Department of Computer Science,
University of Otago

Te Whare Wānanga o Otāgo

Technical Report OUCS-2013-08

Consistency of polychronous neural group activation supports a role as an underlying mechanism for representation and memory: detailed methods and results

Authors:

Mira Guise, Alistair Knott, Lubica Benuskova

Department of Computer Science, University of Otago, New Zealand

Status: This is a document in support of a manuscript submitted to ICONIP 2013.

Department of Computer Science,
University of Otago, PO Box 56, Dunedin, Otago, New Zealand

http://www.cs.otago.ac.nz/research/techreports.php

Consistency of polychronous neural group activation supports a role as an underlying mechanism for representation and memory: detailed methods and results

Mira Guise*, Alistair Knott, Lubica Benuskova

Dept of Computer Science, University of Otago, Dunedin

Abstract

Izhikevich (2006a) has proposed that certain strongly connected groups of neurons known as polychronous neural groups (or PNGs) might provide the neural basis for representation and memory. Polychronous groups exist in large numbers within the connection graph of a spiking neural network, providing a large repertoire of structures that can potentially match an external stimulus (Izhikevich, 2006a; Izhikevich et al., 2004). In this paper we examine some of the requirements of a representational system and test the idea of PNGs as the underlying mechanism against one of these requirements, the requirement for consistency in the neural response to stimuli. The results provide preliminary evidence for consistency of PNG activation in response to known stimuli, although these results are limited by problems with the current methods for detecting PNG activation.

Keywords: spiking network, polychronous neural group, activation, representation, memory

1. Introduction

It is widely assumed that synaptic plasticity provides the neural basis for long-term memory in the brain (Abraham, 2008; Caporale and Dan, 2008; Martin et al., 2000) although the precise nature of the underlying representation is still unclear (Caroni et al., 2012). Izhikevich (2006a) has proposed that certain strongly connected groups of neurons known as polychronous neural groups (or PNGs) might provide this representational mechanism. Polychronous groups exist in large numbers within the connection graph of a spiking neural network, providing a large repertoire of structures that can potentially match an external stimulus (Izhikevich, 2006a; Izhikevich et al., 2004). In this report we examine some of the requirements of a representational system and test the idea of PNGs as a mechanism of representation

*Corresponding author at: Department of Computer Science, University of Otago, PO Box 56, Dunedin 9054, New Zealand

Email address: mguise@cs.otago.ac.nz (Mira Guise)

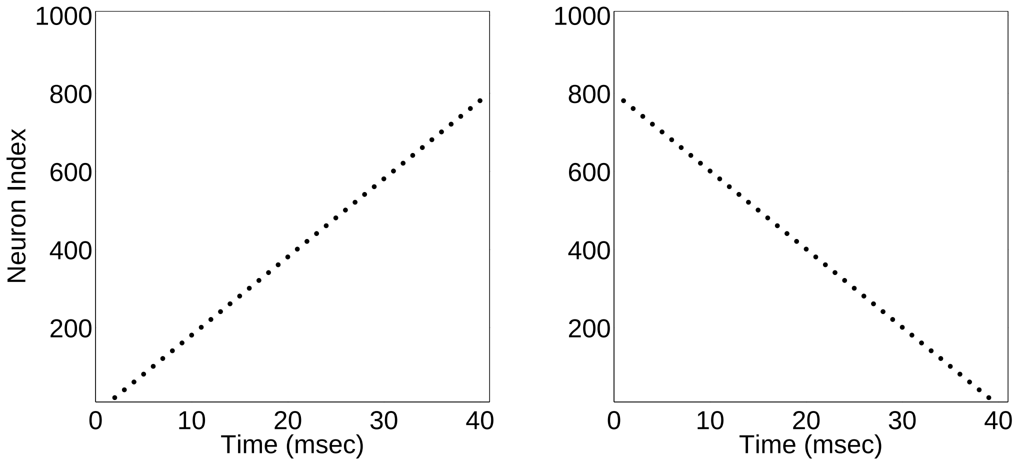
against one of these requirements, the requirement for consistency in the neural response to stimuli.

Polychronous groups exist as distributions of synaptic weights and axonal lengths in the network structure. In order for these *structural PNGs* to provide the basis of a representational system they must be capable of activation in response to stimuli. Activation of structural PNGs is limited to those that match the input pattern (Izhikevich, 2006a) and produces distinct spatio-temporal patterns in the network firing data known as *activated PNGs* (Martinez and Paugam-Moisy, 2009). When activated, the neurons in the polychronous group are said to *polychronize* in a causal chain of firing events that generates a reproducible and precisely timed response pattern (Izhikevich, 2006a; Izhikevich et al., 2004). Analysis of the firing response to stimuli should therefore allow the detection of PNG activation and may also allow the original stimulus to be inferred.

Izhikevich (2006a) observed that the number of PNGs in the network is typically many times larger than the number of neurons. Given this large repertoire of structural PNGs, how might we use it to build a representational system? Several attributes immediately present themselves as necessary for a robust system and we will refer to these with the terms *selectivity*, *consistency*, *stability* and *capacity*. A *selective* system produces a neural response to a stimulus that is sufficiently specific to allow the unique identification of the stimulus. A *consistent* system is able to dependably produce PNG activation on every presentation of the stimulus. A *stable* system is able to maintain a long-term representation, and a system with good *capacity* allows a biologically plausible number of representations.

The selectivity of the neural response to repeated stimulation has been previously examined by Izhikevich in an experiment that tracked the evolution of polychronous groups in response to one of two input patterns (Izhikevich, 2006a). Specific structural groups were found to evolve in response to each input pattern although only a subset of these groups was activated on each presentation of the pattern. Importantly, different groups were activated for each pattern, suggesting that the underlying structural groups might provide a unique long-term representation of each pattern.

Although this experiment provided some initial evidence in support of selective PNG activation, it did not address any of the other attributes necessary for a representational system based on polychronous groups. In addition, the method used in Izhikevich (2006a) for measuring PNG activation is not described, providing some hurdles to the reproduction of these results. The experiment described below employs a template matching technique for detecting PNG activation that is described in detail. In the remainder of this report we will focus on the requirement for a *consistent* representational sys-



(a) Ascending Pattern (b) Descending Pattern

Figure 1: The ascending and descending patterns: each spatio-temporal input pattern is composed of 40 firing events. Note that both patterns share the same neurons, differing only in the temporal order of their firing events.

tem, using the pattern-specific activation of polychronous groups to measure the dependability of the neural response to known stimuli.

