Characterising Neurological Time Series Data using Biologically-Motivated Networks of Coupled Discrete Maps

Michael A. Lones[a,*], Stephen L. Smith[a], Andy M. Tyrrell[a], Jane E. Alty[b], D. R. Stuart Jamieson[b]

[a] *Department of Electronics, University of York, Heslington, York, YO10 5DD, UK*
[b] *Leeds General Infirmary, Leeds, LS1 3EX, UK*

Abstract

Artificial biochemical networks (ABNs) are a class of computational dynamical system whose architectures are motivated by the organisation of genetic and metabolic networks in biological cells. Using evolutionary algorithms to search for networks with diagnostic potential, we demonstrate how ABNs can be used to carry out classification when stimulated with time series data collected from human subjects with and without Parkinson's disease. Artificial metabolic networks, composed of coupled discrete maps, offer the best recognition of Parkinsonian behaviour, achieving accuracies in the region of 90%. This is comparable to the diagnostic accuracies found in clinical diagnosis, and is significantly higher than those found in primary and non-expert secondary care. We also illustrate how an evolved classifier is able to recognise diverse features of Parkinsonian behaviour and, using perturbation analysis, show that the evolved classifiers have interesting computational behaviours.

Keywords: Artificial biochemical networks, Discrete maps, Time series classification, Parkinson's disease.

1. Introduction

Parkinson's Disease (PD) is a neurodegenerative disorder caused by the loss of dopamine-generating neurones in the central nervous system. The symptoms of PD are variable, but all patients develop some form of movement abnormality—such as slowness of movement (*bradykinesia*), tremor, rigidity, and impaired balance. Because of its symptomatic diversity, and symptom overlap with other diseases, PD is sometimes difficult to diagnose, with clinical misdiagnosis rates in the region of 25% (Bajaj et al., 2010; Levine et al., 2003).

Artificial biochemical networks (ABNs) are a class of computational automata whose form and function are modelled upon the biochemical networks found within biological organisms. In (Lones et al., 2010) and (Lones et al., 2011) we developed various ABN models, and showed how they display rich computational behaviours when coupled to a spectrum of dynamical systems. In this work, we apply ABNs to the problem of classifying whether a patient has PD, based on analysis of their movement data. This approach is based on the hypothesis that ABNs can be evolved which will react to the dynamics found within a movement time series, causing them to alter their internal state in an observable manner. It is comparable to other uses of computational dynamical systems to perform time series classification, for example recurrent neural networks (Hüsken and Stagge, 2003) and reservoir computers (Verplancke et al., 2010).

In this paper, we consider two different types of ABN—artificial metabolic networks (AMNs) and artificial genetic networks (AGNs)—and present results showing that ABNs are able to discriminate between the movements of PD patients and control subjects with accuracies in the region of 90%. In particular, we find that discrete map based AMNs provide the best classification accuracy, often using only a small number of discrete maps. To illustrate this, we look at the computational behaviour of an AMN which is composed of only four discrete maps, showing how it uses dampened chaotic dynamics to recognise a range of Parkinsonian behaviours.

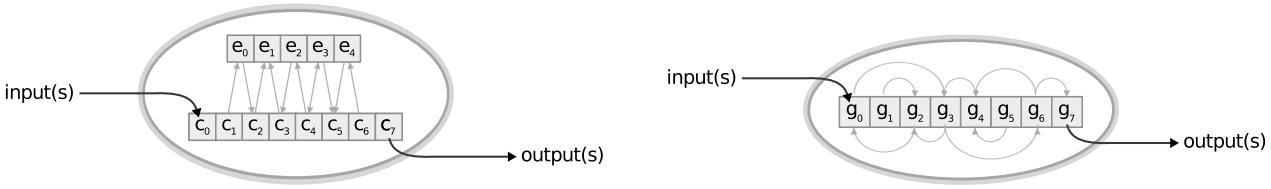
The paper is organised as follows: Section 2 introduces the ABN architectures used in this work; Section 3 gives a summary of materials and methods; Section 4 presents classification results; Section 5 provides a detailed analysis of an evolved classifier; and Section 6 concludes.

2. Artificial Biochemical Networks

We use the two ABN architectures described in Lones et al. (2010), with only minor modifications to reflect the different problem domain addressed in this work.

An artificial metabolic network (AMN) is an abstract model of a cell's metabolism, capturing the idea of a set of enzyme-mediated reactions manipulating the concentrations of a set of chemicals over a period of time. AMNs

*Corresponding author
Email addresses: michael.lones@york.ac.uk (Michael A. Lones), stephen.smith@york.ac.uk (Stephen L. Smith), andy.tyrrell@york.ac.uk (Andy M. Tyrrell), Jane.Alty@hyms.ac.uk (Jane E. Alty), Stuart.Jamieson@leedsth.nhs.uk (D. R. Stuart Jamieson)



(a) AMN: I_C={c_0}, O_C={c_7}

(b) AGN: I_G={g_0}, O_G={g_7}

Figure 1: Artificial biochemical network models.

are comparable to other computational models of cellular metabolism, such as artificial chemistries (Dittrich et al., 2001) and P systems (Păun, 2000), but with a focus on computational efficiency rather than accurate modelling of their biological analogue.

Formally: AMN = $\langle C, E, L_C, I_C, O_C \rangle$, where:

C is the set of chemical concentrations $\{c_0, ..., c_n : \mathbb{R}\}$.

E is the set of enzymes $\{e_0, ..., e_n : e_i = \langle S_i, P_i, m_i \rangle\}$, where:

- $S_i \subseteq C$ is the enzyme's substrates.
- $P_i \subseteq C$ is the enzyme's products.
- $m_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the substrate-product mapping.

L_C is an indexed set of initial chemical concentrations, where $|L_C| = |C|$.

$I_C \subset C$ is the set of chemicals used as external inputs.

$O_C \subset C$ is the set of chemicals used as external outputs.

An artificial genetic network (AGN) is an abstract model of a biological genetic regulatory network, capturing the idea of a set of genes regulating one another's expression levels over a period of time. The AGN model used in this work generalises the more common Boolean network model (Kauffman, 1969) by using continuous-valued expression levels and continuous-valued regulatory functions. This makes the networks more suited to working with continuous-valued data, and also means that smaller networks can have more complex dynamics than Boolean networks of an equivalent size.

