Modular deconstruction reveals the dynamical and physical building blocks of a locomotion motor program

Angela M. Bruno,[1] William N. Frost,[2] Mark D. Humphries[3]

1. Department of Neuroscience, The Chicago Medical School, Rosalind Franklin University of Medicine and Science, North Chicago, Illinois, USA
2. Department of Cell Biology and Anatomy, The Chicago Medical School, Rosalind Franklin University of Medicine and Science, North Chicago, Illinois, USA
3. Faculty of Life Sciences, University of Manchester, Manchester, United Kingdom

Contact : Correspondence should be addressed to M.D.H (mark.humphries@manchester.ac.uk) or W.F. (william.frost@rosalindfranklin.edu)

Running title : Modular deconstruction of a motor program

Abstract

The neural substrates of motor programs are only well understood for small, dedicated circuits. Here we investigate how a motor program is constructed within a large network. We imaged populations of neurons in the *Aplysia* pedal ganglion during execution of a locomotion motor program. We found that the program was built from a very small number of dynamical building blocks, including both neural ensembles and low-dimensional rotational dynamics. These map onto physically discrete regions of the ganglion, so that the motor program has a corresponding modular organization in both dynamical and physical space. Using this dynamic map, we identify the population potentially implementing the rhythmic pattern generator and find that its activity physically traces a looped trajectory, recapitulating its low-dimensional rotational dynamics. Our results suggest that, even in simple invertebrates, neural motor programs are implemented by large, distributed networks containing multiple dynamical systems.

Highlights :

- Neural ensembles code the *Aplysia* locomotion motor program
- Stable and rotational dynamics differentiate classes of ensemble
- Different dynamical classes map to physically separate and discrete regions
- Low-dimensional rotations of activity are recapitulated in physical space

Introduction

The concept of the motor program, a fixed sequence of automatically-executed movements, is widely assumed to underlie automatic motor control in both vertebrates (Mink, 1996; Grillner et al., 2005; Summers and Anson, 2009; Esposito et al., 2014) and invertebrates (Wu et al., 1994; Frost and Katz, 1996; Kupfermann and Weiss, 2001; Jing et al., 2004; Flood et al., 2013; Schoofs et al., 2014). Its neural basis has been most clearly elucidated in the reconstructions of dedicated circuits that initiate, generate, and execute a specific rhythmic behaviour in invertebrates (Selverston, 2010). Deep understanding of these circuits has been possible because each comprises the inter-connections between a few identifiable neurons that are common to every animal. These have illuminated general principles of the neural architectures, dynamics and modulation underlying motor control (Getting, 1989; Katz et al., 1994; Yuste et al., 2005; Selverston, 2010). However, in larger nervous systems the limited genetic capacity for specifying individual neurons and their connectivity means that dedicated circuits give way to stochastically wired networks. The existence of multifunctional motor networks in both *Aplysia* and the medicinal leech, in which the same neural system supports more than one distinct motor program (Tsau et al., 1994; Wu et al., 1994; Briggman et al., 2005; Briggman and Kristan, 2006), suggests that even in simple invertebrates neural motor programs are implemented in large, distributed networks rather than dedicated circuits (Getting, 1989; Wu et al., 1994). Understanding the distributed network implementation of a motor program would thus bridge the gap between dedicated circuits and the general principles of motor control.

How a distributed network implements a single motor program is unclear. Its implementation is potentially built from a mixture of systems (Getting, 1989; Jing et al., 2004; Rokni and Sompolinsky, 2012), including at least one pattern generator for rhythmic output (Selverston, 2010; Rokni and Sompolinsky, 2012; Churchland et al., 2012), a set of motorneurons for translating rhythmic output to muscle commands (Brezina et al., 2000; Rokni and Sompolinsky, 2012), and neuromodulators of both generator and motorneuron output (Getting, 1989; Brezina et al., 2000). Each of these "building blocks" (Getting, 1989) could form a functionally separate population within the network, or two or more could be combined into a single functional population. Each building block could implement a different dynamical system, such as neural ensembles (Wickens et al., 1994; Mattia et al., 2013) or low-dimensional attractors (Schoener and Kelso, 1988; Briggman et al., 2005; Churchland et al., 2012). Consequently, the distributed network implementation of a motor program has many unknowns: whether it is a mixture of functionally independent dynamical building blocks or a single integrated circuit; how these are organised in the network; and what dynamics they implement.

To address these issues, we imaged populations of neurons in the pedal ganglion of the sea-slug *Aplysia* while reliably eliciting its motor program for locomotion. The pedal ganglion contains approximately 1600 neurons (Cash and Carew, 1989), and wholly contains the rhythmic pattern generator (Jahan-Parwar and Fredman, 1979, 1980), motorneurons (Hening et al., 1979; Fredman and Jahan-Parwar, 1980) and associated neuromodulatory neurons (Hall and Lloyd, 1990; McPherson and Blankenship, 1992) for locomotion, thus making it a tractable target for mapping a motor program to the dynamics and structure of its underlying distributed network. This mixture of systems means that population-imaging of the *Aplysia* pedal ganglion is representative of the analytical challenges that will become increasingly common for large-scale recordings of complex neural systems (Cunningham and Yu, 2014), as we know that the recorded populations will have captured multiple dynamical systems within them. We thus had to develop new dimension-reduction approaches to deconstruct population recordings into the motor program's component systems.

In this paper, we report that the locomotion motor program is built from a very small number of dynamical building blocks that are common to every execution. These include both ensembles and low-dimensional dynamics. We show that this dynamical decomposition unexpectedly maps onto physically discrete regions of the ganglion, such that the motor program is built from physical as well as functional building blocks in a distributed network. Using this dynamic map, we identify a population with rotational dynamics potentially implementing the rhythmic pattern generator. We further show that its activity physically traces a looped trajectory. These findings shed light on the general principles of implementing motor programs in distributed networks, place strong constraints on circuit mechanisms underlying locomotion in *Aplysia*, and pave the way for targeted studies of attractor-like dynamics in neural systems.

Results

Optical recordings reveal variation within and between motor program executions

We used a setup that was designed to simultaneously record neural populations at high temporal and spatial resolution during execution of a single motor program (Hill et al., 2010) (Fig. 1; see Experimental Procedures). Aversive tail stimuli elicit escape locomotion in *Aplysia*, which consists of a rhythmic series of head reaches and muscular contraction cycles that roll head to tail down the length of the animal (Jahan-Parwar and Fredman, 1979; Hening et al., 1979). Sharp-electrode studies showed rhythmically bursting motorneurons located mostly in the pedal ganglion's rostro-medial quadrant (Hening et al., 1979) and phase-locked to the rhythmic muscle contractions (Hening et al., 1979; Fredman and Jahan-Parwar, 1980). The ganglion's functional and physical organization during locomotion is otherwise largely unknown.

Using an isolated brain preparation, we reliably elicited fictive locomotion by stimulation of pedal nerve 9, while simultaneously recording the spike-trains of individual neurons across the entire ganglion at high temporal resolution (1.6 kHz) using voltage-sensitive dye imaging with a photodiode array (Fig. 1A) (Hill et al., 2010). We simultaneously recorded 57-125 neurons in each of 12 preparations, a total of 1145 neurons, with each record truncated to the initial 80 seconds in which each preparation consistently showed locomotion-related activity. We accurately mapped the location of every recorded neuron in the 2-dimensional imaging plane (see Supplemental Experimental Procedure 3; Fig. 3A).

Decomposing the motor program required combining recordings to identify dynamical building blocks common to each execution of the program. However, recorded executions varied in both time-scale (Supplemental Fig. 1A-C) and intensity (the distribution of firing rates significantly differed between 55 of the 66 unique pairs of recordings; K-S test at $\alpha = 0.05$). This variation between recordings, combined with the likely mixture of dynamics within each recording, made analysis with standard dimension reduction techniques challenging (Machens, 2010). Illustrating this difficulty, we found that applying principal components analysis (Briggman et al., 2005; Levi et al., 2005) to each recording revealed hints of attractor-like dynamics (Supplemental Fig. 1D-E), but the number of principal components capturing 95% of the variance in a recording ranged from 2-12, indicating that each recording captured different sets of dynamics within the same motor

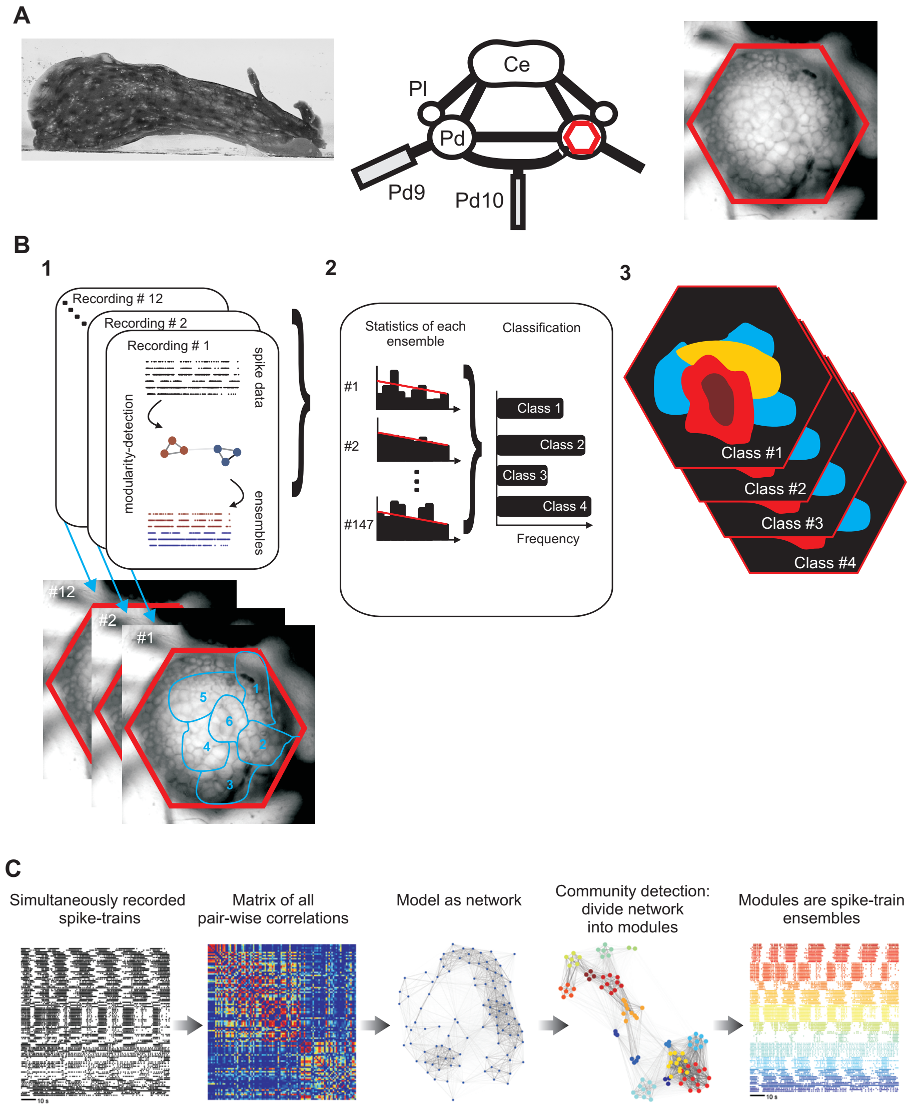

Figure 1: Imaging and analyzing *Aplysia*'s locomotion motor program. **A** Experimental setup. Left: Head reach in *Aplysia*'s rhythmic escape locomotion. Middle: Preparations consisted of the ring ganglia (Ce: cerebral; Pl: pleural; Pd: pedal), with a stimulating electrode connected to peripheral nerve Pd9. Some recordings also used a suction electrode connected to pedal nerve 10 (Pd10) to monitor the neck contraction phase of locomotion. Right: Imaged area of the dorsal pedal ganglion, aligned to the photodiode array of 464 diodes (red outline). **B** Analysis stages for deconstructing the motor program. Fast voltage-sensitive dye recordings captured simultaneous cellular-level activity and the location of every neuron. Step 1: each motor program recording is decomposed into its component ensembles using modularity-detection (panel C). We map the physical location of the ensembles in each recording, seeking a "dynamic map" of each execution of the program. Step 2: ensembles are classified across recordings into groups of statistically similar firing patterns, seeking the dynamical building blocks of the motor program. Step 3: we map the physical location of the ensemble groups over all recordings, seeking the physical layout of the motor program's dynamical building blocks. **C** Modularity detection of ensembles (step 1 in panel B). Schematic illustration of the steps for decoding neural ensembles using community detection with consensus clustering. Key is modelling the pairwise correlation matrix as a network: each node is a neuron, each link's weight is the correlation between that pair of neurons. Community detection algorithms – so-called by analogy with the division of social networks into communities – provide a general solution to the problem of separating an arbitrary network into its component modules: here each module is thus an ensemble of strongly, mutually correlated neurons. Choice of time-scale and type of correlation thus define a "neural ensemble". Here we convolve each spike-train with a Gaussian window whose width is defined by the characteristic period of the locomotion-related activity (Experimental Procedures; Supplemental Fig. 1), and correlate each pair of convolved spike-trains.

program.

To solve these problems, our starting point was to seek the existence of neural ensembles within each recording. Finding these would provide evidence of ensembles within a distributed network implementation of a motor program; as we will show, these also provide a basis for solving the problem of combining information across variable recordings to decompose the dynamic systems within a program. Our analysis strategy for decomposing the motor program is illustrated in Figure 1B; below we detail the new algorithms introduced at each step.

Each motor program execution is comprised of functional neural ensembles

As there are an unknown number and size of ensembles in each execution – potentially zero – we developed an unsupervised algorithm for ensemble detection (Experimental Procedures). Each recording is characterized as a functional network of correlations between neuron pairs (Figure 1C), where each node is a neuron and each link encodes correlation strength. Our algorithm is based on the concept of community detection in arbitrary networks: it separates each functional network into modules ("communities") of nodes by directly optimizing the modularity score Q, which is maximized by grouping nodes into modules that have dense connections within them and sparse connections between them (Newman, 2006b; Humphries, 2011). These modules are thus groups of neurons whose activity patterns are more similar to each other than to any other neuron, and give a quantitative definition of "neural ensemble" (or "cell assembly") (Wehr and Laurent, 1996; Harris, 2005; Peyrache et al., 2009; Niessing and Friedrich, 2010). To solve the problem of reliably clustering each recording for later between-recording comparison, we developed a new consensus-clustering approach to community detection (Supplemental Experimental Procedure 1).

Consensus community detection was highly effective in organizing the raw recording traces into separate ensembles of temporally correlated neurons (illustrated in Fig. 2A-C). We found that each execution of the locomotion motor program decomposed into multiple neural ensembles. The recordings captured between 7 and 18 ensembles each (total of 147), ranging in size from 2–24 neurons (median 8 neurons; Fig. 2D). Every ensemble was highly self-similar (Fig. 2E), indicating that the algorithm was very successful at separating each functional network into its component modules. The modularity of each preparation was approximately the same (mean $Q = 0.18 \pm 0.05$ s.d.), confirming that the recordings were sampling the correlation structure of the circuit in a consistent manner. The number of ensembles scaled with the number of recorded neurons in the preparation (Fig. 2F), suggesting that each recording is sampling a subset of all simultaneously present ensembles.

Supplemental Figure 2 shows that our detected ensemble structures are robust to order of magnitude changes in the time-scale of correlation, and to using an adaptive time-scale that accounts for population firing rate changes over the recording.

Each motor program execution is comprised of physical neural ensembles

To understand how each motor program execution is distributed across the network, we then sought the physical layout of the ensembles in each recording. Our use of voltage-sensitive dye imaging with a photodiode array and independent component analysis to isolate the spike-trains (Hill et al., 2010) allowed us to recover the location of each neuron within the two-dimensional imaging plane of the ganglion (Fig. 3A; Supplemental Exper-

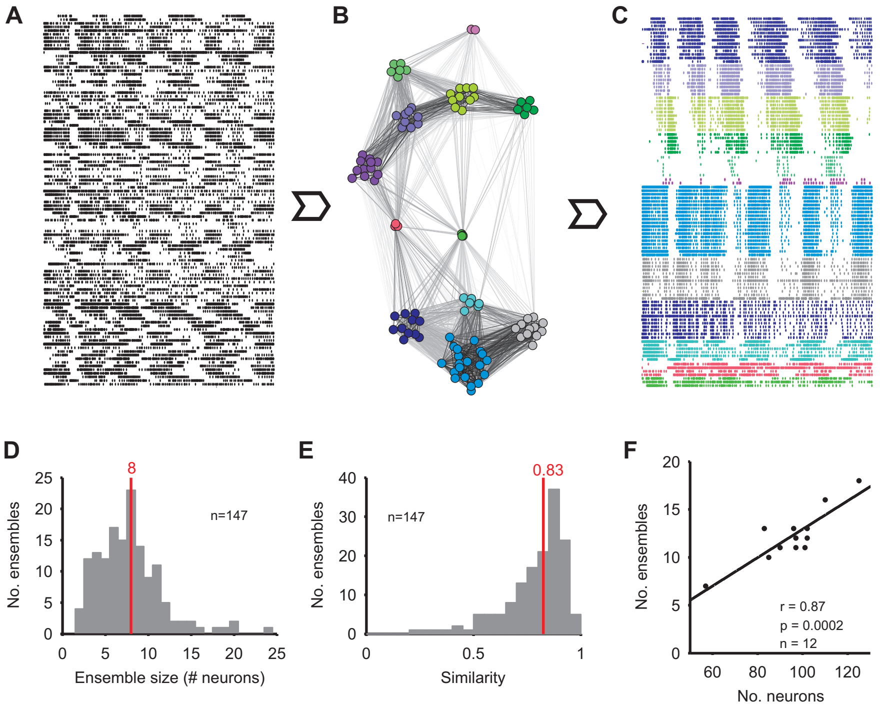

Figure 2: Modular deconstruction of a motor program recording. **A** Raster plot of an example optical recording of 102 neurons over 80 seconds. **B** The corresponding functional network. Each node is a neuron; gray-scale intensity of the links indicates similarity of that pair of neurons. The 12 detected modules within the network are color-coded. Distance between modules indicates their average similarity. **C** Raster plot from panel A ordered and color coded by module membership in panel B, showing the 12 ensembles. **D** Distribution of ensemble size across all recordings. Red line indicates median value. **E** Distribution of median intra-group similarity over all preparations; red line indicates median of distribution. **F** Correlation between number of neurons in each recording and number of ensembles detected (linear regression, $n = 12$). (see also Supplemental Fig. 2).