2. Methods

Twenty independent networks were created for these experiments, each composed of 1000 Izhikevich neurons (800 excitatory and 200 inhibitory) with parameters as described in Izhikevich (2006a). The networks were matured for two hours by exposure to 1 Hz random input generated by a Poisson process. Following maturation, the networks were trained on one of two input patterns or were left untrained. The current experiments reproduce the few known details of the repeated stimulation experiment described in Izhikevich (2006a), namely a twenty minute training period, and the use of an *ascending* or *descending* input pattern as the stimulus (see Fig. 1).

The technique used by Izhikevich (2006a) for detecting PNG activations in the firing data was not described and therefore needed to be redeveloped from the beginning. It was clear that this technique needed to discriminate pattern-specific PNG activations from unrelated PNG activations, and from other spiking events generated by the network. The original method was assumed to make use of the Izhikevich search algorithms (Izhikevich, 2006b) to find structural PNGs in the network, suggesting the use of a template matching technique for the detection of PNG activation.

This assumed template matching technique is reproduced as follows: first, a network is trained with a specified input pattern; next, pattern-specific structural PNGs are isolated from the network using one of the PNG search algorithms; and finally, the isolated PNGs are used as spatio-temporal templates to match the firing data. For convenience, the experiment is split into multiple phases: in an initial *training phase*, the network is repeatedly stimulated with the ascending or descending pattern at 5 or 25 Hz for twenty minutes; in the following *test phase* of the experiment, the network is stimulated with the same ascending and descending patterns at 1 Hz and pattern-specific templates isolated during the training phase are used to probe for group activation.

Throughout the training phase, the isolation of pattern-specific templates occurs at one minute intervals over the course of training. Note that the identification of *pattern-specific* templates in the training phase requires the use of a modified version of the PNG search algorithm that confines the search to groups that are triggered by input pattern firing events. Combinations of these firing events from the input pattern are tested for their ability to initiate PNG activation (see the Detailed Methods section for more details). For performance reasons, the PNG search algorithm limits these triggering spatio-temporal patterns to combinations of three firing events, a *triplet*.

The test phase involves scanning the stream of firing events generated by the stimulated network for template matches. For each temporal offset in the network firing data, each of the templates is matched in sequence (using a matching threshold of 50% and jitter of $\pm$ 2 milliseconds) and successful matches are saved to a file. The use of a 1 Hz stimulation frequency in the test phase creates a well-defined temporal frame for each stimulus and its response. Stimulus onset occurs at $t = 0$ in each one second *response frame*, and the remainder of the frame has sufficient temporal length to include all of the firing events in the resulting neural response. Note that a 1 Hz random background pattern is also presented throughout each test period. Additional method details can be found in the Detailed Methods section.

3. Results

Together the training and testing phases of the experiment produce a large set of data that supports multiple analyses. Training phase data provides a view of the evolution of structural PNGs in response to the stimulus, while test phase data provides a snapshot of the process of PNG activation. Figure 2 uses a combination of both datasets to show a selection of the matching templates following low-intensity (1 Hz) test stimulation of a network.

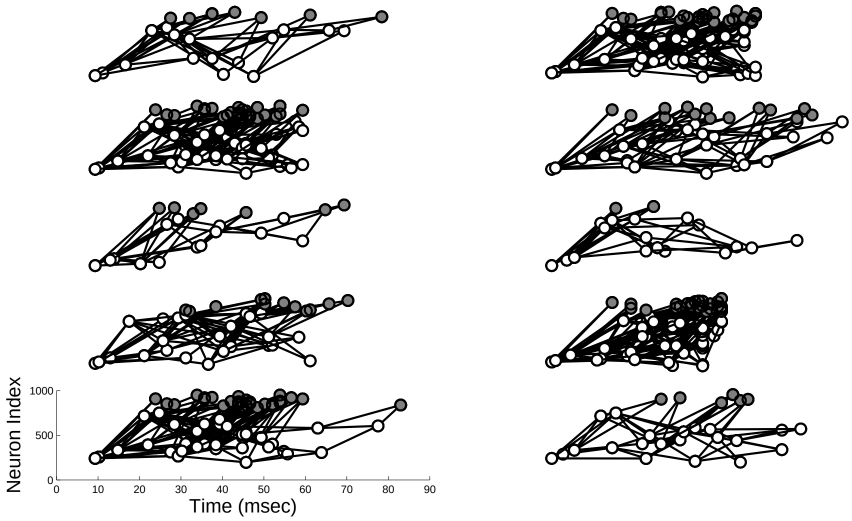


Figure 2: A selection of ten templates that match the firing data following stimulation with the ascending input pattern. The x- and y-axes for each template represent time in milliseconds and neuron index respectively (the y-axis is ordered so that inhibitory neurons are at the top of the graph). Nodes depict firing events generated by excitatory or inhibitory neurons and are drawn using either open circles (excitatory neurons) or gray-filled circles (inhibitory neurons). Lines between nodes represent causal connections between firing events. The network was trained on the ascending pattern at 5 Hz.

These matching templates are sampled from a larger pool of pattern-specific templates whose initial firing event triplets correspond to some triplet combination from the ascending input pattern. Each of the templates in Figure 2 therefore has an upward-sloping initial triplet reflecting its isolation from a network trained on the ascending pattern. Each group consists of multiple convergent connections that support the propagation of neural firing across the members of the group before terminating at an inhibitory neuron (gray-filled circles).

Temporal alignment of just these initial triplets (and with all other firing events removed) produces sloping firing patterns that can be seen in Fig. 3. The gray-scale intensity in this figure encodes the number of times the corresponding firing event acted as a trigger for the initiation of PNG activation, where activation was measured by the number of matching templates accumulated across twenty independent networks. The figure therefore provides a picture of which of the input pattern firing events succeeded or failed at

