INTERFACE

royalsocietypublishing.org/journal/rsif

Research

Cite this article: Messenger DA, Wheeler GE, Liu X, Bortz DM. 2022 Learning anisotropic interaction rules from individual trajectories in a heterogeneous cellular population. *J. R. Soc. Interface* **19**: 20220412.
https://doi.org/10.1098/rsif.2022.0412

Received: 31 May 2022
Accepted: 15 September 2022

Subject Category:
Life Sciences–Mathematics interface

Subject Areas:
biomathematics, computational biology, systems biology

Keywords:
cell migration, cell classification, interacting particle system, equation learning, weak-form sparse identification of nonlinear dynamics

Authors for correspondence:
Daniel A. Messenger
e-mail: daniel.messenger@colorado.edu
David M. Bortz
e-mail: david.bortz@colorado.edu

Learning anisotropic interaction rules from individual trajectories in a heterogeneous cellular population

Daniel A. Messenger[1], Graycen E. Wheeler[2], Xuedong Liu[2] and David M. Bortz[1]

[1]Department of Applied Mathematics, and [2]Department of Biochemistry, University of Colorado, Boulder, CO 80309-0526, USA

DMB, 0000-0003-1163-7317

Interacting particle system (IPS) models have proven to be highly successful for describing the spatial movement of organisms. However, it is challenging to infer the interaction rules directly from data. In the field of equation discovery, the weak-form sparse identification of nonlinear dynamics (WSINDy) methodology has been shown to be computationally efficient for identifying the governing equations of complex systems from noisy data. Motivated by the success of IPS models to describe the spatial movement of organisms, we develop WSINDy for the second-order IPS to learn equations for communities of cells. Our approach learns the directional interaction rules for each individual cell that in aggregate govern the dynamics of a heterogeneous population of migrating cells. To sort a cell according to the active classes present in its model, we also develop a novel ad hoc classification scheme (which accounts for the fact that some cells do not have enough evidence to accurately infer a model). Aggregated models are then constructed hierarchically to simultaneously identify different species of cells present in the population and determine best-fit models for each species. We demonstrate the efficiency and proficiency of the method on several test scenarios, motivated by common cell migration experiments.

1. Introduction

Systems of autonomous agents are ubiquitous in the natural world. Research into their behaviour has led to a plethora of proposed mathematical models, including the agent-based 'boids' model [1], ordinary differential equation models for milling and flocking [2,3], and non-local partial differential equations [4,5], to name a few. A general framework for rigorous analysis of these models is by now very mature [6].

Identifying the rules of interaction between agents is necessary for predicting and influencing the cooperative abilities of any such system, whether composed of autonomous robots, large multi-cellular animals, single-celled organisms or even molecules. Methods for inferring the rules of interaction between agents using observed trajectory data have continued to advance since the early 2000s. Several of the principled techniques include force-matching [7,8], linear regression [9,10], mean-field formulations [11,12], information-theoretic tools [13], underdamped Langevin regression [14,15], Gaussian processes [16] and even a method based on topological rather than metric distances [17].

These and related techniques have been successfully used to identify the dominant drivers of collective behaviour in a variety of social and biological systems [18–20], including schools of fish [21,22], flocks of birds [23,24] and pedestrian traffic [25], all directly incorporating measured trajectory data. While popular methods, such as force-matching, are useful in identifying

THE ROYAL SOCIETY PUBLISHING

fields of vision and spatial statistics of interactions, they cannot easily disentangle the combined effects of multiple forces (e.g. attraction, repulsion and alignment) [26,27], let alone different interactions between multiple species of neighbours. This limits the classes of models they can identify and implies that new methods must be developed for heterogeneous populations.

The field of *equation discovery* is a highly active area of research [28–36], as it offers tools to directly learn governing differential equations. This approach is not only useful in prediction and validation, but can be used to simultaneously identify multiple active modes of inter-agent communication, such as repulsion, velocity alignment and attraction. In this work, we tackle the problem of identifying governing equations for an interacting particle system (IPS) with multiple interacting species. Our proposed approach is completely naive with regard to species membership in order to specifically address problems of heterogeneity in collective cell migration [37]. In accordance with our biological motivations, we refer to agents throughout as 'cells', particle systems as 'populations', and different cell types as 'species', however cell types need not correspond to 'species' in the biological sense (e.g. 'leader' and 'follower' cells could be classified as two different 'species').

Motivated by existing hypotheses regarding the anistropy of cell–cell interactions [38–40], we introduce our framework in the context of *directional interaction* models, as defined below. Moreover, we note that the documented significance of anisotropic interactions in general collective systems [41–43] suggests that our approach may have wide applicability.

1.1. Heterogeneous populations

Many collective populations arising in nature are inherently heterogeneous, with the rules of interaction varying across different subsets of the population. This is readily observable in complex mammalian populations, but is also seen in simpler organisms, such as honeybee swarms, where bees divide into scout and worker bee roles [44]. The advantages of heterogeneity in collective behaviour have even inspired search optimization algorithms [45,46].

At the level of microorganisms, cells have been observed to adopt leader-like and follower-like roles during collective migration events such as wound healing [47,48], without the aid of a central nervous system. Individual cell speed and persistence of motion have also been determined to be functions of the age and size of the cell [49–51], which may lead to heterogeneous responses to stimuli from neighbouring cells. The mechanisms which produce these heterogeneities, and the extent to which heterogeneity is present in a given cell population, are current subjects of debate [52–54]. Data-driven techniques may be useful in formulating accurate mathematical models in the presence of heterogeneity.

Zhong *et al.* [55] develop a highly versatile method for inferring explicit rules of interaction in a heterogeneous population, although it is assumed that species membership is known *a priori*. Several recent works have offered methods of assessing the degree of population heterogeneity [18,53], yet these methods do not provide explicit mathematical models for the different populations. By contrast, the method presented here allows one to classify the given population into different species according to the heterogeneous interaction rules present and produces explicit mathematical models for each species as a by-product.

In this work, we restrict our attention to the case where individuals within the population may follow different interaction rules, but each individual applies only one set of interaction rules to all others members of the population. In other words, individual i applies the same set of rules to individual j and k, while j and k may each apply different interaction rules to particle i. We leave the case of individual i interacting differently with individuals j and k, depending on the species membership of j and k, to future work.

1.2. Directional interaction forces

It is now well known that simple radial interaction models are incapable of explaining many observed collective behaviours in biological settings, and that directionally dependent interaction rules, based on a limited field of view or sensing angle, offer a significant advantage [23,41,56–58]. At the cellular level, directional dependence of cell–cell interaction has been proposed in the context of intracellular polarization [39]; however, the cellular sensing range is not immediately obvious, since a migrating cell does not have an obvious 'field of view'. Recent works have sought to quantify the degree to which interactions are density-dependent [59], but not which directional modes (radial, dipolar, quadrupolar, etc.) are dominant during a collective migration event.

In addition to providing an explanation for certain observed phenomena [43], directional interaction rules are capable of generating *spontaneous migration*, due to the total directional force between particles not being conserved in general. In the modelling of active matter systems (such as migrating cells) [11,60], such symmetry breaking is commonly generated by a combination of Brownian forcing and a self-propulsion device [61]. However, it is not clear that self-propulsion is an appropriate mechanism for modelling cellular movement (in comparison with fish, which are constantly swimming). Directional forces may then be an important mechanism for symmetry breaking and spontaneous cellular migration.

1.3. Weak-form sparse identification of nonlinear dynamics

At its core, our method involves learning ordinary differential equations for cells using available trajectory data. For this we employ the weak-form sparse identification of nonlinear dynamics algorithm (WSINDy), which has been shown to successfully identify governing equations from data at the levels of ordinary different equations [62], partial differential equations [63], first-order interacting particle systems [12] and even works in a small-memory online streaming scenario [64].

A significant advantage of the WSINDy method is that it identifies a single governing equation which can be interpreted, analysed and simulated using conventional techniques of applied mathematics. It does not involve any black-box algorithms or mappings as would be generated in using a neural network-based approach. Another promising direction is a hybrid approach, such as [65], where the authors first learn a neural network model of the potential and then use sparse identification to learn the algebraic form of the potential. Ultimately, an interpretable sparse

model provides the best chance at both describing and modelling the dynamics.

Several alternative methods have been developed to accomplish the equation learning task for particle systems. In particular, Lu *et al.* [10] develop a method for learning general feature-dependent second-order interaction rules for heterogeneous populations, where features may include directional interaction forces, speed dependence and so on. The differences between this and our work are the following. (i) We are performing the *unsupervised* learning task of classifying agents by their interaction rules, whereas Lu *et al.*'s work assumes knowledge of the species membership. (ii) We are interested in *sparse model representations*, in particular selection of the correct modes of interaction (e.g. attractive, repulsive, alignment and drag force), whereas Lu's work assumes knowledge of both the feature-dependence and types of forces present (e.g. for planetary systems, *a priori* knowledge is used to rule out the presence of an alignment force). (iii) Lastly, models are initially extracted from *single-cell trajectories*. As described in the next section, rather than aggregating *data* which may come from multiple cell species, we aggregate *models* which are likely to describe the same species, and then use the aggregate model to perform classification.

1.4. Single-cell learning and model clustering

With a possibly heterogeneous population of cell trajectories available, one is tasked with the problem of deciding how to aggregate the data. If knowledge of the underlying species membership is available, a more accurate model can be inferred by pooling data from all individuals of a given species. On the other hand, pooling data from multiple species into a single model can result in a highly *inaccurate* model if very different interaction rules from multiple species are averaged together. In general, there exists a spectrum of possible pooling strategies, ranging from learning *few models* from *large subsets* of the population, to learning *many models* from *small subsets* of the populations. The former intrinsically produces models with high bias and low variance, while the latter produces models with low bias and high variance. Such pooling strategies have been recently explored in [66], where it is found that identifying a single model can be improved by pooling models learned from subsets of the data. However, this has not been extended to classifying the data itself into species, and finding a model for each species. Moreover, the IPS setting offers a particular advantage on the subject of model validation, as data can easily be assimilated into forward simulations.

In this work, we investigate the extreme case of learning an individual model $\mathcal{M}_i$ for the ith individual trajectory, and then clustering the set of learned models $\mathcal{M} := \{\mathcal{M}_1, \ldots, \mathcal{M}_N\}$ according to their identified modes of interaction. This approach is counterintuitive because there is no guarantee that a single-cell trajectory will provide enough information on the interaction rules of its species. To be able to classify cells using the (potentially) insufficiently informative trajectories, we developed an ad hoc recursive classifier which we show (in §4) accurately clusters and sorts the models into species. This approach prevents any contamination that may result from combining trajectories of multiple species.

Once the models are clustered, an aggregate model $\overline{\mathcal{M}}$ is computed by averaging the models in $\mathcal{M}$ belonging to the most populous cluster. The model $\overline{\mathcal{M}}$ is then used to classify cells via forward simulations which are made highly efficient by directly incorporating the data. In particular, for each trajectory in the dataset, we use $\overline{\mathcal{M}}$ to simulate a new trajectory, but with all neighbour interactions computed using the data. That is, only the new trajectory is propagated forward in time by model $\overline{\mathcal{M}}$, while the rest of the population is simply the data itself. This can then be trivially parallelized, reducing an $\mathcal{O}(N^2)$ computational cost per time step to N cores performing $\mathcal{O}(N)$ updates per time step with no communication overhead.

We show through examples below that this hierarchical model-pooling and validation procedure produces both correct species classification and accurate governing equations, despite individual cell trajectory data carrying low levels of information. For further information on the classification algorithm, see §3.

1.5. Paper outline

In §2, we discuss the general form of directional interacting particle models that will be assumed in the learning process. In §3, we introduce our model selection and classification algorithm, which is composed of the six steps: (a) learn single-cell models, (b) replace inaccurate models, (c) cluster learned models according to active force modes, (d) form an aggregate model by averaging models in the largest cluster, (e) validate the aggregate model using data-driven forward simulations, and (f) classify cells according to performance under the aggregate model. In §4, we examine the performance of the algorithm in learning and classifying homogeneous and heterogeneous populations of one, two and three species. We discuss possible next directions in §5. Some additional information and a summary of notation are included in appendix A.

2. Directional interacting particle models

We use a general second-order directional interaction model framework, where the position and velocity $(x_i, v_i) \in \mathbb{R}^{2d}$ of cell i in d spatial dimensions are governed by the differential equations

$$\begin{aligned}\ddot{x}_i = &\frac{1}{N_{\text{tot}}}\sum_{j=1}^{N_{\text{tot}}} f_{\text{a-r}}(|x_i - x_j|, \theta_{ij})(x_i - x_j)\\ &+\frac{1}{N_{\text{tot}}}\sum_{j=1}^{N_{\text{tot}}} f_{\text{align}}(|x_i - x_j|, \theta_{ij})(v_i - v_j)\\ &+\frac{1}{N_{\text{tot}}}\sum_{j=1}^{N_{\text{tot}}} f_{\text{drag}}(|v_i|, \theta_{ij})v_i. \end{aligned} \tag{2.1}$$

Here, θ_{ij} is the angle between v_i and $x_j - x_i$ (see the diagram in figure 1). The attractive–repulsive force $f_{\text{a-r}}$, alignment force f_{align} and the drag force f_{drag} define the rules by which cell i communicates with the rest of the population. Our primary objective is to identify a set of interaction rules $\{(f_{\text{a-r}}, f_{\text{align}}, f_{\text{drag}})_\ell\}_{1\le\ell\le S}$, one for each of the S species present in the population. We note that additionally the model (2.1) can contain a stochastic noise term to capture random environmental forces; however, we leave this an extension to future work.

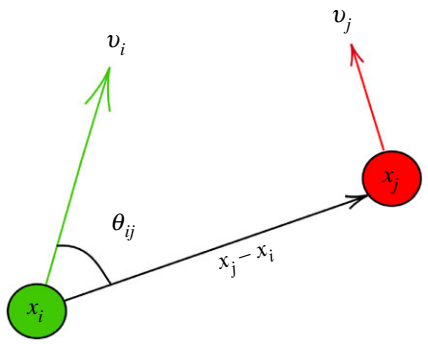


Figure 1. Diagram of social interactions depending on angle θ_{ij} between cell *i*'s velocity and cell *j*'s position relative to *i*.

2.1. Directionality θ_{ij}

As mentioned above, directional variation in the interaction forces between cells can arise from various factors, including intracellular polarization, or heterogeneous distribution of membrane-bound receptors, asymmetry in the protrusion/contraction of lamellopodia as the cell crawls on the substrate, and so on. In the current study, we assume that each of these effects is unobservable, hence we model the aggregate directional effect using the angles θ_{ij}, depicted in figure 1. Dependence on angle θ_{ij} allows for interactions between cell i and cell j to vary depending on the direction of motion. Put another way, in the reference frame of cell i, the polar coordinates $r_{ij} = |x_i - x_j|$ and θ_{ij} allow one to represent any interaction force that varies over the two-dimensional plane.

It should be noted that asymmetric interactions $\theta_{ij} \neq \theta_{ji}$ lead to symmetry breaking and spontaneous cell migration from an initially motionless state. In this study we restrict the angular dependence to $\{1, \cos(\theta_{ij}), \cos(2\theta_{ij})\}$, which allows for a combination of radial, dipolar and quadrupolar interactions (see figure 2 for examples of dipolar (*b*) and quadrupolar (*a*) forces used in this study). Higher-order directionality can usually be assumed to be negligible; however, extension to higher modes is straightforward.

2.2. Attractive–repulsive force f_{a-r}

The interaction force f_{a-r} acts along the vector from cell i to cell j and captures short-range repulsion and long-range signalling. Many IPS models include only an attractive–repulsive force, due to its extensive pattern-forming capabilities [67,68]. Typically f_{a-r} is taken to be the gradient of some potential K, such as the Morse potential $K(r) = C_R e^{-r/L_R} - C_A e^{-r/L_A}$ or power law potential $K(r) = r^{p_R}/p_R - r^{p_A}/p_A$, with r denoting the interparticle distance. The parameters (C_R, L_R, C_A, L_A) or (p_A, p_R) determine the possible long-time behaviours, such as milling, spreading or concentrating.

We impose the following natural constraints on f_{a-r}:

$$\begin{cases} f_{a-r}(r, \theta) \geq 0, & 0 \leq r < r_{nf} \\ f_{a-r}(r, \theta) \leq 0, & r \geq r_{ff} \\ f_{a-r}(r, \theta) \in \text{span}\{1, \cos(\theta), \cos(2\theta)\}, & \text{every } r \text{ fixed.} \end{cases} \tag{2.2}$$

Here r_{nf} is the *near-field threshold*, which can for instance correspond to a cell diameter, and r_{ff} is the *far-field threshold*, i.e. a large distance. The first inequality enforces that f_{a-r} is near-field repulsive, which must be true by volume exclusion. In practice, we define r_{nf} by

$$\mathbb{P}(|x_i - x_j| < r_{nf}) = p_{nf},$$

where in this work we set $p_{nf} = 0.001$, and the dataset is used to compute the probability, taking all interparticle distances over all time points into account. This states that the force must be repulsive over short pairwise distances which are 0.1% likely to be observed, given dataset.

The second equality enforces long-range decay, as well as model stability. Decay is natural since interactions can be expected to be small outside of some large distance r_{ff}. We enforce that interactions are *attractive* at large distances (allowing for decay as well), so that in simulation the particles do not spread to infinity. We set $r_{ff} = 1$ throughout, although r_{ff} can easily be chosen from the data (e.g. $r_{ff} = 50 r_{nf}$ corresponds to an effective interaction range of 50 cell radii). (See appendix A.1 for resulting values of r_{nf} and r_{ff} and other hyperparameters for examples below.)

The third set inclusion simply reiterates the assumptions on directionality described above.

