Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **480,** 371–380 (2018)
Advance Access publication 2018 July 6

doi:10.1093/mnras/sty1788

Modelling the structure of star clusters with fractional Brownian motion

O. Lomax,★ M. L. Bates and A. P. Whitworth
School of Physics and Astronomy, Cardiff University, Cardiff CF24 3AA, UK

Accepted 2018 June 28. Received 2018 June 5; in original form 2018 April 18

ABSTRACT

The degree of fractal substructure in molecular clouds can be quantified by comparing them with fractional Brownian motion (FBM) surfaces or volumes. These fields are self-similar over all length-scales and characterized by a drift exponent H, which describes the structural roughness. Given that the structure of molecular clouds and the initial structure of star clusters are almost certainly linked, it would be advantageous to also apply this analysis to clusters. Currently, the structure of star clusters is often quantified by applying $\mathcal{Q}$ analysis. $\mathcal{Q}$ values from observed targets are interpreted by comparing them with those from artificial clusters. These are typically generated using a box-fractal (BF) or radial density profile (RDP) model. We present a single cluster model, based on FBM, as an alternative to these models. Here, the structure is parametrized by H and the standard deviation of the log-surface/volume density σ. The FBM model is able to reproduce both centrally concentrated and substructured clusters, and is able to provide a much better match to observations than the BF model. We show that $\mathcal{Q}$ analysis is unable to estimate FBM parameters. Therefore, we develop and train a machine learning algorithm that can estimate values of H and σ, with uncertainties. This provides us with a powerful method for quantifying the structure of star clusters in terms that relate to the structure of molecular clouds. We use the algorithm to estimate the H and σ for several young star clusters, some of which have no measurable BF or RDP analogue.

Key words: methods: data analysis – methods: statistical – stars: formation – stars: statistics – ISM: clouds – galaxies: star clusters: general.

1 INTRODUCTION

Recent space-borne instruments have revealed much of the detailed multiscale structure of our own Galaxy. The Herschel submillimetre observatory (Griffin et al. 2010; Poglitsch et al. 2010) has mapped out many of the gas and dust structures in the interstellar medium (ISM; e.g. Molinari et al. 2010). Similarly, the Gaia observatory (Gaia Collaboration et al. 2016, 2018) continues to reveal the spatial and velocity distribution of the stars which accompany this gas and dust. Nevertheless, understanding the link between the structures in the ISM and star clusters remains an ongoing challenge. We are confident that the earliest stages of stellar evolution occur within dense, substructured (i.e. clumpy or filamentary), molecular clouds within the ISM (e.g. Motte, Andre & Neri 1998; André et al. 2010; Smith et al. 2016; Parker 2018). However, the extent to which star clusters retain the structural signatures of their parent molecular clouds is uncertain. Some studies highlight similarities between the distribution of stars and that of the molecular clouds which spawn them (e.g. Elmegreen & Falgarone 1996; Gouliermis, Hony & Klessen 2014). However, numerical studies suggest that gas and stars decouple quickly during the star formation process, erasing structural similarities (e.g. Bate & Bonnell 2005; Parker & Dale 2015). To make headway in this complex field, we require tools that can fulfil two roles. First, we need statistics that can quantify the structure of clouds and clusters, ideally in the same terms. Secondly, in order to simulate these structures, we need initial conditions that statistically match observations.

Stutzki et al. (1998) note that molecular clouds can be compared with surface-density fields generated by fractional Brownian motion (FBM). These are random fractal structures, with well-defined fractal dimension D, which can be analysed using perimeter-area or Δ-variance techniques (e.g. Falgarone, Phillips & Walker 1991; Stutzki et al. 1998; Williams, Blitz & McKee 2000; Elia et al. 2014). Other studies measure the surface-density probability density functions (PDFs) of molecular clouds (e.g. Federrath & Klessen 2012; Schneider et al. 2013). These can provide a measure of a cloud's surface-density dynamic range, which is not necessarily related to its fractal structure. Indeed, a property of fractal distributions is that the density can be rescaled by any one-to-one transform without altering D (Peitgen & Saupe 1988).