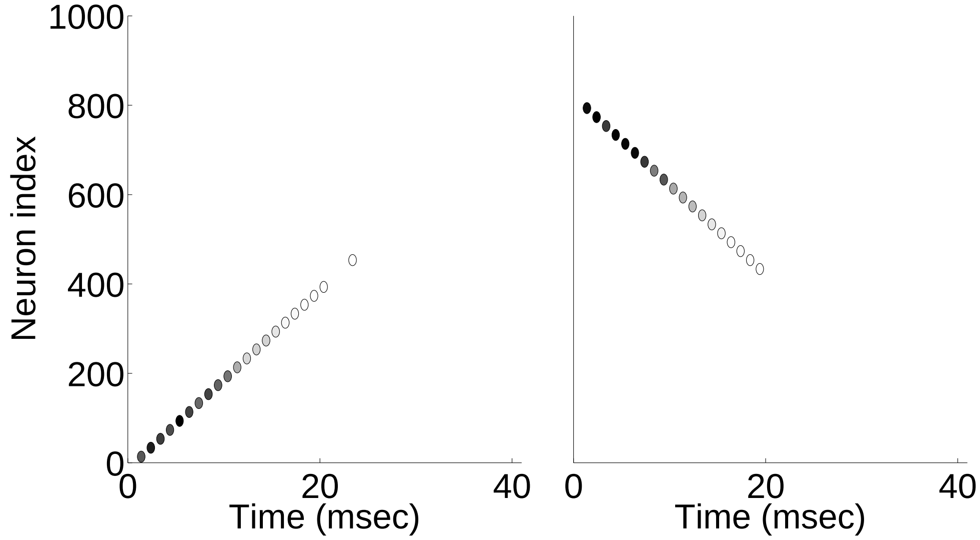


Figure 3: The initial triplets from all templates that match the ascending input pattern (left) or the descending input pattern (right). The first three firing events from each matching template were extracted and aligned in order to show the coverage of the input pattern firing events. Firing events are represented by filled circles; the intensity of the fill color for each firing event represents the number of templates that matched PNG activations triggered by the event. This number, accumulated across twenty independent networks, is greatest in the early stages of each input pattern (darker fill color) and decreases in later stages of the input pattern (lighter fill color). The missing firing events in the later stages correspond to input pattern firing events that failed to initiate a group response during the test period.

initiating PNG activation. As we can see, many of the forty firing events that make up each input pattern failed to initiate a responding group over the ten minutes (six hundred response frames) of the testing phase. Significantly, the majority of these failures are clustered in the later stages of the input pattern, suggesting that group response is concentrated on the early part of each stimulus presentation.

Nevertheless, the PNG activation response as a whole shows a high degree of consistency. Figure 4 shows the activation response of forty networks (twenty trained and twenty untrained) in the first one hundred seconds of the ten minute test run (only the first 100 of 600 response frames are shown in Fig. 4). The stimulus is presented at the start of each frame and any templates that match the firing events in the remainder of the frame are taken as evidence of PNG activation. Each row in Fig. 4 represents a single network and is divided into one hundred segments representing each of the one hundred response frames. The presence of a filled circle in each segment indicates the detection of a PNG activation response in the corresponding response frame. If there was no response, or the method was unable to detect the response, the segment is left empty.

The first 25 frames in this experiment used the ascending pattern, the next 25 used the descending pattern, the third group of 25 frames repeated the use of the ascending pattern, and in the final 25 frames no input pattern was provided (the null pattern). Using a combined pool of all templates to measure the PNG activation response, the twenty trained networks at the top of Fig. 4 show a consistent response to the ascending pattern but little or no response to the descending pattern or the null pattern. In contrast, the twenty untrained networks at the bottom of Fig. 4 show only sporadic activation and no apparent correlation with the type of input pattern. Comparing the activation response of the trained networks with the response of the untrained networks, we see a high degree of consistency in the response to the ascending pattern only where the network has been previously trained on the ascending pattern.

Using the template matching method there is a trade-off in specificity versus sensitivity in using either a single template, or a combined group of templates. A single template has a lower probability of matching the network's firing response to a random spatio-temporal input pattern, and is therefore more specific than a set of templates. However, a combined pool of templates is more likely to match some part of the firing response providing greater sensitivity. In the next figure we compare the activation results of the single best template for each network with the results using a combined pool of all templates for that network. The best template for each network is the template with the most matches over the entire test period.

The PNG activation response using a combined pool of all templates is shown at the top of Fig. 5 while the response to the best single template is shown at the bottom of the figure. The activation response in this figure was measured on networks that had been trained and tested on the ascending pattern, although a similar result is seen for networks that were trained and tested on the descending pattern (results not shown). The results for the combined templates suggest a high degree of response consistency across all networks, with PNG activation detected in nearly every response frame. Consistency in the activation response can also be seen in some of the networks when PNG activation is measured using the single best template, although there is considerable variation between networks.

In order to quantify the results it is useful to calculate the number of response frames that record a template match as a proportion of the total

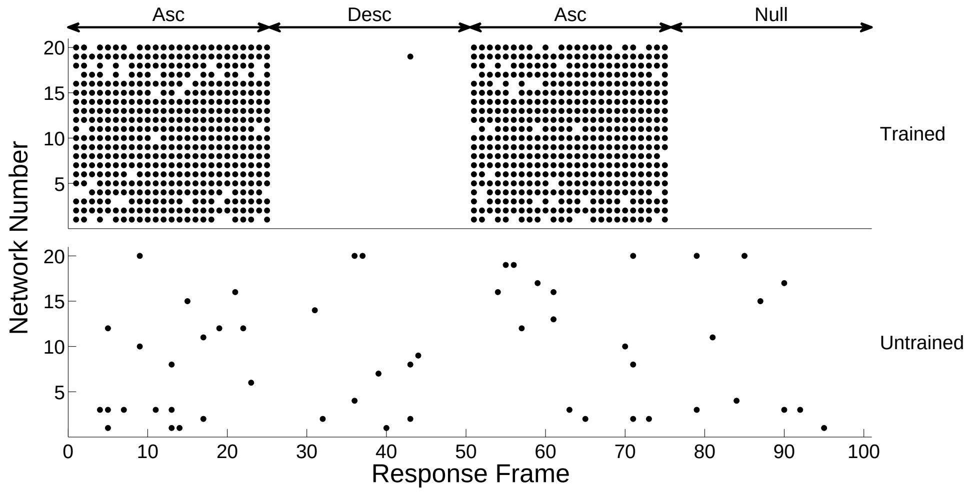


Figure 4: The PNG activation response of twenty trained networks and twenty untrained networks over one hundred response frames. A single stimulus was provided in each frame and the network response was measured. A filled circle represents a positive response to the stimulus while an empty space denotes a lack of response. Four different input patterns were used as stimuli in this experiment: the stimulus for the first and third quarter of the one hundred frames was the ascending pattern and the stimulus for the second quarter was the descending pattern. No stimulus was provided in the fourth quarter (null pattern) although a 1 Hz random background was present in all frames. The response in each frame was measured using the template matching method and a combined pool of all templates. The top figure shows the measured response for a network trained on the ascending pattern at 5 Hz and the bottom figure shows the result using an untrained network. The trained networks in the top figure were derived from the untrained networks in the corresponding row of the bottom figure.