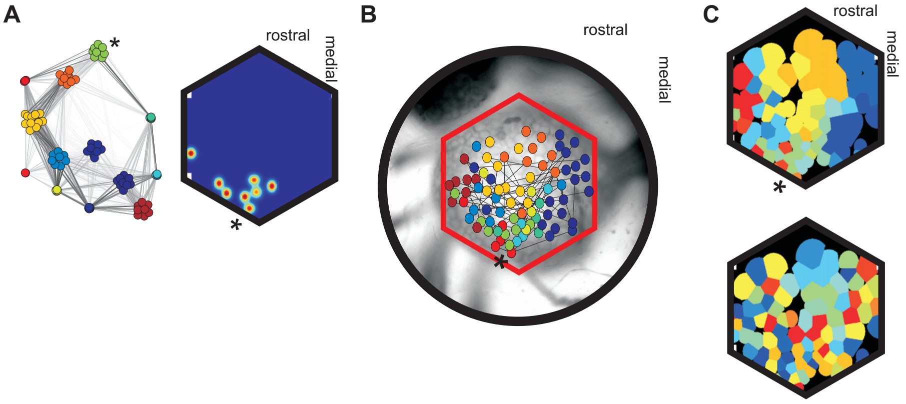

Figure 3: Modular deconstruction of the physical substrate. A Mapping neuron location. Left: layout of a recorded functional network; the asterisk (∗) indicates the example neural ensemble corresponding to physical positions in panels A-C. Right: Estimated position of all 8 neurons in the example ensemble; each position estimate is plotted as a two-dimensional Gaussian, with blue-red indicating minimum-maximum probability of location. **B** Projection of the functional network onto physical space. We plot the mean estimated position of all neurons, color-coded by ensemble membership, onto the approximate extent of the ganglion covered in this recording. Links show a sample of the network (within ± 1 s.d. of mean correlation). **C** Map of neural ensemble locations on the photodiode array (top) showing physical contiguity of the ensembles in a recording: each region of the array is claimed by the closest neuron; all regions belonging to the same neural ensemble are merged (same color). Bottom: example control map, generated by randomly assigning neurons to the same number and size of neural ensembles. Supplemental Figure 3 compares data and randomized maps across all recordings.

imental Procedure 3). Across all recordings we found that most ensembles (100/147) were physically cohesive ($P < 0.05$, permutation test), as illustrated in Figure 3A, such that strongly correlated neurons were physically close.

We then asked whether this physical cohesiveness meant that ensembles comprising a single execution of the program were intermingled or discretely arranged. As each recording sampled a subset of all likely ensembles, to check for discreteness we needed to solve the problems of incomplete data, of unknown, likely irregular physical shapes of ensembles, and of possible noise in the clustering. To do so we developed a parameter-free method for interpolating the physical extent of each ensemble, illustrated in Fig. 3B-C (see Supplemental Experimental Procedure 3).

We found most ensembles (110/147) had at least half their neurons in one continuous patch of space, and so were highly discrete (Supplemental Fig. 3). Most strikingly, as illustrated in Fig. 3C the physical arrangement of ensembles in each recording was also highly discrete (mean proportion of neurons in a single continuous patch was between 52% and 92%; all were greater than 95% confidence interval, permutation test). Functionally-defined ensembles thus correspond to physically separate neuron groups: a single execution of the locomotion motor program is distributed across the network as a set of physically discrete neural ensembles.

Different classes of ensembles correspond to different dynamical systems within the motor program

Having characterized each execution, we turn to the central problem of combining recordings to identify the dynamical building blocks of the motor program (Fig. 1B, Step 2). We had identified a database of 147 ensembles across all recordings. We classified these ensembles into one of four classes defined by significant peaks or troughs in their autocorrelograms (Fig. 4; Experimental Procedures), which allowed us to capture their oscillation pattern invariant to the time-scale or intensity of the motor program execution to which they belonged. We show below that this broad classification allowed us to isolate dynamical systems within each recording.

We found all four possible classes of oscillation pattern existed within the locomotion motor program (Fig. 4). Most ensembles (85%) fell into the three classes of oscillator: "oscillatory" (63%); "bursters" (19%); and "pausers"(3%). The remaining 15% of ensembles had no strong oscillation, yet neurons in these ensembles were recruited by the onset of the motor program (Supplemental Fig. 4). These data suggest that a neural motor program also incorporates neural ensembles that are not manifestations of some oscillation.

We found that the oscillator and burster classes identified separate dynamic systems. For each recording, we selected the subset of ensembles belonging to a particular class, and then projected all neurons in that selected population into a low-dimensional space using PCA, to visualize the population dynamics (Briggman et al., 2005; Levi et al., 2005).

In most recordings we found that the oscillator-class population was dominated by an oscillation of constant amplitude and frequency throughout the program's execution, shown by the first principal component (Fig. 5A). Projecting the population onto the first two principal components showed that this constant oscillation corresponded to a constant rotation of population activity (Fig. 5A). Such consistent rotation in two dimensions indicates that the underlying oscillatory activity in this population was phaselocked throughout the recording

In recordings with sufficient numbers of identified burster-class ensembles, we found that the burster-class population was dominated by evolving activity over the course of the program, shown by the first principal component (Fig. 5B). To confirm this, we correlated the projection onto the first principal component against time, and found a strong relationship (median $r^2 = 0.34$, $n = 11$ recordings), consistent with a change in baseline activity over time. Projecting the population onto the first two principal components showed that this evolving oscillation did not correspond to a constant rotation of population activity (Fig. 5B).

Our identification of four classes of oscillation pattern shows that the locomotion motor program can be decomposed into at least three dynamic systems, one showing correlated firing but not oscillating ("non-oscillators"); one that implements a constant oscillator ("oscillators"); and one that evolves over the program's execution ("bursters"). The pauser class may correspond to a fourth separate dynamic system, but no single recording contained sufficient numbers for us to check.

Dynamic mapping of the motor program

Having identified these four classes of dynamical building blocks of the motor program, we then sought to understand how they are distributed over the physical network (Fig. 1B, Step 3). To do so, we took the map of extrapolated ensemble locations for each recording (such as the example in Fig. 3C) and labeled each location with the oscillatory class of that ensemble. We then combined these re-labeled maps across all recordings. We plot the resulting "dynamic map" for each oscillatory class in Figure 4.

We see that the four classes of ensemble defined solely by oscillation pattern map to different locations on the ganglion. In particular, the oscillator class ensembles are predominantly located in the caudo-lateral quadrant, whereas the burster class ensembles are predominantly located in the rostro-medial quadrant. Thus, the separate dynamical systems identified by the oscillation patterns of neural ensembles correspond to a physical

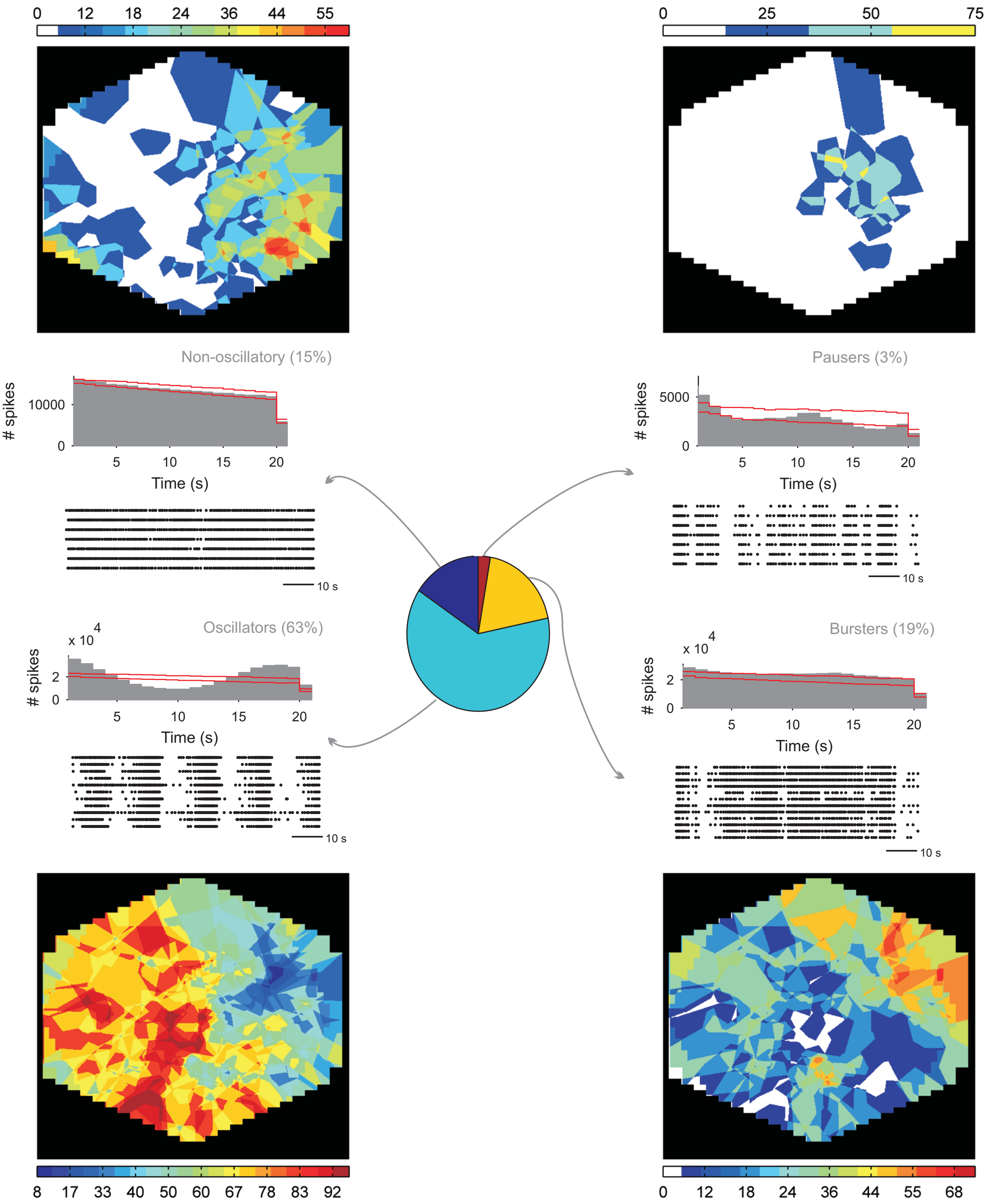

Figure 4: Classes of ensemble. Four classes of ensemble were defined by the presence or absence of significant peaks or troughs in their auto-correlograms: the non-oscillatory class had no significant peaks or troughs; "oscillators" had both; "bursters" had significant peaks but not troughs, indicating repeated phasic firing without repeated silence; "pausers" had significant troughs but not peaks, indicating repeated silence without repeated stereotyped bursts. Auto-correlogram (top) and raster (bottom) plotted for an example ensemble of each class. The red lines plot the upper and lower bounds on expected spike-count predicted by a shuffled inter-spike interval model; peaks are contiguous bins above the upper bound, troughs are contiguous bins below the lower bound. We also plot the "dynamic map" of each oscillatory class, with heat intensity (blue-red) indicating the proportion (%) of recordings in which that location belonged to that oscillatory class of ensemble; white indicates no membership of that class detected. All maps plotted for the left ganglion.

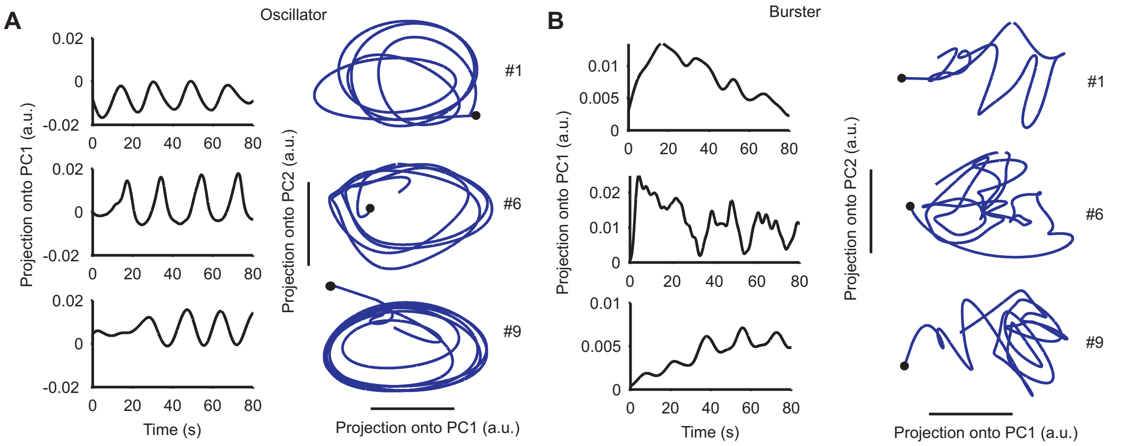

Figure 5: Separate dynamical systems within the locomotion motor program. **A** Oscillator-class ensembles. We plot the projection onto the first principal component (PC1, left) and onto the first two PCs (right) for the population of neurons in oscillator-class ensembles in three recordings (1, 6, and 9). Black dot indicates time zero. **B** Burster-class ensembles. Layout as for panel A, plotted for the population of neurons in burster-class ensembles in the same three recordings.

separation within the pedal ganglion network. We note that this clear separation is not a trivial result of the discreteness of the individual ensembles: the execution of a single program has discrete ensembles, showing that correlated neurons are located within a single, discrete region of the ganglion network; but ensembles of different oscillation classes could themselves be intermingled – our maps show that this is not the case.

Rotational dynamics of oscillator ensembles corresponds to a physical looped trajectory of activity

Aplysia locomotion represents a particular challenge in understanding motor control. In many invertebrates and some vertebrates locomotion is driven by a rolling wave of neural activity that causes sequential muscle contractions. This rolling wave is most often realized by the sequential firing of linked segmental networks (Cacciatore et al., 2000; Eisenhart et al., 2000; Gjorgjieva et al., 2013), but in *Aplysia* the entire neural apparatus is contained within the single pedal ganglion network. It is otherwise unknown how this single network can implement the rhythmic pattern generator necessary to generate the slow rolling wave.

The oscillator ensembles are potentially the observable portion of the unidentified rhythmic pattern generator for locomotion. Our dynamic map localized these to the caudolateral quadrant (Fig. 4) where few motorneurons are found (Hening et al., 1979), and where the single reported example of an oscillatory interneuron was found (Fredman and Jahan-Parwar, 1980). Within each recording, we have shown that the activity of the oscillator-ensemble population of neurons can be well-described by a perfect rotation in a two-dimensional space, despite the variations over a single motor program execution and between each execution (Fig. 5A).

Our mapping of the physical ensembles in each recording allowed us to ask if we could observe directly the rotation in the two-dimensional space as a rolling wave of activity in physical space. We again selected the oscillator ensembles in each recording, and plotted the position of their spatially-averaged activity over time (Fig. 6A; Experimental Procedures). We consistently found that this "activity packet" repeatedly traced a looped

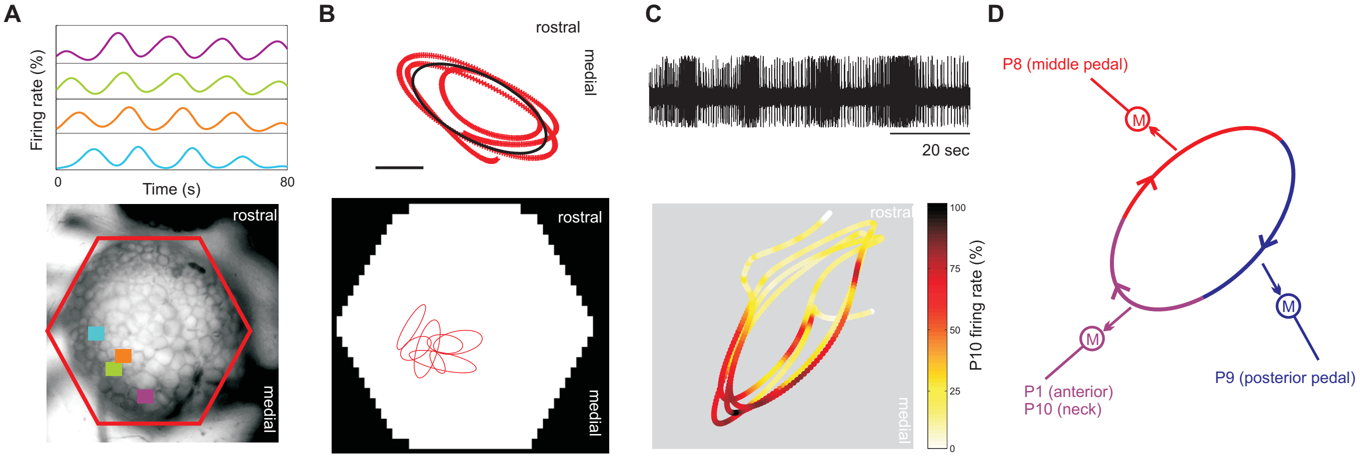

Figure 6: Rotational dynamics correspond to physical rotations of activity. A Projection of oscillator-class activity onto physical space. Top: Sequence of firing of 4 representative oscillator ensembles in recording #1. Bottom: center of each of the 4 ensembles. Eligible recordings had 5-12 oscillator ensembles. **B** Looped trajectories of the activity packet in space. Top: the activity packet from recording #1 (panel A) describes an elliptical orbit (red) on the plane of the ganglion; black line: best-fit ellipse; scale bar: 1 diode spacing ($\sim 60\mu$m). Bottom: best-fit ellipses showing trajectories from all recordings with sufficient oscillator ensembles to uniformly span phases of a motor program cycle. **C** Looped trajectory corresponds to phasic motor output. Top: Extracellular recording from pedal nerve 10 (Pd10). Bottom: firing of Pd10 mapped onto the looped trajectory. Dark red represents peak firing, corresponding to bursts in the upper panel (see Supplemental Fig. 5). **D** Hypothesised control of the pedal wave for locomotion. Cycling activity on the network is plotted schematically as an ellipse. Activity on specific portions of the trajectory is proposed to recruit motorneurons (M) projecting to muscles in the foot and body wall, whose axons contribute to the suggested nerves. One cycle of the trajectory would thus generate the pedal wave for locomotion by sequentially activating the neck/anterior, middle, then posterior nerves.

trajectory in the ganglion network during a single execution of the motor program (Fig. 6B; Supplemental Video 1), recapitulating the low-dimensional rotational dynamics of the population.

If this was a physical realisation of the rhythmic pattern generator, the activity packet trajectory should correlate with specific phases of movement. Consistent with this picture, in further recordings we found that burst firing of pedal nerve 10, correlated with neck contraction (Xin et al., 1996), in turn correlated with a specific portion of the looped trajectory of activity in the ganglion network (Fig. 6C; Supplemental Video 2). Clustering these recordings showed that phasic activity of our detected ensembles was temporally aligned with the nerve's activity, and so correlated with motor output during locomotion (Supplemental Fig. 5). Our results thus suggest the intriguing possibility that ensembles in the caudolateral quadrant are a network implementing a rhythmic pattern generator.