Formally: AGN = $\langle G, L_G, I_G, O_G \rangle$, where:

G is the set of genes $\{g_0, ..., g_n : g_i = \langle \lambda_i, R_i, f_i \rangle\}$, where:

- $\lambda_i : \mathbb{R}$ is a gene's expression level.
- $R_i \subseteq G$ is a gene's regulatory inputs.
- $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is a gene's regulatory function.

L_G is an indexed set of initial expression levels, where $|L_G| = |G|$.

$I_G \subset G$ is the set of genes used as external inputs.

$O_G \subset G$ is the set of genes used as external outputs.

AMNs and AGNs are executed in a similar manner. First, their numerical state components are initialised from L_C or L_G, respectively. During the course of execution, external inputs are delivered by explicitly setting the values of state components indicated in I_C or I_G at appropriate intervals. At each time step, the functional components (enzymes or genes, respectively) synchronously modify the values of the state components. At the end of execution, outputs are captured from the final concentrations of the state components specified in O_C or O_G.

In an AGN, each functional component operates upon one state component. In the AMN, however, the functional components operate upon potentially overlapping subsets of chemical concentrations. In situations where the same chemical is produced by multiple enzymes, the chemical's new concentration is the mean output value of all contributing enzymes.

In addition, AMNs obey a mass conservation law, where the sum of chemical concentrations are constrained to remain constant over time. The use of mass conservation more closely reflects biological systems, where mass balance results in indirect regulatory interactions between chemical reactions. Conservation is carried out after each iteration of the network by uniformly scaling concentrations so that:

$$\left(\sum_{c_i \in C} c_i \right) = 0.5|C| \tag{1}$$

However, chemicals which have reached saturation ($c = 1$) and those which are not present in the chemistry ($c = 0$) remain unchanged, preserving these special states.

2.1. Functional Components

The behaviours of functional components (i.e. enzyme mappings m_i and gene regulatory functions f_i) are chosen from a set of continuous non-linear mappings (Lones et al., 2011). These comprise a standard sigmoidal logistic function and also four discrete maps: the logistic map, Chirikov's standard map, the baker's map, and Arnold's cat map. In previous work, we have found that discrete maps are beneficial when evolving a range of dynamical behaviours. The chosen maps capture a range of dynamical phenomenon which occur in many types of physical and biological systems. Whilst they do not directly represent processes occuring in the nodes of biological biochemical networks, they have been used to simulate biological dynamics (Kaneko, 1992), and have been used as computational elements within other kinds of computational dynamical system (Andersson and Nordahl, 1998).

The logistic map (May, 1976) is a model of population growth which displays both ordered and chaotic behaviour depending upon the value of a parameter $r \in [0, 4]$:

$$x_{n+1} = r x_n (1 - x_n) \tag{2}$$

Chirikov's map (Chirikov, 1969) is a model of Hamiltonian systems whose phase spaces have co-existing ordered and chaotic regimes. The dynamics move from majority-ordered to majority-chaotic as the parameter $k \in [0, 10]$ increases:

$$\begin{aligned} x_{n+1} &= (x_n + y_{n+1}) \bmod 1 \\ y_{n+1} &= (y_n - \frac{k}{2\pi} \sin(2\pi x_n)) \bmod 1 \end{aligned} \tag{3}$$

The baker's map (Tél and Gruiz, 2006) and Arnold's cat map (Arnold and Avez, 1968) are both archetypal models of chaotic phenomenon that occur in a range of systems. They both map points within the unit square and are defined, respectively:

$$(x_{n+1}, y_{n+1}) = \begin{cases} (2x_n, y_n/2) & 0 \leq x_n \leq \frac{1}{2} \\ (2 - 2x_n, 1 - y_n/2) & \frac{1}{2} \leq x_n < 1 \end{cases} \tag{4}$$

$$(x_{n+1}, y_{n+1}) = ([2x_n + y_n] \bmod 1, [x_n + y_n] \bmod 1) \tag{5}$$

3. Materials and Methods

3.1. Movement Data

Movement data was collected at the Leeds Teaching Hospitals NHS Trust, UK, from 49 PD patients and 41 age-matched controls as they performed a finger tapping task, a standard clinical means of measuring bradykinesia. The study was granted approval by the National Research Ethics Service and Medicines and Healthcare Products Regulatory Agency. Written consent was obtained from all subjects and their medications were not altered for the study. There was no history of neurological disease amongst the control subjects.

Each subject was asked to tap their thumb and index finger repeatedly for a duration of 30 seconds, using their dominant hand. Subjects were asked to carry out this exercise as rapidly as possible, separating the finger and thumb as far as they could comfortably achieve. Movement data was collected using a Polhemus Patriot electromagnetic motion tracking device, whose probes were attached to the subject's thumb and index finger whilst carrying out the task. Based on the displacement between thumb and index finger, an acceleration time series was then calculated for each subject. These were divided into training and test sets in the ratio 2:1, with the training data used for fitness evaluation and the test set used to measure classifier generality.

3.2. Preprocessing

Each displacement time series was truncated to one standard deviation around the mean, scaled to the interval [0,1], down-sampled by a factor of 2, and smoothed using a moving average filter of size 2. We have found this to be an effective method for removing absolute amplitude information, which acts as a local optimum in the classification landscape, and for emphasising the shape of the signal, which encourages the evolution of more general classifiers.

3.3. Classification

An acceleration time series is input to an AMN by setting the concentration of the first chemical (c_0), and to an AGN by setting the expression of the first gene (λ_0). The time series is delivered to a network one value at a time, each followed by t_b iterations of the network. Once the whole time series has been delivered, the network is executed for another t_a iterations in order to allow the dynamics to settle. At this point a single output value is read from the final concentration of the last chemical (c_n) or the final expression of the last gene (λ_n). Using a suitable threshold, this output value can then be interpreted as the network's classification for the time series.