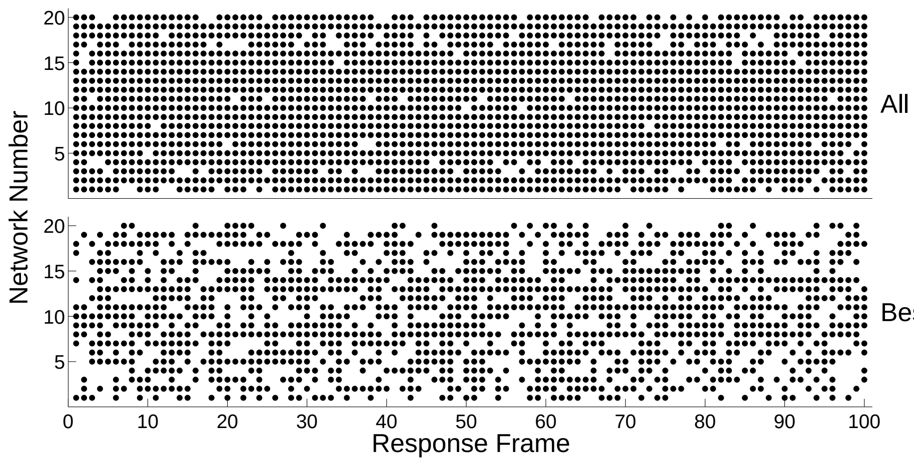


Figure 5: Measurement of the PNG activation response using either a combined pool of all templates or the single best template for each network. The PNG activation response to the ascending input pattern was measured over one hundred frames on twenty networks trained on the ascending pattern at 5 Hz. The top figure shows the measured response using a combined pool of all templates and the bottom figure shows the result using a single template, the best (highest responding) template for each network. Test data was generated using the twenty trained networks used for Fig. 4. A filled circle represents a positive response to the stimulus while an empty space denotes a lack of response.

number of frames (the *template match ratio*). When calculated for each frame, the match ratio provides a measure of the empirical likelihood of a PNG activation given the stimulus. However, the template match ratio can also be calculated at a finer temporal resolution, providing an empirical measure of the likelihood of a response at each temporal offset in the response frame following the stimulus. Using this procedure provides some insight into the temporal evolution of PNG activation following the stimulus and allows the delay between stimulus onset and PNG activation to be determined.

To compute this measure, each one second response frame is sliced into 1000 consecutive sub-frames and the number of template matches at each one millisecond sub-frame is counted. The template match ratio for each offset is then computed by aggregating the number of matches for each offset across all response frames. Using this procedure we expect to see an isolated peak in the number of matches at a short delay following the stimulus at time $t = 0$, reflecting the transient activation of a responding PNG. However, due to limitations in the template matching method the delay can only be calculated to within half the length of each template (i.e. ± 15 milliseconds), depending on where on each template the match occurs.

Figure 6 shows the template match ratios for each network distributed over the first twenty sub-frames of each response frame. As predicted there is an isolated peak that consistently occurs in the first ten milliseconds following the stimulus. Within this small temporal window the likelihood of a template match typically reaches 50% or more, indicating that PNG activation is in full swing. As PNG activation comes to an end, the likelihood of a template match decreases to zero and remains at zero for the remainder of the response frame.

We can also measure the template match ratio at the level of each frame. The proportion of response frames in which a PNG activation is detected provides an overall measure of the empirical likelihood of a response given the presentation of a known stimulus at the start of each frame. Figure 7 shows this response likelihood for each of the trained networks in Fig. 5. As before the response measured using a combined pool of templates is shown in the top of the figure, and the response as measured using the single best template is shown in the bottom of the figure. The template match ratio as a measure of the response likelihood is represented by the vertical axis. Although the ratios computed from the single best template are quite variable, many of the networks respond with a near perfect consistency (i.e. a template match ratio of 1.0) when measured using combined templates.

Although these positive results support the consistency of PNG activation, it is worth noting that the majority of templates are ineffective in matching the firing data. Here, we define an effective template as one that is able to

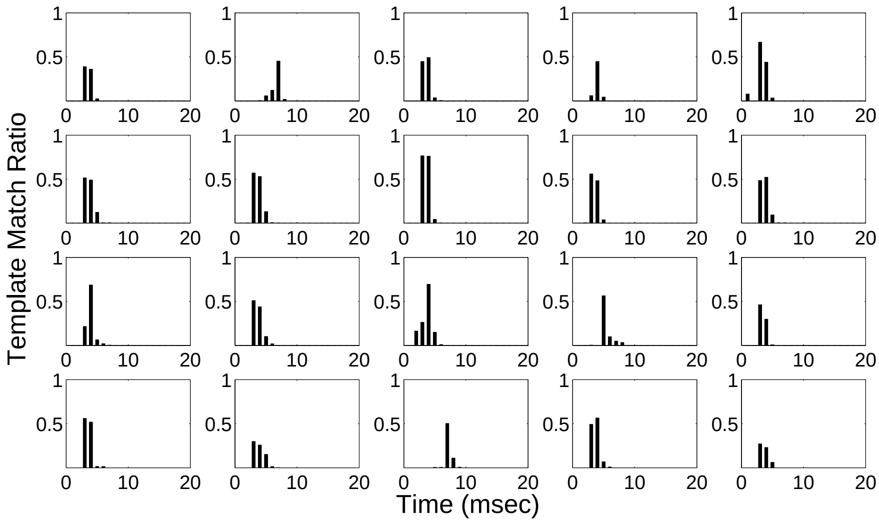


Figure 6: Template match ratios distributed over each one second frame for each of twenty independent networks. The template match ratio was computed for each one millisecond slot in the response frame, accumulated over multiple frames. The response for each network is confined to the first ten milliseconds following the stimulus and therefore only the first twenty milliseconds of the frame are shown. The network ordering is top-to-bottom and left-to-right allowing comparison of individual networks across Figs. 5, 6 and 7. Each network was trained on the ascending pattern at 5 Hz and tested on the same pattern at 1 Hz. Note that the data shown here does not show the precise temporal offset of PNG activation relative to stimulus onset as this will depend on whether each template matches at the beginning or near the end of the template.

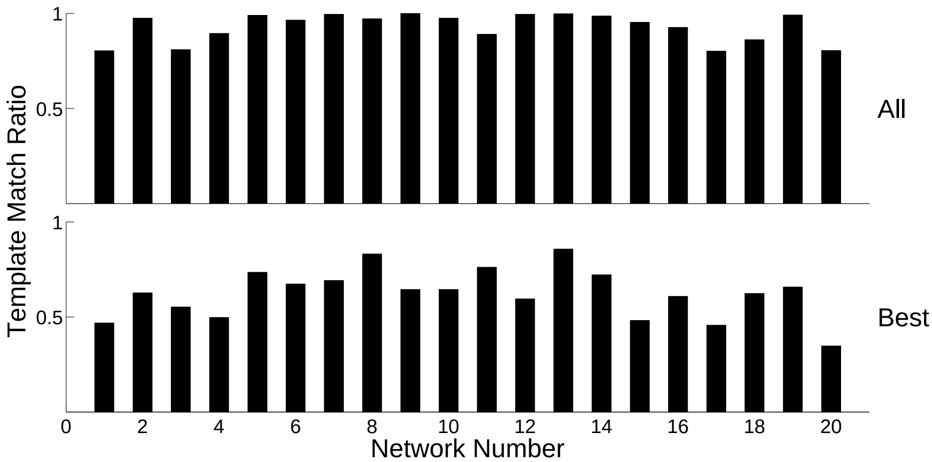


Figure 7: The template match ratio for each of the trained networks in Fig. 5. The template match ratio measures the number of frames in which a template matched as a proportion of the total number of frames. A template match ratio of one indicates perfect consistency in the response to the repeated stimulus. Each network was trained on the ascending pattern at 5 Hz and tested on the same pattern at 1 Hz.