The motor program is comprised of a few dynamical building blocks

The classification of the 147 ensembles into four classes of oscillator allowed us to decompose and map the dynamical building blocks of the motor program, but had some limitations. Such a classification forced all ensembles into four pre-defined classes, thus pre-determining that there could only be a few dynamical building blocks. In addition, visual inspection of our recordings revealed a wider variety of spike-train structure than

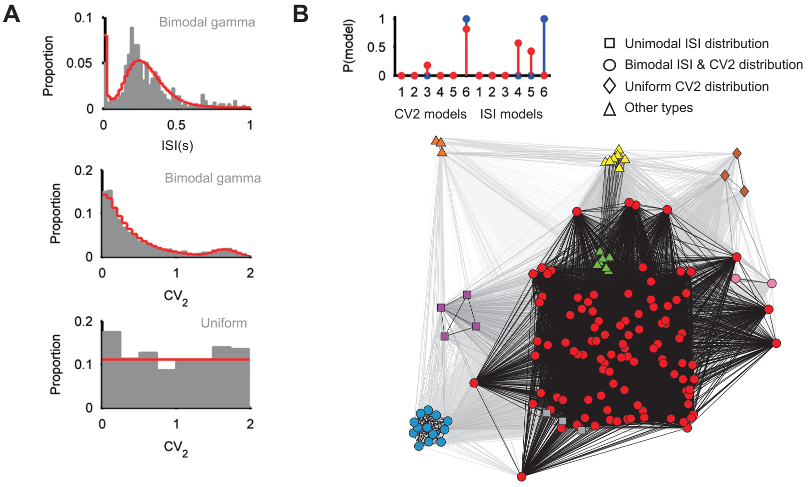

Figure 7: Unsupervised clustering of ensembles in the "fit-space". A Ensemble characterization. Histograms show ISI or CV_2 distributions for three example ensembles, and red lines plot their best-fit models. A distribution for an ensemble was created by pooling distributions for each spike-train in the ensemble. **B** Clustering ensemble groups by spike-train structure. Top-left we plot two examples of concatenated $P(model)$ vectors defining the 12-dimensional "fit-space". Bottom-right: visualization of the network defined by distances between ensembles in this fit-space, and the nine identified ensemble-groups within it (colors); symbols indicate some common properties of the nine groups. Grayscale intensity of lines indicates similarity of nodes.

could be captured by oscillation pattern, and it is likely that each recording captured one or more entirely unknown types of ensemble. To address these issues, starting with the same database of 147 ensembles we repeated steps 2 (classify) and 3 (map) of the analysis (Fig. 1B) using a new unsupervised classification approach based on spike-train structure, thus not presuming any number of dynamical building blocks. We thus develop here a solution to the general problem of clustering ensembles *across* preparations.

To capture the wide variety of spike train structure we used a statistical approach (Compte et al., 2003; Wohrer et al., 2013). Each ensemble was described by its firing rate and regularity, respectively captured by the distribution of inter-spike intervals (ISIs) and the distribution of the local regularity measure CV_2 (Fig. 7A). Capturing each ensemble's dynamics using these complete distributions precludes the use of conventional clustering methods to classify ensembles. To solve this, we developed an approach that clusters in a "fit space" (Experimental Procedures). We fit N models to each distribution and compute the probability that each model was the best fit to the distribution, combined into the length N vector $P(model)$. Here we use $N = 6$ models, covering unimodal (exponential, normal, gamma, log-normal) and bimodal (normal, gamma) distributions (see Supplemental Experimental Procedures 2 for specification). By concatenating the vectors for its ISI distribution and for its CV_2 distribution, each ensemble is then represented as a point in the $(2 \times N)$-dimensional space of rate and regularity properties. Distances between pairs of ensembles in this fit-space are then used as the basis for unsupervised clustering, resulting in groups of ensembles defined by the statistical similarity of their spike-train structure.

We found that the 147 ensembles reduced to nine ensemble groups defined by their spike-train structure (Fig. 7B), supporting the hypothesis that the locomotion motor program is constructed from a limited number of dynamical building blocks. No single recording captured ensembles from all groups (median 4 types of group per recording, range 2-6), showing that our unsupervised "fit space" approach was able to integrate information from across recordings. Figure 8 summarizes the properties of each group. Three groups have bimodal distributions of both ISIs and CV_2 (Groups 1, 2 & 8, totalling 119 of 147 ensembles in the database), consistent with the strongly oscillatory firing in the system. Together these account for the majority of "oscillator" and "burster" classes (Supplemental Fig. 6). A unique ensemble-group (Group 7; 3/147 ensembles) have a bimodal distribution of ISIs, implying that these ensembles are bursting, yet had a uniform distribution of regularity meaning that each burst is a Poisson process. Such highly-irregular firing may indicate that these neurons' afferent network is in the "balanced" state of many asynchronous excitatory and inhibitory inputs (Berg et al., 2007).

Strikingly, two groups of ensembles have a unimodal ISI distribution (Groups 4 & 9, 7/147 ensembles), implying tonic firing throughout their recordings. These confirm that ensembles of neurons with correlated but non-oscillatory firing are a dynamical system present during the locomotion motor program. The lack of oscillatory firing implies that such ensembles do not directly encode the program's sequence of movements, but may play a modulatory role. We further discuss the relationship between the spike-train properties of the nine groups detected here and the known properties of the pedal ganglion in the Supplemental Text.

Using the same approach as the oscillator classes, we plot the resulting "dynamic map" for each ensemble group in Figure 8. We see that these groups are also physically segregated in the ganglion. As an example of the consistency in the mapping across recordings, we see that the ensembles of the Poisson-burst group (Group 7) are found in the same, precise location of the rostro-medial quadrant (Fig. 8). These maps show that, when defined through unsupervised clustering of their spike-train statistics, the dynamical building blocks of the locomotion motor program are physically segregated in the ganglion's network.

Discussion

The implementation of a single motor program in a distributed network is modular, deconstructable into separate dynamical building blocks that are recapitulated in physical space (Figs. 4 and 8). These modules exist at two scales: the individual, discrete ensembles of neurons; and the larger-scale discrete mapping of the different dynamical systems within the pedal ganglion. This doubly-modular nature represents a dramatic dimension reduction: here our data-set of 1145 neurons comprised 147 neural ensembles, which in turn reduced to 9 groups of ensembles with statistically-similar firing. Our results thus support the idea (Koch, 2012) that detecting neural modularity will be highly effective in reducing the dimensional complexity of the brain.

Modular deconstruction of large-scale neural recordings

In tackling the problem of decomposing a motor program, we have developed an analysis pipeline for dynamical mapping of a neural circuit (Fig. 1B). As the pedal ganglion network's implementation of the locomotion motor program contained many unknowns, the key here was to use entirely unsupervised analysis that allowed us to discover components within the large-scale recordings. Placed in sequence (Fig. 1B), the unsupervised steps of ensemble detection and classification allow for a data-driven breakdown of any set of large-scale, cellular-level recordings. Supplemental Fig. 2 shows that ensemble detection using our consensus clustering algorithm is robust to order-of-magnitude or dynamic changes in correlation time-scale, suggesting our methods are generalizable

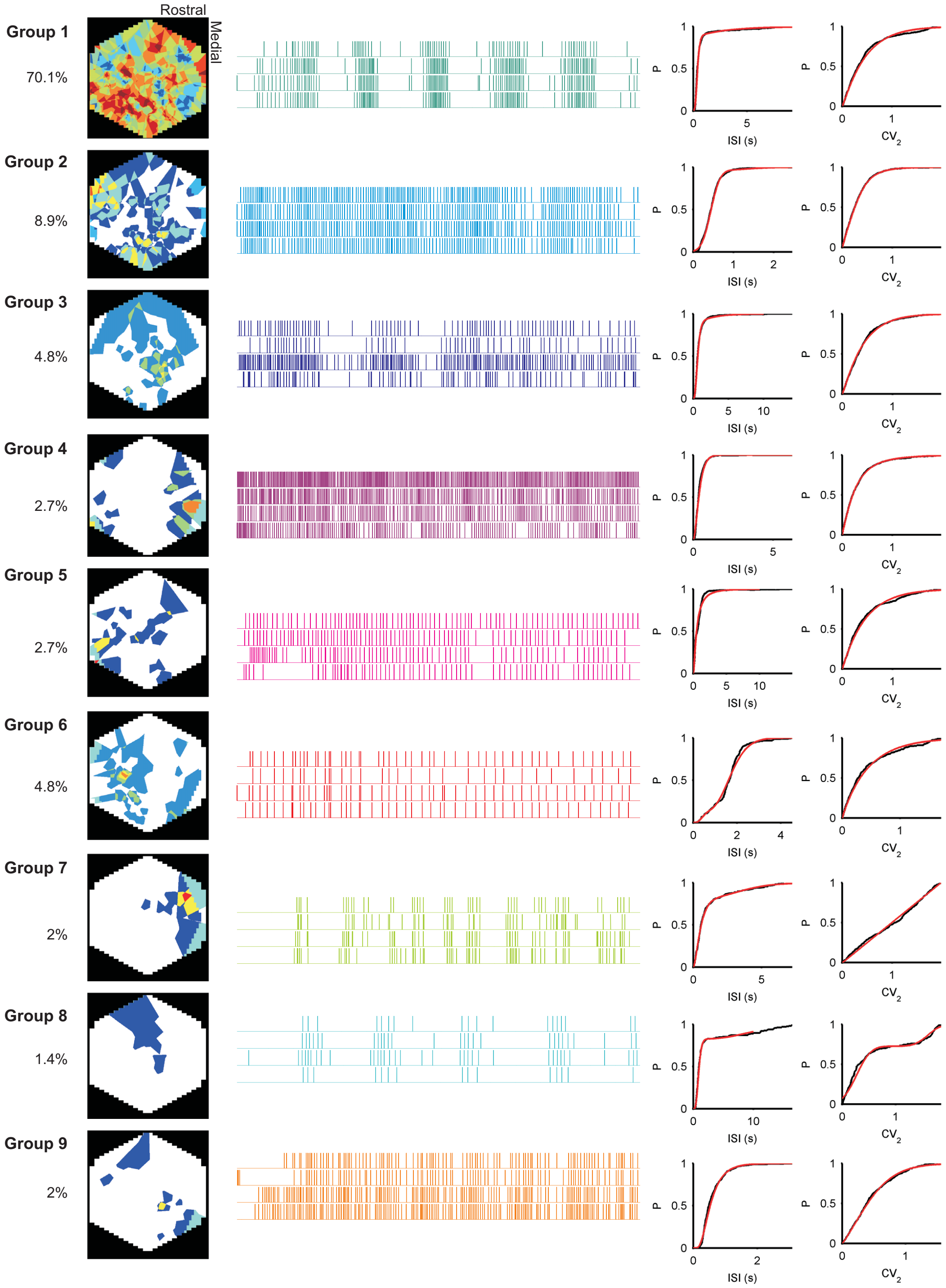

Figure 8: Groups of ensembles defined by spike-train structure. We found nine groups of ensemble distinguished by their firing rate and regularity; percentages on the far left give the proportion of ensembles in each group. In the left-most column we plot the map of all locations containing that ensemble group, with heat intensity (blue-red) indicating the proportion of recordings in which that location contained an ensemble of that group; white indicates no membership of that group detected. Further columns illustrate an example ensemble of each group: the four most-similar spike-trains in that ensemble (80 s duration); and the cumulative distributions for the interspike intervals (ISIs) and the irregularity metric (CV_2) for the ensemble. Black line: data; red line: best fitting model (highest $P(model)$ out of the 6 candidates). Note different scaling of x-axes for cumulative plots. Supplemental Figure 6 maps these ensemble groups defined by spike-train structure onto the distribution of ensembles according to oscillator-class.

to systems with richer, faster dynamics. Code implementing all stages of the analysis is supplied in the Supplemental Software; Supplemental Experimental Procedures 1-3 outline the supplied functions. Ongoing developments of the MATLAB toolboxes are available at: for spike-train community detection: `https://github.com/mdhumphries/SpikeTrainCommunitiesToolBox`; and for analysis and classification of neural ensembles: `https://github.com/mdhumphries/NeuralEnsembleAnalysis`.

We used two different approaches to classifying ensembles, as they give complementary insights. Categorization by oscillation pattern (Fig. 4) allowed us to isolate dynamical systems within a single population recording, and yet identify them across all recordings. The unsupervised classification by spike-train statistics (Figs. 7-8) allowed us to both show that the groupings of ensembles exist independent of any experimenter assumptions, and use the extra statistical power gained by pooling across all recordings to identify unique ensembles that appear in only a small subset of recordings. Both are generally applicable to the output of any ensemble detection method.

Identified hypotheses for locomotion control in *Aplysia*

Generating rhythmic motor behavior requires solving the general problem of generating the underlying repeating cycle of neural activity. The solution for locomotion adopted by many invertebrate brains is to link many segmental ganglia following the longitudinal axis of the animal (Cacciatore et al., 2000; Eisenhart et al., 2000; Gjorgjieva et al., 2013). Their sequential firing then sequentially recruits motorneurons, causing a wave of muscle contractions. The single network of the *Aplysia* pedal ganglion must solve the problem in a different way.

The pattern generator and motorneurons appear to be distinct components in the pedal ganglion (Fredman and Jahan-Parwar, 1980). The motorneurons are not synaptically coupled (Hening et al., 1979; Fredman and Jahan-Parwar, 1980), ruling out the possibility that they are the internal pattern generator, generating and co-ordinating their own bursting activity (Fredman and Jahan-Parwar, 1980). The dominance of excitatory post-synaptic potentials underlying the firing of motorneurons (Hening et al., 1979; Fredman and Jahan-Parwar, 1980) and pedal-peptide neurons (Hall and Lloyd, 1990) points to an excitatory interneuron network within the ganglion. As there are around 1600 neurons in the pedal ganglion (Cash and Carew, 1989), and communication between neurons is predominantly via chemical synapses, a large network of synaptically-connected interneurons could feasibly exist. With this limited prior knowledge of the network, our unsupervised analyses rapidly identified a set of testable functional hypotheses for *Aplysia* locomotion control.

First, our data supports the hypothesis that the rhythmic pattern generator for *Aplysia* locomotion is realized by a network of interneurons in the caudo-lateral quadrant of the pedal ganglion. Here we identified a set of oscillator ensembles (Fig. 4), whose population activity traces a perfect rotation in low-dimensional dynamical space (Fig. 5), recapitulated in the physical space of the ganglion (Fig. 6). These rotations arise because individual neurons within the population had phase-locked oscillatory activity throughout the recordings. It is possible that the richness of observed oscillatory activity across this quadrant is not wholly intrinsic to the pattern generator, as inputs from the intact pleural and cerebral ganglion are known to respectively modulate the amplitude of oscillation (Jahan-Parwar and Fredman, 1979, 1980) and contain the command neurons for self-initiation of locomotion (Fredman and Jahan-Parwar, 1983).

From these data we can sketch a model for locomotion control in *Aplysia*, in which activity in each portion of the loop recruits motorneurons projecting to different muscle groups (as we showed for nerve P10 in Fig. 6C). We illustrate this idea schematically in Figure 6D, concentrating on the known nerves that project from the pedal ganglion to different portions of the foot (and thus control the pedal wave component of locomotion). One prediction of this model is that simultaneous suction electrode recordings from the identified nerves should map onto relatively discrete portions of the trajectory of activity. We do not expect them to be literally as discrete as shown; rather, pedal nerves innervate overlapping areas of the foot, presumably to allow smooth execution of the pedal wave (Fredman and Jahan-Parwar, 1980). As *Aplysia* locomotion includes both crawling and faster galloping (Jahan-Parwar and Fredman, 1979), this model suggests that these locomotion modes correspond to the rate of oscillation in the activity loop.

A second hypothesis is that the rotational dynamics recapitulated in dynamical and physical space are consistent with the direct observation of a cyclical attractor network (Eliasmith, 2005; Rokni and Sompolinsky, 2012; Yoon et al., 2013). This class of networks generate and self-sustain a moving packet of activity (consistent with our observations in Fig. 6, and Supplemental Videos 1-2), and have been proposed to underlie a wide-range of neural systems' dynamics, including grid-cell generation (Yoon et al., 2013) and head-direction representation (Zhang, 1996). Compared to the cyclical attractors embodied by small, fixed central pattern generator circuits (Selverston, 2010), the theoretical advantage for locomotion of a network implementation lies in the ability to produce flexible motor programs of differing speed and/or muscle recruitment by the speed and trajectory of activity in the attractor network (Eliasmith, 2005; Rokni and Sompolinsky, 2012). This cyclical attractor could be representative of the general solution to generating rhythmic activity in a large neuronal network (Yuste et al., 2005; Yoon et al., 2013).

Under this hypothesis, our oscillator ensembles are the observable manifestations of the attractor network's moving activity packet. An alternative explanation of the physical trajectory is that the pattern generator comprises a series of independently oscillating neuron groups: if they were physically arranged in phase order, the location of the activity packet would follow the layout of the groups. To be consistent with our observations, the independent oscillators would be laid out in a loop. Future testing of the attractor-network hypothesis will thus require making use of critical tests for dynamical systems using perturbation. One crucial test would be to perturb the activity of the oscillator-ensembles during an ongoing locomotion program: if they do form an attractor network, then the perturbed activity should return to the same cyclical trajectory. Another crucial test would be to target stimulation to a section of the oscillator-ensemble region at rest: if that section was part of a cyclical attractor network, then sufficiently intense stimulation should initiate the sustained activity packet.

A third hypothesis is that the burster-class ensembles are integrators, as they reliably form an evolving dynamical system over the motor program (Fig. 5B). These ensembles were in the ganglion's rostro-medial quadrant, where motorneurons are predominantly located (Hening et al., 1979). One possibility is that the motorneurons captured in these ensembles are integrating the output of the putative pattern generator together with other modulatory inputs to turn the constant oscillation into a smoothly varying pattern of muscle contractions. Another is that the burster-class represents a subset of neurons integrating input for or against the decision to engage a different behavior (Briggman et al., 2005). Testing these hypotheses in future work requires more thorough sampling of the motorneuron population in this quadrant.

Outlook

Traditionally, functional mapping studies were based on sharp electrodes, typically placing them into at most a few neurons at a time to assess firing patterns during motor programs of interest. Reconstructing networks this way is labor-intensive and takes immense time, often involving successive generations of scientists focused on single model preparations. Locked within the large-scale cellular-level recordings during behavior newly available from a variety of species (Seelig et al., 2010; Ahrens et al., 2012, 2013; Larsch et al., 2013) are the data necessary to reconstruct neural circuits' functional connectivity from single sets of recordings. The data-driven, unsupervised analysis pipeline developed here could unlock this data's potential, reducing the time needed to identify key neuronal components for a given neural circuit by orders of magnitude.

Here we have demonstrated the use of modularity as a static concept to deconstruct the mapping between a motor program and its neural implementation. However, in common with many other neural populations' activity (Briggman et al., 2005; Churchland et al., 2012), the locomotion motor program evolves over its time-course (Hill et al., 2012), displaying moment-to-moment functional reconfiguration. The next, greater challenges are to fully characterize such functional dynamic modularity and the mechanisms underlying it.

Experimental Procedures

Voltage-sensitive dye imaging in the fictive escape locomotion preparation.

The cerebral, pleural and pedal ganglia were dissected out and pinned to the bottom of a Sylgard (Dow Corning) lined chamber containing Instant Ocean artificial seawater (Aquarium Systems). The loose connective tissue surrounding the ganglia and nerves was removed while keeping the sheath intact. The preparation was then transferred and pinned to the floor of a Sylgard-lined recording chamber with a coverslip bottom used for optical recording (PC-H perfusion chamber, Siskiyou). To maximize the number of neurons in focus, the ganglion to be imaged was partially flattened by lightly pressing a shard of cover slip upon it. The shard was held in place with small bits of silicon earplug (Mack's) adhered to the recording chamber floor.