3.4. Evolutionary Algorithm

We used a standard generational evolutionary algorithm (EA) (Luke, 2009) to find ABNs that can classify the movement data. The number of genes/enzymes, the connectivity between genes/enzyme/chemicals, the functional components (including their parameters), and the settling parameters (t_b and t_a) are all encoded as a linear string and evolved (Lones et al., 2010). The EA uses tournament selection of size 4, a single elite, a point mutation rate of 6% and uniform crossover with a crossover probability of 15%. Initial solution sizes were made intentionally short, between 2 and 10 genes/enzymes, to encourage parsimony. Since evolutionary algorithms are stochastic algorithms, we carried out 50 runs for each parameter set to measure the distribution of classifier performance. Each evolutionary run had a population size of 200 and a generation limit of 100.

3.5. Fitness Function

Each evolved ABN was stimulated with movement time series from the training set, producing a real-valued output for each subject. A Receiver Operating Characteristic (ROC) (Fawcett, 2006) curve was then constructed, showing the different trade-offs between specificity and sensitivity for different thresholds on the ABN's output range. From this, the area under the ROC curve (AUC) was calculated, and used as the ABN's fitness score. An AUC of 0.5 is no better than randomly assigning movement sequences to classes, and AUCs of 0 and 1 both indicate fully correct class assignments, but with opposite orderings of the classes within the classifier's output range. During

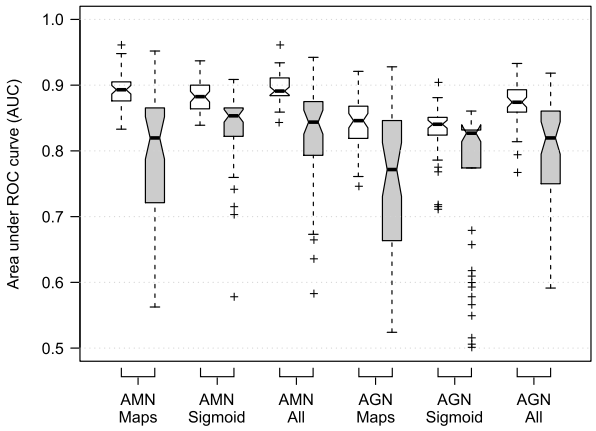


Figure 2: Diagnostic power of evolved ABNs on both the training (white) and test (grey) sets. Notched box plots show summary statistics over 50 runs. Overlapping notches indicate when median values (thick horizontal bars) are not significantly different at the 95% confidence level.

tournament selection, AUCs below 0.5 are normalised to the range $[0.5, 1]$ by subtracting from 1.

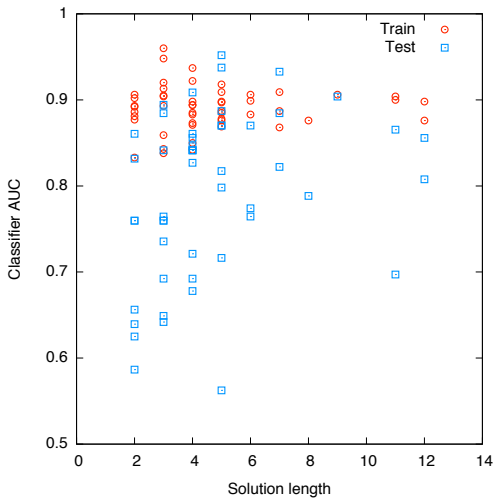
More generally, AUC is equivalent to the probability that a classifier will generate a higher output value for a patient than for a control subject. This relationship to probability means that AUC is easy to interpret, making it a popular metric in medicine (Kraemer et al., 2003).

4. Results

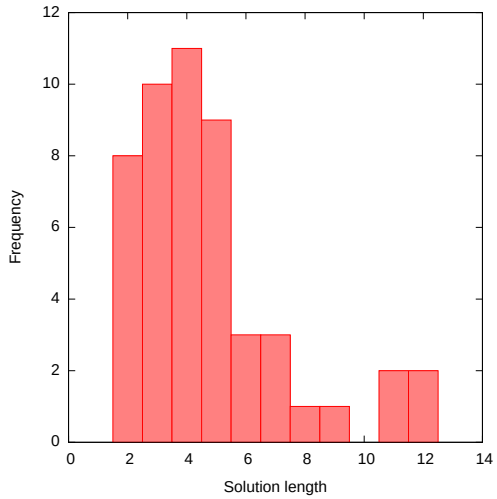
We first compared the ability of the evolutionary algorithm to find AMN and AGN classifiers which could correctly distinguish between PD patients and controls. Fig. 2 shows the resulting distribution of training and test scores for the best classifiers from each of 50 evolutionary runs for AMNs and AGNs with various function sets. This shows that, on average, AMNs perform better than AGNs in terms of both maximum and mean training and test set scores. It is also evident that the best ABNs with discrete maps perform better than the best ABNs containing only sigmoids. Moreover, sigmoidal AGNs—which closely resemble recurrent neural networks (Lones et al., 2013)—perform least well at this task.

The best overall classifier is a discrete map AMN with a test set AUC of 0.95. This corresponds to classification accuracies of around 90%. Whilst slightly lower than the 92-94% accuracy of diagnosis performed by experts in movement disorders, this is considerably higher than the diagnostic accuracies found in non-expert secondary care (75%) and community care (47%) (National Institute for Health and Clinical Excellence, 2006).

Most AMNs have high classification accuracy on the training set. The best AMN classifiers also generalise well to the test set, as shown by the upper part of the distributions in Fig. 2. However, the wide distribution of test set AUCs suggests that a significant proportion of the



(a) Training and test fitnesses



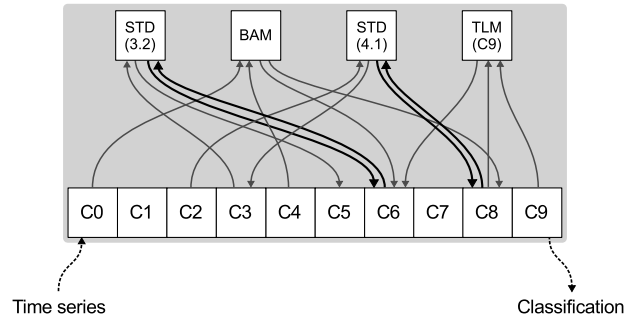
(b) Length distribution

Figure 3: Solution lengths of the best discrete map AMNs from 50 evolutionary runs, showing (a) the effect that solution length has upon generality, and (b) the frequency of evolved solution lengths.