Techniques also exist which estimate the fractal properties of star clusters. Cartwright & Whitworth (2004, hereafter CW04) were the first to use minimum spanning trees (MSTs) to estimate

★ E-mail: oliver.lomax@astro.cf.ac.uk

D for clusters. The application of this method has since become widespread in the field star formation (e.g. Schmeja & Klessen 2006; Cartwright 2009; Cartwright & Whitworth 2009; Lomax, Whitworth & Cartwright 2011; Parker et al. 2014; Parker 2018). However, this analysis assumes that substructured clusters can be described by a box-fractal (BF) (Goodwin & Whitworth 2004) or a radial density profile (RDP) model. The BF model is parametrized by D only. Here, altering D also changes the surface-density dynamic range; the two properties cannot be varied independently. A more recent study by Jaffa, Whitworth & Lomax (2017, hereafter JWL17) expands the BF model to include variable surface-density scaling. This model provides a better likeness to observed clusters, at the cost of two additional parameters.

In this paper, we present a method of generating model FBM star clusters. This provides a parametrization of cluster structure, which matches that of clouds. We demonstrate that BF clusters do not always match observations, and, therefore, should not be used to infer quantitative results. We show that FBM clusters overcome this problem, and we use machine learning to estimate the structural parameters of test clusters and observations. In Section 2 of this paper, we define different star cluster models. In Section 3, we review parameter estimators and apply them to observations. In Section 4, we compare and discuss the results of the estimators. Finally, we summarize our conclusions in Section 5.

2 MODEL STAR CLUSTERS

Here, we present a method for generating artificial star clusters from FBM density fields. Peitgen & Saupe (1988) provide multiple methods for generating the underlying field; we follow the spectral synthesis technique used by Stutzki et al. (1998). In addition, we define the BF and RDP cluster models used by CW04 to calibrate the $\mathcal{Q}$ estimator. These two models have a cross-over point where they generate clusters with a uniform distribution. For a more in-depth discussion of the structural properties of the BF and RDP models, we refer the reader to CW04 and JWL17.

The generation of all three models relies heavily on pseudo-random number generation. Throughout this section, we define $\mathcal{U}$ as a random variate drawn from the uniform distribution in the interval [0,1], and $\mathcal{G}$ as a variate from the Gaussian distribution with zero mean and unit variance. These models can be extended to any E-dimensional space. We use the shorthand E_2 and E_3 to indicate two- and three-dimensional spaces, respectively.

2.1 FBM clusters

We generate FBM clusters by generating an FBM probability density distribution. From this, we randomly sample E-dimensional variates, i.e. stellar positions. FBM is an E-dimensional generalization of classical Brownian motion, parametrized by a drift exponent H (sometimes referred to as the Hurst index), which may take a value between 0 and 1. The field's power spectrum is related to H via the spectral index $\beta = E + 2H$. For a one-dimensional FBM curve $f(x)$, the value at $x + \Delta x$ is given by $f(x + \Delta x) = f(x) + \Delta f$, where Δf is a random Gaussian increment. When $H = 1/2$, i.e. classical Brownian motion, Δf is uncorrelated with $f(x)$. When $H > 1/2$, the curve is smoother, i.e. Δf is correlated with $f(x)$. When $H < 1/2$, the curve is rougher, i.e. Δf is anticorrelated with $f(x)$. In E dimensions, FBM structures have fractal dimension $D = E - H$. When D is close to $E - 1$, the structure is smooth and coherent (e.g. a single sheet, filament or core). When D is close to E, the structure consists of multiple sub-clumps that are evenly distributed in space.