For imaging, the preparation was maintained at $15 - 17°$C, using Instant Ocean passed through a feedback-controlled in-line Peltier cooling system (Model SC-20, Warner Instruments). Temperature was monitored with a BAT-12 thermometer fitted with an IT-18 microprobe (Physitemp, Inc) positioned near the ganglion being imaged. For staining, the room was darkened and the perfusion saline was switched to saline containing the fast voltage sensitive absorbance dye RH-155 (Anaspec). The preparation was perfused with 0.03 mg/ml RH-155 in Instant Ocean for 1 hour at 15°C. An Olympus BX51WI microscope equipped with a 10X 0.6NA water immersion objective was used for imaging. Trans-illumination was provided with light from a 735 nm collimated LED (ThorLabs) which was passed through a 0.9 NA flip top achromat Nikon condenser. The light from the objective was 100% directed either to an Optronics Microfire digital camera used for focusing or to the parfocal focusing surface of a 464-element photodiode array (NeuroPDA-III, RedShirtImaging) sampled at 1600 Hz.

Optical data from the 464 elements were bandpass filtered in Neuroplex (5 Hz high pass and 100 Hz low pass Butterworth filters), and then spike-sorted with independent component analysis (ICA) in MATLAB to yield single neuron action potential traces (the independent components), as detailed in Hill et al. (2010). We recently validated the accuracy of ICA spike sorting using simultaneous intracellular and optical recordings from neurons in various central ganglia of *Tritonia diomedea* and *Aplysia californica* (Hill et al., 2010).

Our set-up and spike-sorting minimizes potential issues caused by light-scattering in optical imaging studies. In invertebrate tissue, our combination of a photodiode array and light wavelength are known to give a scattering radius close to the diameter of single neurons (Cohen and Lesher, 1986; Zochowski et al., 2000). Scattering is minimized further by our imaging with a very shallow depth-of-field, high NA lens, which limits the recording mostly to those neurons visible on the surface, and by our use of a coverslip pressed down onto the ganglion surface, which flattens it and maximizes the number of neurons in focus. The further potential issue of merging neurons with similar activity into one signal is unlikely to arise as individual neurons are defined in our study, not by user outlined regions of interest, but in unsupervised fashion by our ICA spike sorting method.

Rhythmic locomotion motor programs were elicited using 5-10 V 5 ms monophasic pulses delivered in trains from 2-10 Hz for 2-5 s via suction electrode to pedal nerve 9. In some experiments, a separate suction electrode was attached to pedal nerve 10 to continuously monitor the locomotion rhythm (Xin et al., 1996).

Time-scale of each motor program execution

To characterize the time-scale of each motor program execution we sought the time-scale δ that maximized both in- and anti-phase correlation between the whole population's activity, and so captured the time-scale of repeated states of the motor program. Each recording was divided into vectors of neuron activity at time $[t, t+\delta]$, each vector containing one entry for each neuron's spike count in that time window. We correlated all pairs of vectors and computed the dispersion of correlation as $D(\delta) = \{$mean absolute deviation of correlation matrix$\}$ / $\{$ median of correlation matrix $\}$; we picked δ giving the maximum dispersion (Supplemental Fig. 1).

To represent individual spike-trains on this characteristic time-scale, each was convolved with a Gaussian of width σ and unit area over support $\pm 5\sigma$, giving the "spike density" function f. We converted $\sigma = \delta/\sqrt{12}$ (see Humphries, 2011).

PCA of motor program activity

We used standard principal components analysis (PCA) on both the entire set of spike-density functions for each recording, and on sub-sets chosen according to the oscillatory classes found in each recording. See (Briggman et al., 2005) for a tutorial on using PCA to study neural population dynamics. The first few principal components define a set of orthogonal axes that maximally account for the covariance in the data. Each principal component U_i of an analyzed set of spike-density functions contains one entry per neuron in the analyzed population, whose magnitude gives the contribution of that neuron to that component: thus pairs of neurons with anti-phase activity will have large contributions to two different principal components. Note that in such recordings of mixed dynamics, neurons without strongly phasic activity are not strictly orthogonal to others, and so would not be well isolated by PCA. Projections of data onto the ith principal component allow us to visualize the population's dynamics, and are defined at time t by: $z_i(t) = \sum_{k=1}^{n} U_i^k f^k(t)$, where the sum is taken over all n neurons in the analyzed population.

Consensus algorithm for detection of neural ensembles.

We defined functional networks of zero-lag correlations between each pair of neurons. To do so, we computed the correlation coefficient C_{ij} between the spike-density functions (f) for each pair of neurons. The weighted functional network was then: $W_{ij} = C_{ij}$ if $C_{ij} > 0$ and $W_{ij} = 0$ otherwise.

Maximizing modularity Q of this network decomposed the recording into its constituent neural ensembles (Fig. 1C), where modularity is informally defined (Newman, 2006a) as $Q = \{\text{number of links within a module}\}$ - $\{\text{expected number of such links}\}$. An important challenge for us was to produce solutions that individually have high Q yet that can also later be combined across recordings (in steps 2 and 3 of Fig. 1B). To solve this, we extended our previous algorithm (Humphries, 2011) for maximizing Q to produce a consensus clustering of modules for each recording (see Supplemental Experimental Procedure 1). Key advantages of this algorithm over standard clustering approaches include that it automatically determines the number and size of modules; and it can give null answers, as $Q \leq 0$ signals the absence of ensembles (Humphries, 2011). A MATLAB toolbox for this family of algorithms is part of the Supplemental Software. Development of these spike-train community detection algorithms is ongoing at: `https://github.com/mdhumphries/SpikeTrainCommunitiesToolBox`.

Network visualization was done using the MATLAB toolbox of Traud et al (Traud et al., 2009): `http://netwiki.amath.unc.edu/VisComms/VisComms`.

Physical organization of the neural ensembles.

The ICA weight matrix allowed estimation of the two-dimensional co-ordinates of each neuron (Supplemental Experimental Procedures 3). For each recording, we discretized the diode array and constructed ensemble maps by assigning grid squares to neurons in the same ensemble. Contiguous patches of space were defined by all adjacent squares assigned to the same ensemble. See Supplemental Experimental Procedure 3 for details; the code is also part of the Supplemental Software. Dynamic maps of classes and groups were created on a template of the left ganglion oriented with rostral north, so each individual recording's ensemble map was rotated and/or mirrored to that template if necessary.

Finding types of functional ensembles.

We classified ensembles according to their pattern of oscillation. We computed an autocorrelogram A with 1 s bins up to a maximum lag of 20 s; the bin-size was chosen to be smaller than the characteristic time-scale of the motor program in every recording (Supplemental Fig. 1). Significance of oscillations in the auto-correlogram was assessed using a permutation test. We shuffled the inter-spike intervals of the spike train 100 times to produce 100 permuted auto-correlograms A^*, and determined their mean and standard deviation in each bin i. To remove rate-dependence, A was Z-transformed: $Z(i)$ = $[A(i)$ - mean$(A^*(i))]$ / std$(A^*(i))$. We considered a Z-scored bin significant if $Z(i) > 3$ for positive-valued bins and $Z(i) < -2$ for negative-valued bins; note the asymmetry necessary to detect negative correlations in point processes. To further account for noise, two consecutive bins had to meet criterion to qualify as a significant peak or trough.

The"fit-space" approach to clustering each ensemble using the full statistical distributions of its spike-trains is detailed in Supplemental Experimental Procedure 2. This method generalizes to an arbitrary number N of models fit to any choice of S spike-train metrics, giving N vectors for $P(model)$ and an $N \times S$ vector defining each ensemble's position in the "fit space". One key advantage is that the vector acts both as a space in which to classify types and as a mixture-of-models – if the true model is not amongst the N specified, $P(model)$ uniquely weights the contribution of each specified model to the true model. The code for fitting distributions and clustering in the fit-space is part of the Supplemental Software. Further development of these ensemble analysis and classification tools are ongoing at: `https://github.com/mdhumphries/NeuralEnsembleAnalysis`.

Trajectories of motor program activity.

We checked the motor program's evolution in space using the oscillator class ensembles. Each ensemble's activity was characterized by its average spike density function f^*, obtained by taking the mean over the spike density functions f of its neurons. To ensure uniform sampling of the motor program's cycle, we computed the phase-lag between f^*s for all pairs of oscillator ensemble in a recording; phase spread was estimated by von Mises κ, and we chose only recordings with $\kappa = 0$ indicating uniform spread of phase (7/12 recordings). The estimated trajectory of the activity packet on the pedal ganglion was then computed. To make ensemble activity comparable, each f^* was normalized to its maximum rate, giving $\hat{f^*}$. We found the center (x, y) of the largest physical patch of each oscillator ensemble, and computed the current x-axis position $x^*(t)$ of the activity packet by taking the weighted center-of-mass at time t as $x^*(t) = \sum_i^n x_i \hat{f^*}_i(t)$ over all n oscillator ensembles, and similarly computed $y^*(t)$ for y-axis position. Ellipses were fit to trajectories using the direct least-squares method (Fitzgibbon et al., 1999). Mapping of pedal nerve 10 activity was also based on its spike density function f, generated by convolving its spikes with a Gaussian of the same width used for the population recording.

Author Contributions A.B. & W.F. designed the experiments; A.B. performed the experiments; M.D.H. designed the analysis pipeline with contributions from A.B., and developed and coded the analytical tools; M.D.H & A.B. analysed the data; M.D.H., A.B. & W.F. discussed the results; M.D.H. wrote the paper with contributions from A.B. & W.F.

Acknowledgements We thank R. Baines, C. Machens, B. Mensh, and R. Petersen for comments on the manuscript, A. Singh, Y. Billeh & M. Evans for discussions, and J. Wang for technical assistance. M.D.H was supported by a Medical Research Council Senior Fellowship, and the Agence Nationale de la Recherche. W.F. was supported by NIH R01NS060921 and NSF 1257923. A.B. was supported by NIH F31NS079036.

References

Ahrens, M. B., Li, J. M., Orger, M. B., Robson, D. N., Schier, A. F., Engert, F. and Portugues, R. (2012). Brain-wide neuronal dynamics during motor adaptation in zebrafish. Nature *485*, 471–477.

Ahrens, M. B., Orger, M. B., Robson, D. N., Li, J. M. and Keller, P. J. (2013). Whole-brain functional imaging at cellular resolution using light-sheet microscopy. Nat Methods *10*, 413–420.

Berg, R. W., Alaburda, A. and Hounsgaard, J. (2007). Balanced inhibition and excitation drive spike activity in spinal half-centers. Science *315*, 390–393.

Brezina, V., Orekhova, I. V. and Weiss, K. R. (2000). The neuromuscular transform: the dynamic, nonlinear link between motor neuron firing patterns and muscle contraction in rhythmic behaviors. J Neurophysiol *83*, 207–231.

Briggman, K. L., Abarbanel, H. D. I. and Kristan, Jr, W. (2005). Optical imaging of neuronal populations during decision-making. Science *307*, 896–901.

Briggman, K. L. and Kristan, Jr, W. B. (2006). Imaging dedicated and multifunctional neural circuits generating distinct behaviors. J Neurosci *26*, 10925–10933.

Cacciatore, T. W., Rozenshteyn, R. and Kristan, Jr, W. (2000). Kinematics and modeling of leech crawling: evidence for an oscillatory behavior produced by propagating waves of excitation. J Neurosci *20*, 1643–1655.

Cash, D. and Carew, T. J. (1989). A quantitative analysis of the development of the central nervous system in juvenile Aplysia californica. J Neurobiol *20*, 25–47.

Churchland, M. M., Cunningham, John P.and Kaufman, M. T., Foster, J. D., Nuyujukian, P. and Ryu, Stephen I.and Shenoy, K. V. (2012). Neural population dynamics during reaching. Nature *487*, 51–56.

Cohen, L. B. and Lesher, S. (1986). Optical monitoring of membrane potential: methods of multisite optical measurement. Soc Gen Physiol Ser *40*, 71–99.

Compte, A., Constantinidis, C., Tegner, J., Raghavachari, S., Chafee, M. V., Goldman-Rakic, P. S. and Wang, X.-J. (2003). Temporally irregular mnemonic persistent activity in prefrontal neurons of monkeys during a delayed response task. J Neurophysiol *90*, 3441–3454.

Cunningham, J. P. and Yu, B. M. (2014). Dimensionality reduction for large-scale neural recordings. Nat Neurosci , in press.

Eisenhart, F. J., Cacciatore, T. W. and Kristan, Jr, W. (2000). A central pattern generator underlies crawling in the medicinal leech. J Comp Physiol A *186*, 631–643.

Eliasmith, C. (2005). A unified approach to building and controlling spiking attractor networks. Neural Comput *17*, 1276–1314.

Esposito, M. S., Capelli, P. and Arber, S. (2014). Brainstem nucleus MdV mediates skilled forelimb motor tasks. Nature *508*, 351–356.

Fitzgibbon, A. W., Pilu, M. and Fisher, R. B. (1999). Direct least-squares fitting of ellipses. IEEE Transactions on Pattern Analysis and Machine Intelligence *21*, 476–480.

Flood, T. F., Iguchi, S., Gorczyca, M., White, B., Ito, K. and Yoshihara, M. (2013). A single pair of interneurons commands the Drosophila feeding motor program. Nature *499*, 83–87.

Fredman, S. M. and Jahan-Parwar, B. (1980). Role of pedal ganglia motor neurons in pedal wave generation in Aplysia. Brain Res Bull *5*, 179–193.

Fredman, S. M. and Jahan-Parwar, B. (1983). Command neurons for locomotion in Aplysia. J Neurophysiol *49*, 1092–1117.

Frost, W. N. and Katz, P. S. (1996). Single neuron control over a complex motor program. Proc Natl Acad Sci U S A *93*, 422–426.

Getting, P. A. (1989). Emerging principles governing the operation of neural networks. Annu Rev Neurosci *12*, 185–204.

Gjorgjieva, J., Berni, J., Evers, J. F. and Eglen, S. J. (2013). Neural circuits for peristaltic wave propagation in crawling Drosophila larvae: analysis and modeling. Front Comput Neurosci *7*, 24.

Grillner, S., Hellgren, J., Menard, A., Saitoh, K. and Wikstrom, M. A. (2005). Mechanisms for selection of basic motor programs - roles for the striatum and pallidum. Trends Neurosci *28*, 364–370.

Hall, J. D. and Lloyd, P. E. (1990). Involvement of pedal peptide in locomotion in Aplysia: modulation of foot muscle contractions. J Neurobiol *21*, 858–868.

Harris, K. D. (2005). Neural signatures of cell assembly organization. Nat Rev Neurosci *6*, 399–407.

Hening, W. A., Walters, E. T., Carew, T. J. and Kandel, E. R. (1979). Motorneuronal control of locomotion in Aplysia. Brain Res *179*, 231–253.

Hill, E. S., Moore-Kochlacs, C., Vasireddi, S. K., Sejnowski, T. J. and Frost, W. N. (2010). Validation of independent component analysis for rapid spike sorting of optical recording data. J Neurophysiol *104*, 3721–3731.

Hill, E. S., Vasireddi, S. K., Bruno, A. M., Wang, J. and Frost, W. N. (2012). Variable neuronal participation in stereotypic motor programs. PLoS One *7*, e40579.

Humphries, M. D. (2011). Spike-train communities: finding groups of similar spike trains. J Neurosci *31*, 2321–2336.

Jahan-Parwar, B. and Fredman, S. M. (1979). Neural control of locomotion in Aplysia: role of the central ganglia. Behav Neural Biol *27*, 39–58.

Jahan-Parwar, B. and Fredman, S. M. (1980). Motor program for pedal waves during Aplysia locomotion is generated in the pedal ganglia. Brain Res Bull *5*, 169–177.

Jing, J., Cropper, E. C., Hurwitz, I. and Weiss, K. R. (2004). The construction of movement with behavior-specific and behavior-independent modules. J Neurosci *24*, 6315–6325.

Katz, P. S., Getting, P. A. and Frost, W. N. (1994). Dynamic neuromodulation of synaptic strength intrinsic to a central pattern generator circuit. Nature *367*, 729–731.

Koch, C. (2012). Systems biology. Modular biological complexity. Science *337*, 531–532.

Kupfermann, I. and Weiss, K. R. (2001). Motor program selection in simple model systems. Current Opinion in Neurobiology *11*, 673–677.

Larsch, J., Ventimiglia, D., Bargmann, C. I. and Albrecht, D. R. (2013). High-throughput imaging of neuronal activity in Caenorhabditis elegans. Proc Natl Acad Sci U S A *110*, E4266–E4273.

Levi, R., Varona, P., Arshavsky, Y. I., Rabinovich, M. I. and Selverston, A. I. (2005). The role of sensory network dynamics in generating a motor program. J Neurosci *25*, 9807–9815.

Machens, C. K. (2010). Demixing population activity in higher cortical areas. Front Comput Neurosci *4*, 126.

Mattia, M., Pani, P., Mirabella, G., Costa, S., Del Giudice, P. and Ferraina, S. (2013). Heterogeneous attractor cell assemblies for motor planning in premotor cortex. J Neurosci *33*, 11155–11168.

McPherson, D. R. and Blankenship, J. E. (1992). Neuronal modulation of foot and body-wall contractions in Aplysia californica. J Neurophysiol *67*, 23–28.

Mink, J. W. (1996). The basal ganglia: Focused selection and inhibition of competing motor programs. Progress in Neurobiology *50*, 381–425.

Newman, M. E. J. (2006a). Finding community structure in networks using the eigenvectors of matrices. Phys Rev E *74*, 036104.

Newman, M. E. J. (2006b). Modularity and community structure in networks. Proc Natl Acad Sci U S A *103*, 8577–8582.

Niessing, J. and Friedrich, R. W. (2010). Olfactory pattern classification by discrete neuronal network states. Nature *465*, 47–52.

Peyrache, A., Khamassi, M., Benchenane, K., Wiener, S. I. and Battaglia, F. P. (2009). Replay of rule-learning related neural patterns in the prefrontal cortex during sleep. Nat Neurosci *12*, 916–926.

Rokni, U. and Sompolinsky, H. (2012). How the brain generates movement. Neural Comput *24*, 289–331.

Schoener, G. and Kelso, J. A. (1988). Dynamic pattern generation in behavioral and neural systems. Science *239*, 1513–1520.

Schoofs, A., Hckesfeld, S., Schlegel, P., Miroschnikow, A., Peters, M., Zeymer, M., Spie?, R., Chiang, A.-S. and Pankratz, M. J. (2014). Selection of motor programs for suppressing food intake and inducing locomotion in the Drosophila brain. PLoS Biol *12*, e1001893.

Seelig, J. D., Chiappe, M. E., Lott, G. K., Dutta, A., Osborne, J. E., Reiser, M. B. and Jayaraman, V. (2010). Two-photon calcium imaging from head-fixed Drosophila during optomotor walking behavior. Nat Methods *7*, 535–540.

Selverston, A. I. (2010). Invertebrate central pattern generator circuits. Philos Trans R Soc Lond B Biol Sci *365*, 2329–2345.

Summers, J. J. and Anson, J. G. (2009). Current status of the motor program: revisited. Hum Mov Sci *28*, 566–577.

Traud, A. L., Frost, C., Mucha, P. J., and Porter, M. A. (2009). Visualization of communities in networks. Chaos *19*, 041104.