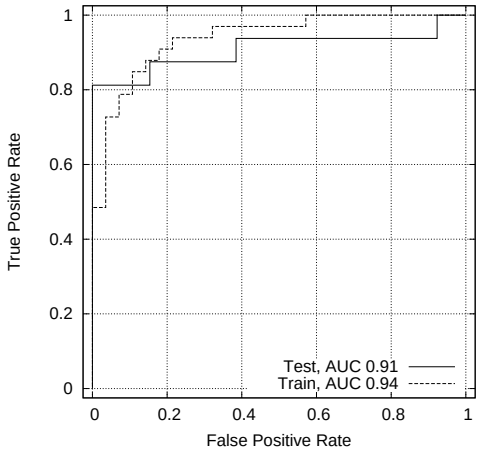
evolved AMNs do not generalise well to unseen data. Fig. 3a shows that this is largely due to a strong correlation between an AMN's length (i.e. number of enzymes) and the AMN's ability to generalise to the test set. This is exacerbated by the evolutionary algorithm's tendency to explore shorter solutions, as shown in Fig. 3b. This is a surprising result, since poor generality is often associated with over-learning, which is normally present in over-sized, rather than under-sized, classifiers.

5. Analysis of an Evolved Classifier

However, it is also notable that a number of relatively short AMNs did generalise well to the test set, suggesting that relatively small networks can effectively classify



(a) The internal structure of the AMN, showing how the discrete maps are coupled together through the chemical concentrations. STD=standard map (with value of k in parentheses), BAM=baker's map, TLM=tunable logistic map (chemical concentration used to set r in parentheses).



(b) ROC curves, showing the trade-off between false positive and false negative predictions as the threshold on the classifier's output, c_9, is varied.

Figure 4: An example of an evolved AMN.

Parkinson's disease. In order to gain a better understanding of how these evolved classifiers work, we analysed the behaviour of one of the best performing short AMNs, a length-4 AMN with an AUC of 0.92 across the whole data set. Fig. 4b shows ROC curves for the AMN's classification of the training and test sets. More generally, the purpose of this case study is to show that (i) we can get some insight into the workings of these evolved artefacts, (ii) they appear to be doing something sensible with respect to the problem domain, and (iii) they are doing something that is interesting from a computational and dynamical systems perspective.

5.1. Network structure

Fig. 4a shows the structure of the evolved network. It comprises four coupled discrete maps. The two standard maps both have k values that place them in a predominantly chaotic phase. The baker's map is always chaotic. The tunable logistic map has its parameter r set by the concentration of the output chemical c_9, whose typical range of values place the map in a cyclic phase. A knock-out analysis confirms that each map plays an important role in the network's overall behaviour: removal of each map in turn reduces the AUC to 0.71, 0.72, 0.78 and 0.61 (with respect to the left-to-right ordering shown in Fig. 4a). Hence, the collective dynamics of the network are determined by the coupled behaviour of three chaotic maps, one ordered map, and the incoming time series.

The network has 10 chemicals. The input time series is introduced a value at a time by setting the concentration of c_0. The network is iterated twice in between each input. Once the input series has been depleted, the network undergoes a settling period of 19 iterations. The classification is then read from the final concentration of c_9.

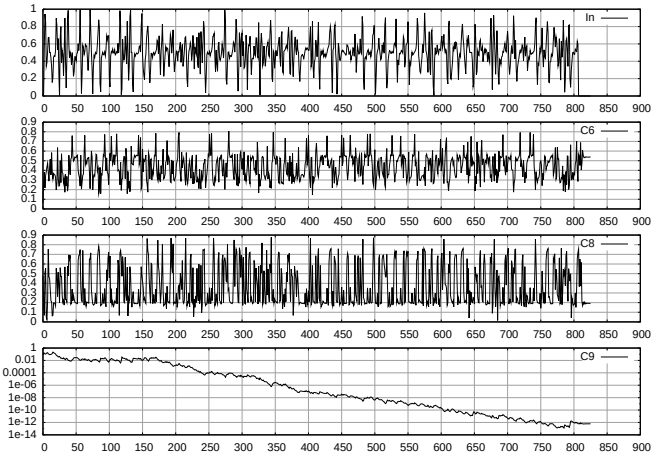
Four of the chemical concentrations are directly manipulated by the discrete maps. In addition, the concentrations of all chemicals are indirectly coupled by the network's conservation law. It is interesting to note that the output chemical c_9 is not directly updated by any of the maps. Instead, its concentration is determined indirectly as a function of dynamical processes operating upon the other chemicals. c_6 and c_8 are both recurrently connected and act as hubs within the network's dynamics. The concentrations of the other chemicals are derived from these, either directly or indirectly.

5.2. Dynamical behaviour

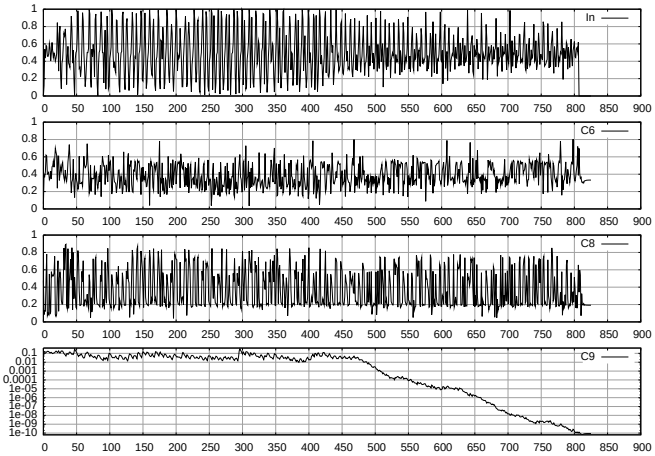
Fig. 5 shows the network's behaviour when it is stimulated with data from a variety of Parkinson's patients and disease-free controls, showing time series plots of chemical concentrations at the network's input (c_0), output (c_9) and its two hubs (c_6 and c_8). In general, the final value of the output chemical is lower for Parkinson's patients and higher for controls, following a logarithmic distribution with an optimal decision threshold around 1.0×10^{-2}.