We generate the periodic field $f(\boldsymbol{r}, H)$ numerically on an E-dimensional Cartesian grid. Along each axis, r has integer values in the range $1 \leq r \leq N_{\rm PIX}$ (for E_2, we set $N_{\rm PIX} = 1000$; for E_3 we set $N_{\rm PIX} = 100$). First, we generate the spectrum

$$\hat{f}(\boldsymbol{k}, H) = A(\boldsymbol{k}, H)\,[\cos\varphi(\boldsymbol{k}) + \mathrm{i}\sin\varphi(\boldsymbol{k})]\,, \tag{1}$$

where $\boldsymbol{k}$ is a grid of wavevectors with integer k values $\lfloor -N_{\rm PIX}/2 \rfloor \leq k \leq \lceil N_{\rm PIX}/2 \rceil$ along each axis. The amplitudes $A(\boldsymbol{k}, H)$ and phases $\varphi(\boldsymbol{k})$ of each component of the spectrum are given by

$$\begin{aligned} A(\boldsymbol{k}, H) &= \begin{cases} \mathcal{P}^{-1/2}\,\|\boldsymbol{k}\|^{-\beta/2} & \text{if } \boldsymbol{k} \neq \boldsymbol{0}\,, \\ 0 & \text{if } \boldsymbol{k} = \boldsymbol{0}\,, \end{cases} \\ \mathcal{P} &= \sum_{\boldsymbol{k}} \|\boldsymbol{k}\|^{-\beta}\,, \\ \beta &= E + 2H\,, \end{aligned} \tag{2}$$

and

$$\begin{aligned} \varphi(\boldsymbol{k}) &= \chi(\boldsymbol{k}) - \chi(-\boldsymbol{k})\,, \\ \chi(\boldsymbol{k}) &= 2\pi\mathcal{U}\,. \end{aligned} \tag{3}$$

The field $f(\boldsymbol{r}, H)$ can be obtained by performing an inverse discrete Fourier transform on $\hat{f}(\boldsymbol{k}, H)$.[1] Note that the first line of equation (3) ensures that $\hat{f}(-\boldsymbol{k}, H)$ is the complex conjugate of $\hat{f}(\boldsymbol{k}, H)$ and therefore $f(\boldsymbol{r}, H)$ is strictly real.

As noted by Peitgen & Saupe (1988) and JWL17, fractal structures in nature are self-similar over a limited range of length-scales. It is therefore appropriate to introduce a length-scale h at which the self-similarity of the structure ceases. This can be easily implemented by convolving $f(\boldsymbol{r}, H)$ with a Gaussian kernel

$$\begin{aligned} f'(\boldsymbol{r}, H, h) &= f(\boldsymbol{r}, H) * w(\boldsymbol{r}, h)\,, \\ w(\boldsymbol{r}, h) &= \frac{1}{h^E\,(2\pi)^{E/2}} \exp\left(-\frac{\|\boldsymbol{r}\|^2}{2h^2}\right)\,. \end{aligned} \tag{4}$$

Here, h is the smoothing length given in pixels widths. This is equivalent to applying a Gaussian filter to $\hat{f}(\boldsymbol{k}, H)$ with standard deviation $k_{\rm max} = N_{\rm PIX}/4h$.

The FBM field cannot directly be used as a PDF because, by construction, the distribution of $f'(\boldsymbol{r}, H, h)$ is roughly Gaussian with $\langle f'(\boldsymbol{r}, H, h)\rangle \approx 0$ and $\langle f'(\boldsymbol{r}, H, h)^2\rangle \approx 1$. However, the fractal properties of a structure remain unchanged when its density is transformed via a one-to-one function. Here, we exponentiate the field

$$g(\boldsymbol{r}, H, h, \sigma) = \exp\left[\frac{\sigma\, f'(\boldsymbol{r}, H, h)}{\sqrt{\langle f'(\boldsymbol{r}, H, h)^2\rangle}}\right]\,, \tag{5}$$

where σ is a free parameter. This changes the Gaussian distribution of densities into a lognormal distribution. Note that σ is the standard deviation of the natural log of $g(\boldsymbol{r}, H, h, \sigma)$.

Finally, We circularly shift the FBM field so that its periodic centre of mass lies at the centre of the grid. This tends to place coherent structures within high H fields at the centre and lower density regions around the edges.