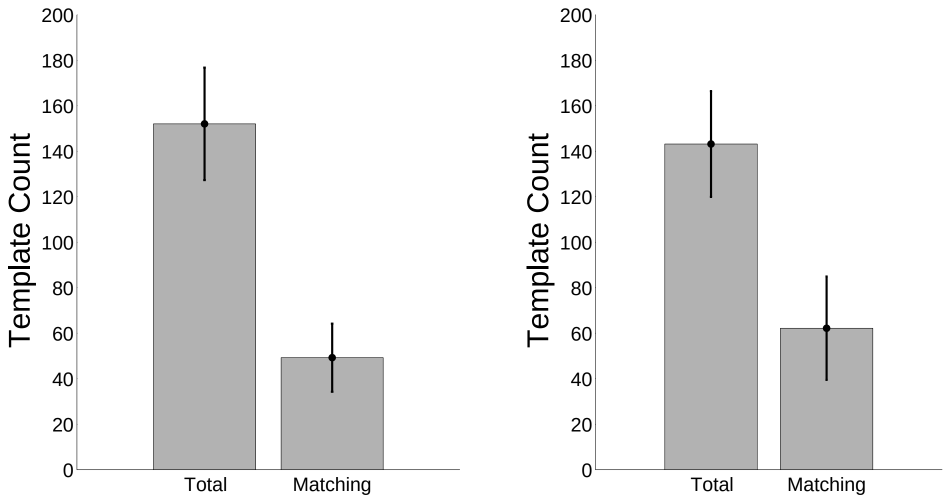


Figure 8: Comparison of the total number of templates and the number of templates that found a match. Results for the ascending pattern are shown in the left figure and results for the descending pattern are on the right. Each bar shows the number of templates averaged across twenty independent networks. Error bars show the upper and lower confidence limits relative to the sample mean (99% confidence interval).

match the firing data at least once during a ten minute period of stimulation with the corresponding input pattern. Figure 8 shows the mean proportion of effective templates relative to total templates for both the ascending and descending input patterns and their corresponding templates. The average number of pattern-specific templates isolated from networks trained on the ascending input pattern (averaged across twenty independent networks) is 152. Of these an average of 49 templates were effective at finding a match over a ten minute period of stimulation with the ascending pattern. The results for the descending pattern are similar with 143 templates in total, of which only 62 found a match. Hence, only 32% of ascending templates and 43% of descending templates were effective at finding a match. It is also worth noting the high variability in template matching performance between networks. As shown in Fig. 9 some networks average as few as three matches in each response frame, suggesting that the template matching method is near to the limit of sensitivity for some networks.

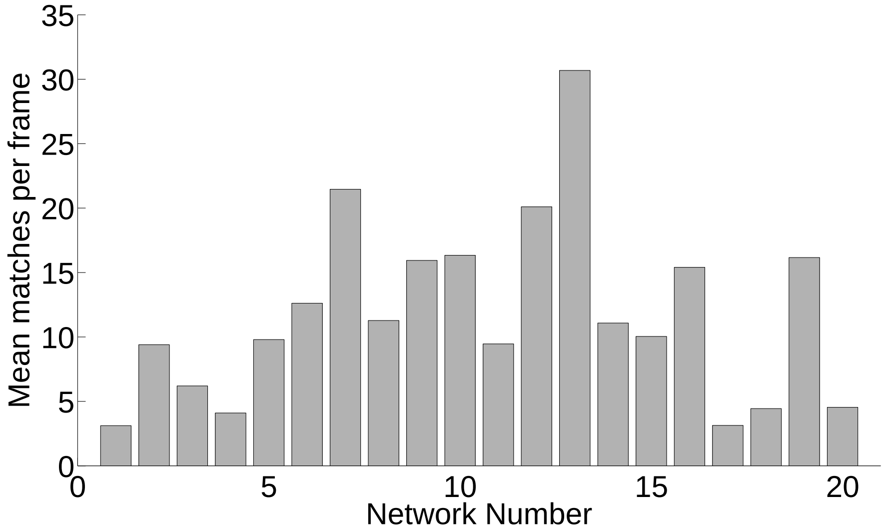


Figure 9: The mean number of template matches per frame for each network.

4. Discussion

The template matching method attempts to match spatio-temporal templates derived from the structural PNGs found in a trained network with the sequence of firing events that are produced when the network is stimulated with the same pattern. For the purposes of this experiment the scope of the PNG search algorithm was limited to searching just for templates derived from triplets that are some combination of the input pattern firing events. These firing event triplets are spatio-temporal *triggering patterns* that evoke a group activation response. The pool of pattern-specific templates that were generated in the training phase of this experiment are therefore all able to be activated by triggering patterns consisting of just three firing events taken from the input stimulus. We can imagine that groups exist in the network that require larger, more complex, triggering patterns although it seems likely that the probability of finding groups with larger triggers decreases with the size of the triggering pattern.

Templates that match the firing data such as those shown in Fig. 2 provide an impression of the corresponding PNG activations that occur in the milliseconds following each stimulus. However, looking at a selection of matching templates creates only a partial picture of the complex pattern of neural firing in response to spatio-temporal stimuli. Visualization of all of the PNG activations that are initiated by combinations of firing events from the input pattern produces a complex graph in which individual PNGs interact and merge (results not shown).

Izhikevich et al. (2004) has proposed that competitive interactions occur within the network, with neurons that are shared by multiple PNGs synchronising their firing times with different polychronising pathways at different times. However, cooperative interactions are also possible in which firing events generated by separate PNG activations together produce the required spatio-temporal initiators for additional PNG activations. The emerging picture is one in which the activation response to complex stimuli is a composition of individual PNG activations that interact and merge in a complex manner.

Interestingly, all of the templates that found a match in the neural response were initiated by triplets made up of firing events from just the early portion of the input pattern. This effect was found across all networks and for both the ascending and descending input patterns. A possible explanation is that competition during PNG formation for use of shared neurons creates an interference effect between early PNG activations and those that come later, with the earlier activating groups forming first and therefore dominating the available neural resources.