2.3. Alignment force f_{align}

The alignment force f_{align} captures cells' tendency to match the velocity of neighbouring cells. There are many theories as to how this arises physically [38,69]. Perhaps protrusions from cells inform the cell about the bulk direction of motion, which would be a very local effect. However, alignment models which have been proven to lead to flocking depend on sufficiently *long-range* alignment. In particular, the Cucker–Smale model involves only an alignment force, which takes the form $f_{align}(r, \theta) = A/(\sigma^2 + r^2)^\beta$. Unconditional flocking occurs for $\beta < 1/2$, and for larger β (leading to a shorter-range alignment force) flocking depends on the initial conditions [3].

We impose the following constraints on f_{align}:

$$\begin{cases} f_{align}(r, \theta) \leq 0, & 0 \leq r \\ f_{align}(r, \theta) \in \text{span}\{1, \cos(\theta), \cos(2\theta)\}, & \text{every } r \text{ fixed.} \end{cases} \tag{2.3}$$

The first inequality enforces that f_{align} is non-positive, which is necessary for the constant velocity state $v_i = v_j = v$ to be a stable configuration. If not, small perturbations away from $v_i = v_j$ result in cells *accelerating* away from each other, which is a redundant force given that cells can be pushed away from each other through f_{a-r} (it is also not hard to see that $f_{align} > 0$ is unphysical). The second constraint restricts the alignment force to be a combination of radial, dipolar or quadrupolar modes, similar to f_{a-r}.

2.4. Drag force f_{drag}

The drag force f_{drag} captures energy expenditure due to general resistance to motion (resulting e.g. from substrate roughness); however, we allow an angular dependence on θ_{ij} to capture possible decreases or increases in drag depending on local neighbour distribution. For this we impose the following constraints:

$$\begin{cases} f_{drag}(s, \theta) \leq 0, & 0 \leq s < \infty \\ f_{drag}(s, \theta) \in \text{span}\{1, \cos(\theta)\}, & \text{every } s \text{ fixed,} \end{cases} \tag{2.4}$$

where s indicates the speed of the cell. The force f_{drag} is

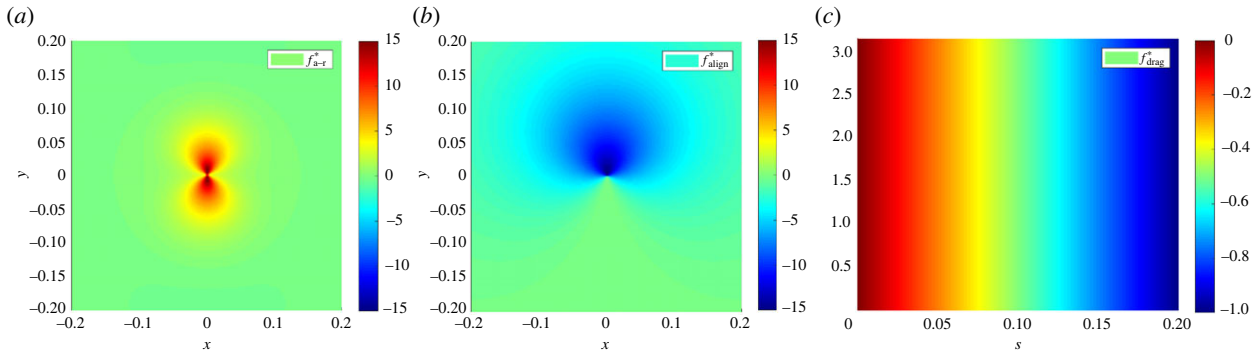


Figure 2. Forces used to generate artificial data, motivated by experiment. (*a*) Quadrupolar attractive-repulsive force $f^{\star}_{\mathrm{a-r}}$. (*b*) Dipolar alignment force $f^{\star}_{\mathrm{align}}$. (*c*) Linear isotropic drag force $f^{\star}_{\mathrm{drag}}$.

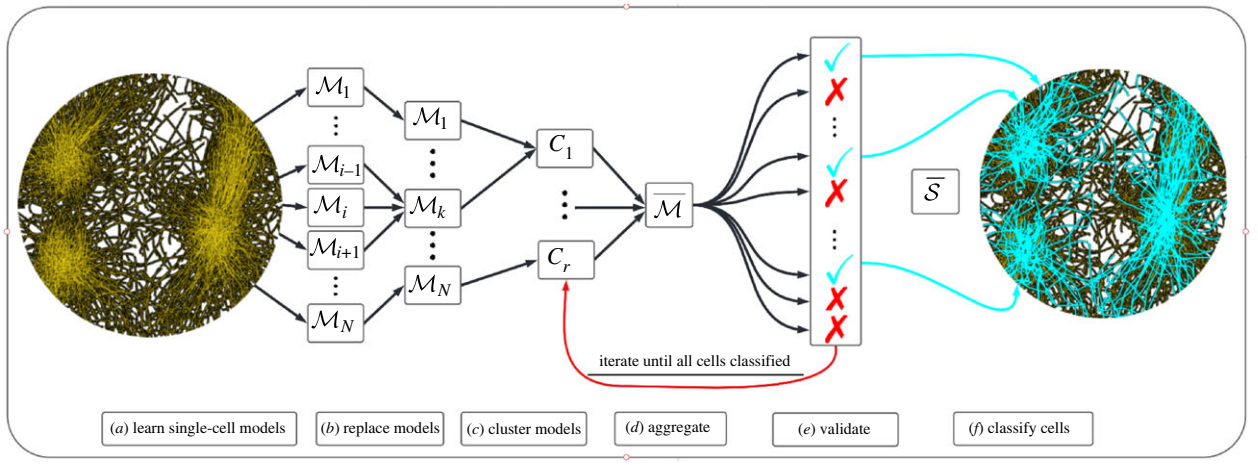


Figure 3. Classification pipeline for cells from heterogeneous populations. (*a*) An ensemble of models $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_N\}$ is learned, each from an individual trajectory; (*b*) models in $\mathcal{M}$ are replaced by neighbouring models with superior performance if any exist; (*c*) $\mathcal{M}$ is partitioned into clusters $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_r\}$ according to active forces in each model; (*d*) models in the largest cluster $\overline{\mathcal{C}}$ are averaged together, giving $\bar{\mathcal{M}}$; (*e*) $\bar{\mathcal{M}}$ is validated along each individual trajectory; (*f*) validation errors are classified, producing an identified species $\overline{\mathcal{S}}$ (cyan checkmarks) and the remaining cells (red X's) are returned to step (*c*) to be clustered again. Steps (*c–f*) repeat until all cells are classified. Note that the number and members of model clusters $\mathcal{C}$ and resulting aggregate model $\bar{\mathcal{M}}$ will change each iteration depending on the identity of remaining unlabelled cells.

chosen to be negative so that cells do not have a 'self-propulsion' device. As mentioned previously, many models of active matter include self-propulsion as a partial mechanism for symmetry breaking and general non-equilibrium effects. To reiterate, we do not expect cells to have a self-propulsion device, in fact, we wish to learn how migration occurs spontaneously, incited by pairwise interactions. In addition, a positive drag force leads to populations spreading outside of the range of meaningful interactions. In this way, negative drag is computationally beneficial, as it leads to improved model stability.

3. Algorithm

Our algorithm involves first learning an ensemble of directional force models $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_N\}$, that is, one model for each of the N *focal cells* selected for learning. Individual models $\mathcal{M}_i$ are then validated on a small neighbourhood of cells, and $\mathcal{M}_i$ is replaced by $\mathcal{M}_j$ if a model $\mathcal{M}_j$ is found that outperforms $\mathcal{M}_i$ on cell i. We next group models into clusters $\mathcal{C} := \{\mathcal{C}_1, \ldots, \mathcal{C}_r\}$ and compute an aggregate model $\bar{\mathcal{M}}$ from the largest cluster, denoted by $\overline{\mathcal{C}}$. A new species $\overline{\mathcal{S}}$ is then identified, with membership in $\overline{\mathcal{S}}$ determined by the accuracy of data-driven forward simulations of model $\bar{\mathcal{M}}$. Cells in the species $\overline{\mathcal{S}}$ are then removed from the population and the remaining cells are returned to the clustering phase. More explicitly, the algorithm is composed of the following steps, which are visualized in figure 3.

(a) **Identify** individual cell models $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_N\}$ using the WSINDy algorithm.
(b) **Replace** models in $\mathcal{M}$ with superior models of 'neighbouring' cells (as described in §3.2).
(c) **Cluster** $\mathcal{M}$ into $\{\mathcal{C}_1, \ldots, \mathcal{C}_r\}$ according to active force modes.
(d) **Aggregate** models in the largest cluster $\overline{\mathcal{C}}$ to arrive at a single model $\bar{\mathcal{M}}$.
(e) **Validate** model $\bar{\mathcal{M}}$ on each remaining unlabelled cell, using the data to calculate neighbour interactions.
(f) **Classify** cells based on simulation error under $\bar{\mathcal{M}}$ and label the lowest-error class as the new species $\overline{\mathcal{S}}$ (remove cells in $\overline{\mathcal{S}}$ from the population and return to step (c)).

The result is a set of S models and species $\{(\overline{\mathcal{M}}_\ell, \overline{\mathcal{S}}_\ell)\}_{\ell=1}^{S}$, where each model $\overline{\mathcal{M}}_\ell$ is constructed from an average of individual models within a cluster. We use the notation[1] of $\mathcal{M}_i$ to be the model for the ith cell, $\mathcal{C}_j$ to be the jth cluster of models

and $\bar{\mathcal{S}}_\ell$ to be the set of cells identified as the ℓth species and which obey model $\bar{\mathcal{M}}_\ell$.

We now give more detail on steps (a)–(f), including stopping criteria, leaving the more technical aspects to the appendix. In appendix A.1, we include a table of notations used throughout (table 5), along with algorithmic hyperparameters and their corresponding values used in the examples below, followed by a brief discussion about the problem-dependent nature of several hyperparameters.

3.1. Learning single-cell models

The first step in the algorithm is to learn an ensemble of *single-cell* models, one for each of the N focal cells selected from the N_{tot} total cells tracked during the experiment.[2] By 'single-cell' model, we mean that the scope of each model is limited to learning only the dynamics of its focal cell; however, data from the remaining tracked cells are incorporated to learn the interaction forces.

3.1.1. Weak-form sparse identification of nonlinear dynamics

The main ingredient in learning single-cell models in $\mathcal{M}$ is the WSINDy algorithm, together with careful choices for the bases used to represent the three main forces $f_{\text{a-r}}$, f_{align}, and f_{drag}. Each cell i is identified by a position and velocity $(x_i(t),v_i(t))$ which we assume is well-approximated by a second-order model of the form (2.1). The dynamics take the general form

$$\ddot{x}_i(t) = F_i(X(t), V(t)), \tag{3.1}$$

where $(X(t), V(t)) \in \mathbb{R}^{2dN_{\text{tot}}}$ denotes the entire population of positions and velocities in the colony at time t. We then assume that we have available a dataset of positions $\mathbf{X} = (\mathbf{x}_1(t_k), \ldots, \mathbf{x}_{N_{\text{tot}}}(t_k))_{k=1}^{L}$ sampled from the system X at L time points. Our goal is to identify F_i using $\mathbf{X}$.

The SINDy approach involves representing F_i as a sparse linear combination of basis elements $\Theta(X, V) := (f_j(X, V))_{1\le j\le J}$, such that

$$F_{i,d'}(X, V) = \sum_{j=1}^{J} \mathbf{w}^{\star}{}_{i,j} f_{j,d'}(X, V),$$

where subscript d' indicates the spatial coordinate ($d' \in \{1, 2\}$ in this study). The basis Θ is chosen by the user and determines the accuracy of the learned model as well as the conditioning of the WSINDy algorithm.

The available cell position data $\mathbf{X}$ is used to approximate velocities $\mathbf{V} := \dot{\mathbf{X}} \approx \dot{X}$ and accelerations $\ddot{\mathbf{X}} \approx \ddot{X}$, using e.g. finite differences, leading from (3.1) to the data-driven linear system

$$\ddot{\mathbf{x}}_i \approx \Theta(\mathbf{X}, \mathbf{V})\mathbf{w}^{\star}{}_i. \tag{3.2}$$

With some abuse of notation, we denote by $\Theta(\mathbf{X}, \mathbf{V})$ the matrix that results from evaluating the basis $\Theta(X, V)$ at the time-series data $(\mathbf{X}, \mathbf{V})$. The entries are $\Theta(\mathbf{X}, \mathbf{V})_{k+(d'-1),j} = f_{j,d'}(\mathbf{X}(t_k), \mathbf{V}(t_k))$.

The data $\mathbf{X}$ are often corrupted by measurement noise, which leads to inaccurate computations of derivatives $\mathbf{V}$. For the current setting, the standard SINDy approach just outlined requires *second-order* derivatives $\ddot{\mathbf{X}}$, which are even less accurate to compute from noisy data. To prevent some of the corruption from noise,[3] we can use the weak form, which leads to WSINDy. Returning to equation (3.1), we convert to the weak form by multiplying by a *test function* $\phi(t)$ and integrating in time,

$$\langle \phi, \ddot{x}_i \rangle := \langle \phi, F_i(X, V) \rangle. \tag{3.3}$$

where the inner product $\langle \cdot, \cdot \rangle$ denotes the time integral

$$\langle f, g \rangle = \int_{-\infty}^{\infty} f(t) g(t)\, \mathrm{d}t.$$

Choosing ϕ to be twice differentiable and zero outside of some interval (a, b), we then integrate by parts twice on the left-hand side to arrive at

$$\langle \ddot{\phi}, x_i \rangle = \langle \phi, F_i(X, V) \rangle,$$

so that the second derivative has been removed from x_i and placed on ϕ. Choosing a basis of test functions $\Phi := (\phi_q)_{1\le q\le Q}$, we build the weak-form linear system

$$\mathbf{b}^{(i)} \approx \mathbf{G}^{(i)} \widehat{\mathbf{w}}^{(i)}, \tag{3.4}$$

where $\mathbf{b}^{(i)}_{q+(d'-1)} = \langle \ddot{\phi}_q, \mathbf{x}_{i,d'} \rangle$ and $\mathbf{G}^{(i)}_{q+(d'-1),j} = \left\langle \phi_q, f_{j,d'}(\mathbf{X}, \mathbf{V}) \right\rangle$.

As well as choosing Θ, in order to compute $(\mathbf{G}^{(i)}, \mathbf{b}^{(i)})$, we need to compute[4] $\mathbf{V}$ from the position data $\mathbf{X}$, choose a test function basis Φ and discretize integrals appearing in the linear system. For simplicity, we compute $\mathbf{V}$ using second-order centred finite difference, although a number of methods exist for numerical differentiation from data [70,71]. For integration, we use the trapezoidal rule, and we use test functions of the form

$$\phi_q(t) = \max\left(1 - \left(\frac{t - t_q}{m\Delta t}\right)^2, 0\right)^p, \tag{3.5}$$

for shape parameters m and p, and timestamps t_q in the range of the available time series. We use the class of test functions (3.5) for its desirable accuracy and robustness properties combined with the trapezoidal rule [62], and refer to [62,63] for methods of choosing (m, p, t_q). In this work, we use the changepoint algorithm in [63] with $\tau = 10^{-10}$ and $\widehat{\tau} = 3$, leading to $m \in \{31, \ldots, 38\}$ and $p \in \{8, 9\}$ (table 6 lists these values used in the examples below). Since the time series below are relatively short ($L = 200$ or $L = 400$), we use all available t_q, i.e. $(t_q)_{q=1}^{Q} = (m\Delta t, \ldots, (L - m - 1)\Delta t)$ so that $Q = L - 2m$.

3.1.2. Trial basis functions

In the case of the directional force model (2.1), we require three bases $\mathcal{F}_{\text{a-r}}$, $\mathcal{F}_{\text{align}}$ and $\mathcal{F}_{\text{drag}}$ for the three proposed forces $f_{\text{a-r}}$, f_{align} and f_{drag}. We seek a sparse model, and so choose global basis functions, rather than a model composed of a large sum over basis functions that are spatially localized.

For the attractive–repulsive basis $\mathcal{F}_{\text{a-r}}$ we choose products of cosines and scaled and weighted Laguerre polynomials,

$$\mathcal{F}_{\text{a-r}} = \{\cos(n\theta) p_\ell(\alpha r)\, \mathrm{e}^{-(\alpha/2)r}\}_{n=0,\ell=0}^{2,17} \tag{3.6}$$

for ℓth degree Laguerre polynomial p_ℓ. The scale α is chosen from $r_{\max}$, the maximum observed distance between cells, such that $\mathrm{e}^{-(\alpha/2)r_{\max}} = \epsilon_{\text{mach}} \approx \mathrm{e}^{-36}$. We set $\alpha = 36$ in all cases below since $r_{\max} \approx 2$.

The pattern of attractive and repulsive regions of the force $f_{\text{a-r}}$ is not known *a priori*, hence the Laguerre basis offers flexibility. The choice of weighted Laguerre

polynomials (with weight $\omega(r) = e^{-r/2}$) is guided by the orthogonality relation

$$\int_0^\infty p_m(r)p_n(r)\omega^2(r)\,dr = \delta_{mn}, \tag{3.7}$$

where δ_{mn} is the Kronecker delta. We find $\mathcal{F}_{a-r}$ leads to a well-conditioned matrix $\mathbf{G}$ despite orthogonality not holding with respect to the data distribution. We use the first 18 such weighted Laguerre polynomials to provide a sufficiently large basis; however, this number is fairly arbitrary and may need to be increased or decreased depending on the complexity of the dynamics.

For the alignment force, we choose a basis of shifted cosines and exponential functions

$$\mathcal{F}_{\text{align}} = \{(1+\cos(n\theta))\,e^{-2^\ell r}\}_{n=0,\ell=-2}^{2,5} \tag{3.8}$$

which is informed by the fact that f_{align} must be negative. This is easily controlled with $\mathcal{F}_{\text{align}}$ by simply enforcing that the coefficients $\widehat{\mathbf{w}}_{\text{align}}$ be negative. For the same reason, we choose the drag force from a basis of monomials and cosines,

$$\mathcal{F}_{\text{drag}} = \{(1+\cos(n\theta))|v|^\ell\}_{n=0,\ell=0}^{1,4} \tag{3.9}$$

as this can also be easily controlled to yield an overall negative f_{drag} force by constraining only the basis elements $\widehat{\mathbf{w}}_{\text{drag}}$. Moreover, monomials capture the physical assumption that resistance to motion should increase with speed.