In summary, we generate a modified FBM field, defined using three parameters: H, h, and σ.[2] This is then used as the PDF from

[1] If $N_{\rm PIX}$ is even, the range of k-values along a given axis is reduced to $-N_{\rm PIX}/2 \leq k \leq N_{\rm PIX}/2 - 1$. Here, the $N_{\rm PIX}/2$ wavenumber is equivalent to $-N_{\rm PIX}/2$. Values of $\hat{f}(\boldsymbol{k}, H)$ with one or more coordinates $k = N_{\rm PIX}/2$ are superposed onto the corresponding $-N_{\rm PIX}/2$ values.

[2] Strictly speaking, the field is defined by five parameters if we include $N_{\rm PIX}$ and the random seed.

which we sample $N_\star$ random positions (see Appendix A for a description of the random sampling technique). E_3 clusters are projected onto E_2 space by marginalizing the distribution along one of its axes. We note that in most practical cases (i.e. $N_\star \leq 10^4$), h is unlikely to have a strong impact on the distribution of points. Essentially, h is a *nuisance parameter* that we include to randomize the field resolution without introducing coarse grid artefacts. For E_2 fields, we randomly pick a value of h from the log-uniform distribution in the interval $[10^{-3}\,N_{\rm PIX}, 10^{-2}\,N_{\rm PIX}]$. For E_3 fields, computational limitations require that we use a coarser grid (both grids have the same number of elements). Here, we skip equation (4) and set $f'(\boldsymbol{r}, H, h) = f(\boldsymbol{r}, H)$.

Fig. 1 shows how the structure of an E_2 FBM cluster varies with H and σ. Here, we have used the same random seed for each realization and set $h = 10^{-3}N_{\rm PIX}$. We see that fixing σ and varying H alter the amount of substructure in a cluster. The outline of the cluster remains roughly the same shape, but the number of internal clumps increases with H. Fixing H and changing σ alter the dynamic range of the cluster surface-density. When σ is high, the clumps are sharply defined. As σ tends towards zero, the cluster structure tends towards uniform density distribution.

2.2 Box-fractal cluster

We generate a BF cluster with approximately $N_\star$ stars by taking an E_3 cube with unit edge-length, and bisecting it along each axis to make 2^E sub-cubes. A random set of 2^D sub-cubes are labelled as *active*, where D has a value in the interval $(0, E]$. In cases where 2^D is non-integer, the number of active sub-cubes if given by

$$N_{\rm act} = \begin{cases} \lfloor 2^D \rfloor \text{ if } \left(2^D - \lfloor 2^D \rfloor\right) < \mathcal{U}, \\ \lceil 2^D \rceil \text{ if } \left(2^D - \lfloor 2^D \rfloor\right) \geq \mathcal{U}. \end{cases} \tag{6}$$

The method is recursively repeated on each active sub-cube a further $\lceil \log_2(N_\star)/D \rceil - 1$ times. Finally, a star is placed at a random position within the volume of each final generation cube. In order to perform the analysis in E_2 space, the BF structure is projected through a random line of sight.

2.3 Radial profile cluster

We construct an E_3 RDP cluster by generating $N_\star$ random coordinates

$$\begin{aligned} \boldsymbol{r} &= r\,\hat{\boldsymbol{u}}\,, \\ r &= \mathcal{U}^{\frac{1}{E-\alpha}}\,, \\ \boldsymbol{u} &= (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_E)\,, \end{aligned} \tag{7}$$

where α is the radial density exponent. The cluster has a density profile $\rho(r) \propto r^{-\alpha}$, where α may have any value in the interval $[0, E)$. Again, the E_3 cluster is projected onto E_2 space through a random line of sight.

3 PARAMETER ESTIMATORS

We examine a family of estimators that use the MST and complete graph (CG) to infer the structural parameters of clusters. The MST is the shortest possible network that connects $N_\star$ vertices with $N_m = N_\star - 1$ edges. The CG is the graph that connects each vertex directly to all the other vertices. The CG has $N_s = N_\star(N_\star - 1)/2$ edges in total. Here, we review the $\mathcal{Q}$ estimator (CW04) and $\bar{m}$–$\bar{s}$ plots (Cartwright 2009, hereafter C09). Next, we present a machine learning algorithm wthat builds and improves upon these two methods. In all three cases, we test each estimator's ability to recover the parameters of artificial clusters and apply them to a selection of observed clusters.

