error as a whole is reduced during training. However, modelling the whole system as AANN is simpler and faster. Next, we explored model simplification by replacing the poorly performing individual linear models with sigmoid neurons and assembled a hybrid linear/nonlinear AANN from individual neurons to produce correct protein dynamics in the system. For the reasons described above, this AANN system behaviour was slightly superior to the hybrid linear/sigmoid AANN whole system. Therefore, hybrid neuron models can simplify whole system models and thus there is value in training individual models and assembling them into a system model.

In the next series of questions, we explored the capacity of perceptron neurons and perceptron AANN for binary representation of proteins and protein networks, respectively. With this, we also tested the possibility of replacing qualitative Boolean models with quantitative Perceptron (binary) models. To test this possibility, we used data from a Boolean model previously developed for the core cell cycle system to train individual perceptron models to represent individual proteins. Individual perceptron modelled correctly predicted protein dynamics and when assembled into a binary (perceptron) AANN, it produced system dynamics identical to the Boolean model and realistic to cell cycle. Thus, perceptron models can replace qualitative Boolean IF-THEN rules, that represent various stages of behaviour of a protein in complex ways, with a simpler quantitative binary representation, and similarly, the assembled perceptron AANN as a system can replace Boolean models with a quantitative whole system representation.

The above perceptron models were developed from data for only 12 time steps. To refine the models with more frequent data, we converted the ODE continuous data into binary format to obtain over 350 data points and trained individual perceptrons to represent individual proteins. These individual models produced very realistic binary protein dynamics as in the converted data and correctly matched the continuous ODE data as well. The assembled perceptron AANN too thus produced realistic binary response of the whole system. Since these modelling exercises revealed that perceptrons can model binary responses, meaning rapid responses, with the final question, we explored the potential of neural models for time scale separation existing in protein interaction networks where some processes such as cyclin protein synthesis are slow, but their activations are rapid. We modelled the four cyclins using sigmoid neurons and their activators using perceptrons. In these models, a mixture of binary and continuous inputs can feed a neuron whose output can be either continuous or binary, depending on the neuron type. Therefore, we used the same set of continuous data obtained from the ODE model for cyclins and converted binary data for rapid activators and redeveloped individual models. The binary neuron models produced accurate binary responses of the cyclin regulators, and the nonlinear neurons correctly approximated cyclins dynamics in a stepped fashion due to binary inputs they receive from their activators. Accordingly, the hybrid binary/nonlinear AANN model realistically represented the time scale separation and correctly captured the trend and timing of fast (binary) and slow (continuous) dynamics of proteins.

This study presented a thorough investigation of the potential of neural networks for modelling the dynamics of protein interaction networks insightfully, meaningfully, simply and effectively, and successfully demonstrated their capacity to do so with auto associative neural network (AANN) system models. Further, it revealed that individual neurons can represent individual proteins with binary, linear and nonlinear (sigmoid) characteristics that supported the modular bottom-up construction systems models. Further, it revealed the potential of hybrid AANN models such as linear/nonlinear and binary/continuous to support model simplicity and, importantly, to develop hybrid models within one (neural) modelling formalism. Further, we demonstrated the potential of neural networks to transform Boolean models as well as

their capacity for time scale separation with rapid and slow activating neurons to represent fast and slow activating processes. Thus, we have presented a neural network modelling formalism for realistic representation of molecules and molecular networks to suit the type and extent of available data and knowledge.

5.1. Future directions

Future studies can apply our AANN approach to other molecular networks. We also plan to probe further into our models to explore their other interesting attributes. Specifically, we intend to explore the nature of the parameters estimated by AANN and biological insights they may provide in a future investigation. We will also explore AANN further to investigate their other emerging properties such as robustness, stability and vulnerabilities (e.g., susceptibility to diseases and effect of mutations) as well as probe further into the ability of AANN to self-correct and hold memory to explore the evolutionary traits of molecular networks through modelling. Further, our models were trained with data generated from an ODE model. As such, our data were smooth and continuous whereas in reality experimental data are noisy and sparser. Also, knowledge of proteins and their connections are often missing for many systems. Therefore, in future, we also plan to put our models to test to ascertain their ability to predict protein dynamics under more real conditions where data contain noise and are temporally sparse. We intend to do this by imposing these conditions on the data we have used here and retraining the models under these conditions. Further, we plan to extend this investigation to explore model robustness against missing proteins and connections to simulate the reality of missing knowledge of proteins and connections in molecular systems. We intend to do this by randomly removing proteins and connections from the models to test the impact on system behaviour. We aim to present the outcomes of all these model investigations in a future communication.