Tsau, Y., Wu, J. Y., Hpp, H. P., Cohen, L. B., Schiminovich, D. and Falk, C. X. (1994). Distributed aspects of the response to siphon touch in *Aplysia*: spread of stimulus information and cross-correlation analysis. J Neurosci *14*, 4167–4184.

Wehr, M. and Laurent, G. (1996). Odour encoding by temporal sequences of firing in oscillating neural assemblies. Nature *384*, 162–166.

Wickens, J., Hyland, B. and Anson, G. (1994). Cortical cell assemblies: a possible mechanism for motor programs. J Mot Behav *26*, 66–82.

Wohrer, A., Humphries, M. D. and Machens, C. (2013). Population-wide distributions of neural activity during perceptual decision-making. Prog Neurobiol *103*, 156–193.

Wu, J. Y., Cohen, L. B. and Falk, C. X. (1994). Neuronal activity during different behaviors in *Aplysia*: a distributed organization? Science *263*, 820–823.

Xin, Y., Weiss, K. R. and Kupfermann, I. (1996). An identified interneuron contributes to aspects of six different behaviors in Aplysia. J Neurosci *16*, 5266–5279.

Yoon, K., Buice, M. A., Barry, C., Hayman, R., Burgess, N. and Fiete, I. R. (2013). Specific evidence of low-dimensional continuous attractor dynamics in grid cells. Nat Neurosci *16*, 1077–1084.

Yuste, R., MacLean, J. N., Smith, J. and Lansner, A. (2005). The cortex as a central pattern generator. Nat Rev Neurosci *6*, 477–483.

Zhang, K. (1996). Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. J Neurosci *16*, 2112–2126.

Zochowski, M., Wachowiak, M., Falk, C. X., Cohen, L. B., Lam, Y. W., Antic, S. and Zecevic, D. (2000). Imaging membrane potential with voltage-sensitive dyes. Biol Bull *198*, 1–21.

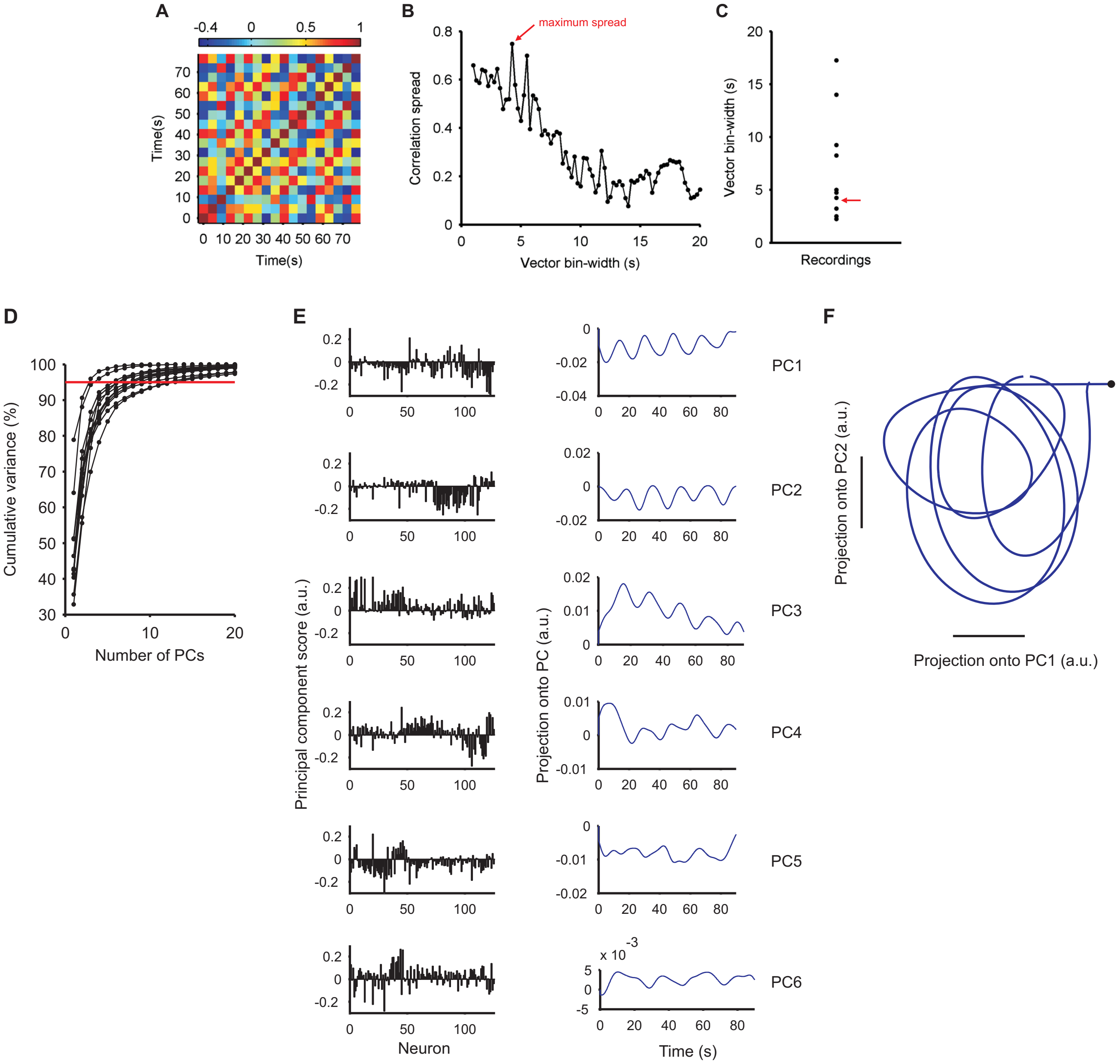

Figure S1: Related to Figure 1. **Variability within and between motor programs. A-C** Characteristic time-scales of motor program execution. We divided each population recording into vectors of neuron activity in some bin-width δ (see Online Methods). **A** Example correlation matrix for population-vectors constructed using a bin-width of 4.25 seconds. **B** Dependency of the spread of correlation in each matrix on the chosen vector bin-width. We measured spread of correlation to simultaneously assess the correlated and anti-correlated state of the population, defined as: {mean absolute deviation of correlation matrix} / { median of correlation matrix }. **C** For each recording, the maximum spread of correlation defined the characteristic time-scale δ for that motor program recording. Each spike-train was then represented on that time-scale by convolving its spikes with a Gaussian of width $\sigma = \delta/\sqrt{12}$; thus this set the time-scale defining a neural ensemble in each motor program. Arrows in panels B and C indicate the maximum spread of correlation for the example recording in panel A. **D-F** Dimensionality of dynamics. We applied PCA to the set of all spike-density functions in a recording. **D** Cumulative variance accounted for by the first 20 principal components (PCs) in each recording, one line per recording. The number of PCs required to account for 95% of the variance (red line) ranged from 2 to 12. **E** Example PCA analysis of recording #5. In this recording 6 PCs account for 95% of the variance. For each PC, we plot its scores (or “factor loadings”) on the left, and the projection of the spike-density functions on to that PC on the right. The mixture of projections shows that the motor program recording likely captured multiple dynamics. **F** Plot of the projection onto the first two PCs; time zero is indicated by the black dot. Rotational dynamics of the population are evident, but cannot be isolated from other dynamical contributions.

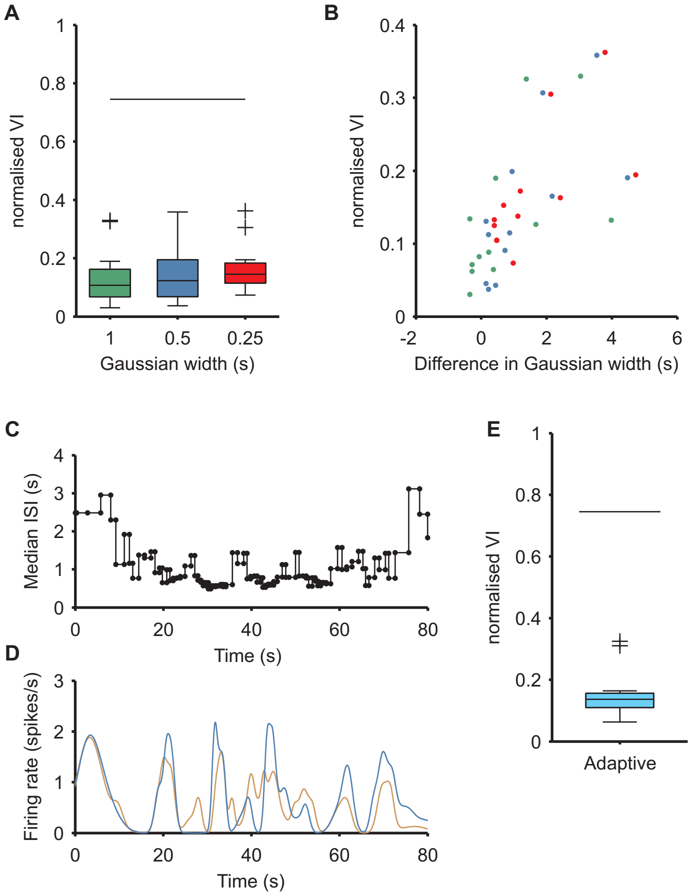

Figure S2: Related to Figure 2. **Robustness of ensemble detection to time-scale of correlation**. In the main text our detected ensembles were found using a different Gaussian width per recording for the spike-train density functions, to account for between-recording variation; here we show that the ensemble structure is robust to a wide range of widths, and so our results do not critically depend upon our choice. **A** Distance between original and new clusterings into ensembles when using the same fixed-width Gaussian window for all recordings. Distance is computed as the normalized variational information (VI), which measures the amount of information needed to turn one set of ensembles into the other; it ranges between 0 (identical ensembles) and 1 (orthogonal clustering). We re-cluster all 12 recordings using 3 fixed-widths; the 0.5 and 0.25 s widths are smaller than the smallest width used in the main text. The black line above the boxplots gives the best-case random assignment for comparison: the same number and size of ensembles, with neurons randomly assigned. We compute the VI between the permuted and original ensembles for 100 random assignments of each recording, and plot the mean here: re-clustering in all cases was thus well above the best-case chance performance. Boxplots show median (center line), interquartile range (IQR; box), data range (1.5 × IQR; whiskers), and outliers. **B** Correlation between the difference in Gaussian width and the resulting distance between the detected ensembles; colours correspond to boxplots in panel A. **C** Time-series of median population inter-spike interval (ISI) for one recording. **D** Example spike-density functions for two neurons generated using an adaptive Gaussian window: the Gaussian's variance was set according to the median population ISI (panel C) at the time of the spike. **E** Distance between original and new clusterings into ensembles when using an adaptive-width Gaussian window for all recordings. Conventions as per panel A. Upper black line gives the result for the same permutation control as in panel A.

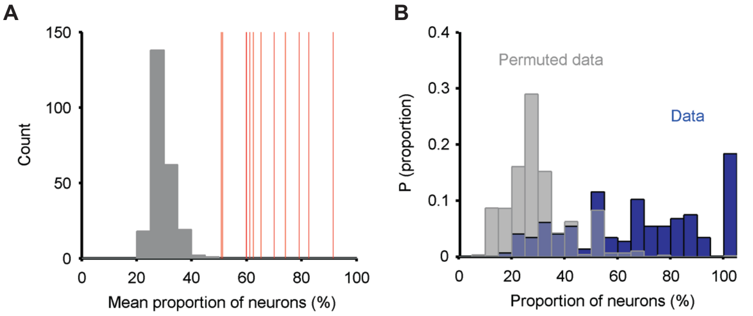

Figure S3: Related to Figure 3. **Physically discrete organization of functional ensembles in *Aplysia* pedal ganglion. A** Map-level discreteness. Red lines correspond to the mean for a map of the proportion of neurons in the largest contiguous patch of each ensemble. The histogram gives the distribution of these map-level means over all permutations of the maps ($n = 240$ permuted maps). All data maps were more discrete than all permuted maps. (Each permutation kept the same number and size of neural ensembles for that recording, but randomly distributed neurons within each, then constructed the maps as per the reconstructions for the data). **B** Ensemble-level discreteness. Histogram in blue gives the distribution over all detected ensembles ($n = 147$) of the proportion of neurons in the largest contiguous patch. Histogram in grey gives the corresponding distribution over all ensembles taken from the permuted maps ($n = 2490$ permuted ensembles). The majority of detected ensembles were more discrete than the most discrete permuted ensemble.

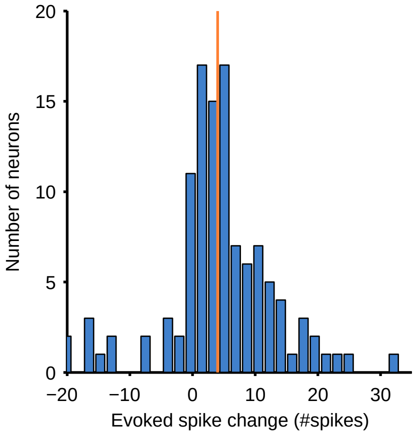

Figure S4: Related to Figure 4. **Evoking the motor program recruits neurons in non-oscillatory ensembles**. Histogram of spike count changes after motor program onset for all neurons in non-oscillatory ensembles. Change is computed as the difference in spike count from the 10 s window before stimulus onset to the 10 s window at the mid-point of the recorded program. The median difference (red line) is significantly greater than zero (Wilcoxon signtest, $P \approx 2.5 \times 10^{-10}$, $n = 114$), showing that neurons in non-oscillatory ensembles were recruited by the motor program.

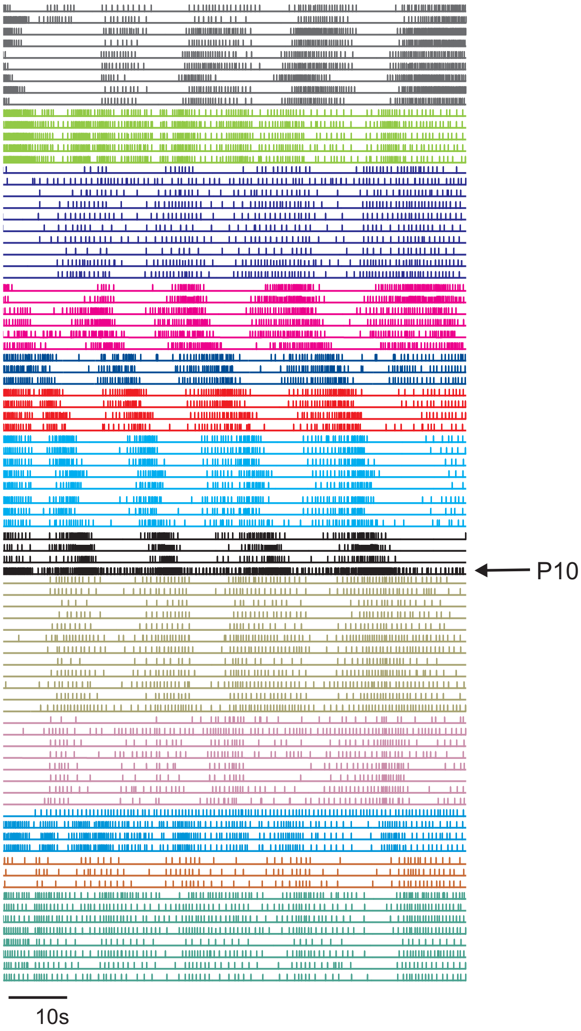

Figure S5: Related to Figure 6. **Simultaneous optical imaging with monitoring of motor output.** Raster plot of all ensembles detected in the optical imaging recording used in Figure 6 and Supplementary Video 2, which had a simultaneous suction-electrode recording from nerve P10. When entered into the clustering algorithm along with all recorded neuronal spike-trains, the spike-train for nerve P10 was assigned to the ensemble in black. The oscillatory firing of other ensembles, such as the four ensembles plotted above the P10 ensemble, is aligned and phase-shifted with respect to P10's firing. The consistent temporal alignment with the firing of this neck-muscle projecting nerve shows that phasic activity of the detected ensembles correlated with the locomotion program.

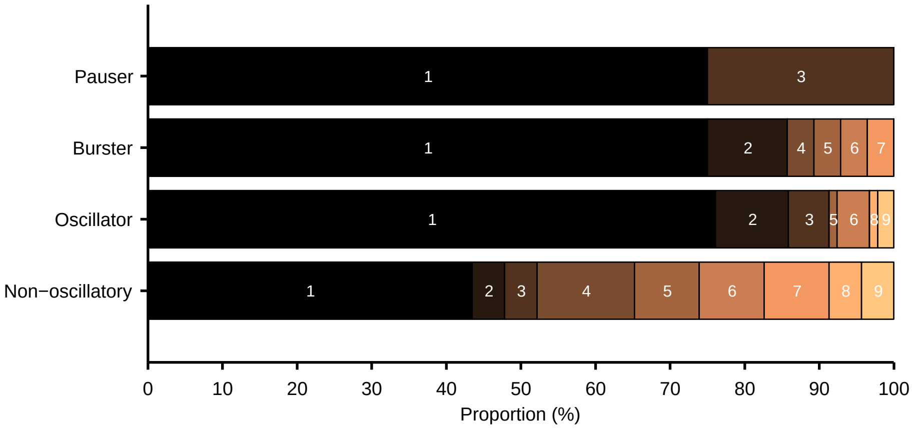

Figure S6: Related to Figure 8. **Breakdown of each oscillator class by identified ensemble groups**. For the set of ensembles in an oscillatory class (Fig. 4 of the main text), we plot the proportion belonging to each of the 9 identified groups of ensembles defined by their spike-train statistics (Fig. 8 of the main text). White numbers indicate the bar segment for each group. Group#1 (with bimodal ISI and CV2) unsurprisingly makes up the majority of the three oscillating classes; but not of the non-oscillatory class, which included the majority of ensembles in groups with unimodal ISI distributions.

Supplemental Video Legends

Supplemental Video 1. Related to Figure 6. **Rotational dynamics in projected and physical space**. For a sample recording, we compare the low-dimensional reconstruction of the population dynamics (right) against the physical trajectory of the same population's activity packet (left). On the right we animate the time-course (in blue) of the projection onto the first two principal components of the population of neurons in oscillator-class ensembles (on the same axes used in Figure 5A). On the left we animate the corresponding trajectory (in red) of the oscillator-class ensembles' activity packet on the plane of the ganglion, superimposed on an image of the ganglion from that recording to show the trajectory's approximate physical location (same axes as Figure 6B). The trajectory was reconstructed for all 80 seconds of the recording.

Supplemental Video 2. Related to Figure 6. **Trajectory rotations correspond to phasic motor output**. Animation of the activity packet's trajectory in a simultaneous recording of population activity and nerve P10. Colour intensity (yellow-red) of the trajectory codes the simultaneous firing rate of nerve P10 (yellow: low; red: high; see Figure 6C). The trajectory was reconstructed for all 80 seconds of the recording.