This explanation has implications for the maximum number of simultaneous activations that a network of a given size is able to support, and might in turn impact the maximum number of representations that can simultaneously be “held in mind” in a representational system based on polychronous groups. However, note that this explanation does not contradict the extraordinary potential capacity of a PNG-based representational system (Izhikevich, 2006a) because any potential limitation in the number of *simultaneous* activations supported by a representational system does not necessarily affect the network capacity i.e. the total number of representations that can be stored within the network.

Despite any interference caused by interactions between simultaneous activations, the template matching method provides good support for the consistency of a PNG-based representational system. Using a combined pool of all templates, one or more template matches are detected in almost every response frame, suggesting a consistent PNG activation response following each presentation of the stimulus. The best single template for each network is also able to show quite a high degree of consistency although most individual templates match only rarely.

Computing the template match ratio for each one millisecond time-slot in the response frame shows that all matches are confined to a narrow temporal window following each stimulus presentation (see Fig. 6). This strong interaction between the time of the stimulus and the time of template matching supports the view that template matching reflects the causal relationship between stimulus onset and subsequent PNG activation.

Summing the template match ratio across all response frame timeslots and across all response frames produces an aggregated result that reflects the empirical likelihood of PNG activation given the stimulus. With the combined templates, this likelihood value approaches certainty for many of the networks (see Fig. 7), although there is considerable variation between networks.

Together these results indicate a high degree of consistency in the PNG activation response following a stimulus. However, despite this consistent response there are occasional response frames where no neural response is detected, despite the presence of a known stimulus. The lack of a detectable response does not mean that PNG activation did not occur and may instead be due to limitations in the template matching method. Examination of the precise timing of the firing events in consecutive response frames shows considerable jitter in the spike times of PNG neurons between frames (results not shown) and competition for neural resources between activating groups may increase this jitter to the point where the corresponding template fails to match.

The lack of tolerance of the template matching method to temporal jitter is just one of the flaws of this method for detecting PNG activations. Although this technique is able to respond selectively to substantially different stimuli (e.g. discriminating between the ascending and descending patterns, or the ascending and null patterns in Fig. 4), the low matching threshold potentially allows templates to match unrelated spatio-temporal patterns and the template matching method may therefore have difficulty in resolving stimuli that are too closely related.

Another problem with the template matching method is that it treats matching as a local process when it is likely to be a global one. The neural response to a complex stimulus is a unique *set of PNG activations*; it is therefore the set as a whole and not individual activations that provide a unique signature of the stimulus. Given a set-oriented view of the neural response, if a single template happens to match a single PNG activation, does this provide good evidence of the presence of the stimulus? For example, two stimuli with partial overlap in their spatio-temporal firing patterns could both match the same template and the two stimuli would therefore not be resolvable. In recognition of a set-oriented view of the neural response, the template matching method makes use of a pool of templates that are able to detect multiple PNG activations. However, the current method does not take into account the number of unique matches in each response frame and is therefore unable to counter the problem of overlapping stimuli.

Each of the templates generated in the training phase contribute to the time it takes to scan the firing data in the testing phase. It is therefore a problem that the majority of templates are ineffective, with less than half of the templates ever able to generate a match. Although the single best template for each network matches the neural response very consistently, the majority of templates that match at all do so only rarely. In addition, the number of matches in each response frame is sometimes very low suggesting that this method is close to the threshold for maximum sensitivity for some networks. For example, although the trained network in the bottom row of Fig. 4 (Network 1) shows a high degree of consistency, the evidence for most of these PNG activations comes from only a few matches and in some cases just a single match. In contrast, the templates from the best performing network averages more than thirty matches per frame (see Fig. 9 for the average number of matches per frame for each network).

It is likely that Izhikevich (2006a) used a similar technique to show selectivity in the neural response, despite the flaws of the template matching method. The issues with this method, while limiting the scope and accuracy of the current results, do not invalidate our overall finding. Here we provide preliminary evidence for the consistency of PNG activation in response to stimuli, suggesting that polychronous groups may be able to meet at least one of the necessary criteria for a representational system. The neural response to complex stimuli appears to involve multiple interacting PNG activations suggesting that an alternative method for measuring the neural response must treat any single PNG activation as only partial evidence in favor of a particular stimulus. Work is in progress on such an alternative technique that will address these limitations.

References

Abraham, W. C., 2008. Metaplasticity: tuning synapses and networks for plasticity. Nature Reviews Neuroscience 9 (5), 387–387.

Caporale, N., Dan, Y., 2008. Spike timing-dependent plasticity: a hebbian learning rule. Annu. Rev. Neurosci. 31, 25–46.

Caroni, P., Donato, F., Muller, D., 2012. Structural plasticity upon learning: regulation and functions. Nature Reviews Neuroscience 13 (7), 478–490.

Guise, M., Knott, A., Benuskova, L., 2013. Spinula: software for simulation and analysis of spiking network models. Tech. rep., Dept of Computer Science, University of Otago, Dunedin, see http://www.cs.otago.ac.nz/research/techreports.php.

Izhikevich, E. M., Feb. 2006a. Polychronization: computation with spikes. Neural computation 18 (2), 245–82.

Izhikevich, E. M., 2006b. Reference software implementation for the Izhikevich model: minimal spiking network that can polychronize.
URL http://www.izhikevich.org/publications/spnet.htm

Izhikevich, E. M., Gally, J. a., Edelman, G. M., Aug. 2004. Spike-timing dynamics of neuronal groups. Cerebral cortex (New York, N.Y. : 1991) 14 (8), 933–44.

Martin, S., Grimwood, P., Morris, R., 2000. Synaptic plasticity and memory: an evaluation of the hypothesis. Annual review of neuroscience 23 (1), 649–711.

Martinez, R., Paugam-Moisy, H., 2009. Algorithms for structural and dynamical polychronous groups detection. Artificial Neural Networks ICANN 2009 5769, 75–84.