There are also other interesting potential methods that can be explored in future. One is spiking neural networks (SNN). SNN emulate the processing in the brain more closely than standard neural networks in that they use membrane potential to generate spiking action in neurons. They incorporate temporal processing in neurons embedded in spatial networks. Some SNN features bear resemblance to protein networks - for example, SNN's temporal and spatial information processing with sparse asynchronous computations and time-dependent neuronal functionality. Protein networks are also spatial networks that perform time-dependent processing (production, degradation, activation, phosphorylation etc.) in asynchronous manner. Accordingly, how concepts of SNN apply to protein networks could be an interesting avenue in future. Some of the challenges of large memory requirement in learning of earlier SNN have been addressed by more recent SNN such as Reversible SNN ([Zhang and Zhang, 2024](#)). Efficient Reversible SNN architectures ([Vicente-Sola et al. \(2022\)](#)) can allow configurations of feedback connections of choice. Further, recently proposed self-supervised learning based on information bottleneck theory ([Yang et al., 2025a](#)) could help SNN learn more abundant and meaningful temporal characteristics from data and could potentially provide a large-scale spatio-temporal SNN learning framework for protein network modelling. Another framework, based on high-order spike-based information bottleneck (HOSIB) leveraging surrogate gradient technique, that explores common latent architecture and sparse spike-based intrinsic information, while discarding superfluous information in data ([Yang and Chen, 2023](#)) could potentially help identify and model protein networks.

The main purpose of our work was to test the applicability of AANN to model a known protein network. However, as a curiosity, we also tested its ability to find unknown connection patterns from data and found that the AANN could not find the correct connection pattern although it could model the system with high accuracy. This indicates that finding PPI from data is a complex task due to model parameter redundancy. As this could be a common pitfall of machine learning models, we cautioned against using data driven models for finding

unknown protein systems. Therefore, methods that can help narrow down the possibilities to the correct connectivity pattern is of extreme benefit, especially in this era of high-throughput data generation. In our AANN, some form of regularization is needed to arrive at the correct pattern in future. From this perspective, specifically designed deep learning approaches based on graph neural networks, could help identify specific connection patterns. Yang et al. (2025b) use a tailored regularization term, that alleviates the over-smoothing issue on graph neural networks, to train a fuzzy clustering integrated graph convolution network. Such approach could be useful in identifying connection patterns in protein networks.

Wang et al. (2023) propose an algorithm based on a mixed membership stochastic block model able to generate a PPI by capturing the latent community structures. It further optimises the membership distributions of proteins over different complexes and assesses the likelihood of two proteins interacting with each other, with promising results shown for five PPI networks from different species. Such approaches can help generate PPI that can be simulated with AANN for continuous system dynamics. In a drug repurposing study, Zhao et al. (2025) use regulation-aware graph representation learning to leverage diverse connectivity patterns to gain new insights into the regulatory mechanisms of drugs acting on target proteins in diseases. They defined a set of meta-paths to reveal different regulatory mechanisms, each corresponding to distinct connectivity patterns. For each meta-path, they constructed a regulation graph through random-walk sampling of its instances in the network to obtain drug and disease embeddings, showing that the approach leverages meaningful connectivity patterns for three benchmark datasets and two disease case studies. This could also be a promising avenue for finding unknown PPI that could be subsequently modelled by AANN.

CRediT authorship contribution statement

Sandhya Samarasinghe: Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Tran Nguyen Minh-Thai:** Visualization, Software, Formal analysis. **Komal Sorthiya:** Visualization, Formal analysis. **Don Kulasiri:** Writing – review & editing, Conceptualization.

Declaration of competing interest

We declare that there are no conflicts of interest pertaining to this manuscript.