3.1.3. Regression

We solve the linear system (3.4) for coefficients $\widehat{\mathbf{w}}^{(i)}$ by approximately solving the following constrained sparse regression problem:

$$\widehat{\mathbf{w}}^{(i)} = \underset{\mathbf{w}\text{ s.t. }\mathbf{C}\mathbf{w}\le\mathbf{d}}{\arg\min}\ \{\|\mathbf{G}^{(i)}\mathbf{w}-\mathbf{b}^{(i)}\|_2^2 + \lambda^2\|\mathbf{w}\|_0\}. \tag{3.10}$$

The linear inequality constraint $\mathbf{C}\mathbf{w}\le\mathbf{d}$ encodes the constraints listed in (2.2), (2.3) and (2.4) on the forces on f_{a-r}, f_{align} and f_{drag}, and λ is the sparsity threshold. We employ the modified sequential thresholding algorithm from [12,63], with least-squares iterations replaced by solving the associated linearly constrained quadratic program.[5]

Since the coefficients $\widehat{\mathbf{w}}^{(i)}$ have no *a priori* absolute magnitude, we threshold only on the magnitudes of the given term relative to the response vector $\mathbf{b}^{(i)}$, namely, we define the thresholding operator $H_\lambda(\mathbf{w})$ by

$$(H_\lambda(\mathbf{w}))_j = \begin{cases} 0, & \dfrac{\|\mathbf{G}_j^{(i)}(\mathbf{w})_j\|}{\|\mathbf{b}^{(i)}\|} \notin [\lambda,\ \lambda^{-1}] \\ (\mathbf{w})_j, & \text{otherwise.} \end{cases} \tag{3.11}$$

The sequential thresholding algorithm for solving (3.10) thus produces iterates $\{\mathbf{w}_0^{(i)}, \ldots, \mathbf{w}_\ell^{(i)}, \ldots, \widehat{\mathbf{w}}^{(i)}\}$ where each $\mathbf{w}_{\ell+1}^{(i)}$ is obtained from $\mathbf{w}_\ell^{(i)}$ by first solving (3.10) with $\lambda = 0$ for $\tilde{\mathbf{w}}$ subject to $\text{supp}(\tilde{\mathbf{w}}) \subset \text{supp}(\mathbf{w}_\ell^{(i)})$, and then setting $\mathbf{w}_{\ell+1}^{(i)} = H_\lambda(\tilde{\mathbf{w}})$. A sweep over 40 equally log-spaced λ values $\boldsymbol{\lambda} = (10^{-4}, \ldots, 1)$ is performed according to [12,62] to choose an appropriate threshold λ.

3.2. Model replacement

Model replacement is akin to 'cross-pollination' and is crucial to increasing the accuracy of the learned models, as it transfers successful learning of few cells with highly informative trajectories to cells with less informative trajectories. As with all validation steps of our algorithm, this approach would be infeasible if not for fast data-driven forward simulations, as discussed further in §3.5.

Once the initial batch of N models $\mathcal{M}$ is learned, we simulate each model $\mathcal{M}_j$ as outlined in §3.5 on K different *validation cells* selected from the data and specific to cell j, where we set $K = 32$ throughout. If $\mathcal{M}_j$ performs better than $\mathcal{M}_i$ on cell i, we replace $\mathcal{M}_i$ with $\mathcal{M}_j$ (specifically, $\mathcal{M}_i$ is replaced with the *best performing* such model, if one exists).

For a given model $\mathcal{M}_i$, we select the K validation cells by finding cells in the population that match well certain statistics of cell i. In particular, we define the following distributions:

$$\rho_{rr}^{(i)}(r) = \frac{1}{T}\int_0^T \mathbb{P}_{x\sim X'}(\|x_i(t)-x(t)\| < r)\,dt, \tag{3.12}$$

$$\rho_{vv}^{(i)}(s) = \frac{1}{T}\int_0^T \mathbb{P}_{v\sim V'}(\|v_i(t)-v(t)\| < s)\,dt \tag{3.13}$$

and

$$\rho_{v}^{(i)}(s) = \mathbb{P}(\|v_i\| < s), \tag{3.14}$$

where (X', V') denotes the remainder of the cell population excluding cell i. Respectively, these denote the distribution of *distances* from cell i to all other cells, the distribution of *velocity differences* between cell i and all other cells, and the distribution of *speeds* that cell i experiences. These statistics are likely to correspond to the information content that cell i carries about its own forces f_{a-r}, f_{align} and f_{drag}, given the force dependencies. We approximate these distributions from the data using histograms with 50 bins. Figures 15 and 16 in the appendix depict species averages of $\rho_{rr}^{(i)}$, $\rho_{vv}^{(i)}$, $\rho_{v}^{(i)}$.

For each cell i, we compute the Kullback–Leibler (KL) divergence between its distributions $\rho_{rr}^{(i)}$, $\rho_{vv}^{(i)}$, $\rho_{v}^{(i)}$ and those of the rest of the population,[6] where the KL divergence between densities ρ and ν is given by

$$\mathcal{D}_{\text{KL}}(\rho|\nu) = -\int \rho(x)\log\left(\frac{\nu(x)}{\rho(x)}\right)dx.$$

The K validation cells used to validate model i are the K cells with smallest cost $\mathcal{L}$, defined by

$$\mathcal{L} := \mathcal{D}_{\text{KL}}(\rho_{rr}^{(i)}|\rho_{rr}^{(j)})^2 + \mathcal{D}_{\text{KL}}(\rho_{vv}^{(i)}|\rho_{vv}^{(j)})^2 + \mathcal{D}_{\text{KL}}(\rho_{v}^{(i)}|\rho_{v}^{(j)})^2.$$

Let the validation error $\Delta V_{i\to j}$ be defined as in (3.19), but indicating $\mathcal{M}_i$ used to validate cell j (i.e. using the initial conditions of cell j). We replace $\mathcal{M}_i$ with $\mathcal{M}_j$ if the following three conditions are met:

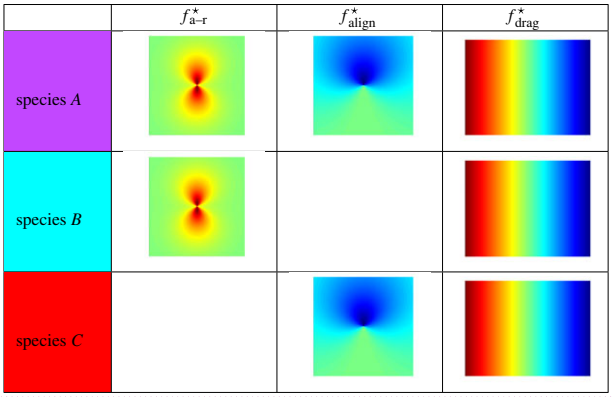
(1) $\Delta V_{i\to i} > \Delta V_{j\to i}$
(2) $\Delta V_{i\to j} > \Delta V_{j\to j}$
(3) $\max\{\Delta V_{j\to i}, \Delta V_{j\to j}\} < \text{tol}$,

where we set $\text{tol} = 0.25$ in this work. In words, $\mathcal{M}_j$ replaces $\mathcal{M}_i$ if (1) $\mathcal{M}_j$ performs better than $\mathcal{M}_i$ on cell i, (2) $\mathcal{M}_j$ performs better than $\mathcal{M}_i$ on cell j, and (3) $\mathcal{M}_j$ achieves a reasonably low error (defined by tol) on both cell i and cell j. (Note that cell i and cell j are required to be mutual validation cells for a model replacement to occur). Furthermore, if $\mathcal{M}_j$ replaces $\mathcal{M}_i$, and another model $\mathcal{M}_k$ replaces $\mathcal{M}_j$, we replace $\mathcal{M}_i$ with $\mathcal{M}_k$ as well, even if cells i and k are not mutual validation cells.

3.3. Cluster

Models are then clustered according to the force modes present. Specifically, using the bases above, we can expand each

Table 1. Species delineation by active force modes.



force according to distinct directional modes

$$f_{\mathrm{a-r}}(r,\theta) = f^{(0)}_{\mathrm{a-r}}(r) + \cos(\theta) f^{(1)}_{\mathrm{a-r}}(r) + \cos(2\theta) f^{(2)}_{\mathrm{a-r}}(r),$$

$$f_{\mathrm{align}}(r,\theta) = f^{(0)}_{\mathrm{align}}(r) + \cos(\theta) f^{(1)}_{\mathrm{align}}(r) + \cos(2\theta) f^{(2)}_{\mathrm{align}}(r)$$

and

$$f_{\mathrm{drag}}(|v|,\theta) = f^{(0)}_{\mathrm{drag}}(|v|) + \cos(\theta) f^{(1)}_{\mathrm{drag}}(|v|).$$

This leads to eight possible force modes, which we order as follows:

$$\{f^{(0)}_{\mathrm{a-r}}, f^{(1)}_{\mathrm{a-r}}, f^{(2)}_{\mathrm{a-r}}, f^{(0)}_{\mathrm{align}}, f^{(1)}_{\mathrm{align}}, f^{(2)}_{\mathrm{align}}, f^{(0)}_{\mathrm{drag}}, f^{(1)}_{\mathrm{drag}}\}. \tag{3.15}$$

We associate the sparsity pattern of the force modes with the set of all 8-bit codes, giving a total of $2^8 = 256$ possible model clusters. Models are partitioned into clusters $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_r\}$ based on their associated codes. For example, species A listed in table 1 is associated with the code 10111010 indicating that $f^{(0)}_{\mathrm{a-r}}$, $f^{(2)}_{\mathrm{a-r}}$, $f^{(0)}_{\mathrm{align}}$, $f^{(1)}_{\mathrm{align}}$ and $f^{(0)}_{\mathrm{drag}}$ are present in the model.

There are several other options for model replacement and clustering, include clustering based on the sparsity pattern of $\widehat{\mathbf{w}}$, or simply on the presence or the absence of each of the three forces $f_{\mathrm{a-r}}$, f_{align} and f_{drag}. The former significantly increases the number of possible clusters, while the latter leads to just 8 possible clusters. Our choice reflects the desire to disentangle directionalities of the governing forces without introducing a strong dependence on the bases used to approximate each force.

3.4. Aggregate

Having formed the model clusters $\mathcal{C}$, let $\overline{\mathcal{C}}$ be the cluster with the most members. We then compute $\overline{\mathcal{M}}$ by averaging[7] over the coefficients from models in $\overline{\mathcal{C}}$ and then performing a final round of thresholding. That is, we compute

$$\left.\begin{aligned} &\overline{\mathbf{w}} = \tfrac{1}{|\overline{\mathcal{C}}|}\sum_{i\in\overline{\mathcal{C}}} \widehat{\mathbf{w}}^{(i)}, \\ &\mathcal{I} = \{i : |\overline{\mathbf{w}}_i| < 10^{-\lambda_{\log}} \max|\overline{\mathbf{w}}|\} \\ \text{and}\quad &\overline{\mathbf{w}}(\mathcal{I}) = 0, \end{aligned}\right\} \tag{3.16}$$

where $|\overline{\mathcal{C}}|$ denotes the number of elements in $\overline{\mathcal{C}}$ and $\lambda_{\log} = 4$ in this work, so that coefficients falling below four orders of magnitude from the maximum absolute coefficient are discarded. Thresholding here is simply to speed up computation, as small coefficients result in unnecessary evaluation of negligible basis functions during forward solves.

3.5. Validate

To validate the aggregate model $\bar{\mathcal{M}}$, we perform forward simulations over the remaining unclassified cells in a highly parallelizable way that uses the experimental data to efficiently march forward in time.

Let $N' \le N$ be the number of remaining unclassified cells. For each $i = 1, \ldots, N'$, we simulate a new trajectory $\{(\overline{\mathbf{x}}_i(t_k), \overline{\mathbf{v}}_i(t_k))\}_{k=1}^{L}$ using the averaged model $\bar{\mathcal{M}}$ with the experimental initial conditions $(\overline{\mathbf{x}}_i(0), \overline{\mathbf{v}}_i(0)) = (\mathbf{x}_i(0), \mathbf{v}_i(0))$. We march forward in time according to the forward Euler update

$$\overline{\mathbf{v}}_i(t_{k+1}) = \overline{\mathbf{v}}_i(t_k) + \Delta t \overline{\mathcal{M}}(\overline{\mathbf{x}}_i(t_k), \overline{\mathbf{v}}_i(t_k), \mathbf{X}'^i(t_k), \mathbf{V}'^i(t_k)) \tag{3.17}$$

and

$$\overline{\mathbf{x}}_i(t_{k+1}) = \overline{\mathbf{x}}_i(t_k) + \Delta t \overline{\mathbf{v}}_i(t_k), \tag{3.18}$$

where $(\mathbf{X}'^i(t_k), \mathbf{V}'^i(t_k))$ indicates $(\mathbf{X}(t_k), \mathbf{V}(t_k))$ with the ith cell removed.[8] Since the time resolution of the data Δt is assumed to be coarse, we perform the simulation on a finer grid with time step $\Delta t' = 2^{-5}\Delta t$, and use piecewise cubic hermite interpolation to generate positions and velocities of neighbours $(\mathbf{X}'^i, \mathbf{V}'^i)$ at intermediate times. We stress that we do not update the neighbour cells using the model $\bar{\mathcal{M}}$, which would be much more costly; we merely use neighbour positions and velocities from the data to compute interactions that govern the motion of cell i. The resulting trajectories $\{(\overline{\mathbf{x}}_i, \overline{\mathbf{v}}_i)\}_{i=1}^{N'}$ can then be computed in a trivially parallel manner.[9]

We then define the validation error for cells $i = 1, \ldots, N'$ as the relative velocity difference

$$\Delta V_i := \sqrt{\frac{\sum_{k=1}^{L'} \|\overline{\mathbf{v}}_i(t_k) - \mathbf{v}_i(t_k)\|_2^2}{\sum_{k=1}^{L'} \|\mathbf{v}_i(t_k)\|_2^2}}, \tag{3.19}$$

where $L' \le L$ is a subset of the time series over which the simulation is expected to remain accurate. In particular, for chaotic systems the trajectories cannot be expected to remain close for large times; however, the correct model will be initially accurate. In this work, we choose $L' = 0.25L$. In other words, with $L = 200$ time steps (as in most examples below), we compare with the data over the first 50 time steps at the original scale Δt, or equivalently 1600 time steps on the finer scale $\Delta t'$.

3.6. Classify

Let VE be the set of validation errors, $\mathrm{VE} = \{\Delta V_1, \ldots, \Delta V_{N'}\}$. An empirical observation used in this work is that when $\bar{\mathcal{M}}$ approximates well an underlying model for a true species, the log-transformed validation errors $\log_{10}(\mathrm{VE})$ are fit well by a Gaussian mixture model (GMM) with two mixtures (see figures 4–6). We thus use a two-mixture GMM to classify the remaining cells. Cells are granted membership into the mixture that yields the highest posterior probability of generating its log-validation error, and the class with lowest mean error is labelled as a species. This can be thought of as a sequential binary classification scheme.

For example, in each plot of figure 4, a representative GMM resulting from a two-species population, the left-most mixture corresponds to low validation errors under the model $\bar{\mathcal{M}}$ and is classified as a species $\overline{\mathcal{S}}$ (in this case, species

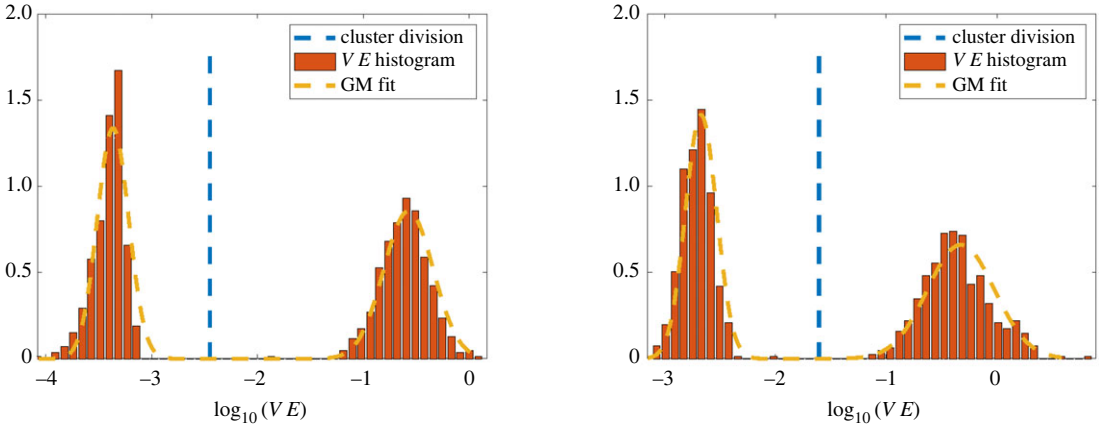


Figure 4. Distribution of log-validation errors for heterogeneous cell experiments $\mathbf{X}_{A,C}$ and $\mathbf{X}_{B,C}$. In each case, species C is identified in the first iteration, and a clear separation between the two species allows for accurate clustering using Gaussian mixture models.