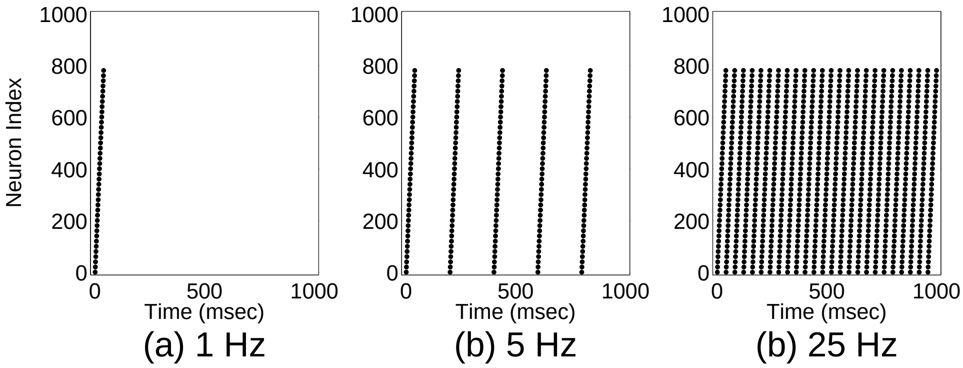


Figure 10: Input stimulation patterns used in the PNG activation experiment at 1, 5 and 25 Hz. Note that the 25Hz pattern is continuous in that at least one input pattern neuron is fired in every millisecond.

5. Detailed Methods

The methods in this report are described via a series of scripts that allow the reproduction of each experiment. Execution of these scripts requires the installation of the Spinula software package (Guise et al., 2013), a small suite of dynamic link libraries that provide functions for network construction, execution and analysis.

There is one script each for the training and testing phases of the experiment and a modified version of the testing phase script that supports the use of multiple input patterns. An additional script is used to generate structural descriptions of each matching template for the purpose of visualizing the PNG structure. All experiments were performed on a set of twenty independent spiking networks, each composed of 1000 Izhikevich neurons with parameters taken from Izhikevich (2006a). The stimuli used in these two phases (at either 1, 5 or 25 Hz) are shown for the ascending pattern in Figure 10.

5.1. Training Phase

The primary purpose of the training phase was to train each network on a selected input pattern so that subsequent exposure to that pattern would produce PNG activation. The training phase began with twenty untrained networks that had previously been matured for two hours in the presence of 1 Hz random background stimulation. Each of these networks was trained with either the ascending or descending input patterns at either 5 Hz or 25 Hz for 20 minutes with continuous 1 Hz random background stimulation. At the end of the training period the network state was saved for use in the testing phase of the experiment.

Structural groups were accumulated over the course of the training phase for later use as templates to probe for group activation. At regular intervals during training the network was scanned for these templates and the accumulated templates were saved to a file. The search for templates used the slower, more precise version of the Izhikevich search algorithm (Izhikevich, 2006b) although a modified version of this PNG search algorithm was created to confine the search to *pattern-specific* structural groups.

5.1.1. Pattern-specific templates

The Izhikevich search algorithms find structural PNGs in the network by searching for spatio-temporal *triplets* of three neurons whose non-synchronous firing is able to trigger a causal chain of firing along pathways of a minimum length. One such triplet in the ascending pattern is (1, 1), (2, 21), (3, 41) where each tuple is of the form (time, neuron index).

By default, the search algorithms find all structural PNGs in the network that are initiated by triplets composed from any combination of three neurons in the network. If the network has been previously trained on a known pattern then it is assumed that some of these structural groups will be pattern-specific such that presentation of the pattern will produce group activation. In order to select these pattern-specific PNGs we restrict the search algorithm to the use of triplet initiators that are subsets of the input pattern. Although the search for theoretical PNGs is based on *triplets* only, it is likely that other pattern-specific groups could be found if the search was extended to include larger combinations e.g. quintuplets etc. It is also worth noting that, in addition to these pattern-specific groups, the network structure may also simultaneously contain many thousands of groups that respond to spatio-temporal patterns unrelated to the input pattern.

5.1.2. Training phase script

A script that implements the training phase of the experiment is shown in Listing 5.1. The first argument to the script is the pattern type (ascending or descending) followed by the input pattern stimulation frequency and the background stimulation frequency. The next argument specifies the path to a state file representing a mature network state, and the remaining arguments set the output folder, where the trained network and the isolated structural PNGs will be saved, and the base name from which the names of save files will be derived.

Lines 13 and 14 of the training phase script define the network specifier. We will use this specifier to load a mature network state and then run this network in the presence of coherent stimulation. All possible triplet combinations are generated from the input pattern in lines 19 to 22. Note that the

```
1  // Run a mature network while providing coherent external stimulation to build new groups
2  // Search for structural PNGs for later testing
3  let TrainWithCoherentStimulation patternType patternStimulationsPerSecond backgroundFrequency
4          stateFilePath pathToOutputFolder basename =
5
6      let numExcitatoryNeurons = 800
7      let numInhibitoryNeurons = 200
8      let numSynapsesPerNeuron = 100
9      let maxDelay = 20
10     let runSeconds = 1200               // 20 minutes
11     let saveState = true                // flag for saving of network state
12
13     let networkSpecifier = new IzhikevichNetworkSpecifier(
14             numExcitatoryNeurons, numInhibitoryNeurons, numSynapsesPerNeuron, maxDelay)
15
16     let patternStep = if patternType = InputPatternType.Ascending then 1 else -1
17
18     // get all triplet combinations from a linear ascending or descending input pattern
19     let triplets =
20         // single repeat of an ascending or descending pattern
21         let inputPattern = Span.CreateLinearInputPattern(1, 1, patternStep, 40)
22         Span.CreateTripletListFromPattern(inputPattern)
23
24     // create an input pattern composed from a 40 msec ascending or descending pattern
25     // continuously repeated over each second at a rate specified by patternStimulationsPerSecond
26     let inputPattern =
27         if patternStimulationsPerSecond = 0 then
28             null
29         else
30             Span.CreateLinearInputPattern(patternStimulationsPerSecond, 40, 1, patternStep, 40)
31
32     // run the network and collect any PNGs that match the initial triplets
33     Span.ScanForTheoreticalGroups(networkSpecifier,
34         triplets, inputPattern, patternStimulationsPerSecond, backgroundFrequency, runSeconds,
35         stateFilePath, pathToOutputFolder, basename, saveState)
```

Listing 5.1: Training phase script

firing times and neural indices in each triplet need not be consecutive, and therefore the number of triplet combinations is high (nearly ten thousand). Lines 26 to 30 define the stimulation pattern, and lines 33 to 35 perform the actual training. Internally, this function runs the network in the presence of external stimulation, pausing every minute to scan the network structure for PNGs; any discovered PNGs are saved into a file indexed by the time (in minutes) of their discovery. Following training, the network state is saved to a file in the output folder.

5.2. Testing Phase

In the testing phase of the experiment, each of the saved networks from the training phase is stimulated with either the ascending or descending input pattern and the accumulated templates are used to probe the resulting firing data for matches. The algorithm for the detection of group activation in the original Izhikevich experiment has not been published, and therefore a new template matching algorithm was developed (Guise et al., 2013). Broadly, the procedure is to generate a spike raster from the network firing data and then slide a set of template-specific windows across the spike raster and score each match. At each window position in the scan, each of the templates discovered during the training phase is sequentially matched using a matching threshold of 50% and jitter of ± 2 milliseconds. The selected input pattern and a 1 Hz random background pattern are both presented at 1 Hz throughout each test period (typically 600 seconds). The use of a 1 Hz test stimulus allows the test period to be divided into one second stimulus-response frames which are of sufficient length to ensure that PNG activation in each frame is complete before the next stimulus presentation.