The time series trends for c_9 suggest that the network is integrating local deviations from normality that are present in the input, using these to push down the concentration of the output. Examples are shown for four different patients, each displaying different tapping behaviours. In Fig. 5a, the patient's tapping is highly irregular, and this causes the output to be pushed down to one of the lowest levels seen within the data set. In Fig. 5b, the patient's tapping is only slightly impaired, but displays fatigue about half way through the time series. This also causes the output to be pushed down relatively low. Figs. 5c–d are less visibly impaired, with only minor deviations in terms of frequency and consistency. These exert less pressure upon the output level, but still cause it to be pushed down significantly below the decision threshold. Hence, there does not appear to be a single feature that is used by the network as a basis for classification, but rather a combination of features.

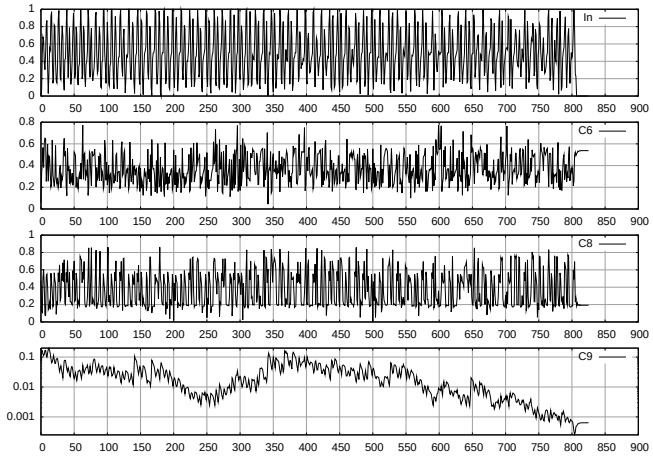
The two examples of controls (Figs. 5e–f) both appear visibly normal, and the output in both cases is considerably above the decision threshold. However, for Fig. 5f,



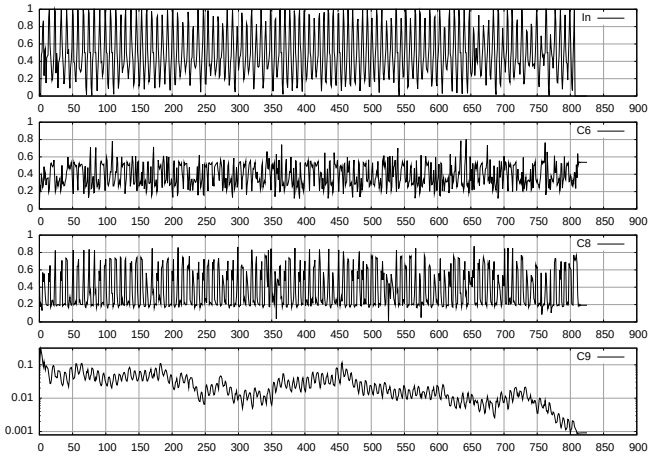
(a) Patient, irregular, output=6.0×10^{-13}



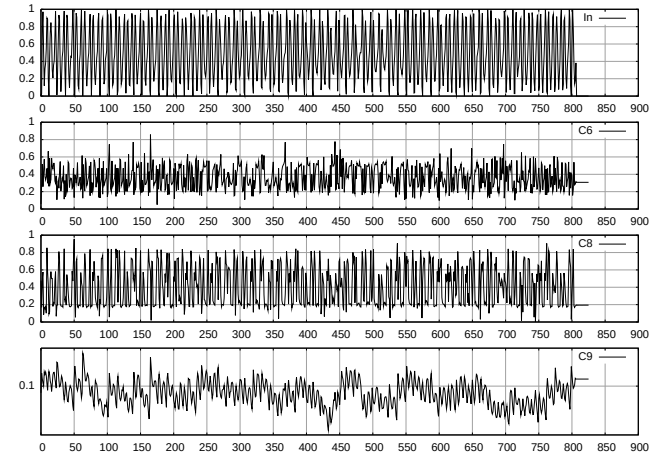
(b) Patient, amplitude fatiguing, output=8.8×10^{-11}



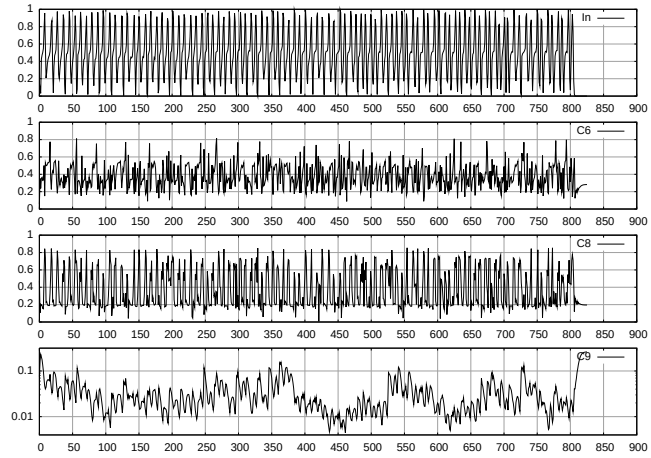
(c) Patient, slight impairment, output=6.4×10^{-4}