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.biosystems.2025.105613>.

Data availability

Data will be made available on request.

References

- Abroudi, A., Samarasinghe, S., Kulasiri, D., 2017. A comprehensive complex systems approach to the study and analysis of Mammalian cell cycle control system in the presence of DNA damage stress. *J. Theor. Biol.* 429, 204–228.
- Abroudi, A., Samarasinghe, S., Kulasiri, D., 2020. Towards abstraction of computational modelling of Mammalian cell cycle: model reduction pipeline incorporating multi-level hybrid petri nets. *J. Theor. Biol.* 496, 110212.
- Alsharaiah, M., 2018. Qualitative and Semi Quantitative Systems Approaches to Complex Systems Modelling – Intuitive and Flexible Models of Biological Systems. Lincoln University, New Zealand. PhD thesis.
- Alsharaiah, M., Samarasinghe, S., Kulasiri, D., 2023. Proteins as fuzzy controllers: auto tuning a biological fuzzy inference system to predict protein dynamics in complex biological networks. *Biosystems* 224, 104826, 2023.
- Behl, C., Ziegler, C., 2014. Cell cycle: the life cycle of a cell. In: *Cell Aging: Molecular Mechanisms and Implications for Disease*. Springer, Berlin Heidelberg, pp. 9–19, 10.1007/978-3-642-45179-9_2.
- Bodenstein, M., 1913. Eine theorie der photochemischen reaktionsgeschwindigkeiten. *Z. Phys. Chem.* 85 (1), 329–397.
- Brightman, F.A., Fell, D.A., 2000. Differential feedback regulation of the MAPK cascade underlies the quantitative differences in EGF and NGF signalling in PC12 cells. *FEBS Lett.* 482 (3), 169–174.
- Danos, V., Feret, J., Fontana, W., Harmer, R., Krivine, J., 2007. Rule-Based Modelling of Cellular Signalling. Springer Berlin Heidelberg, Berlin, Heidelberg. Abstract retrieved from doi: 10.1007/978-3-540-74407-8_3. 2007.
- Dong, P., Maddali, M.V., Srimani, J.K., Thélot, F., Nevins, J.R., Mathey-Prevot, B., 2014. Division of labour between Myc and G1 cyclins in cell cycle commitment and pace control. *Nat. Commun.* 2014.
- Farr, K.J., Gallaway, P.J., Hongu, N., 2017. Breast Cancer Prevention: Exercise and Healthy Diet. The University of Arizona Cooperative Extension, 2017.
- Faure, A., Naldi, A., Chaouiya, C., Thieffry, D., 2006. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics* 22 (14), e124–e131.
- Fujita, S., Matsui, M., Matsuno, H., Miyano, S., 2004. Modeling and simulation of fission yeast cell cycle on hybrid functional petri net. *IEICE Trans. Fund. Electron. Commun. Comput. Sci.* 87 (11), 2919–2928.
- Herajy, M., Heiner, M., 2012. Hybrid representation and simulation of stiff biochemical networks. *Nonlinear Analysis* 6 (4), 942–959.
- Hopfield, J.J., 1982. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* 79 (8), 2554–2558.
- Kitano, H., 2007. Towards a theory of biological robustness. *Mol. Syst. Biol.* 3 (1).
- Limas, J., Cook, J.G., 2019. Preparation for DNA replication: the key to a successful S phase. *FEBS (Fed. Eur. Biochem. Soc.) Lett.* 593 (2019), 2853–2867.
- Ling, H., Kulasiri, D., Samarasinghe, S., 2010. Robustness of G1/S checkpoint pathways in cell cycle regulation based on probability of DNA-damaged cells passing as healthy cells. *Biosystems* 101 (3), 213–221.
- Ling, H., Samarasinghe, S., Kulasiri, D., 2013. Novel recurrent neural network for modelling biological networks: oscillatory p53 interaction dynamics. *Biosystems* 114 (3), 191–205.
- Matlab 2022. Mathworks Inc., USA.
- Morales, A., Froese, T., 2019. Self-optimization in a Hopfield neural network based on the *C. elegans* connectome. *Artificial Life Conference Proceedings* 31, 448–453.