Contents

Supplemental Procedures 1–3 outline the three main new analytical tools for the unsupervised, parameter-free analysis pipeline. Together they form the user guide for the code supplied in the Supplemental Software. Supplemental Procedure 1 introduces the basic concepts behind community detection algorithms, the spectral clustering algorithm we introduced in (Humphries, 2011), and the key new extension of this algorithm to consensus clustering. Supplemental Procedure 2 describes the motivation and technical details behind the new “fit-space” approach to unsupervised clustering using distributions – here, of spike-train metrics. Supplemental Procedure 3 describes the localisation of each neuron’s physical position, and the motivation and implementation of the new Voronoi tessellation methods for estimating the physical extent of neural ensembles. The Supplemental Text places our new results on the types of neural ensembles and their locations into the context of prior work on the *Aplysia* pedal ganglion.

Supplemental Procedure 1. Community detection and consensus clustering algorithms

Network theory provides a powerful framework for modelling the functional correlations of neurons. By representing each neuron as a node and each correlation as a weighted link, we define a functional network of cellular-level activity (Humphries, 2011). A key challenge in network analysis is how to divide a network into groups — or *modules* — such that the nodes in each module are densely connected with each other but weakly connected to nodes in other modules (for review see (Fortunato, 2010; Newman, 2012)). This is usually termed the problem of "community detection", by analogy with detecting meaningful groupings in social networks. The division of our functional network into modules thus defines potential neural ensembles.

We first briefly recap our community detection algorithm from (Humphries, 2011), which provides a fast, scalable approach that directly maximises a quantitative measure of modularity. We then introduce our novel algorithm for producing consensus clusterings using that community detection algorithm, which allows for highly reproducible groupings of the same data-set.

1.1 Defining the functional network

Our starting point is the weighted network matrix $\mathbf{W}$ whose entries W_{ij} give the positive correlation between neurons i and j. We emphasise that the modular-deconstruction approach to such cellular-level functional networks is highly general. The matrix W can be constructed from a wide range of spike-train correlation measures, on time-scales ranging from milliseconds to seconds (as we demonstrated in (Humphries, 2011)). For example, one could first convolve each spike-train with an exponential, Gaussian, Epanechnikov, or other kernel, then compute pairwise correlations using cosine similarity, the correlation coefficient, or other linear measure. Alternatively, one could compute the pair-wise correlations using metrics specific to spike-train structure, such as the Victor-Purpura distance (Victor and Purpura, 1996), Lyttle-Fellous burst metric (Lyttle and Fellous, 2011) or the parameter-free ISI-distance (Kreuz et al., 2007). Modules within the functional network then quantitatively define a "neural ensemble" independently of the specific time-scale and choice of pair-wise correlation measure.

In the main text we used a rectified correlation coefficient computed between pairs of Gaussian-convolved spike-trains at zero-lag. These choices were made as we were interested in simultaneous spiking of neurons in common phases of the overall oscillatory motor program.

1.2 Community detection using spectral decomposition

Our goal is to divide the functional network into its component modules. We seek to optimise the modularity Q of this matrix, informally defined as Q = {number of links within a module} - {expected number of such links}. Optimising the modularity thus requires the placement of all nodes into an unknown number of modules of unknown sizes.

To begin, we form the modularity matrix (Newman, 2006a)

$$\mathbf{B} = \mathbf{W} - \mathbf{P}, \tag{1}$$

where $\mathbf{P}$ is a matrix encoding a suitable null model network; following (Newman, 2006a; Humphries, 2011) we use the model with entries between nodes (i, j) of $P_{ij} = k_i k_j / m$, where k_i, k_j are the total weight of links made by nodes i and j, and m is the total weight of the network. This model gives the ensemble average of networks with the same expected distribution of weights per node as the data network, ignoring any correlation between nodes that may arise from modular structure in the data network.

Assume we are given some arbitrary assignment of all nodes n to a set of M modules, encoded by the $n \times M$ binary matrix $\mathbf{S}$: each of the M columns encodes the membership of that module, so that entry $S_{ij} = 1$ for node i in module j, and is 0 otherwise. From this, we can quantitatively define the informal notion of modularity as (Newman, 2006a)

$$Q = \text{Tr}(\mathbf{S}^T \mathbf{B} \mathbf{S}), \tag{2}$$

where Tr indicates the matrix trace, and ^{T} is the matrix transpose.

We thus seek some assignment of nodes in $\mathbf{S}$ that maximises Equation 2. Our starting point is an eigenspectra decomposition of $\mathbf{B}$: as $\mathbf{B}$ is a centred matrix with mean of zero, the number of p positive eigenvalues indicates the maximum number of detectable modules in the network (Newman, 2006a).

We proceed as follows (Humphries, 2011). The p eigenvectors corresponding to all positive eigenvalues form a p-dimensional projection of the modularity matrix, in which entry i in each eigenvector together specify the p-dimensional co-ordinates of node i. We thus cluster in this space to determine the module membership. We consider in turn each possible number of modules between $k = 2$ and $p + 1$ (as the first eigenvector denotes two modules, given by the sign of its entries). For each k, we repeat k-means clustering 100 times, using Euclidean distance. Each k-means clustering returns an assignment of each node to a module, for which we compute Q.

Out of all $100 \times p$ clusterings, the one that maximises Q is a viable definition of the neural ensembles present in the recording. In (Humphries, 2011) we showed how this can give insights into the evoked firing of dopaminergic neurons, and the spontaneous activity of cortex in vivo. However, the need to combine modules from across recordings presented a stern new challenge.

1.3 Consensus community detection

One distinct advantage of using modularity optimisation is that not only is it a demonstrably powerful approach to the analysis of networks (Newman, 2006a,b; Nadakuditi and Newman, 2012), but also its limitations are very well characterised. This both allows us to interpret its output correctly, and to understand how to solve specific problems. The search space for modules is astronomically large: for N neurons, the number of possible groups ranges between 1 (all neurons in one group) and N (each neuron in its own group), and so the total number of group permutations is given by the Bell number for N; for 100 neurons, the number of possible groupings is $\sim 4.75 \times 10^{115}$. Given the astronomical size, sampling this space usually requires that the algorithms are stochastic in that each single iteration of the algorithm starts from a different initial condition (as in the k-means step from the above algorithm) and/or starts with a different initialisation of the random number generator.

Like all known benefit or cost functions for stochastic unsupervised clustering, maximising Q is a NP-complete problem (Brandes et al., 2006); thus all algorithms can only be heuristic approximations to finding the solution with maximum Q within this astronomical search space. Moreover, it is known that many solutions with high values of Q exist, and often form plateaus around the maximum-valued solution(s) (Good et al., 2010). Consequently, each run of any good stochastic unsupervised clustering algorithm, such as the above, is likely to return good solutions that are related but different sets of modules.

In practical contexts, such variability can be undesirable. In many applications, we have a set of networks representing a set of related samples from the system under study, and we wish to combine the analyses of those networks to build a picture of the whole system. We thus would like to combine the results of community detection on each network, but this is particularly difficult to do if applying an algorithm to each network will return different good clusterings. A method for finding both a good solution and a consistent solution is needed to allow such combinations.

In this regard neural recording studies, whether cellular-level as here or at the macroscopic scale of fMRI and EEG (Bullmore and Sporns, 2009), are particularly challenging because each data-set that defines a single network is a noisy incomplete sample from a large, relatively unknown neural system. Understanding the neural system thus requires the combination of all sampled networks. In the present study, the challenge is to define neural ensembles consistently between recordings, so that we can combine those recordings to quantitatively examine the common functional and physical properties of the *Aplysia* pedal ganglion.

Our solution here is to use the consensus clustering approach (Monti et al., 2003; Nguyen and Caruana, 2007). The general idea is that a stochastic clustering algorithm will return a total of g clusterings, and each clustering G contains some information about the most reliable partition of the data-set into groups. Lancichinetti and Fortunato (Lancichinetti and Fortunato, 2012) recently introduced the use of this idea into the problem of community detection. Inspired by this, we developed a novel consensus community detection algorithm that, crucially, is parameter-free: it self-determines both the appropriate clustering and its convergence.

Our consensus algorithm is generally applicable to arbitrary networks. For our present purposes, it provides a powerful solution to the problem of generating reliable neural ensembles before comparison between and merging across recordings.

1.3.1 Consensus algorithm

Given the initial weighted network $\mathbf{W}$, we retain all $g = 100 \times p$ clusterings in set $\mathbb{G}$ produced by the above spectral clustering algorithm of modularity maximisation. We then iteratively proceed as follows:

1. From set $\mathbb{G}$ remove all clusterings that have $Q \leq 0$ as these provide no information about modularity, giving set $\mathbb{G}^*$ with g^* clusterings in total.

2. Produce the *consensus matrix* $\mathbf{C}$: entry $C_{ij} = n_{ij}/g^*$ is the proportion of times each pair of nodes (i, j) are placed in the same module.

3. Check for convergence of the consensus matrix (see below):

   (a) If converged, stop; return the single clustering defined by the consensus matrix (see below)

   (b) If not converged, cluster the consensus matrix using the spectral algorithm giving a new set of clusterings $\mathbb{G}$; repeat from step (1).

   (c) If not converged after 50 iterations, exit with current best answer.

Note that this algorithm is simple because the consensus matrix defines another weighted network that can be directly input to the spectral algorithm. Figure P1 shows this algorithm in action on one of the recordings in this paper.

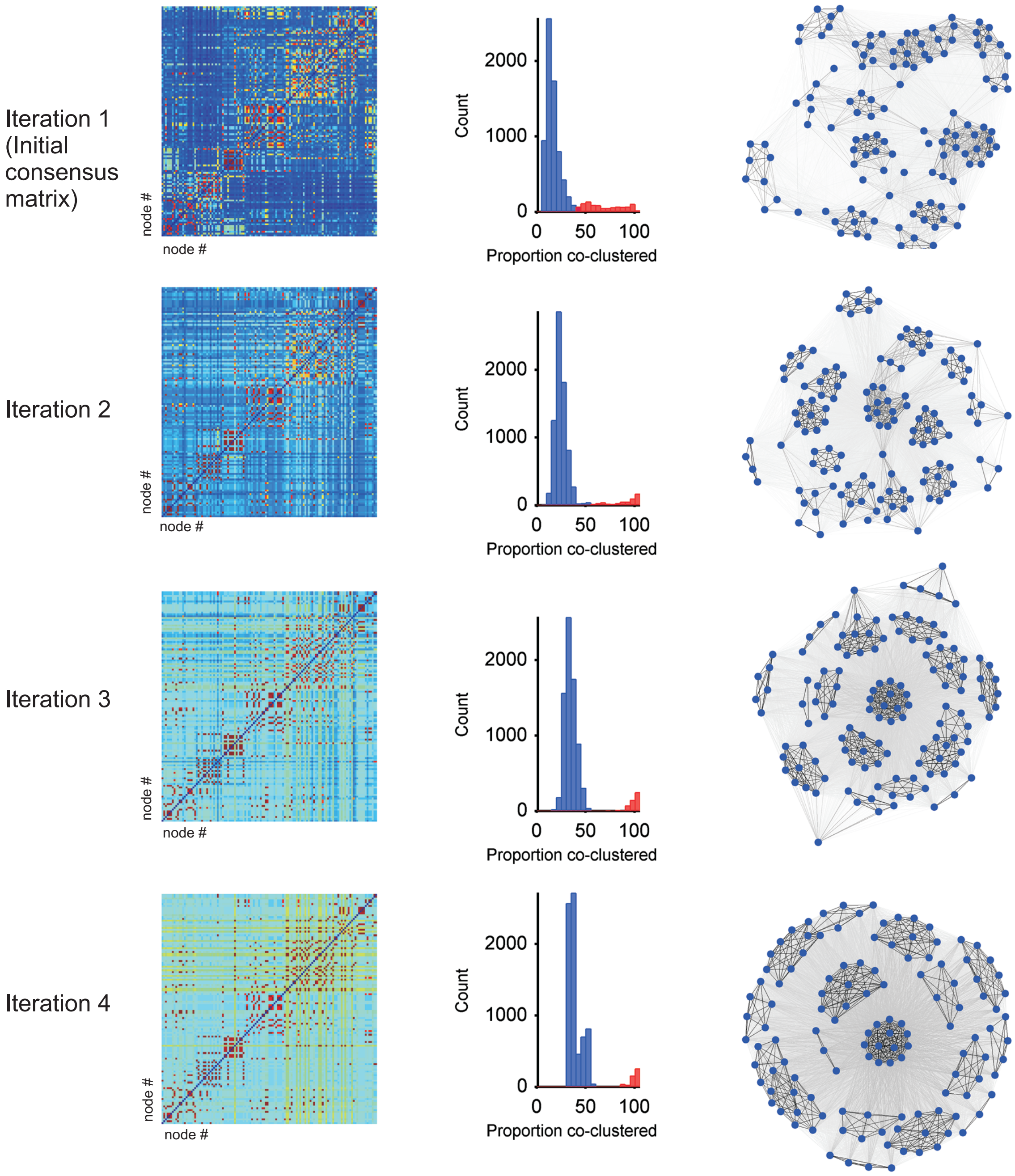

Figure P1: Example output of the consensus algorithm for community detection. Each row shows properties of the consensus matrix **C** for one iteration of the algorithm, from the initial consensus matrix built from the clustering of the recorded data, to the final converged matrix after four iterations. The rows depict: left, the consensus matrix visualised as a pseudo-colour plot, colour intensity (blue-red) of each square coding the proportion of times that pair of nodes (i, j) are placed in the same module; middle: histogram of entries in the consensus matrix, colour-coded to show the two distributions ($\mathbb{C}^{low}$,$\mathbb{C}^{high}$) found by the k-means clustering of those entries; right: network visualisation of the consensus matrix, with each link's greyscale intensity proportional to the consensus matrix entry for that pair of nodes. The network visualisation clearly shows how the increasing bimodality of the consensus matrix over the iterations corresponds to the separation of the original data-set into a single set of modules.

Convergence The values of the consensus matrix will become bimodal in the presence of any detectable modular structure: the distribution around the upper mode will comprise entries corresponding to pairs of nodes consistently clustered together because of their shared membership of a common module; the distribution around the lower mode will comprise entries corresponding to pairs of nodes clustered together at random (Figure P1). Thus, our convergence condition is that the entries of the upper distribution define a single, unique clustering of the nodes.

On each iteration, we check this criterion as follows:

- We cluster the entries of $\mathbf{C}$ into two groups $\mathbb{C}^{low}$ and $\mathbb{C}^{high}$ corresponding to the two modes. We used k-means with $k = 2$ and initial centroid values of [0.4,0.9].
- We then check for the presence of a single clustering in the upper distribution. We first create two lists: an empty list of assigned nodes, and a list of unassigned nodes that initially contains all nodes.

  1. Picking any node at random from the unassigned list, we create a module containing that node and all other nodes with which it has a non-zero entry in $\mathbb{C}^{high}$.
  2. If any of these nodes are already in the assigned list, then the upper distribution does not define a unique clustering. We exit and continue the consensus algorithm.
  3. Otherwise these nodes are moved to the assigned list.
  4. Repeat from step (1) until either we have exited (at step 2) or all nodes are assigned.

Thus, if each node is assigned, then we have produced a single consensus clustering of all nodes into modules.

This convergence criterion was reliable for every recorded data-set from this study, as well as all further example networks we have tested (the catch condition to exit after 50 iterations was never reached). Every data-set converged to a bimodal distribution, and the upper distribution always uniquely defined a single consensus clustering. For the 12 networks in the data-set, the mean number of iterations to convergence was 2.25.

1.4 Code

All MATLAB functions for running the consensus community detection algorithm are in the ConsensusCommunityDetectionToolbox folder of the Supplemental Software. Each function has extensive help text within it; here we give an overview of each function's role. Ongoing development of this toolbox is hosted at:
https://github.com/mdhumphries/SpikeTrainCommunitiesToolBox.

Main functions:

consensus_cluster_spike_data_binless The top-level function: this takes a dataset of spike-trains, and executes all steps necessary to run the algorithm on spike-train data – convolution of each train with a function, computation of the similarity matrix from all pairs of convolved trains, and executing the consensus algorithm itself.

allevsplitConTransitive The consensus algorithm function: this takes a similarity matrix, and executes the algorithm described above. This calls the helper functions:

expectedA Computes the null model matrix **P**

spike_train_from_times Expands spike-train representation for convolution

Useful functions:

plot_clusters Visualisation of detected ensembles. Takes the spike-train data-set and a grouping of that data-set; plots the spike-trains as a raster, ordered into their detected groups

sortbysimilarity Computes various measures of each cluster's similarity, and sorts the group IDs into order of similarity

MIpartitions Computes the normalised mutual information between a pair of clusterings of the same data-set

VIpartitions Computes the variational information between a pair of clusterings of the same data-set

The toolbox is used by top-level scripts in the Supplemental Software:

Consensus_Cluster_Dataset uses the toolbox to detect the ensembles present in each recordings in the supplied data-set

Supplemental Procedure 2. Fit-space: finding types of neurons and ensembles

Having recorded from many individual neurons and detected the neural ensembles in each recording, the key challenge is building a picture of the neural system using this data. Each recording represents a partial, noisy sample of the system, so we need robust tools to integrate across the samples. The first step is to identify common characteristics of firing properties across the recordings, to build a database of types of ensembles in the neural system.

Our goals here include:

1. Ensemble types, existence: ensembles are defined by the temporal correlations of their neurons' spike-trains; but each ensemble will then also have a characteristic set of spike-train statistics; how many types of ensemble exist when we take these statistics into account? (e.g. tonic, bursting, oscillatory, other?)

2. Ensemble types, use of: by identifying ensemble types, we can then refine all further analyses of ensemble-level dynamics by selecting the type of ensembles we are interested in (e.g. for sequential firing, we can select out only oscillatory types).

3. Cataloguing neurons and ensembles: large-scale recordings allow us to rapidly sample across a large subset of simple neural systems, reducing the time to gather data by orders of magnitude compared to traditional single cell extra- or intra-cellular recordings. To date, the only well-characterised large sample of neuron types in the *Aplysia* pedal ganglion are the motorneurons of the rostro-medial quadrant (Hening et al., 1979) and neurons contributing axons to pedal nerve 9 (Fredman and Jahan-Parwar, 1980). So we have the opportunity to map the largely unexplored majority of the ganglion.

4. Novelty of neurons and ensembles: we would like to know if there exists any particularly unusual types of ensembles and neurons in the pedal ganglion, which would merit further investigation on their own.

To achieve these goals, we want to characterise the ensembles using a range of measurements, and use unsupervised clustering on these measurements to identify groups of similar ensembles. That is, we might imagine there exists a small set of "prototype" ensembles of which the actual recorded ensembles are noisy realisations.

We should note that this problem of unsupervised electrophysiological classification is of very general interest in neuroscience – a well-known example being the efforts of cortical electrophysiologists to automate categorisation of the broad range of interneurons and pyramidal neurons (Cauli et al., 2000; Petilla Interneuron Nomenclature Group, 2008; Defelipe et al., 2013). A particular challenge facing the use of large-scale recording technology (Bartho et al., 2004; Fujisawa et al., 2008) is that we only have access to spike-trains, and not other electrophysiological parameters that usefully distinguish neuron types.

Thus the challenge here is to use only measures of spike-train structure to define these types, and find a combination of these measures' values to uniquely define each type. The key problem to solve is that each neuron's spike-train potentially has a complex structure, so we need a way to succinctly capture this structure without obscuring differences between neurons, yet in a way that allows for unsupervised clustering.