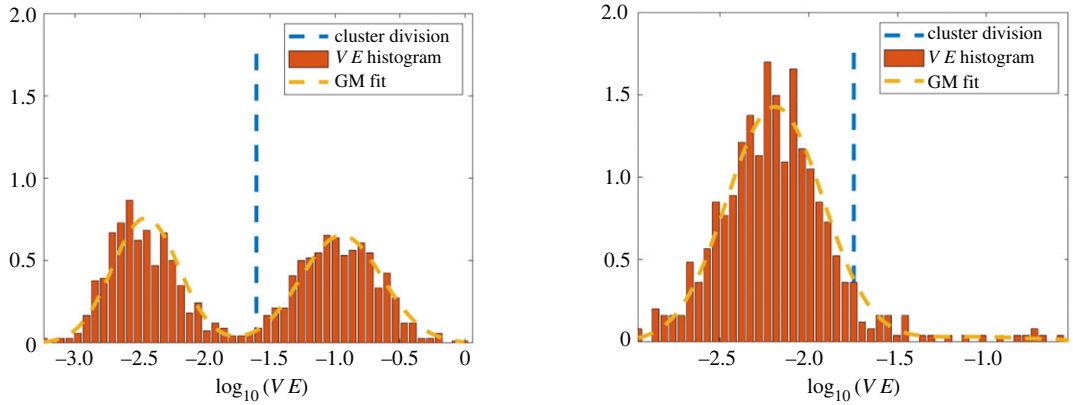


Figure 5. Log-validation errors for two-species population $\mathbf{X}_{A,B}$ (long). Strong similarities between the two species present an initial challenge to identification, which is overcome by taking a longer time series. The initial Gaussian mixture model (left) identifies a majority species B cluster. In the second iteration (right), a cluster with all species A cells is identified, and a small group of cells remains which is then partitioned correctly (see row 5, columns CS(A) and CS(B) of table 3).

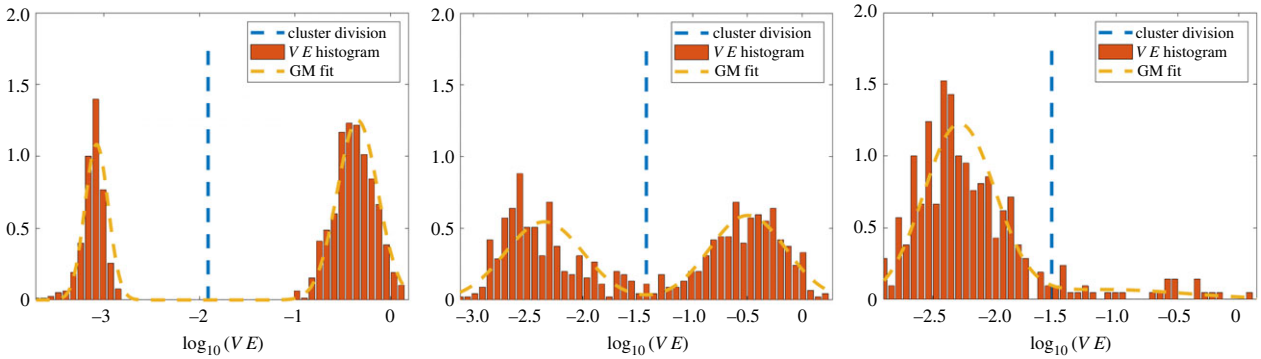


Figure 6. Gaussian mixture models for classifying the three-species experiment $\mathbf{X}_{A,B,C}$(long) (see table 4 row 3 for details). We see an initial complete separation of species C (left), followed by a mixed cluster containing 96.1% of the species B cells and 0.9% of the species A cells (middle). The next iteration classifies an entirely species A cluster (right). Clusters 4 and 5 are effectively outliers and contain the remaining 31/1000 cells.

C in table 1 is identified). The cells in $\overline{\mathcal{S}}$ are subsequently removed from the population, and cells in the right-most mixture are returned to the cluster phase (c).

3.7. Stopping conditions

Steps (c)–(f) are repeated until one the following conditions is met.

(1) Less than $N'_{\min}$ cells remain.
(2) More than $(1-\delta_{\text{gmm}})\times 100\%$ of remaining cells have less than $\epsilon_{\text{gmm}}\times 100\%$ validation error: $\mathbb{P}(\text{VE} < \epsilon_{\text{gmm}}) \geq (1-\delta_{\text{gmm}})$.
(3) The maximum allowable number of species has been reached: $S = S_{\max}$.

The first case is an obvious criterion to prevent infinite looping over outlier cells for which there is not enough

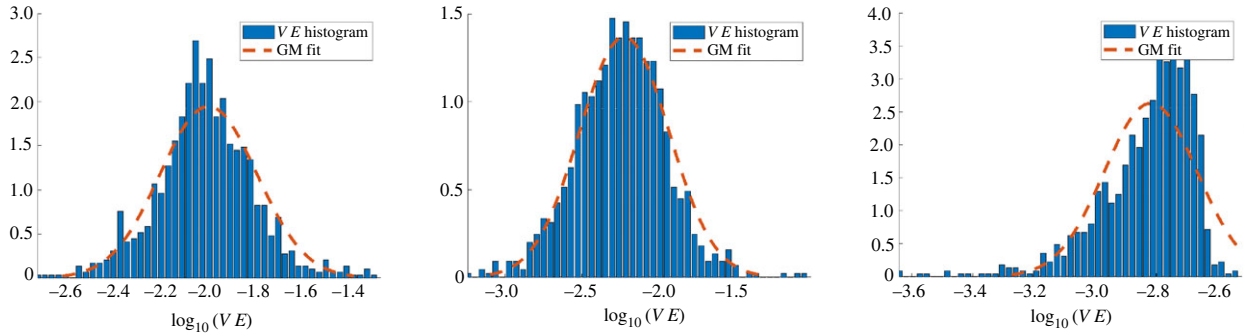


Figure 7. Distribution of log-validation errors for homogeneous cell experiments $\mathbf{X}_A$, $\mathbf{X}_B$, $\mathbf{X}_C$. Distributions for $\mathbf{X}_A$ and $\mathbf{X}_B$ are fit well by a single Gaussian, indicating a single species is present. The distribution for $\mathbf{X}_C$ has a non-Gaussian tail, although all errors are below 1%, indicating that the candidate model fits the population up to the specified error tolerance.

Table 2. Performance of model learning and classification algorithm of homogeneous populations.

experiment	Δf_{a-r}	Δf_{align}	Δf_{drag}	CS(A)	CS(B)	CS(C)	ΔV
$\mathbf{X}_A$	0.0211	0.0384	0.0382	1.000	—	—	0.0100
$\mathbf{X}_B$	0.0112	—	0.0125	—	0.997	—	0.0076
$\mathbf{X}_C$	—	0.0007	0.0016	—	—	1.000	0.0016

information to learn an adequate model. We set $N'_{\min} = 2$. The second condition skips the GMM fitting process when all cells have sufficiently low error. If the condition is met, all cells with error less than ϵ_{gmm} are assigned to a new species, while the remaining cells are left as outliers without a model. We choose $(\epsilon_{gmm}, \delta_{gmm}) = (0.05, 0.01)$, such that if 99% of the cells achieve less than 5% error, the algorithm terminates. This is necessary to account for the case of *high-accuracy* recovery, where it is observed that *VE* is no longer approximately lognormal, leading to an inaccurate GMM partition (see e.g. the rightmost plot of figure 7). Finally, for N very large, it may be necessary to restrict the total number of species, which is encapsulated in the third condition. We set $S_{\max} = 10$ throughout, although we did not observe the number of iterations exceeding 5 in any trials with $N \leq 1000$ and $S \leq 3$ true species.

4. Results: artificial cells

We examine artificial cell cultures with combinations of 1–3 distinct cell types, denoted by species *A*, species *B* and species *C*. Each species has a unique combination of the following forces:

$$f^{\star}_{a-r}(r, \theta) := (15 + 10\cos(2\theta))(e^{-20r} - 0.25\, e^{-10r}), \quad (4.1)$$

$$f^{\star}_{align}(r, \theta) := -(8 + 8\cos(\theta))\, e^{-8r} \quad (4.2)$$

and

$$f^{\star}_{drag}(s, \theta) := -5s. \quad (4.3)$$

The forces $f^{\star}_{a-r}$ and $f^{\star}_{align}$ are depicted in figure 2, and force combinations for species *A*, *B* and *C* are specified in table 1. The forces include a *quadrupolar* attractive–repulsive force $f^{\star}_{a-r}$, a *dipolar* alignment force $f^{\star}_{align}$, and a *monopolar* drag force $f^{\star}_{drag}$ which is linear in its speed argument. As we will see below, species *A* and species *B* share the force f_{a-r} and hence result in similar dynamics, which presents a challenge to identification. It turns out that using a longer time series results in correct classification.

We let $\mathbf{X}_P$ denote a simulation with individuals from species *P*, for example, $\mathbf{X}_A$ is a simulation with only individuals from species *A* and $\mathbf{X}_{A,B}$ is a simulation with a mixed population of species *A* and species *B*. Each simulation has 1000 individuals and the same number of members in each species (up to rounding). More details on the simulations, including plots of initial and final states and several statistics, can be found in appendix A.2.

We refer to species identified by the algorithm as 'clusters' to disambiguate between the true species (*A*, *B*, *C*) present in the data. We are particularly interested in three traits of our learning algorithm: (1) was the classification successful? (2) are the learned forces close to the true forces? (3) are simulated trajectories using the learned model close to the original trajectories? To assess (1) we report the *classification success* CS(*i*) for $i \in \{A, B, C\}$ as the fraction of individuals from species *i* that ended up in the cluster in question, where clusters are listed as subrows (rows not separated by horizontal lines) within each row in tables 2–4, in the order they were identified. For example, in row 2 of table 3, two clusters are identified from the two-species data $\mathbf{X}_{A,C}$, with the first cluster containing 100% of the species *C* cells, indicated by CS(*C*) = 1.000, and the second cluster containing 100% of the species *A* cells, indicated by CS(*A*) = 1.000, with no outliers.

To assess the accuracy of learned forces with respect to the ground truth forces, for each of the three forces we compute the relative L^2 error over a square grid discretized with 1000 points in each direction. We denote these quantities by Δf_{a-r}, Δf_{align} and Δf_{drag} in tables 2–4. For f_{a-r} and f_{align} we use $(x, y) = (r\cos\theta, r\sin\theta) \in [-2, 2] \times [-2, 2]$, since $r_{\max} \approx 2$ for all examples, and for f_{drag} we use $(s, \theta) \in [0, s_{\max}] \times [0, \pi]$, where $s_{\max}$ is the maximum speed attained during the experiment. It is worth mentioning that for f_{a-r} we use a force that

Table 3. Performance of model learning and classification algorithm for two-species populations. $\mathbf{X}_{A,B}$(long) is simply the continuation of $\mathbf{X}_{A,B}$ to twice the time horizon, and significantly improves classification over $\mathbf{X}_{A,B}$. Note that identified species are listed within each delinearted row as subrows (rows not separated by horizontal lines) in the order they were identified.

experiment	Δf_{a-r}	Δf_{align}	Δf_{drag}	CS(A)	CS(B)	CS(C)	ΔV
$\mathbf{X}_{A,C}$	—	0.0011	0.0002	0	—	1.000	0.0005
	0.0226	0.0941	0.0472	1.000	—	0	0.0200
$\mathbf{X}_{B,C}$	—	0.0077	0.0051	—	0	1.000	0.0023
	0.0328	—	0.0461	—	1.000	0	0.0339
$\mathbf{X}_{A,B}$	0.0341	—	0.0133	0.102	0.628	—	0.1201
	0.0075	—	0.0538	0.084	0.340	—	0.0362
	0.4780	0.3660	—	−0.814	0.012	—	0.3945
$\mathbf{X}_{A,B}$(long)	0.0018	—	0.0042	0.002	0.978	—	0.0045
	0.0034	0.0023	0.0067	0.994	0	—	0.0070
	0.0071	—	0.0044	0	0.023	—	0.0568
	0.0034	0.0051	0.0144	0.004	0	—	0.0199

Table 4. Performance of model learning and classification algorithm of three-species populations. $\mathbf{X}_{A,B,C}$(long) is simply the continuation of $\mathbf{X}_{A,B,C}$ to twice the time horizon.

experiment	Δf_{a-r}	Δf_{align}	Δf_{drag}	CS(A)	CS(B)	CS(C)	ΔV
$\mathbf{X}_{A,B,C}$	—	0.0005	0.0020	0	0	1.000	0.0016
	0.0287	—	0.0191	0.091	0.988	0	0.0620
	0.0243	0.2321	—	0.542	0.006	0	0.5010
	0.0022	0.0001	0.0050	0.332	0	0	0.0406
	0.0478	—	0.118	0.009	0.006	0	0.3462
$\mathbf{X}_{A,B,C}$(long)	—	0.0014	0.0013	0	0	1.000	0.0009
	0.0082	—	0.0008	0.009	0.961	0	0.0064
	0.0041	0.0014	0.0010	0.937	0	0	0.0067
	0.0065	0.0077	0.0074	0.045	0	0	0.0482
	0.0151	—	0.0054	0.009	0.039	0	0.347

does not have a sparse representation in the basis $\mathcal{F}_{a-r}$. In this case, we see that the algorithm correctly classifies individuals despite having the truncation error that results from representing the force over the basis $\mathcal{F}_{a-r}$.

Lastly, we assess the difference in learned and true trajectories using the average validation error $\Delta V = (1/|\overline{\mathcal{S}}|)\sum_{i\in\overline{\mathcal{S}}}\Delta V_i$, where ΔV_i is computed from model $\overline{\mathcal{M}}$ associated with identified species $\overline{\mathcal{S}}$ using (3.19).

4.1. Homogeneous populations

As an initial benchmark, we detect single-species populations from homogeneous data. While simpler than the heterogeneous case, this is a non-trivial task due to the variability of single-cell trajectories and local environments within the population. Our method successfully identifies the models for species A, B, and C from homogeneous simulations, achieving less than 1% mean validation errors in each case, and less than 4% relative force errors Δf (table 2). In simulation $\mathbf{X}_B$, three cells are identified as outliers (appearing in the right tail of figure 7 (middle)), and all other cells in $\mathbf{X}_A$, $\mathbf{X}_B$, and $\mathbf{X}_C$ are correctly classified. A comparison between original and learned trajectories is depicted in figure 8, with learned trajectories overlapping original trajectories in each case.

4.2. Two-species populations

Next we examine the ability of the learning algorithm to detect two-species populations along with accurate aggregate models. Figure 4 displays two representative Gaussian mixture fits to the log-validation errors for $\mathbf{X}_{A,C}$ (left) and $\mathbf{X}_{B,C}$ (right). In both cases, the log-errors are well approximated by Gaussian mixtures with wide separations between mixtures. This allows for complete classification in both cases, as indicated by CS(A), CS(B), and CS(C) in rows 2 and 3 of table 3. Force differences Δf are less than 5% in all but one case (estimation of f_{align} in cluster 2 of experiment $\mathbf{X}_{A,C}$), with trajectory validation errors less than 3.5%. In particular, species C achieves less that 0.3% validation error, which is due to the true force $f^{\star}_{\text{align}}$ existing in the span of the library $\mathcal{F}_{\text{align}}$, whereas $f^{\star}_{a-r}$ is approximated using a truncated

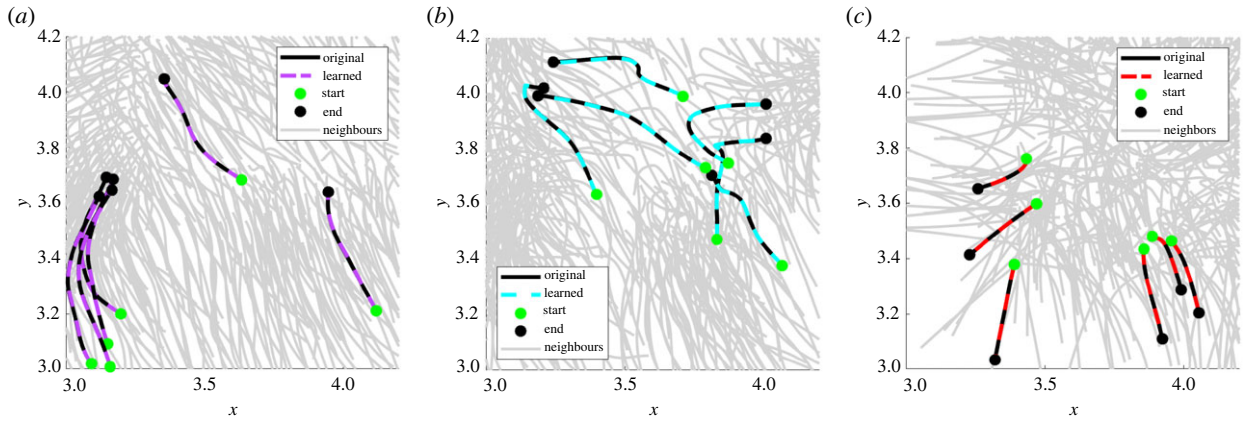


Figure 8. Examples of learned and original trajectories from homogeneous populations. (*a*) $\mathbf{X}_A$, (*b*) $\mathbf{X}_B$, (*c*) $\mathbf{X}_C$.

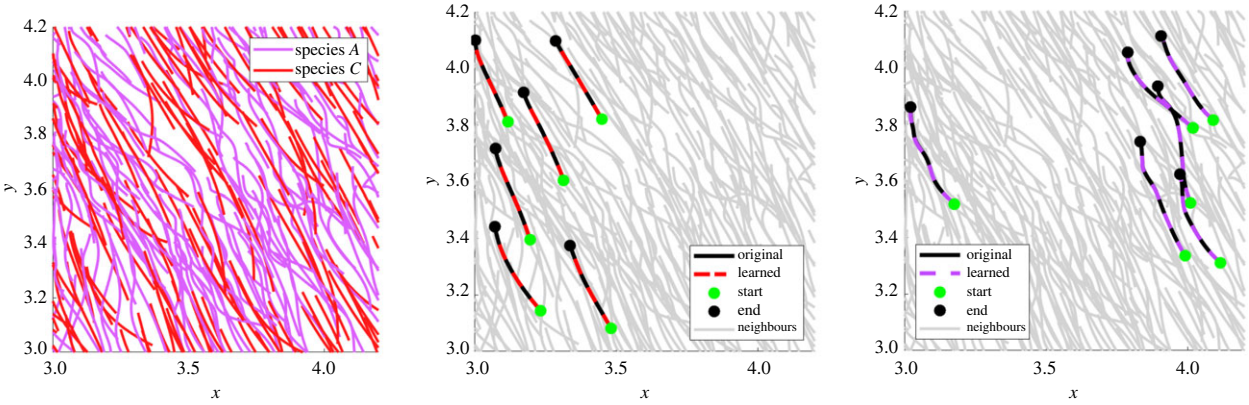


Figure 9. Example trajectories from experiment $\mathbf{X}_{A,C}$. Cells with true colour labels are depicted on the left, but are passed into the algorithm unlabelled. The algorithm then classifies the population into different species and returns accurate models for each species. Classified cells from species *C* (middle) and species *A* (right) are highlighted showing excellent agreement between data and simulation.