(d) Patient, slight impairment, output=9.1×10^{-4}

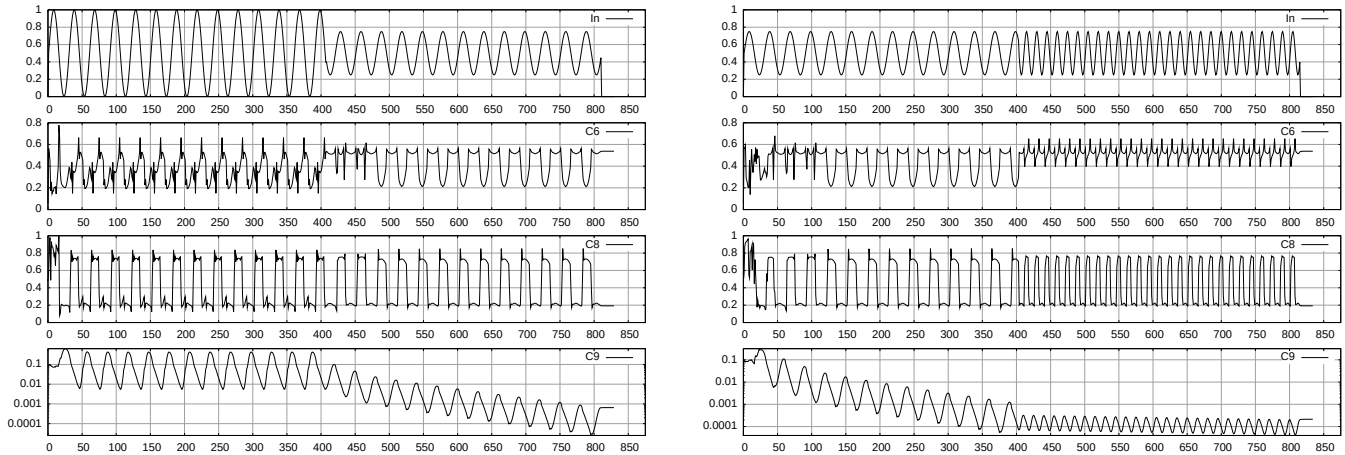


(e) Control, normal, output=1.0×10^{-1}



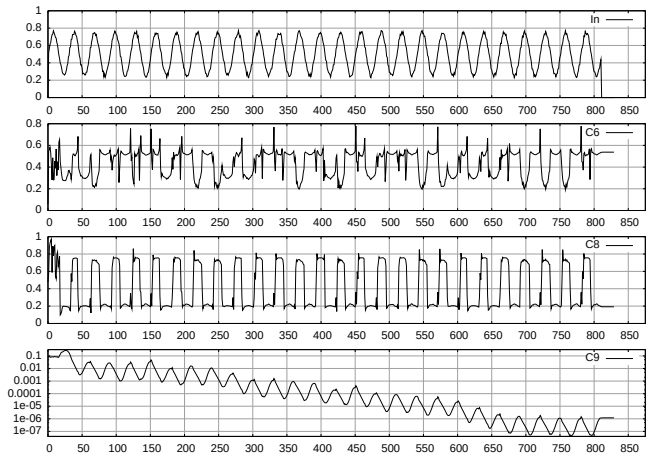
(f) Control, normal, output=2.6×10^{-1}

Figure 5: Example of the evolved AMN in Fig. 4 processing movement data (In) from subjects (a–d) with and (e–f) without Parkinson's. The lower three plots in each case show how chemical concentrations at important points in the network change as the input sequence is processed. c_9 (lowest plot) is the designated output chemical, whose final concentration is interpreted as the classifier's output. Note that the plots for c_9 use a logarithmic axis. The examples are in reverse order of output magnitude. The optimal threshold for separating patients from controls is $\sim 1.0 \times 10^{-2}$, assuming equal weight given to specificity and sensitivity.

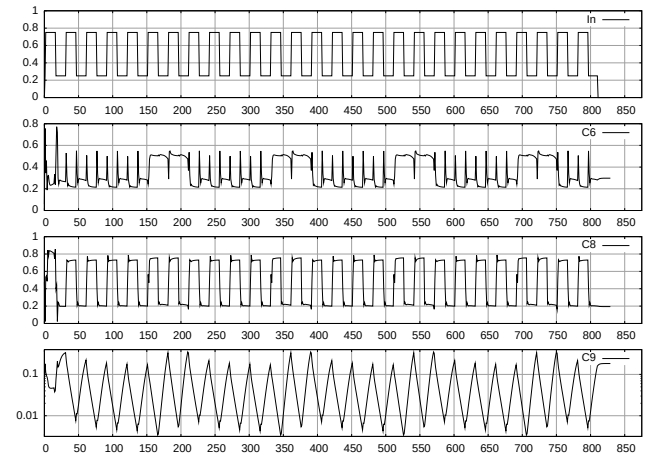


(a) Amplitude response (changing from 0.5 to 0.25)

(b) Frequency response (changing from 1Hz to 2Hz)



(c) Added noise



(d) Square wave

Figure 6: Perturbation analysis of the evolved AMN in Fig. 4, showing the network's time series behaviour when perturbed with waveforms of different shape, amplitude and frequency.

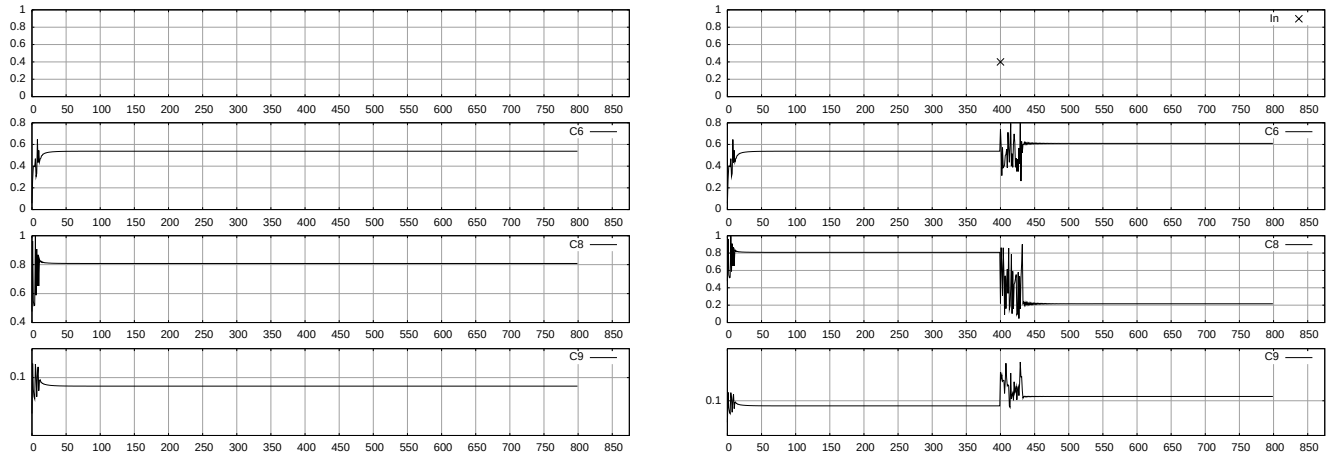
the output level is near, and sometimes below, the threshold for much of the network's time evolution. The output classification, in this case, is determined by the network's behaviour during the final settling period, which brings it well above the threshold. This is a relatively uncommon case, but it does suggest that the network is not merely counting occurrences of a particular pattern but also taking into account its global dynamics.