- Naldi, A., Remy, E., Thieffry, D., Chaouiya, C., 2011. Dynamically consistent reduction of logical regulatory graphs. *Theor. Comput. Sci.* (21), 2207–2218.
- Samarasinghe, S., 2006. *Neural Networks for Applied Sciences and Engineering: from Fundamentals to Complex Pattern Recognition*. CRC Press.
- Samarasinghe, S., 2023. Collective computational intelligence in biology – emergence of memory in somatic tissues. *Biosystems* 223, 104816, 2023.
- Schoeberl, B., Eichler-Jonsson, C., Gilles, E.D., Müller, G., 2002. Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. *Nat. Biotechnol.* 20 (4), 370–2002.
- Singhania, R., Sramkoski, R.M., Jacobberger, J.W., Tyson, J.J., Beard, D., 2011. A hybrid model of mammalian cell cycle regulation. *PLoS. Comput. Biol.* 7 (2), e1001077.
- Tyson, J.J., 1991. Modeling the cell division cycle: cdc2 and cyclin interactions. *Proc. Natl. Acad. Sci.* 88 (16), 7328–7332.
- Vicente-Sola, A., Manna, D.L., Kirkland, P., Caterina, G.D., Bihl, T., 2022. Keys to accurate feature extraction using residual spiking neural networks. *Neuromorph. Comput. Eng.* 2. <https://doi.org/10.1088/2634-4386/ac8bef>. Number 4.
- Waidyaratne, P., Samarasinghe, S., 2018. Boolean calcium signalling model predicts calcium role in acceleration and stability of abscisic acid-mediated stomatal closure. *Sci. Rep.* 8, 17635. <https://doi.org/10.1038/s41598-018-35872-9>.
- Wang, X., Yang, W., Yang, Y., He, Y., Zhang, J., Wang, L., Hu, L., 2023. PPISB: a novel network-based Algorithm of predicting protein-protein interactions with mixed membership stochastic blockmodel. *IEEE ACM Trans. Comput. Biol. Bioinf.* 20 (2), 1606–1612. <https://doi.org/10.1109/TCBB.2022.3196336>, 2023 Mar-Apr.
- Watson, R.A., Mills, R., Buckley, C.L., 2011. Global adaptation in networks of selfish components: emergent associative memory at the system scale. *Artif. Life* 17 (3), 147–166.
- Yang, S., Chen, B., 2023. Effective surrogate gradient learning with high-order information bottleneck learning for spike-based machine intelligence. *IEEE Transact. Neural Networks Learn. Syst.* 36 (1), 1734–1748. <https://doi.org/10.1109/TNNLS.2023.3329525>. Jan. 2025.
- Yang, S., Linares-Barranco, B., Wu, Y., Chen, B., 2025a. Self-Supervised high-order information bottleneck learning of spiking neural network for robust event-based optical flow estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* 47 (4), 2280–2297. <https://doi.org/10.1109/TPAMI.2024.3510627>. April 2025.
- Yang, Y., Li, G., Li, D., Zhang, J., Hu, P., Hu, L., 2025b. Integrating fuzzy clustering and graph convolution network to accurately identify clusters from attributed graph. *IEEE Trans. Netw. Sci. Eng.* 12 (2), 1112–1125. <https://doi.org/10.1109/TNSE.2024.3524077>.
- Zarco, M., Froese, T., 2018a. Self-modeling in Hopfield neural networks with continuous activation function. *Procedia Comput. Sci.* 123, 573–578.
- Zarco, M., Froese, T., 2018b. Self-optimization in continuous-time recurrent neural networks. *Front. Robotics and AI* 5 (96).
- Zhang, H., Zhang, Y., 2024. Memory-efficient reversible spiking neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*. <https://doi.org/10.1609/aaai.v38i15.29616>, 2024.
- Zhao, B., Su, X., Yang, Y., Li, D., Li, G., Hu, P., You, Z., Luo, X., Hu, L., 2025. Regulation-aware graph learning for drug repositioning over heterogeneous biological network. *Inf. Sci.* 686, 121360. <https://doi.org/10.1016/j.ins.2024.121360>. ISSN 0020-0255.