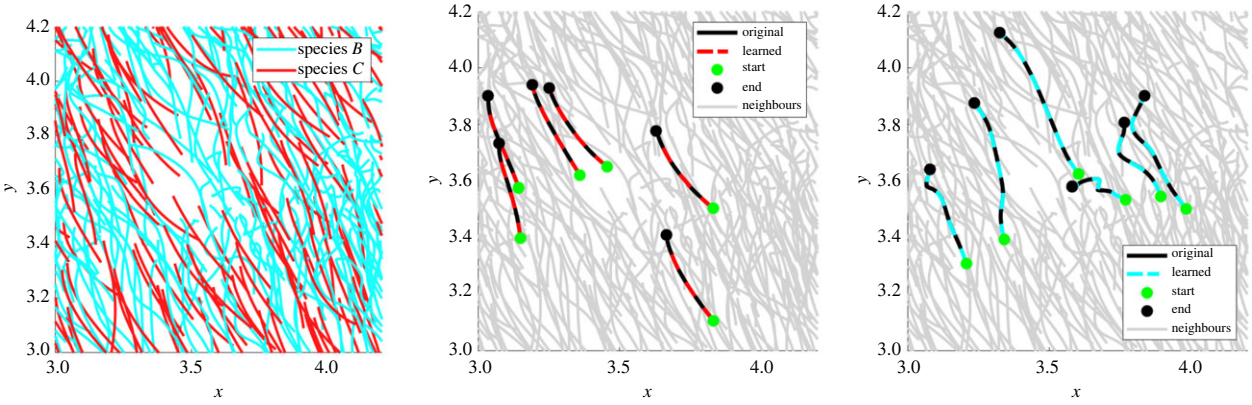


Figure 10. Example trajectories from experiment $\mathbf{X}_{B,C}$. As in figure 9, cells with true colour labels are depicted on the left. Classified cells from species *C* (middle) and species *B* (right) are highlighted showing excellent agreement original data and output of the learned models.

series expansion, resulting in larger errors. See figures 9 and 10 for comparison between original and learned trajectories.

For experiment $\mathbf{X}_{A,B}$, initially the method is incapable of correctly classifying cells into species *A* and species *B*. Three clusters are identified with suboptimal models (table 3 row 4). Accurate classification is achieved by running the algorithm with a longer experiment $\mathbf{X}_{A,B}$(long) (table 3 row 5) which is the continuation of $\mathbf{X}_{A,B}$ for twice the total time points, at the same temporal resolution. An initial cluster is identified containing 97.8% of species *B* along with 0.2% (a single cell) of the existing species *A* cells, followed by a second cluster with 99.4% of the species *A* cells and no cells from species *B*. The last two clusters correctly partition the remaining cells (12 in total), again finding accurate models, allowing for recombination with the first two cluster during post-processing.

Figure 11 shows a comparison between original and learned trajectories for $\mathbf{X}_{A,B}$(long) and figure 5 depicts

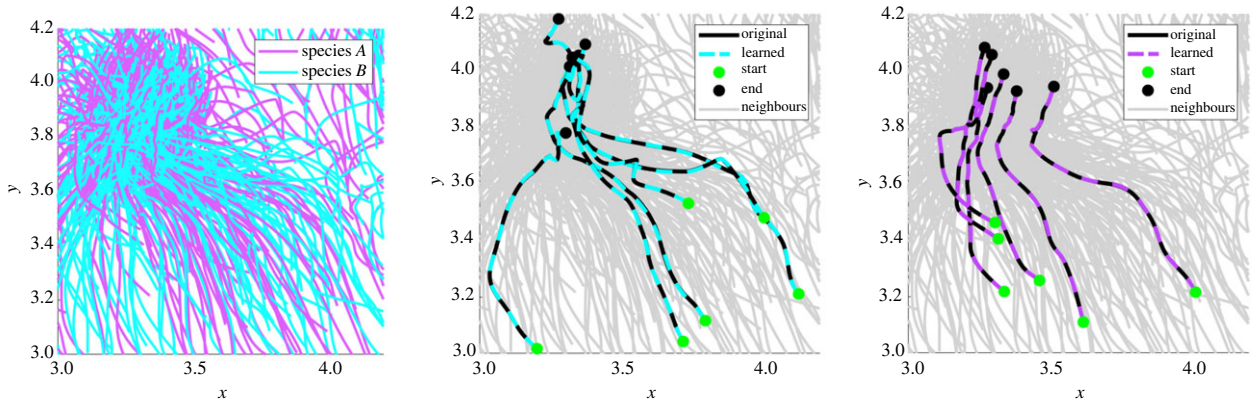


Figure 11. Example trajectories from experiment $\mathbf{X}_{A,B}$(long). Cells with true labels are depicted on the left and classified cells from species *B* (middle) are species *A* (right) are depicted with model output overlapping the input data in each case.

representative Gaussian mixture models. In particular, figure 5 (left) shows increased overlap between the two Gaussian mixtures in the first iteration, compared with figure 4; however, model performance is still sufficiently different as to classify approximately 98% of cells correctly.

4.3. Three-species population

As a final test we identify species from the three-species experiment $\mathbf{X}_{A,B,C}$. Similar to the case $\mathbf{X}_{A,B}$, we see improvements with a longer time-series $\mathbf{X}_{A,B,C}$(long). For the initial experiment $\mathbf{X}_{A,B,C}$, species *C* is completely identified in the first cluster (table 4 row 2), and in the second cluster 98.8% of species *B* cells are identified along with 9.1% of species *A* cells, leading to a fairly inaccurate model ($\Delta V \approx 0.06$). The subsequent clusters divide the remaining species *A* and *B* cells.

Doubling the time series with $\mathbf{X}_{A,B,C}$(long), we find the majority of each species residing in its own cluster (table 4 row 3). Cluster 1 contains all of the species *C* cells, cluster 2 consists of 96.1% of species *B* cells and 0.9% of species *A*, and cluster 3 consists of 93.7% of species *A*. Moreover, the aggregate models for each of these first three clusters result in validation errors under 1%.

Clusters 4 and 5 of $\mathbf{X}_{A,B,C}$(long) contain the remaining 31 *A* and *B* cells (3.1% of the total population); however, the learned forces in each cluster are still accurate: we find that $\Delta f < 2\%$ for all forces and all clusters (row 3 of table 4). The validation error is high for cluster 5, reaching $\Delta V = 35\%$, which indicates that the cells in cluster 5 have trajectories that are particularly sensitive to perturbations. Given the complexity of the dynamics (one can observe sharp turns taken by cells in the bottom two plots of figure 12), trajectories cannot be expected to remain close for all time, and in this case the validation error (3.19) may be too strong a metric.[10] It is thus remarkable that the aggregate models for clusters 1, 2 and 3 produce accurate learned trajectories.

For a more in-depth statistical view of the algorithm for $\mathbf{X}_{A,B,C}$(long), in figure 17 we depict the average pointwise error and variance of the learned forces $f_{\mathrm{a-r}}$ and f_{align} across all individual learned models for cells in cluster 3. Moreover, we compare the effects of computing the aggregate model (given by the coefficients $\overline{\mathbf{w}}$) as a raw cluster average versus first performing the model replacement step and then thresholding the final coefficients. For each force, we see that both methods produce satisfactory models to the eye, yet examining the pointwise error and variance reveals that the model replacement + thresholding step reduces errors by orders of magnitude.

5. Discussion

We have introduced a method for performing a combined classification and model selection task relevant to heterogeneous systems of autonomous agents. Specifically, we have shown that learning an ensemble of interacting particle models (one for each agent) allows iterative classification of agents into species according to their forward simulation accuracy. This is surprising due to the limited information carried in a single trajectory. Fortunately, the validation errors empirically approximate a log-normal distribution, hence Gaussian mixture model classification arises as the appropriate tool for identifying species membership.

Computational feasibility of this approach is grounded in the parallel nature of both the learning algorithm and the simulation component. Learning each single-trajectory model is cheap, with linear systems of size $m \times n$ with m and n not exceeding several hundreds (see table 7 for wall times). In the simulation step, each trajectory is validated separately and in parallel, and neighbour interactions are computed using the measurement data itself, resulting in $\mathcal{O}(N)$ pairwise interaction computations per time step instead of an $\mathcal{O}(N^2)$ full simulation.[11]

As previously mentioned, this approach is widely applicable to heterogeneous interacting particle systems, but there are many opportunities for extension. New techniques will need to be developed to effectively model particles which switch behaviours over time, or for non-conserved particle number (due to e.g. cell division or death). Multiple species which share force modes but vary in their magnitudes will also be challenging to identify using the proposed method. It is also possible that for highly diffusive particles the validation metric (3.19) is too restrictive, and a purely position-based metric should be used to classify species.

Lastly, several aspects of this approach deserve a more technical analysis, which may lead to improvements. We aim in future work to undertake more rigorous study of each component of the algorithm as outlined below.

5.1. Learning single-cell models

(I) **Information content**: It would be beneficial to quantify the information content in each cell trajectory,

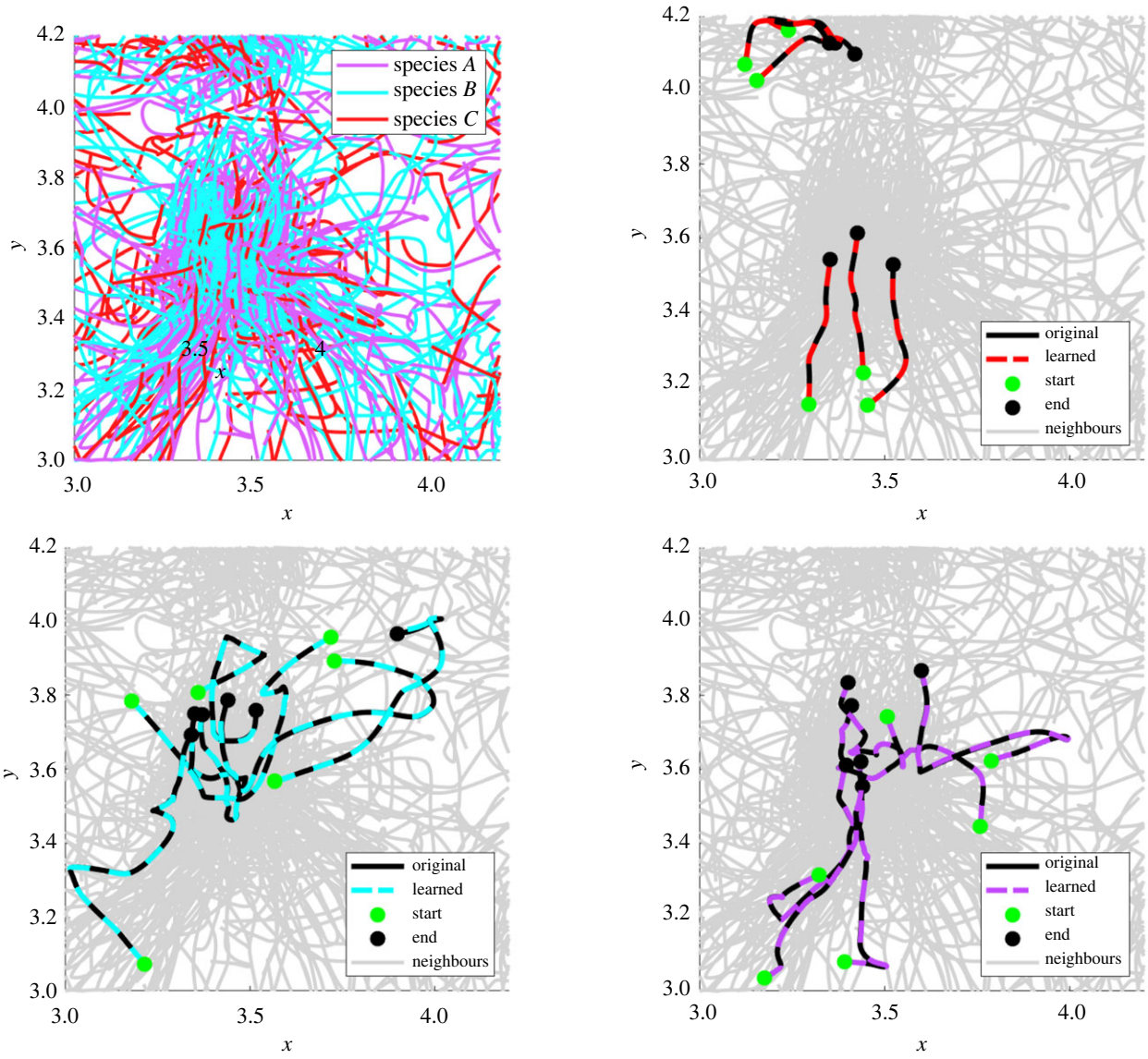


Figure 12. Example trajectories simulated using learned models from $\mathbf{X}_{A,B,C}$(long). Top left: example cells with true labels. Top right, bottom left, bottom right: example trajectories from clusters 1–3 (see table 4 row 3 for details). Note in particular that learned models for clusters 2 and 3 are able to capture sharp turns in the true dynamics.

possibly eliminating trajectories that do not provide sufficient information. Model replacement, as outlined in §3.2, is an initial step in this direction. We saw in the experiments $\mathbf{X}_{A,B}$ and $\mathbf{X}_{A,B,C}$ that increasing the length of the trajectory leads to better classification when two species exhibit similar dynamics. It may be possible to use existing techniques, such as force matching, to identify highly informative cells from forces magnitudes, neighbour distributions, etc.

(II) **Noisy trajectories**: To focus on the classification task and equation learning methodology, we have neglected to add noise to trajectories in this work. However, it is reasonable to anticipate that measurement noise will be filtered out by the weak form as previously demonstrated on ODEs [62], PDEs [63] and first-order IPS [12]. We leave full examination of the robustness to both intrinsic (e.g. Brownian) and extrinsic (e.g. measurement) noise to future work.

(III) **Model library**: We chose the force bases and constraints to reflect physical properties, namely short-distance repulsion, long-distance decay, negative alignment, and negative drag. Directional modes enforce bilateral symmetry, and are low order (monopole, dipole, quadrupole). These can easily be adapted to incorporate other known information; moreover, the bases themselves may be adapted to the data (note this is partially done, using the neighbour distance distribution ρ_{rr} to restrict the range of interactions). One major assumption here is that there is no propulsion force, that energy is increased only through anisotropic interactions with neighbours. It would be interesting to examine whether this assumption holds true.

(IV) **Regression approach**: We employ modified sequential thresholding, which looks for an overall sparse solution, although we threshold *only* on the term magnitude $||\mathbf{G}_j\mathbf{w}_j||$ and not that raw coefficient $\mathbf{w}_j$. This in particular allows f_{a-r}, f_{align} and f_{drag} to have equal opportunity to enter the model despite different scales and bases used. The effect is that the resulting model is sparse in the *force modes* (as described in §3.3), while each force mode may have many components (in fact f_{a-r} is usually not sparse on a given directional mode). It may be more appropriate to use

a group sparsity-enforcing method, such as constrained group LASSO. In general, this lies at the intersection of approximation and selection, where sparse selection is required to select the correct modes; however, the force content in each mode requires approximation. Explorations of the appropriate balance between selection and approximation would be valuable.

5.2. Cluster and aggregate

(I) Here we cluster models based on the directional modes present. This can easily be extended to the full pattern of non-zero elements in the model vector $\widehat{\mathbf{w}}$, although this depends on the number of resulting clusters.

(II) We have used a simple uniformly weighted average (3.16) to aggregate models; however, the use of model replacement (see §3.2) implies that the average is implicitly weighted according to *model generalizability*. Results may be improved if other criteria (e.g. information criteria) are incorporated into the weighted average, or if the median is taken instead of the mean.

(III) In the examples above, the aggregate model is used as the final model for the give class. Instead, one could further refine the model by performing an additional regression combining all data from the identified species.

(IV) Each cell experiences a different total number and duration of interactions with other cells. Accordingly, the models identified for each cell have varying levels of reliability, depending on the amount of information acquired to inform the model. This necessitated the development of our ad hoc classification scheme as we were unable to identify a suitable approach for sorting models with varying degrees of trustworthiness.

5.3. Validate and classify

In practice, the validation errors can easily be checked to satisfy lognormalcy *a posteriori*. If this is not satisfied, it may not be straightforward to cluster based on the validation error. In particular, chaotic trajectories cannot be expected to achieve a low validation error, in which case another metric is needed. In this case, it is reasonable to require that trajectories to be long enough to compute statistics. We aim to investigate the requirements for performing classification with chaotic interacting particles in future work.

Data accessibility. All software used to generate the results in this work is available at this repository: https://github.com/MathBioCU/WSINDy_CellCluster.git.

Authors' contributions. D.A.M.: conceptualization, formal analysis, investigation, methodology, software, validation, visualization, writing—original draft, writing—review and editing; G.E.W.: data curation, investigation, software, writing—review and editing; X.L.: project administration, supervision, writing—review and editing; D.M.B.: conceptualization, funding acquisition, methodology, project administration, resources, supervision, validation, writing—review and editing.

All authors gave final approval for publication and agreed to be held accountable for the work performed therein.

Conflict of interest declaration. We declare we have no competing interests.

Funding. This research was supported in part by the NSF/NIH Joint DMS/NIGMS Mathematical Biology Initiative grant R01GM126559, in part by the NSF Mathematical Biology MODULUS grant 2054085, and in part by the NSF Computing and Communications Foundations grant 1815983. This work also utilized resources from the University of Colorado Boulder Research Computing Group, which is supported by the National Science Foundation (awards ACI-1532235 and ACI-1532236), the University of Colorado Boulder and Colorado State University.