As mentioned earlier, the concentration of the output chemical is a result of the dynamical processes occurring at c_6 and c_8. A visual inspection of the time series plots suggests that there is a relationship between the concentrations of these chemicals and the input data, but the relationship appears to be highly non-linear. This is unsurprising, given that these concentrations are a result of interactions between the input signal and 4 non-linear maps, 3 of which are chaotic. However, it provides little insight into the computational behaviour of the network.

5.3. Waveform Response

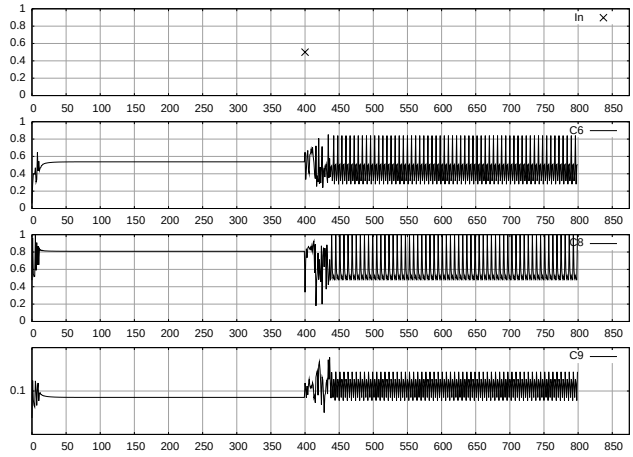
Another way of understanding the dynamical properties of the network is to remove the input signal and analyse the network's response when perturbed using known functions.

Fig. 6 shows how the network responds to a variety of waveforms, illustrating the effect that amplitude, frequency, noise and shape have upon the network's output response. Changes of amplitude have the largest effect in general, with an exponential downward trend in the output level as the amplitude drops below ∼0.35, for a 1Hz sine wave (Fig. 6a). This supports the interpretation that the AMN responds to the amplitude fatiguing sometimes observed in PD patients. Frequency also has an effect upon the output level, although less so than amplitude. For instance, in Fig. 6b, a doubling in frequency significantly decreases the downward trend. This may indicate a response to bradykinesia or tremor components, both of which can affect a subject's tapping frequency. The shape of the waveform also has a significant influence upon the output response. In Fig. 6c, we deformed the sine wave

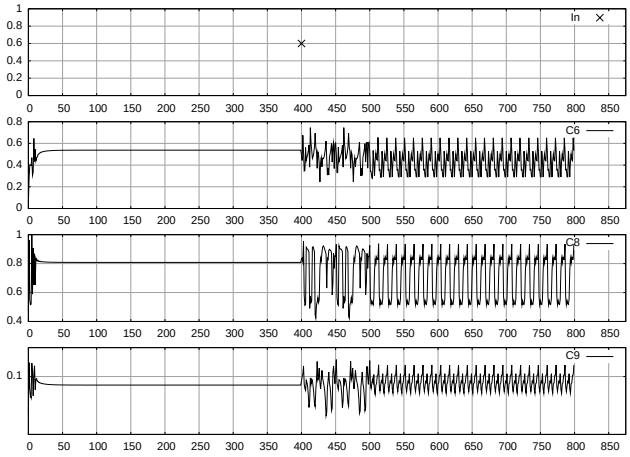


(a) No input

(b) Impulse magnitude 0.4



(c) Impulse magnitude 0.5



(d) Impulse magnitude 0.6

Figure 7: Perturbation analysis of the evolved AMN in Fig. 4, showing the network's time series behaviour (a) when input is removed, and (b–d) when the network is perturbed with an impulse approximately half way through the network's time evolution.

with a small amount of noise. In response, the output level decreased by another order of magnitude, with respect to the undistorted sine wave. This could indicate a response to the irregular movements of PD patients.

It is notable that the dynamical activity of the network is generally cyclic, with the same frequency as the stimulating waveform. The shape of the waveform at c_8 is relatively stable. The waveform at c_6, by comparison, changes shape considerably depending upon the properties of the stimulating waveform. Its response to noise is particularly interesting, with small deformations of the sine wave leading to large spikes in the time series of c_6. These spikes presumably propagate through the network, resulting in the lowering of the output response.

This analysis suggests that the network responds to a combination of signal properties—including amplitude, frequency and noise—which supports the analysis from real human data. However, the acceleration waveforms produced by humans are not sinusoidal, so there is a limit to the generality of lessons learnt from this analysis. The response to a square wave (Fig. 6d), for instance, is qualitatively different, with c_6 displaying a cyclic response over a longer time period, in addition to the stimulating frequency.

5.4. Impulse Response

To further characterise the effect of noise, we carried out an analysis of the network's impulse response. Fig. 7a shows the dynamics of the network in the absence of an input signal, i.e. when the concentration of c_0 is not manipulated externally. After an initial transient period, during which the network displays chaotic behaviour, the network converges to a fixed-point attractor. Figs. 7b–d show what happens if an impulse is then introduced to the network after it has reached this steady-state, i.e. when the concentration of c_0 is set to a particular value for a single time step. In all three cases, this causes the network to ring chaotically, then synchronise and settle to an ordered attractor state.

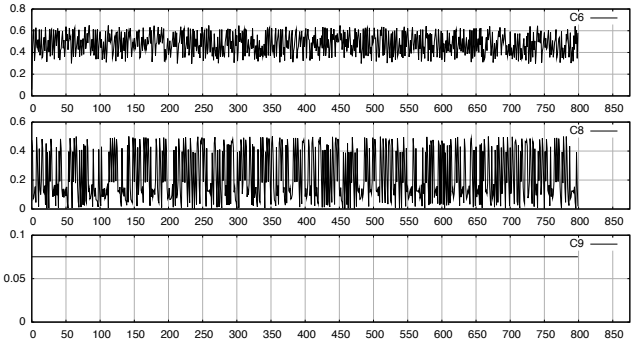


Figure 8: Time series behaviour of the evolved AMN in Fig. 4 with no input and no conservation.