2.1 Spike-train measures for defining types

We defined types of neural ensembles based on the structural properties of their spike trains in terms of the firing rate and firing regularity. The former gives us clues about the excitability of the neuron, the latter gives us clues about the intrinsic dynamics of the neuron; together they form a picture of how a neuron integrates its inputs into its output. We refer to this throughout as the rate-regularity type of an ensemble.

For firing rate, we simply use the inter-spike interval (ISI), and calculate all ISIs for each spike-train. Typically spike-train rates are summarised by the mean ISI or its reciprocal f = 1/mean(ISI) spikes/s. However, the mean ISI is not broadly distributed for the neurons in the pedal ganglion, and so is a poor indicator of differences in spike-train structure.

Many measures of spike-train regularity are available. The most common choice is to use the coefficient of variation (CV) of the ISI: CV(ISI) = std(ISI)/mean(ISI). This has the nice property that CV(ISI) = 1 for a Poisson point process, and so CV(ISI) < 1 indicates greater regularity and CV(ISI) > 1 indicates greater irregularity. However, as it is measured from ISIs taken over the whole recording, it is sensitive to changes in firing rate over the recording because the standard deviation of the ISIs will reflect those changes.

To overcome this problem, we used the so-called CV_2 (Holt et al., 1996), which is both less sensitive to rate changes and allows for examination of the changes of irregularity over time (Ponce-Alvarez et al., 2010; Wohrer et al., 2013). This is a local measure of regularity between adjacent ISIs I_i and I_{i+1}:

$$CV_2(i) = \frac{2|I_i - I_{i+1}|}{I_i + I_{i+1}}, \tag{3}$$

which is the ratio of the twice the difference to the sum of those adjacent ISIs. Typically the spike-train is then summarised by taking the mean $\bar{CV}_2$ over all $CV_2(i)$ with $i = 1 \dots n-1$ for n ISIs. Again this has the nice property that $\bar{CV}_2 = 1$ for a Poisson point process, and so $\bar{CV}_2 < 1$ indicates greater regularity and $\bar{CV}_2 > 1$ indicates greater irregularity. Moreover, we also have the option of examining the distribution of the $CV_2(i)$s themselves, which we use below.

2.2 Defining types by distributions of metrics

We computed the above measures for every spike-train in the data-set. Spike-train structure is difficult to summarise using simple aggregates of these measures. Our data-set indeed showed clear problems with using simple aggregate measures to characterise neurons: visual inspection showed that many neurons had bimodal distributions of ISIs and of $CV_2(i)$s (e.g. Figure 7 in the main text), and so mean rate (mean ISI) or mean regularity ($\bar{CV}_2$) did not capture even the central peak of the data, let alone the shape of the distribution. Consequently, we wished to use the complete distribution of ISIs and of $CV_2(i)$s to capture the complexity of each neuron's spike-train structure.

To identify ensemble types we needed to also characterise their rate and regularity properties. One option would be to use an aggregate over all neurons in the ensemble – such as the mean firing rate – but given the problems with using such aggregate measures for individual neurons, pooling them across multiple neurons would cause further issues in capturing the structure of the ensembles' spike-trains. We thus characterised each ensemble by a single ISI and a single $CV_2(i)$ distribution created by pooling the distributions for each neuron in the ensemble.

2.3 Summarising distributions of spike-train metrics

To directly use these distributions of rate and regularity to characterise each ensemble's spike-train structure we must first quantitatively summarise them. We thus fit models for a range of possible distributions (see below). However, this seems incompatible with our ultimate goal of finding ensemble types by unsupervised clustering: first, for unsupervised clustering each object must be defined by the same length N vector of properties; for each ensemble that would be something like: [mean(rate), std(rate), mean(regularity), std(regularity),...]. We then cluster in this N-dimensional space. But fitting a complete distribution model would provide either different numbers of parameters or incomparable parameters for each distribution, depending on the fitted model (for example, one free parameter for an exponential distribution versus two for a normal distribution; the mean and standard deviation for the normal distribution versus the a,b parameters for the Gamma distribution). Second, even if we arrange for each distribution to be characterised by the same set of parameters, that leaves the problem of how to choose the right distribution model(s).

2.4 Fit space

We introduce here the idea of "fit space" to solve both problems simultaneously: we fit a range of m models to a distribution, and find for each model M the probability $p(M)$ that it is the best fitting of the m candidates. For that distribution we now have a length m vector $P(model)$ defining the relative fits of all the models – for example for $m = 5$ we might have $P(model) = [0, 0.1, 0, 0.7, 0.2]$. By repeating for all distributions, we then have every ensemble represented as a point in this m-dimensional space of "fits". We can then cluster to determine if there are distinct types of ensemble present in that space.

This approach has two particularly appealing properties. First, we can go on to include other properties we wish to use to define "types" (for example, the mean phase of spiking with respect to some global oscillation) by simply concatenating their measured values to the length m vector and increasing the dimensions of the space. We can even concatenate two or more fit-spaces, as we do in the main text for the rate and regularity vectors for $P(model)$. Second, amongst the m candidates we do not need to guess the correct model for every single distribution in the data. The m-dimensional vector of probabilities can also be interpreted as mixture weights for combining the candidate distributions to obtain the true distribution. If a particular type of ensemble has a distribution of, say, ISIs that is not in the m candidates, then it is likely that all ensembles of that type will have approximately the same vector of probabilities giving the mixture of models closest to the true distribution. Thus all ensembles of that "type" will still have points in approximately the same region of fit space, and will still be found by the unsupervised clustering.

2.4.1 Choosing and fitting candidate models

Our choice of models was guided by the twin needs to broadly sample the range of possible distributions (to allow a mixture interpretation to make sense) and to capture the evident dominant distributions of ISIs and $CV_2(i)$s in the data from visual inspection of a sample of neurons. Our final set of $m = 6$ candidates were: (1) Normal ($n = 2$ parameters); (2) Log-normal ($n = 2$); (3) Gamma ($n = 2$); (4) Uniform ($n = 2$); (5) Bimodal Normal ($n = 5$); (6) Bimodal Gamma ($n = 5$). Note that both bimodal models include a mixture parameter.

Two are particularly unusual. The uniform distribution was included because this is the expected distribution of CV_2(i)s for a Poisson process (as each possible pair of adjacent

ISIs occurs with equal probability, so their ratios as expressed in $CV_2(i)$ must also occur with equal probability). The bimodal distributions of ISIs and $CV_2(i)$s visually observed in the data often were not symmetric around the two modes, and thus a bivariate Gamma model was included in an attempt to capture this asymmetry.

All models were fitted to each distribution using maximum likelihood estimation (MLE). Models (1)-(4) have analytical MLE solutions for their parameters, and are also available in MATLAB's Statistics Toolbox. Models (5) and (6) do not have analytical MLE solutions for their parameters, so we used numerical optimisation of their maximum-likelihood fits to the distribution using MATLAB's mle and fmincon functions. Examples fits are shown in Figure 7 and Figure 8.

2.4.2 Calculating $P(model)$

For each distribution the fitting process gives us a vector of negative log-likelihoods L, one per model. One option is to interpret the model with the lowest L as the best-fit to the distribution. However, this does not take into account that the models have different numbers of parameters, and we must always take care that a model with more parameters is not simply over-fitting the data. Thus we choose some measure of model-selection that computes the trade-off between the goodness-of-fit and the number of parameters. A wide-range of model-selection approaches are available (see e.g. (Wasserman, 2004)). Here we will use the Bayesian Information Criterion (BIC) as it tends to more heavily penalise the number of parameters, and thus makes the selection of the bimodal models (5-6) more convincing. But equally we could have used Akaike's Information Criterion (AIC) or others.

Given that L is the negative log-likelihood of model M_i, the BIC is

$$\text{BIC}(M_i) = 2L + k\ln(n) \tag{4}$$

where k is the number of free parameters and n is the number of data-points in the distribution. The model with the lowest BIC score is thus considered the best-fit in that it best trades-off the goodness-of-fit (as measured by L) with the number of free parameters (as given by the second term above). However, the absolute BIC values cannot tell us whether that model is notably better than the next-best model: as these values scale with the number of data-points n, so their differences might be small on the scale of the BIC values. Instead, using the BIC values we can compute the approximate posterior probability that each model is correct (Wasserman, 2004):

First for each model we compute the difference between its BIC value and the minimum of the BIC values: $\Delta_i = \text{BIC}(M_i) - \min_{j\in m}\text{BIC}(M_j)$.

Second, we compute the posterior probability that this model is correct by:

$$p(M_i) = \frac{\exp(-\Delta_i/2)}{\sum_{j=1}^{m}\exp(-\Delta_j/2)}. \tag{5}$$

Thus we now have our fit-space vector: $P(model) = [p(M_1), p(M_2), ..., p(M_m)]$ computed for each of the m models applied to the distributions. We do this for the distribution of ISIs and the distributions of $CV_2(i)$s for each ensemble.

2.4.3 Finding "types" in the fit-space

Our overarching goal was to find properties of ensemble activity that uniquely define "types". This is another unsupervised clustering problem in an N-dimensional space: we take N metrics for spike-train structure, and n objects (here, ensembles) with scores for each metric, and we ask if this n-dimensional data-set can be reduced to C clusters, where each cluster is a defined "type". In the main text we use a 12-dimensional vector, made by concatenating the 6-dimensional $P(model)$ vector for rate (ISI distribution) with the 6-dimensional $P(model)$ vector for regularity ($CV_2(i)$ distribution).

We use our consensus community detection algorithm to also tackle this unsupervised clustering problem. We construct a weighted network $\mathbf{D}$ for the entire dataset using the Euclidean distance between the fit-space vectors for each pair of objects. Note that we use Euclidean distance and not a correlation-type measure due to the likely small number of non-zero entries in the vectors. We then convert this to a similarity matrix $\mathbf{W}$ by $W_{ij} = \exp(-D^2_{ij})$. This similarity matrix is then passed to the consensus community detection algorithm. As noted in the main text, we constructed an ensemble-level network for all 147 ensembles in the data-set.

2.5 Code

All MATLAB functions for creating and clustering in the "fit-space" are in the Supplemental Software. Each function has extensive help text within it; here we give an overview of each function's role. Ongoing development of this ensemble analysis and classification toolbox is hosted at: https://github.com/mdhumphries/NeuralEnsembleAnalysis.

Main functions:

fitMLEdistribution Takes a distribution of some variable, and fits the specified models to it using maximum likelihood; also computes the AIC and BIC scores for the fits, and corresponding estimates of $P(model)$; the latter specify position of that distribution in the fit-space. This function calls:

- **bimodalGaussianPDF** computes the PDF for a bimodal Gaussian model
- **bimodalGaussianCDF** computes the cumulative distribution function (CDF) for a bimodal Gaussian model
- **bimodalGammaPDF** computes the PDF for a bimodal Gamma model
- **bimodalGammaCDF** computes the CDF for a bimodal Gamma model
- **BICL** Compute BIC scores from a set of (negative) log-likelihoods
- **AICL** Compute AIC scores from a set of (negative) log-likelihoods
- **Akwgts** Compute the approximate posterior probability $P(model)$

These functions are used by top-level scripts in the Supplemental Software:

Analyse_Spike_Train_Properties computes statistical properties of each neuron's and ensemble's spike-trains, and fits models to their distributions using fitMLEdistribution

Types_of_Ensemble_Across_Dataset Clusters in the fit-space. Computes the similarity between each pair of ensembles, and uses the consensus community detection algorithm to cluster ensembles in that space.

Supplemental Procedure 3. Building maps of ensemble locations

The following outlines the construction process for the physical map of the neuron locations and extent of each neural ensemble from each recording. The ultimate goals were to understand: (1) how cohesive each functionally-identified ensemble was in physical space; and (2) the tessellation of physical space by the ensembles.

3.1 Neuron location

We first had to establish the location of each neuron in the two-dimensional recording plane. The starting point was the mixing matrix resulting from the Independent Component Analysis (ICA) of the time-series recorded from the photo-diode array.

3.1.1 ICA output

The ICA analysis of the 464 recorded diodes gives the N sources (individual neuron time-series) and a weight matrix $\mathbf{W}$ (the matrix inverse of the mixing matrix $\mathbf{M}$ i.e. $\mathbf{W} = \mathbf{M}^{-1}$). The weight matrix has N columns; the jth column of the weight matrix has 464 entries, entry i being the contribution of that diode's time-series to the jth neuron. Thus it is possible to estimate the position of the neuron in two-dimensional space by the relative weightings of each diode location.

The first step is to normalise each row of the weight matrix: the weights in each row are proportional to the amplitude of the original diode's time-series (i.e. small amplitudes would lead to large weights and vice-versa). So each row needs to be on the same scale. We thus normalised each row to the range (min:max) of the diode's time-series.

3.1.2 Estimating (x,y) position

The diodes are arranged in an hexagonal pattern on a 50×50 grid. To recover the (x, y) position of a neuron on this grid, we proceed as follows:

1. Make a weight map $\mathbf{Z}$: assign the weight of each diode's contribution (from the corresponding column of $\mathbf{W}$) to its location on the grid.

2. Threshold the weight map: keep only weights greater than the mean+3 S.D.s of all non-zero weights; all other weights are then set to zero (this removes all very small weights that are simply noise; without this, very wide estimates of a neuron's extent are occasionally obtained).

3. Fit a two-dimensional Gaussian to estimate the neuron's $(\bar{x}, \bar{y})$ position and the variance (σ_x^2, σ_y^2) in that estimate. For the x-position this is (similarly for the y-position):

   (a) let S_x with $x = 1 \ldots 50$ be the sum of each column of $\mathbf{Z}$ (i.e. the total weight of each possible x-location) [for S_y this is the sum of each row of $\mathbf{Z}$]

   (b) then $p(x) = S_x / \sum_{x=1}^{50} S_x$

   (c) estimated x-location is then: $\bar{x} = \sum_{x=1}^{50} x p(x)$

   (d) estimated variance $\sigma_x = \sum_{x=1}^{50} p(x)[x - \bar{x}]^2$

4. Evaluate Gaussian on a 500×500 grid (Figure P2A). This discretisation of the diode array then forms the basis for all further processing.

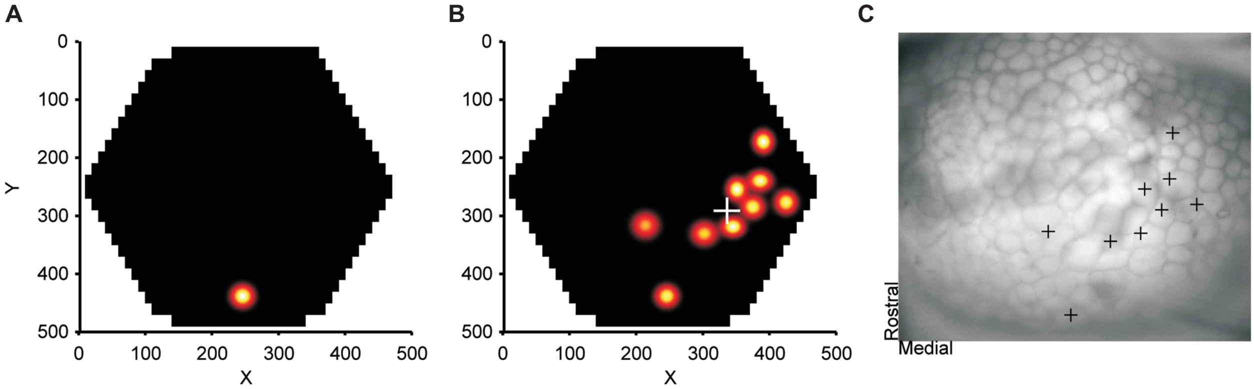

Figure P2: Estimated positions of neurons and ensembles. **A** Probability distribution for the location of a neuron on the two-dimensional recording plane; colour-intensity is proportional to $P(location)$ according to the fitted two-dimensional Gaussian distribution. **B** Probability distribution for location of each neuron in an ensemble on the two-dimensional recording plane. The white cross shows the estimated physical centre of the ensemble. **C** Location of the ensemble in panel B within the pedal ganglion. Crosses plot the $(\bar{x}, \bar{y})$ position of each neuron shown in panel B on an image of the recorded pedal ganglion preparation.

We repeat the above to obtain the estimated $(\bar{x}, \bar{y})$ position of every neuron in the recording.

3.2 Ensemble cohesiveness

The consensus community-detection algorithm returns G ensembles in a recording, each ensemble comprising neurons grouped by similar firing characteristics. For each ensemble, we estimate its cohesiveness in physical space by measuring how far each neuron is from the physical centre of the ensemble. We take the ensemble's centre (x_G, y_G) as the average of the $(\bar{x}, \bar{y})$ positions of each neuron (Figure P2B-C).

For the n neurons in the ensemble, the cohesiveness measure is then: given $d_i =$ distance of neuron i from (x_G, y_G), cohesiveness $C = \text{Median } \forall_i \ (d_i)$.

This measure of cohesiveness (median distance from centre) deals more smoothly with an irregularly-shaped physical distribution of neurons in an ensemble than other tested options (e.g. distance of the neurons in an ensemble from each other).

To interpret the cohesiveness measure we need to test the significance of the cohesion against a null model that the ensembles are not cohesive. We use a null model in which the same number and size of ensembles are maintained, but ensemble membership is permuted by selecting neurons with uniform random probability from across those recorded on the two-dimensional plane. For each ensemble, we generate a P-value for cohesiveness by:

1. An ensemble of the same size is created by randomly selecting n neurons from the data-set

2. A new centre (x^*_G, y^*_G) is computed for the randomly-assigned ensemble

3. From this, a cohesiveness measure C^* is computed for the random ensemble

4. Steps 1-3 are repeated 1000 times to give $C^*_1, C^*_2, \ldots, C^*_{1000}$

5. $P = [\#C_i < C]/1000$

This P-value is then the probability that the ensemble would be that physically cohesive by chance.

3.3 Physical extent of ensembles

We need to estimate the physical extent of each ensemble, in order to assess whether or not ensembles correspond to discrete physical regions that do not overlap.

The problems to solve here are three-fold:

1. *Missing data.* Even if a fully discrete region corresponding to a neural ensemble exists, we will not typically have recorded all neurons in that ensemble (due to a combination of occluded neurons and small-bodied neurons producing smaller signals). Therefore, the distance between two neurons of the same ensemble is not indicative of whether or not the ensemble is physically contiguous.

2. *Irregularity.* The physical shape of an ensemble is likely to not be simple, even if it is a single discrete region. Given the small number of neurons in each ensemble, this makes boundary-drawing a rather unconstrained problem

3. *Noise.* Even if we could record every neuron in a "true" ensemble, there will likely be neurons assigned to wrong ensembles simply due to the size of the search space. Consequently, the physical mapping needs to be robust to noise in the clustering.

We solved these problems using (discretised) Voronoi diagrams. These allow us to identify possible contiguities between neurons spaced far apart, while also respecting the irregular physical shape of the ensembles. Moreover, we can concentrate on the largest single region identified as belonging to one ensemble to circumvent the noise problem.