Acknowledgements. The authors wish to thank Prof. Vanja Dukić (Department of Applied Mathematics, University of Colorado, Boulder) for insightful comments about statistical aspects of this work as well as the mathematical form of the directional interaction kernel.

Disclaimer. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Institutes of Health or the National Science Foundation.

Endnotes

[1]See table 5 for a complete list of notations used.

[2]For the artificial data used here, the set of focal cells is the entire population ($N = N_{\text{tot}}$); however, in practice imperfectly tracked cells should be removed from the set of focal cells.

[3]In order to isolate the classification task, in this work we do not examine noisy data other than numerical errors arising from coarse time-sampling.

[4]In some cases, we can use weak-form integration to eliminate this step but we do not pursue this in the current work.

[5]Details on this implementation, in particular $\mathbf{C}$ and $\mathbf{d}$, are given in appendix A.3.

[6]To limit the computational overhead, we only compute these KL divergences to cell i's nearest 200 neighbours in the Euclidean sense.

[7]As explored in [66], in some cases it may be more appropriate to use the coefficient median, or take a weighted average; however, the model replacement step already induces a weighted average. We leave these concerns for future work.

[8]By some abuse of notation, $\bar{\mathcal{M}}(x, v, X, V)$ is used to denote the instantaneous force on a particle (x, v) from neighbouring cells (X, V) under model $\bar{\mathcal{M}}$

[9]See table 7 for simulation wall times.

[10]We discuss this further in the conclusion but leave a complete investigation to future work.

[11]We are not aware of other simulation approaches in the context of interacting particles that use the measurement data for direct calculation of the forces.

[12]In fact each of these parameters was chosen merely to reach a heuristic level of sufficiency. For example, we let $\Delta t'/\Delta t = 2^{-5}$ because $\Delta t'/\Delta t = 2^{-6}$ did not improve simulation accuracy, and we chose $K = 32$ to provide a sufficiently large neighbourhood of cells for model replacement. We did not test whether $\Delta t'/\Delta t = 2^{-5}$ and/or $K = 16$ would also be sufficient, although these would provide computational savings and thus are worth examining in a future work.

[13]In the case of Brownian cells, one modification could be to estimate the strength of the Brownian motion and validate models by averaging over multiple forward simulations of the corresponding stochastic differential equation. We leave this approach to a future work.

Appendix A

A.1. Notation, algorithm hyperparameters and wall times

In table 5, we include notation used throughout the article along with locations in the text where each given symbol first appears and values of hyperparameters used in examples where applicable (last column). While the list of hyperparameters is quite long (rows of table 5 with entries in last column, 18 in total), many can be taken sufficiently large (N, K, $S_{\max}$, n_{gmm}), sufficiently small ($\Delta t'$, $\lambda_{\log}$, $N'_{\min}$), or sufficiently dense ($\boldsymbol{\lambda}$), depending purely on available computational resources.[12] The family of test functions Φ given by (3.5) has been demonstrated to work in a wide range of scenarios [12,62,63], and so may be taken as a fixed

Table 5. Summary of notation used throughout.

symbol	definition	location	value used here
$\mathcal{C}$	set of model clusters based on force modes	§3.3	—
$\overline{\mathcal{C}}$	cluster with the most members	§3.4	—
$(\mathbf{C}, \mathbf{d})$	linear inequality constraint system	§A.3	equation (2.2)–(2.4)
$(\epsilon_{\text{gmm}}, \delta_{\text{gmm}})$	halt classification if $\mathbb{P}(VE < \epsilon_{\text{gmm}}) \geq 1 - \delta_{\text{gmm}}$	§3.7	(0.05, 0.01)
$(f_{\text{a-r}}, f_{\text{align}}, f_{\text{drag}})_{\ell}$	forces obeyed by cells in species ℓ	§§2.2–2.4	—
$f^{(i)}_{\text{force}}$	directional force modes (force ∈ {a−r, align, drag})	§3.3	equation (3.15)
$(\mathbf{G}^{(i)}, \mathbf{b}^{(i)})$	WSINDy linear system for learning model for cell i	equation (3.4)	—
K	number of neighbour cells chosen for model replacement	§3.2	32
L	number of time steps in data	§3.1	—
L'	number of time steps chosen for validation	§3.5	$0.25L$
$\boldsymbol{\lambda}$	set of sparsity thresholds to sweep over	§3.1.3	$(10^{-4}, \ldots, 1)$
λ_{log}	small threshold applied to abs. val of $\overline{\mathbf{w}}$	§3.4	10^{-4}
$\mathcal{M}$	set of single-cell models	§3	—
$\bar{\mathcal{M}}$	model associated with coefficients $\overline{\mathbf{w}}$	§3.4	—
$\bar{\mathcal{M}}(x, v, X, V)$	total force on (x, v) from neighbours (X, V) using model $\bar{\mathcal{M}}$	§3.5	—
N	number of focal cells selected for learning	§3	N_{tot}
N_{tot}	total number of cells in the population	§2	—
N'_{min}	minimum allowable number of cells in a species	§3.7	2
n_{gmm}	number of 2-GMM fits to average over	§A.4	20
p_{nf}	probability used to determine near-field radius r_{nf} for $f_{\text{a-r}}$	§2.2	0.001
$\Phi = \{\phi_q\}_{1 \leq q \leq Q}$	test functions to compute weak time derivatives	equation (3.5)	equations (3.5)
r_{ff}	far-field radius, above which $f_{\text{a-r}}$ is attractive	§2.2	1
s	independent variable for cell speed	§2	—
$\overline{\mathcal{S}}$	species identified as obeying the model $\bar{\mathcal{M}}$	§3.6	—
S_{max}	maximum allowable number of species	§3.7	10
$(\tau, \widehat{\tau})$	test function hyperparameters	§3.1.1	$(10^{-10}, 3)$
θ_{ij}	angle between v_i and $x_i - x_j$	§2.1	—
$\Theta = (\mathcal{F}_{\text{a-r}}, \mathcal{F}_{\text{align}}, \mathcal{F}_{\text{drag}})$	library of force functions for learning	§§3.1.1, 3.1.2	equations (3.6)–(3.9)
Δt	time step of data	§3.1.1	—
$\Delta t'$	time step for validation simulations	§3.5	$2^{-5}\Delta t$
ΔV_i	validation error of cell i	equation (3.19)	equation (3.19)
VE	set of validation errors	§3.5	—
$\mathbf{w}^{\star}$	true model coefficients	§3.1.1	—
$\widehat{\mathbf{w}}^{(i)}$	learned model coefficients for cell i	equation (3.10)	—
$\overline{\mathbf{w}}$	coefficients obtained from averaging the models in cluster	equation (3.16)	—
(X, V)	cell population position and velocity	§2	—
(x_i, v_i)	position and velocity of cell i	§2	—
$(\mathbf{X}, \mathbf{V})$	position and velocity time-series data	§3.1	—
$(\mathbf{x}_i, \mathbf{v}_i)$	position and velocity of cell i in time-series data	§3.1	—
$(\overline{\mathbf{x}}_i, \overline{\mathbf{v}}_i)$	cell i validation data simulated with model $\bar{\mathcal{M}}$	equations (3.17), (3.18)	—
$(\mathbf{X}'^i, \mathbf{V}'^i)$	cell data with ith cell removed	§3.5	—

Table 6. Hyperparameters used for each example.

experiment	m	p	r_{nf}	r_{ff}
$\mathbf{X}_A$	38	8	0.0497	1
$\mathbf{X}_B$	38	8	0.0365	1
$\mathbf{X}_C$	32	9	0.0219	1
$\mathbf{X}_{A,C}$	35	9	0.0494	1
$\mathbf{X}_{B,C}$	35	9	0.0499	1
$\mathbf{X}_{A,B}$	38	8	0.0365	1
$\mathbf{X}_{A,B}$(long)	31	9	0.0219	1
$\mathbf{X}_{A,B,C}$	38	8	0.0569	1
$\mathbf{X}_{A,B,C}$(long)	31	9	0.0253	1

Table 7. Wall times for main components of one iteration of the algorithm, recorded for the $\mathbf{X}_{A,B,C}$(long) experiment ($N_{tot} = 1000$ cells and $L = 400$ time points). Each component contains an inner iteration which may be trivially parallelized. The reported times are for one step of the respective inner iteration. Specifically, it takes 5–10 s to learn each model $\mathcal{M}_i$, while computation time for the full set of models $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_N\}$ depends on the number of available CPUs. To form the model clusters $\mathcal{C} = \{\mathcal{C}_1, \ldots \mathcal{C}_r\}$, find the largest cluster $\overline{\mathcal{C}}$, and compute the averaged model $\bar{\mathcal{M}}$, forward simulations of each model $\mathcal{M}_i$ (see §3.2) are performed which take 10–30 s (depending on the complexity of $\mathcal{M}_i$). Similarly, computation of each ΔV_i requires one forward simulation (10–30 s). Lastly, to identify $\overline{\mathcal{S}}$, it takes less than 1 second to perform GMM classification of $\log_{10}$(VE), and the results of 20 rounds of classification are averaged. If full parallelization is available, the wall time is less than 2 min per outer iteration, or less than $2S$ minutes in total, where S is the number of identified species. For comparison, the cost of generating the data for $\mathbf{X}_{A,B,C}$(long) takes 5–6 h.

$\mathbf{x}_i \to \mathcal{M}_i$	$\mathcal{M} \to (\overline{\mathcal{C}},$	$(\mathbf{x}_i,$	$VE \to \overline{\mathcal{S}}$
5–10 s	10–30 s	10–30 s	<1 s

hyperparameter with minor tuning of the test function hyperparameters $(\tau, \hat{\tau})$, although it is possible that new application areas will require a different choice of test function class.

The remaining hyperparameters are problem-dependent. The force library Θ, along with constraints ($\mathbf{C}$, $\mathbf{d}$, which depend on p_{nf}, r_{ff}) may be altered to include additional knowledge of the possible force modes present along with their constraints. The modes $f^{(i)}_{force}$ over which models are partitioned (taken to be based on directional dependence in this work, see equation (3.15)) could be subdivided in other ways, as discussed in §3.3. The length L' of time points over which to validate model simulations, along with the error function used to define the validation error ΔV_i (equation (3.19)) and hyperparameters related to the GMM fitting process (ϵ_{gmm}, δ_{gmm}), may need tuning in the case of severely noisy data or Brownian cells, or chaotic trajectories, where forward simulations may not remain accurate for long.[13] We conjecture that each of these hyperparameters may be learned from the given dataset (plus available domain knowledge), although we leave derivation of a direct map from data to hyperparameters to future work.

Several hyperparameter choices for the examples above which were not covered in §§3–4 are listed in table 6.

We include wall times for the main components of the algorithm in table 7, recorded in Matlab using an AMD Ryzen 7 pro 4750u processor. Detailed description of the implementation with respect to possible parallelism is included in the figure caption.

A.2. Simulation details

All example data were generated from the exact models using forward Euler with a time step of $\Delta t^* \approx 0.00042$ up until a final time of $T \approx 26$. The time series was then coarsened to a resolution of $\Delta t = 0.13$, resulting in a total of $L = 200$ time points for learning. Experiments $\mathbf{X}_{A,B}$(long) and $\mathbf{X}_{A,B,C}$(long) are simply extensions of $\mathbf{X}_{A,B}$ or $\mathbf{X}_{A,B,C}$ by an additional 200 time points at the resolution $\Delta t = 0.13$. Initial positions were generated using Latin hypercube sampling in a box of side length 2. Initial velocities were drawn i.i.d. from a Gaussian distribution, where for $\mathbf{X}_{A,B,C}$, $\mathbf{X}_{A,B,C}$(long) and $\mathbf{X}_C$ we chose a mean velocity of $\bar{v} = (0, 0)$ and covariance $\Sigma = 0.0025\mathbf{I}_2$, and for all other experiments we used $\bar{v} = (-0.02, 0.035)$, $\Sigma = \begin{bmatrix} 0.0014 & 0.0005 \\ 0.0005 & 0.0012 \end{bmatrix}$. The two sets of initial conditions represent different migratory stages, correlated or uncorrelated motion. See figures 13 and 14 for initial and final (time step 200) states of each experiment.

In figures 15 and 16, we include statistical information for species A, B and C in homogeneous experiments as well as the heterogeneous experiment $\mathbf{X}_{A,B,C}$(long). In each figure, the top row contains averages of the distributions used to select validation cells in the model replacement step (equations (3.12)–(3.14)). The bottom rows contains average polarization and angular momentum, respectively, defined as

$$P(t) = \left| \frac{\sum_{i \in I_S} v_i(t)}{\sum_{i \in I_S} |v_i(t)|} \right| \quad \text{and}$$
$$M_{ang}(t) = \left| \frac{\sum_{i \in I_S} r_i(t) \times v_i(t)}{\sum_{i \in I_S} |r_i(t)||v_i(t)|} \right| \qquad \text{(A 1)}$$

where I_S represents all cells in species $S \in \{A, B, C\}$ and $r_i(t) = x_i(t) - x_c(t)$ where x_c is the center of mass of the entire population. The order parameters $P(t)$ and $M_{ang}(t)$ are used to characterize phenotypes such as milling and spreading behaviour (e.g. [72]); however, they do not appear to indicate strong agreement with either phenotype in our datasets. Species A and B have nearly identical speed distributions (top right of each figure), in contrast to species C, with cells travelling slower and exhibiting stronger alignment (top middle of each figure). Notice that the algorithm is able to distinguish between A and B cells despite overall statistical similarity between the two species.

A.3. Constrained sparse regression

The constrained sequential thresholding algorithm requires solving at each thresholding iteration ℓ a linearly constrained quadratic program of the form

$$\mathbf{w}^{(\ell+1)} = \underset{\substack{\text{ws.t.}\mathbf{Cw} \le \mathbf{d} \\ \text{supp}(\mathbf{w}) \subset I^{(\ell)}}}{\arg\min} \|\mathbf{Gw} - \mathbf{b}_2^2\| \qquad \text{(A 2)}$$

where $\mathcal{I}^{(\ell)}$ is the set of coefficients of $\mathbf{w}^{(\ell)}$ satisfying (3.11). The constraint system $\mathbf{Cw} \le \mathbf{d}$ has the following four components.

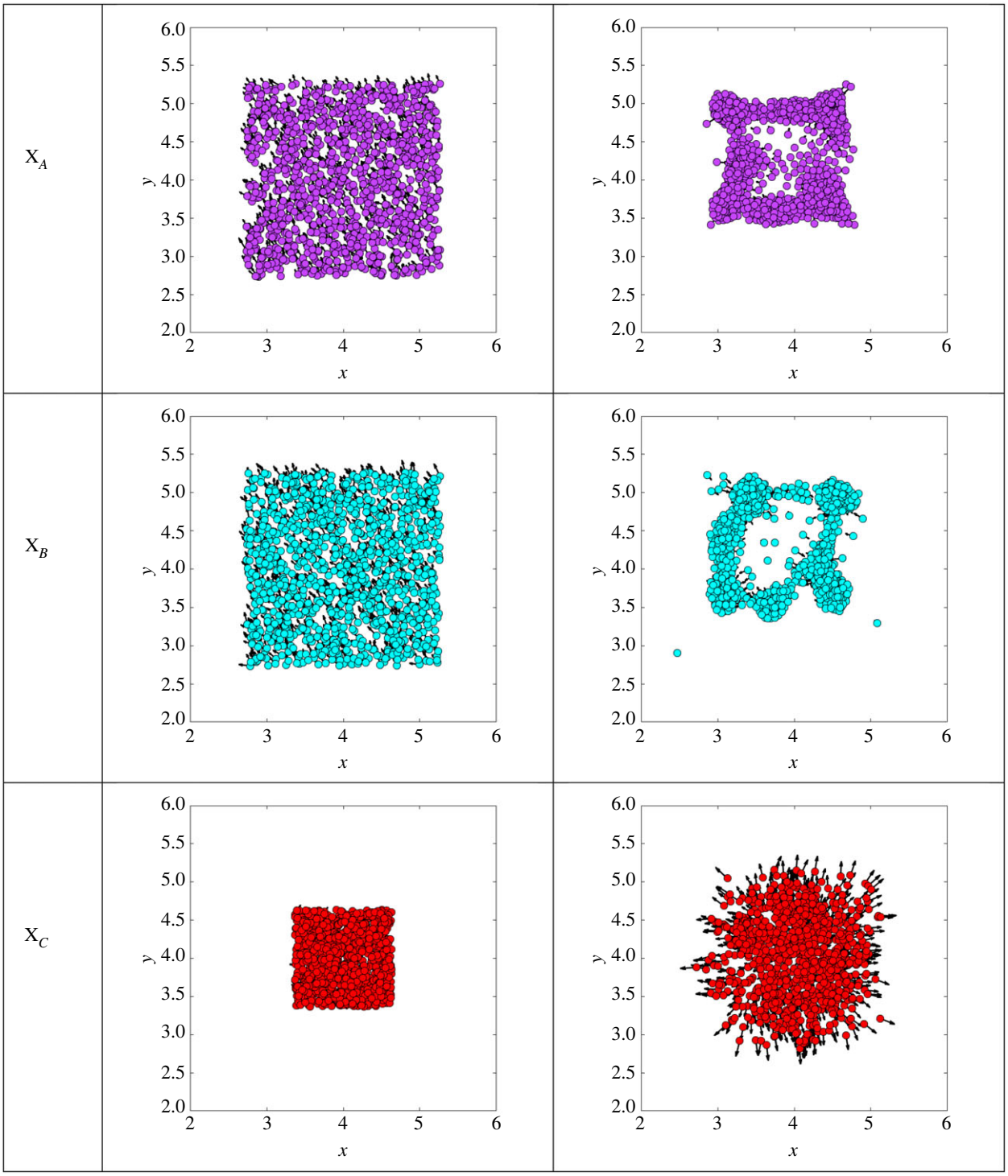


Figure 13. Plots of artificial data for homogeneous experiments at initial (left) and final (right) times.