It is particularly interesting to note that relatively small changes to the impulse magnitude leads to the network entering different attractors. Overall, the network's responses suggest that, unperturbed, it enters a state that is stable yet highly sensitive to input. Small changes in the input then lead to relatively large dynamical responses. This behaviour has certain commonalities with reservoir computers (e.g. liquid state machines (Maass et al., 2002), echo state networks (Jaeger, 2003)), where the general aim is to create a dynamical system with dampened internal dynamics but which maximally separates incoming signals. This is despite the large structural differences between reservoir computers and ABNs: reservoir computers are large and only the linear output nodes are trained; ABNs are small, and all the parameters are trained. In the case of reservoir computers, damping is a property built into the construction of the dynamical system; in the case of ABNs, it is a property that is evolved.

Chaos is an innately controllable phenomenon, and we can hypothesis that the sensitivity of this particular ABN is a consequence of using chaotic discrete maps. This may explain why discrete map-based AMNs work better than sigmoidal AMNs for this particular problem: although networks of sigmoids are capable of expressing chaotic dynamics, it is much easier to evolve chaotic behaviour when chaos is readily expressed by individual primitive functions. However, chaotic behaviour can also be a disruptive phenomenon. In this case, conservation appears to play an important role in controlling chaos. As an illustration of this, Fig. 8 shows the time evolution of the network when the conservation law is not used. Unlike in Fig. 7a, the chaotic dynamics do not become dampened, and dominate the time evolution of the connected components of the network.

6. Conclusions

In this paper, we have shown that artificial biochemical networks can be used to recognise abnormal motor function associated with Parkinson's disease. The evolved classifiers perform an objective diagnosis based upon data collected from simple movement tasks, and have accuracies comparable to trained clinicians. Analysis of the classifiers suggests that diagnosis can be performed by relatively simple evolved networks, and that chaotic dynamics may play an interesting role. In future work, we hope to investigate whether this approach can also be applied to other forms of neurological disorder, such as Alzheimer's and Huntington's disease.

Acknowledgements

This work was funded, in part, by the EPSRC grant "Artificial Biochemical Networks: Computational Models and Architectures" (ref: EP/F060041/1). The authors would also like to thank Susan Stepney, Leo Caves, Alexander Turner and Luis Fuente for their contributions to discussions about this work.

References

Andersson, C., Nordahl, M., 1998. Evolving coupled map lattices for computation. In: Banzhaf, W., et al. (Eds.), Genetic Programming, Proc. 1st European Workshop on Genetic Programming. Vol. 1391 of Lecture Notes in Computer Science. Springer, pp. 151–162.

Arnold, V., Avez, A., 1968. Ergodic problems in classical mechanics. Benjamin, New York.

Bajaj, N. P. S., Gontu, V., Birchall, J., Patterson, J., Grosset, D. G., Lees, A. J., Nov 2010. Accuracy of clinical diagnosis in tremulous parkinsonian patients: a blinded video study. J Neurol Neurosurg Psychiatry 81 (11), 1223–1228.

Chirikov, B. V., 1969. Research concerning the theory of nonlinear resonance and stochasticity. Tech. rep., Institute of Nuclear Physics, Novosibirsk.

Dittrich, P., Ziegler, J., Banzhaf, W., 2001. Artificial chemistries—a review. Artificial Life 7, 225–275.

Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recognition Letters 27, 861–874.

Hüsken, M., Stagge, P., 2003. Recurrent neural networks for time series classification. Neurocomputing 50, 223–235.

Jaeger, H., 2003. Adaptive nonlinear system identification with echo state networks. In: Becker, S., Thrun, S., Obermayer, K. (Eds.), Advances in Neural Information Processing Systems 15. MIT Press, Cambridge, MA, pp. 593–600.

Kaneko, K., 1992. Overview of coupled map lattices. Chaos 2 (3), 279 – 282.

Kauffman, S. A., Mar 1969. Metabolic stability and epigenesis in randomly constructed genetic nets. J Theor Biol 22 (3), 437–467.

Kraemer, H. C., Morgan, G. A., Leech, N. L., Gliner, J. A., Vaske, J. J., Harmon, R. J., Dec 2003. Measures of clinical significance. J Am Acad Child Adolesc Psychiatry 42 (12), 1524–1529.

Levine, C. B., Fahrbach, K. R., Siderowf, A. D., Estok, R. P., Ludensky, V. M., Ross, S. D., May 2003. Diagnosis and treatment of Parkinson's disease: a systematic review of the literature. Evid Rep Technol Assess (Summ) (57), 1–4.

Lones, M. A., Fuente, L. A., Turner, A. P., Caves, L. S. D., Stepney, S., Smith, S. L., Tyrrell, A. M., 2013. Artificial biochemical networks: Evolving dynamical systems to control dynamical systems. Evolutionary Computation, IEEE Transactions onTo appear.

Lones, M. A., Tyrrell, A. M., Stepney, S., Caves, L. S. D., 2010. Controlling complex dynamics with artificial biochemical networks. In: Esparcia-Alcázar, A. I., et al. (Eds.), Proc. 2010 European Conference on Genetic Programming (EuroGP 2010). Vol. 6021 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 159–170.

Lones, M. A., Tyrrell, A. M., Stepney, S., Caves, L. S. D., August 2011. Controlling legged robots with coupled artificial biochemical networks. In: Lenaerts, T., et al. (Eds.), Advances in Artificial Life, ECAL 2011: Proc. 11th European Conference on the Synthesis and Simulation of Living Systems. MIT Press, pp. 465–472.

Luke, S., 2009. Essentials of Metaheuristics. Lulu, available for free at http://cs.gmu.edu/∼sean/book/metaheuristics/.

Maass, W., Natschläger, T., Markram, H., 2002. Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural Computation 14 (11), 2531–2560.

May, R. M., 1976. Simple mathematical models with very complicated dynamics. Nature 261, 459–467.

National Institute for Health and Clinical Excellence, 2006. Parkinson's disease: diagnosis and management in primary and secondary care. Royal College of Physicians.
URL `http://www.nice.org.uk/CG035`

Păun, Gh., 2000. Computing with membranes. Journal of Computer and System Sciences 61 (1), 108–143.

Tél, T., Gruiz, M., 2006. Chaotic Dynamics: An Introduction Based on Classical Mechanics. Cambridge Press.

Verplancke, T., Van Looy, S., Steurbaut, K., Benoit, D., De Turck, F., De Moor, G., Decruyenaere, J., 2010. A novel time series analysis approach for prediction of dialysis in critically ill patients using echo-state networks. BMC Medical Informatics and Decision Making 10 (1), 4.