5.2.1. Testing phase script

The testing phase script is shown in Listing 5.2. The first argument to this script determines the input pattern type, either ascending or descending. The second and third arguments specify the training time and the path to the trained state file. Remaining arguments specify the stimulation intensity, the location of the saved templates, and the output file respectively.

Lines 13 and 14 define the specification for the network that will later be reloaded with the selected network state. The ascending or descending input pattern is created in lines 18 to 20 and line 23 loads the saved templates. The critical part of the script occurs in the function call on the last two lines. Internally this function loads the selected state file into a new network and runs it for (typically) six hundred seconds while collecting firing data. It then scans the firing data using the specified templates, saving any matches. Note that longer templates are clipped to a maximum of thirty firing events, the justification being that while the head of each activated group can be quite stable over multiple stimulations, the trailing portion shows considerable variation (Izhikevich, 2006a).

5.2.2. Modified testing phase script for multiple stimuli

Figure 4 uses a modified version of the testing phase script (shown in Listing 5.3) that allows the input stimuli to change over the course of the test period. This script was run for each of twenty untrained networks, and again with the same twenty networks after they had been trained on either the ascending or descending input patterns. The test period for this script was limited to 100 seconds (100 response frames with one stimulus per frame), allowing 25 frames per input pattern. Note the creation of multiple input patterns in lines 19 to 25 and an alternative scanning function in the last two lines that allows the network to be stimulated with multiple input patterns.

5.3. Script for generating PNG structural diagrams

Listing 5.4 shows a script that takes a file of matches and extracts a structural description of each of the matching templates, allowing the structure of the templates to be graphically displayed. Line 8 retrieves the matching

```
1  // Provide coherent stimulation to a network loaded from the specified state file
2  // and scan the firing data for evidence of PNG activation
3  let ScanNetworkForActivatedGroups patternType stateSaveTime stateFilePath
4          patternStimulationsPerSecond (groupsFilePath:string) outputFilePath =
5
6      let numExcitatoryNeurons = 800
7      let numInhibitoryNeurons = 200
8      let numSynapsesPerNeuron = 100
9      let maxDelay = 20
10     let maxTemplateSize = 30            // trim large templates to this size
11     let runSeconds = 600
12
13     let networkSpecifier = new IzhikevichNetworkSpecifier(
14             numExcitatoryNeurons, numInhibitoryNeurons, numSynapsesPerNeuron, maxDelay)
15
16     // create an input pattern from a 40 msec ascending pattern
17     // continuously repeated over each second at the specified frequency
18     let inputPattern =
19             let patternStep = if patternType = InputPatternType.Ascending then 1 else -1
20             Span.CreateLinearInputPattern(patternStimulationsPerSecond, 40, 1, patternStep, 40)
21
22     // read the groups file and prepare PNG templates
23     let templateList = Span.GetTemplates(groupsFilePath, maxTemplateSize)
24
25     // load the network state file and run the network with stimulation
26     // scan the firing data for template matches
27     Span.ScanNetworkForActivatedGroups(stateSaveTime, stateFilePath, templateList,
28             networkSpecifier, inputPattern, runSeconds, outputFilePath)
```

Listing 5.2: Testing phase script

templates, filtering out those few template matches that do not match the firing events in the *initial* triplet of each template. The majority of templates that include the initial triplet in their matching firing events are likely to have matched spatio-temporal firing patterns that are initiated by the input pattern (pattern-specific PNG activation). In lines 11 and 12 the retrieved templates are saved to a new file. A flag on the Save method specifies that duplicates be removed prior to saving. At line 15 structural descriptors are generated for each matching template using linkage data from the specified groups file. The set of descriptors is then saved to a file in lines 18 and 19.

```
// Provide a selection of stimuli to a network loaded from the specified state file
// and scan the firing data for evidence of PNG activation
let ScanForActivationsMultiPattern stateSaveTime stateFilePath patternStimulationsPerSecond
        (groupsFilePath:string) outputFilePath =

        let numExcitatoryNeurons = 800
        let numInhibitoryNeurons = 200
        let numSynapsesPerNeuron = 100
        let maxDelay = 20
        let maxTemplateSize = 30            // trim large templates to this size
        let runSeconds = 100
        let secsPerPattern = 25

        let networkSpecifier = new IzhikevichNetworkSpecifier(
                numExcitatoryNeurons, numInhibitoryNeurons, numSynapsesPerNeuron, maxDelay)

        // create four different 40 msec input patterns
        // continuously repeated over each second at the specified frequency
        let patterns =      // ascending, descending, ascending, null
            [
                Span.CreateLinearInputPattern(patternStimulationsPerSecond, 40, 1, 1, 40);
                Span.CreateLinearInputPattern(patternStimulationsPerSecond, 40, 1, -1, 40);
                Span.CreateLinearInputPattern(patternStimulationsPerSecond, 40, 1, 1, 40);
                null
            ]

        // read the groups file and prepare PNG templates
        let templateList = Span.GetTemplates(groupsFilePath, maxTemplateSize)

        // load the network state file and run the network with stimulation
        // scan the firing data for template matches
        Span.ScanNetworkForActivatedGroupsMultiPattern(stateSaveTime, stateFilePath, templateList,
                networkSpecifier, patterns, secsPerPattern, runSeconds, outputFilePath)
```

Listing 5.3: Modified testing phase script for multiple stimuli

```
// Get matching templates for the specified sample time and generate structural descriptors
let GetPNGMatchDescriptors stateSaveTime matchesFilePath (groupsFilePath:string)
        pathToOutputFolder templateFileName descriptorSetFileName =

    let distinctOnly = true

    // get the matching templates at this sample time
    let matches = Span.GetMatchingTemplatesAtTime(stateSaveTime, matchesFilePath)

    // save the matching templates to a file
    let templateOutputPath = PathDescriptor.Create(pathToOutputFolder, templateFileName)
    matches.Save(distinctOnly, templateOutputPath)

    // generate a set of linkage-data-based descriptors for each matching template
    let descriptorSet = matches.CreatePNGDescriptorSet(groupsFilePath)

    // save the descriptors to a file
    let descriptorOutputPath = PathDescriptor.Create(pathToOutputFolder, descriptorSetFileName)
    descriptorSet.Save(descriptorOutputPath)
```

Listing 5.4: Script for generating structural descriptors