(i) $f_{a-r} \geq 0$ when $0 \leq r < r_{nf}$: for the near-field repulsion of f_{a-r} we discretize the region $\{(r, \theta) : 0 \leq r < r_{nf},\ \theta \in [0, 2\pi)\}$ choosing five equally spaced points in r from 10^{-6} to r_{nf} and five equally spaced points in θ from 0 to π. Evaluating each of the basis function for f_{a-r} at this grid results in a constraint system $\mathbf{C}_{a-r,nf}\mathbf{w}_{a-r} \leq \mathbf{0}$ of dimension $25 \times J_{a-r}$ where J_{a-r} is the number of basis functions used to approximate f_{a-r} and $\mathbf{w}_{a-r}$ is the restriction of $\mathbf{w}$ to coefficients of f_{a-r}.

(ii) $f_{a-r} \leq 0$ when $r \geq r_{ff}$: similarly for the far-field region, we choose 10 equally spaced points in r from r_{ff} to r_{max}, where r_{max} is the maximum observed neighbour–neighbour distance in the simulation, and θ over the same points as previous. This results in a constraint system $\mathbf{C}_{a-r,ff}\mathbf{w}_{a-r} \leq \mathbf{0}$ of dimensions $50 \times J_{a-r}$.

(iii) $f_{align} \leq 0$: since the basis is positive, the constrain system here is simply $\mathbf{I}_{J_{align}}\mathbf{w}_{align} \leq \mathbf{0}$, where $\mathbf{I}_n$ indicates the identity on $\mathbb{R}^n$.

(iv) $f_{drag} \leq 0$: similarly the basis is positive, so the constraint system is $\mathbf{I}_{J_{drag}}\mathbf{w}_{drag} \leq \mathbf{0}$.

Altogether we get $\mathbf{d} = \mathbf{0}$ and

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{a-r,nf} & 0 & 0 \\ \mathbf{C}_{a-r,ff} & 0 & 0 \\ 0 & \mathbf{I}_{J_{align}} & 0 \\ 0 & 0 & \mathbf{I}_{J_{drag}} \end{bmatrix}$$

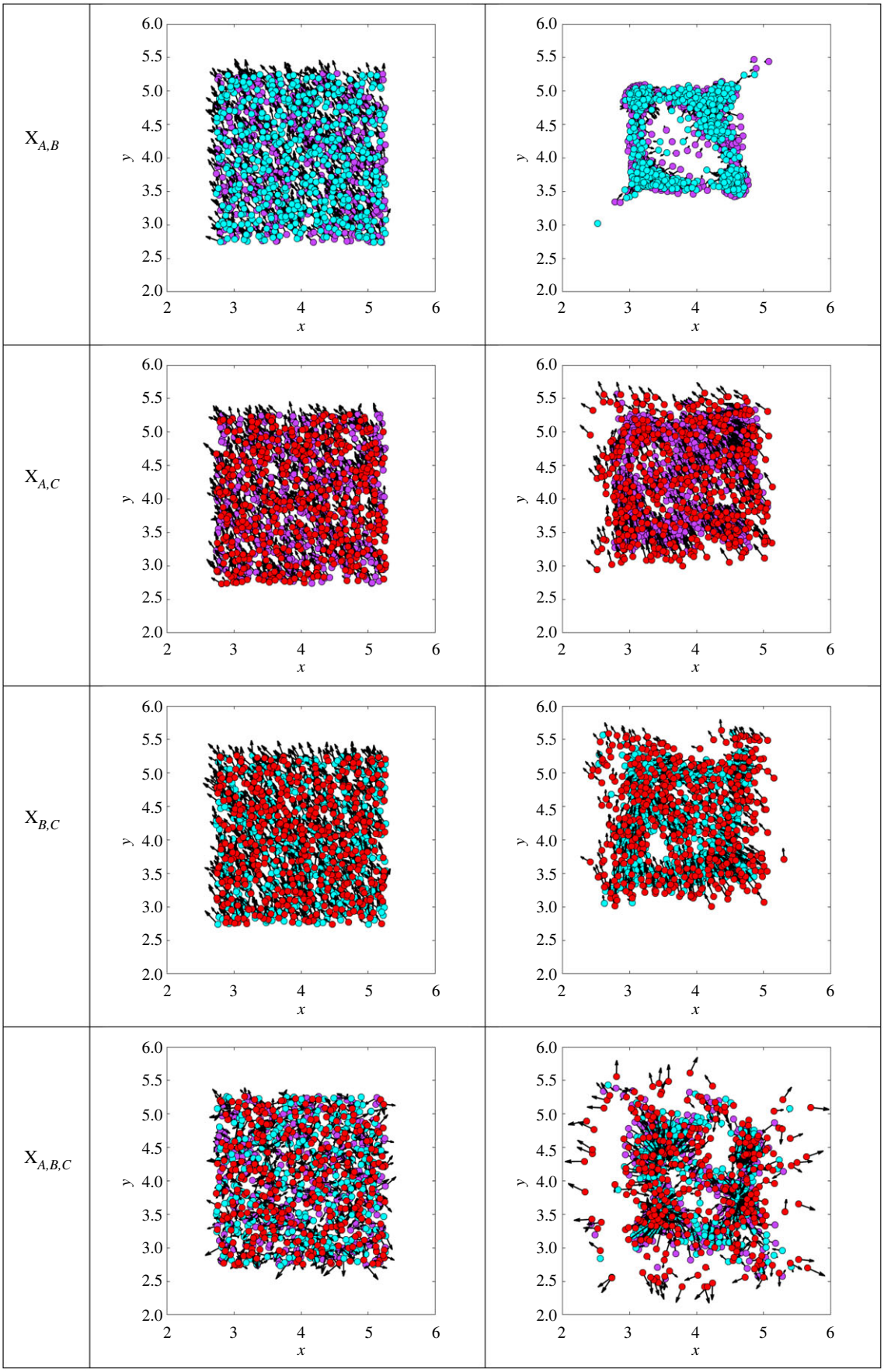


Figure 14. Similar to figure 13 but for heterogeneous experiments. Species are colour-coded as in figure 13.

We use Matlab's `quadprog` with constraint tolerance 10^{-10} and maximum iterations set to 1000. Note that since $\mathbf{d} = \mathbf{0}$, we do not lose feasibility during the thresholding step, which is possible in general. However, it is possible to arrive at the zero solution. This further necessitates the parameter sweep over λ values to select an appropriate threshold.

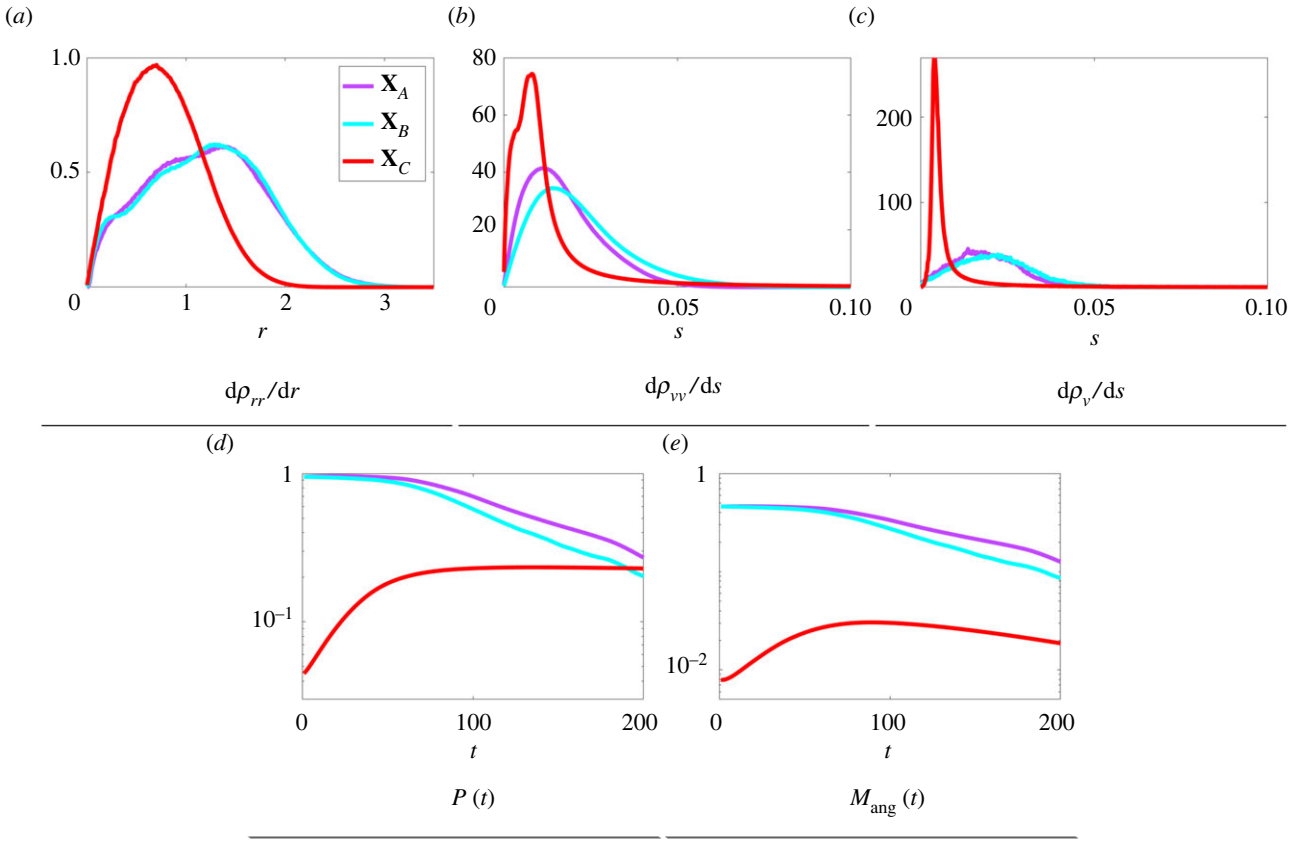


Figure 15. Statistical information for members of species *A*, *B* and *C* within the homogeneous experiments $\mathbf{X}_A$, $\mathbf{X}_B$, $\mathbf{X}_C$. Top row: pairwise distance density, pairwise velocity density and speed density (averages of distributions in (3.12)–(3.14) used for model replacement). Bottom row: average polarization and angular momentum over time (equation (A1)).

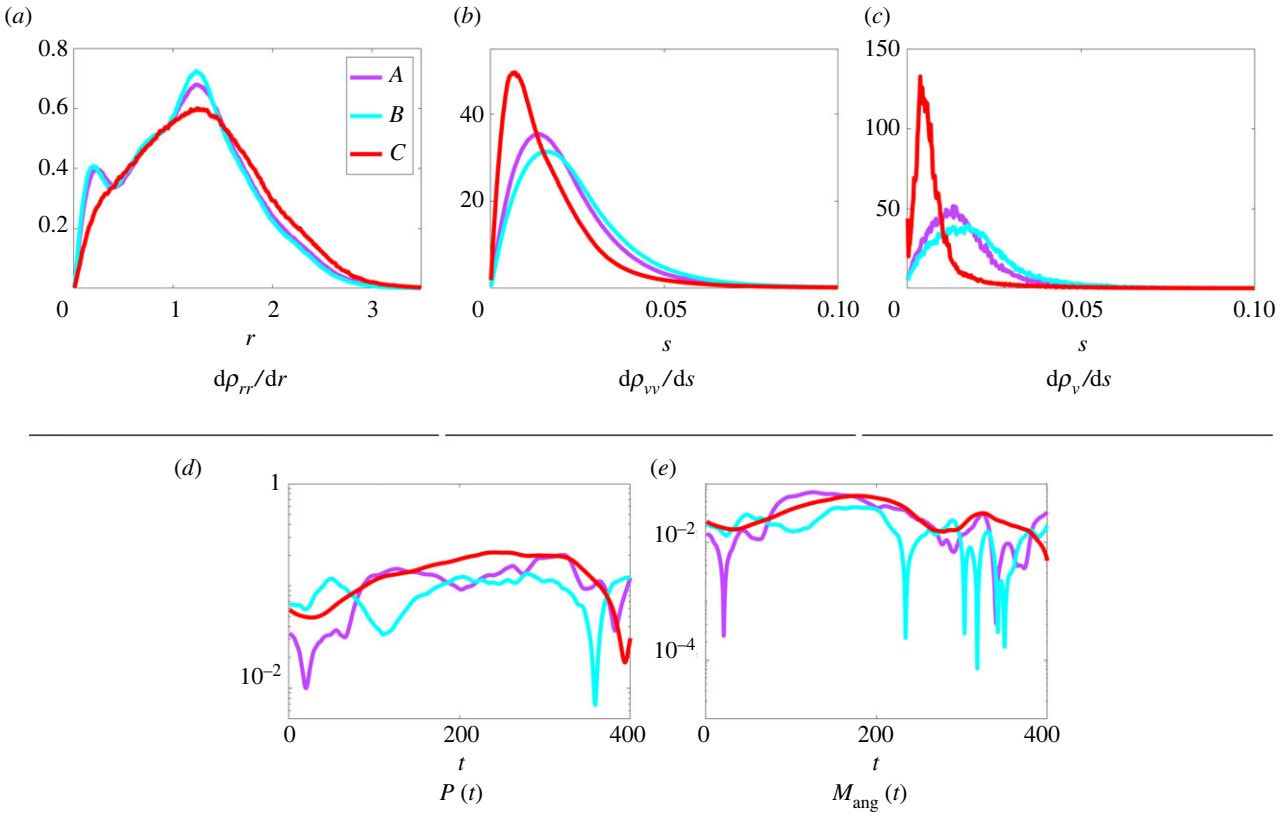


Figure 16. Similar to figure 15, only for members of species *A*, *B* and *C* within the single experiment $\mathbf{X}_{A,B,C}$(long). In the case of pairwise distributions ρ_{rr} and ρ_{vv}, the average is taken over pairs $\{(x_i, v_i), (x_j, v_j)\}$ where *i* ranges over only the species in question and *j* ranges over the entire population.

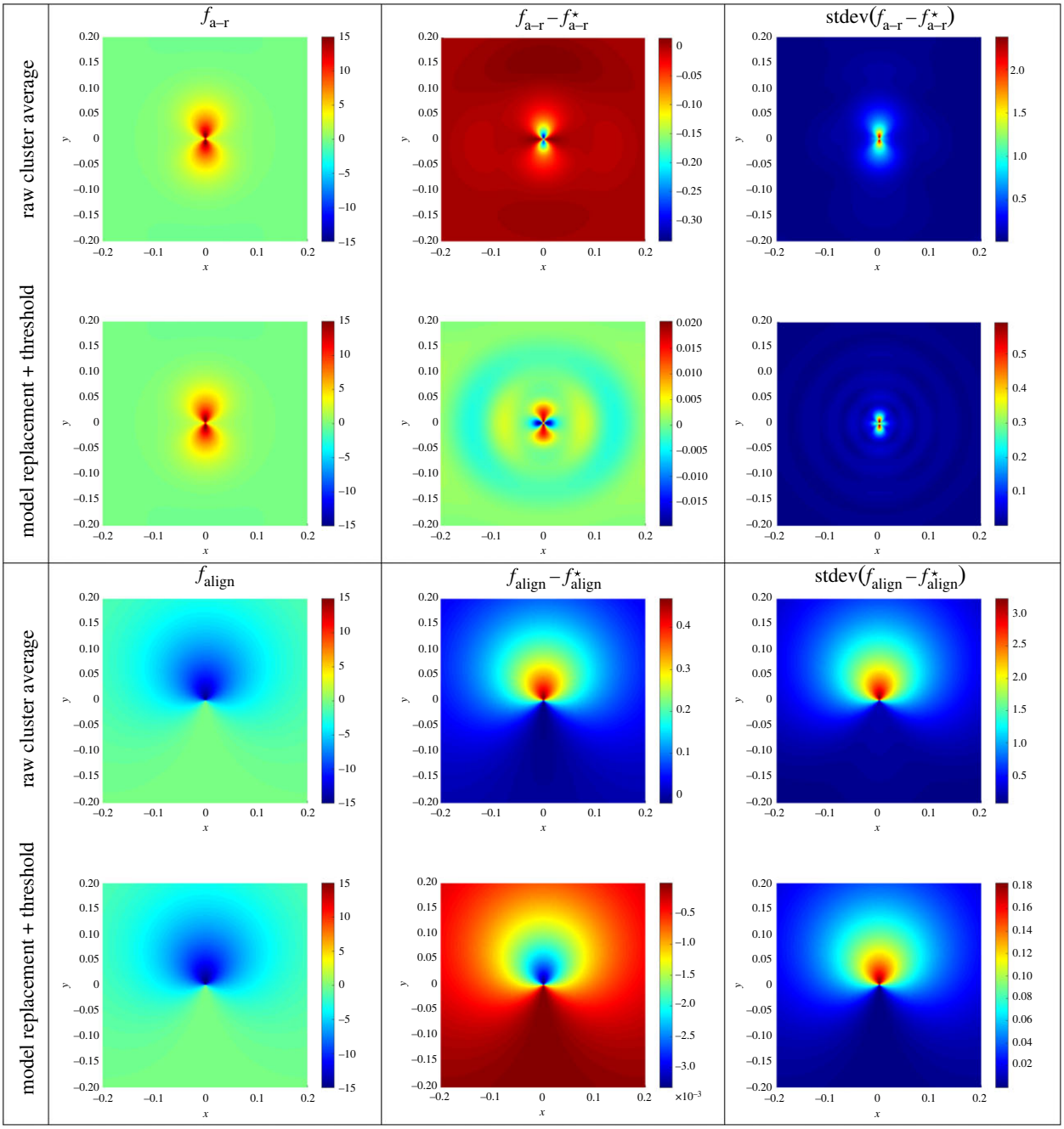


Figure 17. Statistical comparison of learned forces f_{a-r} and f_{align} from cluster 3 of the $\mathbf{X}_{A,B,C}$(long) data with and without the model replacement step followed by a final round of thresholding as in equation (3.16). Left to right: learned force, difference between true and learned force, and pointwise standard deviation of the individual models. To compute standard deviations for the model replacement + threshold case, individual models are modified to have the same coefficient support as the aggregate coefficients $\overline{\mathbf{w}}$. While each learned force agrees well with the true force in the 'eyeball norm' (left column, compare with true forces in figure 2), it is clear from the middle and right columns that model replacement and a final thresholding step significantly reduce pointwise error and variance.