Starting with the same 500×500 grid above, for each recording we proceeded as follows:

1. Neuron diagram: For each grid square within the diode array, assign that square to the closest neuron (given each neuron's $(\bar{x}, \bar{y})$ position estimate). [Figure P3A]. We solve this by brute force checking each viable grid square, i.e. all on the diode array.

2. Ensemble diagram: for each grid square on the neuron diagram, assign that square the ID number of the ensemble to which that neuron belongs. [Figure P3B]. Again we solve this by brute force checking of each viable grid square (all on the diode array).

3. Find each patch of physical space belonging to each ensemble: a *patch* is defined as a single contiguous region of space. We defined contiguous as any of the 8 surrounding grid squares belonging to the same ensemble. We solve this using a breadth-first search of adjacent squares.

We note that the above algorithm works equally well in three-dimensional tissue, based on a cuboid divided into voxels, and defining contiguous as any of the 26 voxels surrounding the current voxel.

Based on these identified patches, we defined a metric for physical discreteness as "density-per-ensemble" D: the proportion of the ensemble's neurons in the largest patch. Thus, a $D \geq 50\%$ means that at least half that ensemble's neurons are physically contiguous. For each recording, we summarised the discreteness of the map ($\tilde{D}$) by taking the average of D over all ensembles.

While this metric is useful, it is also good to compare it against a null model for no physical discreteness at the level of the whole map. For this null model we again assumed

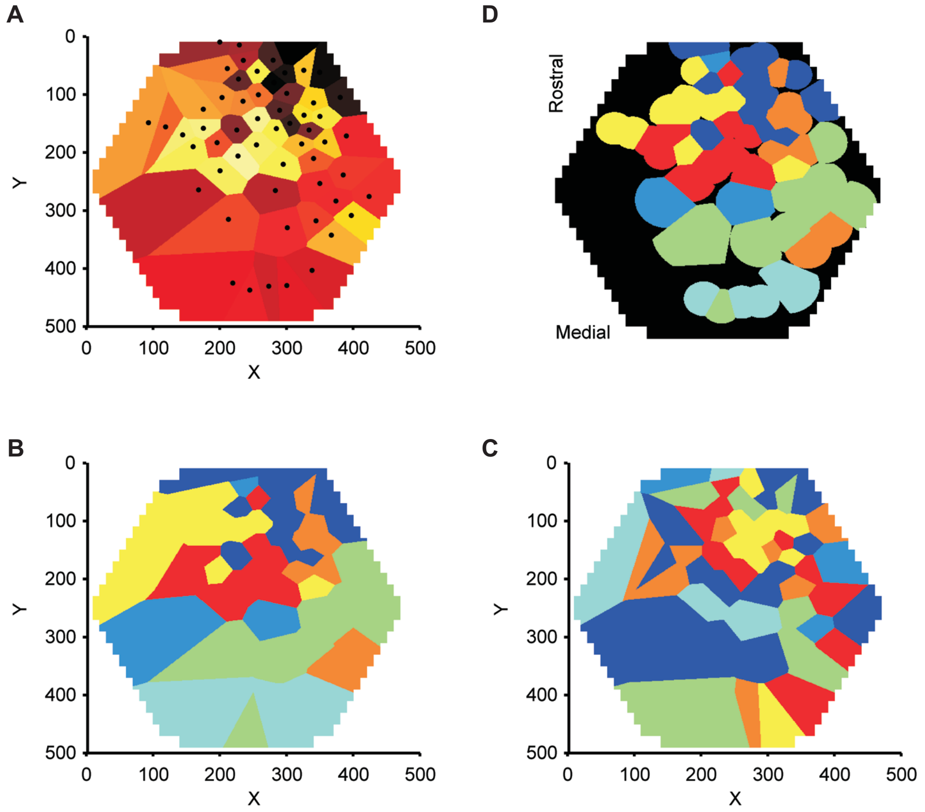

Figure P3: Voronoi diagrams for estimating the physical extent of ensembles. **A** Neuron-level Voronoi diagram. Each patch of the same colour is the grid region assigned to one neuron; the black dots give the estimated $(\bar{x}, \bar{y})$ position of each neuron. **B** Ensemble-level Voronoi diagram. Each *patch* of the same colour is the grid region belonging to neurons in that ensemble. This recording had 7 identified ensembles. All have one dominant patch of space. **C** Example control diagram for the physical distribution of ensembles. Using the same number and size ensembles, each neuron is randomly assigned to one of the ensembles. The resulting physical distribution is more fragmented than the actual ganglion (panel B). **D** Visualisation of the ensemble location maps using sparse Voronoi diagrams. As before, each patch of the same colour is the grid region assigned to the same ensemble, but black indicates areas unclaimed by any ensemble. Compare to panel B.

a uniformly-random assignment of neurons to ensembles of the same size as those detected by the community detection algorithm. This model thus tests for the likelihood that these contiguous patches containing the clustered neurons arose by chance.

Due to the computational bottleneck of the breadth-first search in producing the patch assignments, we cannot do thousands of permutations to produce accurate estimations of P-values. Instead, we simply establish whether or not the metric is above the 95% confidence interval by sampling. The procedure is:

1. Randomly assign neurons to the same size ensembles as in the original clustering of the data-set
2. Construct a randomised ensemble diagram using that random assignment to ensembles (Figure P3C).
3. Find all patches belonging to each randomised ensemble.
4. Compute D^* for each randomised ensemble and $\tilde{D}^* = \text{mean}(D^*)$ for the whole randomised map
5. Repeat steps 1-4 20 times. If the average discreteness $\tilde{D}$ for the recording is greater than maximum value of $\tilde{D}^*$, then the functional ensembles are more discretely organised in physical space than expected by chance at the 95% confidence interval.

In Supplemental Figure 3A, we plot the ensemble-level discreteness in two histograms: the blue histogram plots the distribution of D obtained over all ensembles in the database; the grey histogram plots the distribution of D^* obtained over all ensembles in all the permuted maps of all recordings. In Supplemental Figure 3B, we plot the map-level discreteness: the average $\tilde{D}$ of each data map as a red line, and the distribution of $\tilde{D}^*$ obtained over all the permutations of all maps in a histogram.

3.4 Visualisation and combination of maps

The above provides a general solution to the three problems of missing data, irregularity (of ensembles' physical shape), and noise. However, the plots of the Voronoi diagrams in panels A and B of Figure P3 have the drawback that visually they imply coverage of the entire diode array by the recording, whereas there is an uneven spread of neuron locations in most recordings, often with few neurons recorded in the medial-rostral quadrant ("Sector II" in Hening et al., 1979) of the ganglion, due to their small size. (Note that the claiming of all grid squares by the Voronoi-based algorithm is nonetheless necessary for estimating the physical contiguity of the ensembles). Moreover, when combining each individual recording's map to produce the dynamic maps (Figures 4 and 8 of the main text), the assignment of grid squares far from any neuron may produce results biased in favour of the most common types of neuron or ensemble encountered.

To avoid both this visual illusion and allow for unbiased combination of maps, we can define a sparse type of Voronoi map: we proceed as above, but allow for the existence of unclaimed grid squares on the diode array itself. A square is unclaimed if it is further from its closest neuron than that neuron is from its closest neuron. In this way, areas of the diode array with a low density of recorded cells will have many unclaimed regions. Figure P3D shows that this "sparse" Voronoi map successfully conveys the lack of recorded neurons in the most rostral portions of the ganglion, and successfully avoids colouring to the edge of the diode array in all directions.

3.5 Code

All MATLAB functions for creating and quantifying the Voronoi maps are in the Supplemental Software. Each function has extensive help text within it; here we give an overview of each function's role.

Main functions:

Voronoi_grid Takes a set of points and discretised volume containing them (in any number of dimensions), and constructs the Voronoi diagram on that volume.

find_contiguous_patches Search algorithm for detecting contiguity between Voronoi patches belonging to neurons in the same ensemble.

These functions are used by top-level scripts in the Supplemental Software:

Map_topology_using_Voronoi Computes physical properties of each recording: the cohesiveness of its ensembles, their Voronoi diagrams, and their discreteness.

Supplemental Text. Interpretation of functional modules and physical locations

Here we discuss the further possible mappings of the identified types of ensemble to known neuron types of the pedal ganglion, and to the locomotion motor program. The discussion here is in reference to the four classes of ensemble defined by their oscillations (Figure 4 in the main text), and to the nine groups of ensemble defined by the rate and regularity properties of their spike-trains (Figure 8 in the main text).

4.1 Tonically firing ensembles

One of our most intriguing observations was the existence of tonically firing neural ensembles (the "non-oscillatory" class in Figure 4 and Groups 4 and 9 in Figure 8), groups of neurons with correlated firing yet not correlated with the cycle of oscillations in the motor program.

We consistently found Group 4 and 9 ensembles at the medial edge of the caudal-medial quadrant, near the pedal commissure (sector IB and edge of 3c in map of (Hening et al., 1979)). A population of neurons here likely use serotonin as a neurotransmitter (McPherson and Blankenship, 1992; Marinesco et al., 2004). Some of these serotonergic neurons are oscillatory during fictive locomotion (McPherson and Blankenship, 1992), and thus were likely identified as Group 1 ensembles when recorded (note presence of ensembles in this region of the Group 1 atlas). We thus suggest the possibility that these Group 4 and 9 neurons may be further populations of non-oscillatory serotonergic neurons.

4.2 Phasic Poisson ensembles

We found a unique type of ensemble (Group 7) that was correlated with the cycle of the motor program but each cycle's burst was a Poisson process, consisting of spikes produced at random intervals. These ensembles were consistently located in the rostro-medial quadrant (Sector II of (Hening et al., 1979)) predominantly occupied by motorneurons, yet this pattern of firing has not been previously reported in motorneurons (Hening et al., 1979; Fredman and Jahan-Parwar, 1980). One possibility is that this ensemble is the sub-set of serotonergic neurons with burst-like activity just above the pedal commissure observed by (McPherson and Blankenship, 1992); however, in the closely related *Aplysia brasiliana* these serotonergic neurons have highly regular burst patterns during fictive motion (McPherson and Blankenship, 1991).

A stronger possibility is that Group 7's irregular, Poisson-like firing pattern implies the afferent network to these neurons is in the so-called "balanced state". Originally observed for single-cell *in vivo* recordings in primate visual cortex, the existence of irregular, Poisson-like firing presented a challenge for standard models of how single neurons integrate their input, as such models predict a regularising operation whereby summing excitatory post-synaptic potentials caused by the input of irregular spike trains produces a highly regular output spike-train (Softky and Koch, 1993). The leading explanation for the irregular firing of a single neuron is a network effect in which each neuron receives an approximately balanced input from many asynchronously firing excitatory and inhibitory neurons (van Vreeswijk and Sompolinsky, 1996; Shadlen and Newsome, 1998; Salinas and Sejnowski, 2000; Renart et al., 2010). Far from being restricted to cortex, evidence for such a "balanced" asynchronous input state has been reported for burst-firing in turtle spinal motorneurons during fictive movement (Berg et al., 2007). Consequently, it is possible that our Group 7 ensembles represent a set of neurons driven by a balanced input regime during their burst activity – confirmation of this would represent, to our knowledge, the first evidence that such a balanced regime exists at the ensemble level in an invertebrate network, and thus that the balanced state is a general principle of operation for neural circuits. Equally intriguing is why such a balanced input state exists only for this highly specific group of neurons in the entire recorded portion of the ganglion.

4.3 Unknown types

We have observed a number of types which have no clear correlate with electrophysiological signatures or physical locations of neuron classes reported in the literature, though it is possible that within these ensemble types are included the rare GABAergic (Diaz-Rios et al., 1999) and nitric oxide synthase (NOS) neurons (Moroz, 2006) within the ganglion. These are:

- Group 2: smoothly oscillating ensembles (with a bimodal Gaussian distribution of ISIs), found mainly in the rostro-lateral quadrant.
- Group 3: noisy bursting ensembles, in the centre of ganglion.
- Group 5: ensembles with a unimodal distribution of CV_2, found in caudo-lateral quadrant
- Group 6: sparsely firing single spikes, but unexpectedly modulated by ongoing oscillation (bimodal Gamma distribution of ISIs). Located predominantly lateral of the ganglion's centre.
- Group 8: sparsely firing bursters, rostral of the ganglion's centre. Their bimodal distribution of CV_2 implies regularly spaced spikes within bursts (first mode) with long intervals between bursts (second mode).

References

Bartho, P., Hirase, H., Monconduit, L., Zugaro, M., Harris, K. D. and Buzsaki, G. (2004). Characterization of neocortical principal cells and interneurons by network interactions and extracellular features. J Neurophysiol *92*, 600–608.

Berg, R. W., Alaburda, A. and Hounsgaard, J. (2007). Balanced inhibition and excitation drive spike activity in spinal half-centers. Science *315*, 390–393.

Brandes, U., Delling, D., Gaertler, M., Goerke, R., Hoefer, M., Nikoloski, Z. and Wagner, D. (2006). Maximizing modularity is hard. Technical Report arXiv:physics/0608255v2.

Bullmore, E. and Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. Nat Rev Neurosci *10*, 186–198.

Cauli, B., Porter, J. T., Tsuzuki, K., Lambolez, B., Rossier, J., Quenet, B. and Audinat, E. (2000). Classification of fusiform neocortical interneurons based on unsupervised clustering. Proc Natl Acad Sci U S A *97*, 6144–6149.

Defelipe, J., Lpez-Cruz, P. L., Benavides-Piccione, R., Bielza, C., Larraaga, P., Anderson, S., Burkhalter, A., Cauli, B., Fairn, A., Feldmeyer, D., Fishell, G., Fitzpatrick, D., Freund, T. F., Gonzlez-Burgos, G., Hestrin, S., Hill, S., Hof, P. R., Huang, J., Jones, E. G., Kawaguchi, Y., Kisvrday, Z., Kubota, Y., Lewis, D. A., Marn, O., Markram, H., McBain, C. J., Meyer, H. S., Monyer, H., Nelson, S. B., Rockland, K., Rossier, J., Rubenstein, J. L. R., Rudy, B., Scanziani, M., Shepherd, G. M., Sherwood, C. C., Staiger, J. F., Tams, G., Thomson, A., Wang, Y., Yuste, R. and Ascoli, G. A. (2013). New insights into the classification and nomenclature of cortical GABAergic interneurons. Nat Rev Neurosci *14*, 202–216.

Diaz-Rios, M., Suess, E. and Miller, M. W. (1999). Localization of GABA-like immunoreactivity in the central nervous system of Aplysia californica. J Comp Neurol *413*, 255–270.

Fortunato, S. (2010). Community detection in graphs. Physics Reports *486*, 75–174.

Fredman, S. M. and Jahan-Parwar, B. (1980). Role of pedal ganglia motor neurons in pedal wave generation in Aplysia. Brain Res Bull *5*, 179–193.

Fujisawa, S., Amarasingham, A., Harrison, M. T. and Buzski, G. (2008). Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. Nat Neurosci *11*, 823–833.

Good, B. H., de Montjoye, Y.-A. and Clauset, A. (2010). Performance of modularity maximization in practical contexts. Phys Rev E *81*, 046106.

Hening, W. A., Walters, E. T., Carew, T. J. and Kandel, E. R. (1979). Motorneuronal control of locomotion in Aplysia. Brain Res *179*, 231–253.

Holt, G. R., Softky, W. R., Koch, C. and Douglas, R. J. (1996). Comparison of discharge variability in vitro and in vivo in cat visual cortex neurons. J Neurophysiol *75*, 1806–1814.

Humphries, M. D. (2011). Spike-train communities: finding groups of similar spike trains. J Neurosci *31*, 2321–2336.

Kreuz, T., Haas, J. S., Morelli, A., Abarbanel, H. D. I. and Politi, A. (2007). Measuring spike train synchrony. J Neurosci Methods *165*, 151–161.

Lancichinetti, A. and Fortunato, S. (2012). Consensus clustering in complex networks. Scientific Reports *2*, 336.

Lyttle, D. and Fellous, J.-M. (2011). A new similarity measure for spike trains: Sensitivity to bursts and periods of inhibition. J Neurosci Methods *199*, 296–309.

Marinesco, S., Kolkman, K. E. and Carew, T. J. (2004). Serotonergic modulation in aplysia. I. Distributed serotonergic network persistently activated by sensitizing stimuli. J Neurophysiol *92*, 2468–2486.

McPherson, D. R. and Blankenship, J. E. (1991). Neural control of swimming in Aplysia brasiliana. III. Serotonergic modulatory neurons. J Neurophysiol *66*, 1366–1379.

McPherson, D. R. and Blankenship, J. E. (1992). Neuronal modulation of foot and body-wall contractions in Aplysia californica. J Neurophysiol *67*, 23–28.

Monti, S., Tamayo, P., Mesirov, J. and Golub, T. (2003). Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. Mach. Learn. *52*, 91–118.

Moroz, L. L. (2006). Localization of putative nitrergic neurons in peripheral chemosensory areas and the central nervous system of Aplysia californica. J Comp Neurol *495*, 10–20.

Nadakuditi, R. R. and Newman, M. E. J. (2012). Graph spectra and the detectability of community structure in networks. Phys Rev Lett *108*, 188701.

Newman, M. E. J. (2006a). Finding community structure in networks using the eigenvectors of matrices. Phys Rev E *74*, 036104.

Newman, M. E. J. (2006b). Modularity and community structure in networks. Proc Natl Acad Sci U S A *103*, 8577–8582.

Newman, M. E. J. (2012). Communities, modules and large-scale structure in networks. Nature Physics *8*, 25–31.

Nguyen, N. and Caruana, R. (2007). Consensus Clusterings. In Proceedings of the 2007 Seventh IEEE International Conference on Data Mining ICDM '07 pp. 607–612, IEEE Computer Society, Washington, DC, USA.

Petilla Interneuron Nomenclature Group (2008). Petilla terminology: nomenclature of features of GABAergic interneurons of the cerebral cortex. Nat Rev Neurosci *9*, 557–568.

Ponce-Alvarez, A., Kilavik, B. E. and Riehle, A. (2010). Comparison of local measures of spike time irregularity and relating variability to firing rate in motor cortical neurons. J Comput Neurosci *29*, 351–365.

Renart, A., de la Rocha, J., Bartho, P., Hollender, L., Parga, N., Reyes, A. and Harris, K. D. (2010). The asynchronous state in cortical circuits. Science *327*, 587–590.

Salinas, E. and Sejnowski, T. J. (2000). Impact of correlated synaptic input on output firing rate and variability in simple neuronal models. J Neurosci *20*, 6193–6209.

Shadlen, M. N. and Newsome, W. T. (1998). The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. J Neurosci *18*, 3870–3896.

Softky, W. R. and Koch, C. (1993). The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. J Neurosci *13*, 334–350.

van Vreeswijk, C. and Sompolinsky, H. (1996). Chaos in neuronal networks with balanced excitatory and inhibitory activity. Science *274*, 1724–1726.

Victor, J. D. and Purpura, K. P. (1996). Nature and precision of temporal coding in visual cortex: a metric-space analysis. J Neurophysiol *76*, 1310–1326.

Wasserman, L. (2004). All of Statistics: A Concise Course in Statistical Inference. Springer, New York.

Wohrer, A., Humphries, M. D. and Machens, C. (2013). Population-wide distributions of neural activity during perceptual decision-making. Prog Neurobiol *103*, 156–193.