A.4. Gaussian mixture model classification

Since the Gaussian mixture model (GMM) fitting is performed using the expectation–maximization algorithm with random conditions, we perform the GMM fitting for $n_{gmm} = 20$ trials and identify $\overline{\mathcal{S}}$ as the cells that in more than half of the trials appear in the mixture with lowest error.

At some stage in the algorithm, all the remaining cells will be homogeneous. In this case, a two-species Gaussian is the wrong model. To account for this, we do an initial fit to a single Gaussian and compute its Bayesian information criterion (BIC). We accept the two-mixture GMM if the average BIC of all 20 trial GMM fits with two mixtures is lower than that of the single Gaussian.

A.5. Visualization tools

In the repository https://github.com/MathBioCU/WSINDy_CellCluster.git we include the following scripts for visualization. (Note that the data can be downloaded at https://doi.org/10.5281/zenodo.6968448.)

— plot_true_forces.m: plots true forces, as in figure 2.

— visualize_trajectories.m: plots learned trajectories versus true trajectories, as in figures 8–12.
— plot_modelForce_stats.m: plots statistics related to individual models, as in figure 17.
— plot_gmm_script.m: plots Gaussian mixture models approximating the log-validation errors, as in figures 4–6.
— plot_individual_models.m: view subset of individual model forces (not included in the figures here).

References

1. Reynolds CW. 1987 Flocks, herds and schools: a distributed behavioral model. In *Proc. of the 14th Annual Conf. on Computer graphics and Interactive Techniques*, pp. 25–34. New York, NY: Association for Computing Machinery.
2. Levine H, Rappel W-J, Cohen I. 2000 Self-organization in systems of self-propelled particles. *Phys. Rev. E* **63**, 017101. (doi:10.1103/PhysRevE.63.017101)
3. Cucker F, Smale S. 2007 Emergent behavior in flocks. *IEEE Trans. Autom. Control* **52**, 852–862. (doi:10.1109/TAC.2007.895842)
4. Mogilner A, Edelstein-Keshet L. 1999 A non-local model for a swarm. *J. Math. Biol.* **38**, 534–570. (doi:10.1007/s002850050158)
5. Toner J, Tu Y. 1998 Flocks, herds, and schools: a quantitative theory of flocking. *Phys. Rev. E* **58**, 4828–4858. (doi:10.1103/PhysRevE.58.4828)
6. Carrillo JA, Choi Y-P. 2021 Mean-field limits: from particle descriptions to macroscopic equations. *Arch. Ration. Mech. Anal.* **241**, 1529–1573. (doi:10.1007/s00205-021-01676-x)
7. Eriksson A, Jacobi MN, Nyström J, Tunstrøm K. 2010 Determining interaction rules in animal swarms. *Behav. Ecol.* **21**, 1106–1111. (doi:10.1093/beheco/arq118)
8. Katz Y, Tunstrøm K, Ioannou CC, Huepe C, Couzin ID. 2011 Inferring the structure and dynamics of interactions in schooling fish. *Proc. Natl Acad. Sci. USA* **108**, 18 720–18 725. (doi:10.1073/pnas.1107583108)
9. Lukeman R, Li Y-X, Edelstein-Keshet L. 2010 Inferring individual rules from collective behavior. *Proc. Natl Acad. Sci. USA* **107**, 12 576–12 580. (doi:10.1073/pnas.1001763107)
10. Lu F, Maggioni M, Tang S. 2021 Learning interaction kernels in heterogeneous systems of agents from multiple trajectories. *J. Mach. Learn. Res.* **22**, 1013–1067. (doi:10.1007/s10208-021-09521-z)
11. Supekar R, Song B, Hastewell A, Mietke A, Dunkel Jörn. 2021 Learning hydrodynamic equations for active matter from particle simulations and experiments. (https://arxiv.org/abs/2101.06568)
12. Messenger DA, Bortz DM. 2022 Learning mean-field equations from particle data using WSINDy. *Physica D* **439**, 133406. (doi:10.1016/j.physd.2022.133406)
13. Pilkiewicz KR *et al.* 2020 Decoding collective communications using information theory tools. *J. R. Soc. Interface* **17**, 20190563. (doi:10.1098/rsif.2019.0563)
14. Brückner DB, Arlt N, Fink A, Ronceray P, Rädler JO, Broedersz CP. 2021 Learning the dynamics of cell–cell interactions in confined cell migration. *Proc. Natl Acad. Sci. USA* **118**, e2016602118. (doi:10.1073/pnas.2016602118)
15. Brückner DB, Ronceray P, Broedersz CP. 2020 Inferring the dynamics of underdamped stochastic systems. *Phys. Rev. Lett.* **125**, 058103. (doi:10.1103/PhysRevLett.125.058103)
16. Feng J, Ren Y, Tang S. 2021 Data-driven discovery of interacting particle systems using Gaussian processes. (https://arxiv.org/abs/2106.02735)
17. Ballerini M *et al.* 2008 Interaction ruling animal collective behavior depends on topological rather than metric distance: evidence from a field study. *Proc. Natl Acad. Sci. USA* **105**, 1232–1237. (doi:10.1073/pnas.0711437105)
18. Schaerf TM, Herbert-Read JE, Ward AJW. 2021 A statistical method for identifying different rules of interaction between individuals in moving animal groups. *J. R. Soc. Interface* **18**, 20200925. (doi:10.1098/rsif.2020.0925)
19. Sumpter DJT, Szorkovszky A, Kotrschal A, Kolm N, Herbert-Read JE. 2018 Using activity and sociability to characterize collective motion. *Phil. Trans. R. Soc. B* **373**, 20170015. (doi:10.1098/rstb.2017.0015)
20. Akaike H. 1974 A new look at the statistical model identification. *IEEE Trans. Autom. Control* **19**, 716–723. (doi:10.1109/TAC.1974.1100705)
21. Ward AJW, Schaerf TM, Herbert-Read JE, Morrell L, Sumpter DJT, Webster MM. 2017 Local interactions and global properties of wild, free-ranging stickleback shoals. *R. Soc. Open Sci.* **4**, 170043. (doi:10.1098/rsos.170043)
22. Tunstrøm K, Katz Y, Ioannou CC, Huepe C, Lutz MJ, Couzin ID. 2013 Collective states, multistability and transitional behavior in schooling fish. *PLoS Comput. Biol.* **9**, 1002915. (doi:10.1371/journal.pcbi.1002915)
23. Chen D, Xu B, Zhu T, Zhou T, Zhang H-T. 2017 Anisotropic interaction rules in circular motions of pigeon flocks: an empirical study based on sparse Bayesian learning. *Phys. Rev. E* **96**, 022411. (doi:10.1103/PhysRevE.96.022411)
24. Paranjape AA, Chung S-J, Kim K, Shim DH. 2018 Robotic herding of a flock of birds using an unmanned aerial vehicle. *IEEE Trans. Rob.* **34**, 901–915. (doi:10.1109/TRO.2018.2853610)
25. Warren WH. 2018 Collective motion in human crowds. *Curr. Dir. Psychol. Sci.* **27**, 232–240. (doi:10.1177/0963721417746743)
26. Mudaliar RK, Schaerf TM. 2020 Examination of an averaging method for estimating repulsion and attraction interactions in moving groups. *PLoS ONE* **15**, e0243631. (doi:10.1371/journal.pone.0243631)
27. Escobedo R, Lecheval V, Papaspyros V, Bonnet F, Mondada F, Sire C, Theraulaz G. 2020 A data-driven method for reconstructing and modelling social interactions in moving animal groups. *Phil. Trans. R. Soc. B* **375**, 20190380. (doi:10.1098/rstb.2019.0380)
28. Lagergren JH, Nardini JT, Lavigne GM, Rutter EM, Flores KB. 2020 Learning partial differential equations for biological transport models from noisy spatio-temporal data. *Proc. R. Soc. A* **476**, 20190800. (doi:10.1098/rspa.2019.0800)
29. Nardini JT, Baker RE, Simpson MJ, Flores KB. 2021 Learning differential equation models from stochastic agent-based model simulations. *J. R. Soc. Interface* **18**, 20200987. (doi:10.1098/rsif.2020.0987)
30. Schmidt M, Lipson H. 2009 Distilling free-form natural laws from experimental data. *Science* **324**, 81–85. (doi:10.1126/science.1165893)
31. Rudy SH, Brunton SL, Proctor JL, Kutz JN. 2017 Data-driven discovery of partial differential equations. *Sci. Adv.* **3**, e1602614. (doi:10.1126/sciadv.1602614)
32. Brunton SL, Proctor JL, Kutz JN. 2016 Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **113**, 3932–3937. (doi:10.1073/pnas.1517384113)
33. Schaeffer H. 2017 Learning partial differential equations via data discovery and sparse optimization. *Proc. R. Soc. A* **473**, 20160446. (doi:10.1098/rspa.2016.0446)
34. Udrescu S-M, Tegmark M. 2020 AI Feynman: a physics-inspired method for symbolic regression. *Sci. Adv.* **6**, eaay2631. (doi:10.1126/sciadv.aay2631)
35. Long Z, Lu Y, Ma X, Dong B. 2018 Pde-net: learning pdes from data. In *Int. Conf. on Machine Learning*, pp. 3208–3216. PMLR.
36. Champion K, Lusch B, Kutz JN, Brunton SL. 2019 Data-driven discovery of coordinates and governing equations. *Proc. Natl Acad. Sci. USA* **116**, 22 445–22 451. (doi:10.1073/pnas.1906995116)
37. Tang DG. 2012 Understanding cancer stem cell heterogeneity and plasticity. *Cell Res.* **22**, 457–472. (doi:10.1038/cr.2012.13)
38. Sarkar D, Gompper G, Elgeti J. 2021 A minimal model for structure, dynamics, and tension of monolayered cell colonies. *Commun. Phys.* **4**, 1–8. (doi:10.1038/s42005-020-00515-x)
39. Holmes WR, Park J, Levchenko A, Edelstein-Keshet L. 2017 A mathematical model coupling polarity signaling to cell adhesion explains diverse cell migration patterns. *PLoS Comput. Biol.* **13**, e1005524. (doi:10.1371/journal.pcbi.1005524)
40. Notbohm J *et al.* 2016 Cellular contraction and polarization drive collective cellular motion. *Biophys. J.* **110**, 2729–2738. (doi:10.1016/j.bpj.2016.05.019)
41. Eftimie R, de Vries G. 2007 Complex spatial group patterns result from different animal communication mechanisms. *Proc. Natl Acad. Sci.*

USA **104**, 6974–6979. (doi:10.1073/pnas.0611483104)

42. Bernardi S, Eftimie R, Painter KJ. 2021 Leadership through influence: what mechanisms allow leaders to steer a swarm? *Bull. Math. Biol.* **83**, 69. (doi:10.1007/s11538-021-00901-8)
43. Zmurchok C, de Vries G. 2018 Direction-dependent interaction rules enrich pattern formation in an individual-based model of collective behavior. *PLoS ONE* **13**, e0198550. (doi:10.1371/journal.pone.0198550)
44. Seeley TD, Visscher PK, Passino KM. 2006 Group decision making in honey bee swarms: when 10 000 bees go house hunting, how do they cooperatively choose their new nesting site? *Am. Sci.* **94**, 220–229. (doi:10.1511/2006.59.220)
45. Engelbrecht AP. 2010 Heterogeneous particle swarm optimization. In *Int. Conf. on Swarm Intelligence*, pp. 191–202. Berlin, Germany: Springer.
46. Kengyel D, Hamann H, Zahadat P, Radspieler G, Wotawa F, Schmickl T. 2015 Potential of heterogeneity in collective behaviors: a case study on heterogeneous swarms. In *Int. Conf. on Principles and Practice of Multi-Agent Systems*, pp. 201–217. Berlin, Germany: Springer.
47. Chapnick DA, Liu X. 2014 Leader cell positioning drives wound-directed collective migration in TGFβ-stimulated epithelial sheets. *Mol. Biol. Cell* **25**, 1586–1593. (doi:10.1091/mbc.e14-01-0697)
48. Vishwakarma M, Di Russo J, Probst D, Schwarz US, Das T, Spatz JP. 2018 Mechanical interactions among followers determine the emergence of leaders in migrating epithelial cell collectives. *Nat. Commun.* **9**, 1–12. (doi:10.1038/s41467-018-05927-6)
49. Lan T *et al.* 2021 Decomposition of cell activities revealing the role of the cell cycle in driving biofunctional heterogeneity. *Sci. Rep.* **11**, 1–15.
50. Bonneton C, Sibarita J-B, Thiery J-P. 1999 Relationship between cell migration and cell cycle during the initiation of epithelial to fibroblastoid transition. *Cell Motil. Cytoskeleton* **43**, 288–295. (doi:10.1002/(SICI)1097-0169(1999)43:4<288::AID-CM2>3.0.CO;2-Y)
51. Boehm M, Nabel EG. 2001 Cell cycle and cell migration: new pieces to the puzzle. *Circulation* **103**, 2879–2881. (doi:10.1161/01.CIR.103.24.2879)
52. Nardini JT, Chapnick DA, Liu X, Bortz DM. 2016 Modeling keratinocyte wound healing dynamics: cell–cell adhesion promotes sustained collective migration. *J. Theor. Biol.* **400**, 103–117. (doi:10.1016/j.jtbi.2016.04.015)
53. Schumacher LJ, Maini PK, Baker RE. 2017 Semblance of heterogeneity in collective cell migration. *Cell Systems* **5**, 119–127.e1. (doi:10.1016/j.cels.2017.06.006)
54. Haeger A, Wolf K, Zegers MM, Friedl P. 2015 Collective cell migration: guidance principles and hierarchies. *Trends Cell Biol.* **25**, 556–566. (doi:10.1016/j.tcb.2015.06.003)
55. Zhong M, Miller J, Maggioni M. 2020 Data-driven discovery of emergent behaviors in collective dynamics. *Physica D* **411**, 132542. (doi:10.1016/j.physd.2020.132542)
56. Cai AQ, Landman KA, Hughes BD. 2006 Modelling directional guidance and motility regulation in cell migration. *Bull. Math. Biol.* **68**, 25–52. (doi:10.1007/s11538-005-9028-x)
57. Carrillo JA, Fornasier M, Toscani G, Vecil F. 2010 Particle, kinetic, and hydrodynamic models of swarming. In *Mathematical modeling of collective behavior in socio-economic and life sciences* (eds G Naldi, L Pareschi, G Toscani), pp. 297–336. Berlin, Germany: Springer.
58. Evers JHM, Fetecau RC, Ryzhik L. 2015 Anisotropic interactions in a first-order aggregation model. *Nonlinearity* **28**, 2847–2871. (doi:10.1088/0951-7715/28/8/2847)
59. Browning AP, Jin W, Plank MJ, Simpson MJ. 2020 Identifying density-dependent interactions in collective cell behaviour. *J. R. Soc. Interface* **17**, 20200143. (doi:10.1098/rsif.2020.0143)
60. Ramaswamy S. 2010 The mechanics and statistics of active matter. *Annu. Rev. Condens. Matter Phys.* **1**, 323–345. (doi:10.1146/annurev-conmatphys-070909-104101)
61. Vicsek T, Zafeiris A. 2012 Collective motion. *Phys. Rep.* **517**, 71–140. (doi:10.1016/j.physrep.2012.03.004)
62. Messenger DA, Bortz DM. 2021 Weak SINDy: Galerkin-based data-driven model selection. *Multiscale Model. Simul.* **19**, 1474–1497. (doi:10.1137/20M1343166)
63. Messenger DA, Bortz DM. 2021 Weak SINDy for partial differential equations. *J. Comput. Phys.* **443**, 110525. (doi:10.1016/j.jcp.2021.110525)
64. Messenger DA, Dall'Anese E, Bortz DM. In press. Online weak-form sparse identification of partial differential equations. In *Proc. of the 3rd Mathematical and Scientific Machine Learning Conf.* Proceedings of Machine Learning Research.
65. Bhat HS. 2020 Learning and interpreting potentials for classical Hamiltonian systems. In *Machine learning and knowledge discovery in databases*, pp. 217–228. Berlin, Germany: Springer International Publishing.
66. Fasel U, Kutz JN, Brunton BW, Brunton SL. 2022 Ensemble-SINDy: robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proc. R. Soc. Math. Phys. Eng. Sci.* **478**, 20210904. (doi:10.1098/rspa.2021.0904)
67. Topaz CM, Bertozzi AL. 2004 Swarming patterns in a two-dimensional kinematic model for biological groups. *SIAM J. Appl. Math.* **65**, 152–174. (doi:10.1137/S0036139903437424)
68. Fetecau RC, Huang Y, Kolokolnikov T. 2011 Swarm dynamics and equilibria for a nonlocal aggregation model. *Nonlinearity* **24**, 2681–2716. (doi:10.1088/0951-7715/24/10/002)
69. Tambe DT *et al.* 2011 Collective cell guidance by cooperative intercellular forces. *Nat. Mater.* **10**, 469–475. (doi:10.1038/nmat3025)
70. Knowles I, Renka RJ. 2014 Methods for numerical differentiation of noisy data. *Electron. J. Differ. Equ.* **21**, 235–246.
71. Van Breugel F, Kutz JN, Brunton BW. 2020 Numerical differentiation of noisy data: a unifying multi-objective optimization framework. *IEEE Access* **8**, 196 865–196 877. (doi:10.1109/ACCESS.2020.3034077)
72. Bhaskar D, Manhart A, Milzman J, Nardini JT, Storey KM, Topaz CM, Ziegelmeier L. 2019 Analyzing collective motion with machine learning and topology. *Chaos* **29**, 123125. (doi:10.1063/1.5125493)