3.1 Observations

We apply the estimators to the clusters examined by CW04 and JWL17. Table 1 lists their properties and original references. The stellar positions of each cluster are plotted in Fig. 2. Each of the cluster has been pre-processed to remove probable multiple systems. Here, any star with neighbours closer than 5×10^{-3} pc is removed, along with its neighbours, and replaced by a single star at the original stars' centre of mass. This minimum length-scale reflects the widest separations typically observed amongst multiple systems in young clusters (King et al. 2012).

3.2 $\mathcal{Q}$ parameter

CW04 define the statistic $\mathcal{Q} = \bar{m}/\bar{s}$, where $\bar{m}$ and $\bar{s}$ are, respectively, the normalized mean edge-lengths of the MST and the CG:

$$\begin{aligned} \bar{m} &= \left(\frac{1}{\left(\pi R^E [N_m + 1]^{E-1}\right)^{1/E}} \right) \sum_{i=1}^{N_m} m_i\,, \\ \bar{s} &= \left(\frac{1}{N_s R} \right) \sum_{i=1}^{N_s} s_i\,. \end{aligned} \tag{8}$$

Here, m_i and s_i are graph edge-lengths, and R is a characteristic length-scale of the system. Note that the R terms cancel when calculating $\mathcal{Q}$.

The CW04 calibration of $\mathcal{Q}$ involves calculating the statistic for BF and RDP clusters. A uniform density cluster (i.e. $D = 3$ or $\alpha = 0$) returns $\mathcal{Q} \sim 0.8$. BF clusters have $\mathcal{Q} \lesssim 0.8$ and RDP clusters have $\mathcal{Q} \gtrsim 0.8$. $\mathcal{Q}$ increases monotonically with both D and α. Fig. 3 shows the relationship between $\mathcal{Q}$ and D, and between $\mathcal{Q}$ and α.

The $\mathcal{Q}$ values of Lupus 3, IC 348, and ρ Oph suggest that they have RDPs with $\alpha \gtrsim 1.5$. The $\mathcal{Q}$ values of Cha I and Taurus suggest they are similar to BF structures with $D \lesssim 2.5$. The $\mathcal{Q}$ value of IC 2391 lies near a plateau on the plot, making its structural type difficult to determine.

Fig. 3 also shows how $\mathcal{Q}$ relates to the parameters of FBM clusters. Here, we see that there is a slight positive correlation between H and $\mathcal{Q}$. However, the scatter introduced by σ exceeds the dynamic range of the correlation. There is no noticeable correlation between $\mathcal{Q}$ and σ. Therefore, $\mathcal{Q}$ is a poor predictor of H and/or σ.

3.3 $\bar{m}$ – $\bar{s}$ plots

C09 suggest that plots of $\bar{m}$ versus $\bar{s}$ provide a more robust diagnostic tool than $\mathcal{Q}$ alone. They show that BF and RDP clusters with fixed parameters fill distinct regions of the $\bar{m}$–$\bar{s}$ plot. However, there is a lack of agreement on which length-scale R should be used to normalize $\bar{m}$ and $\bar{s}$. In the original C09 publication, R is set to the distance between the cluster's centre of mass and its outer most point. This measure is problematic as a single outlying star can dominate the length-scale and this value is not representative of the area of a cluster with a high aspect ratio. Both of these issues can add significant noise to the normalization of $\bar{m}$ and $\bar{s}$ (see Parker 2018, for a review of different R normalization methods). Instead, we use the Schmeja & Klessen (2006) scheme, which sets R to the square root of the area of the convex hull of the set of stars. This lessens (although does not necessarily eliminate) the issues with outliers and the aspect ratio.

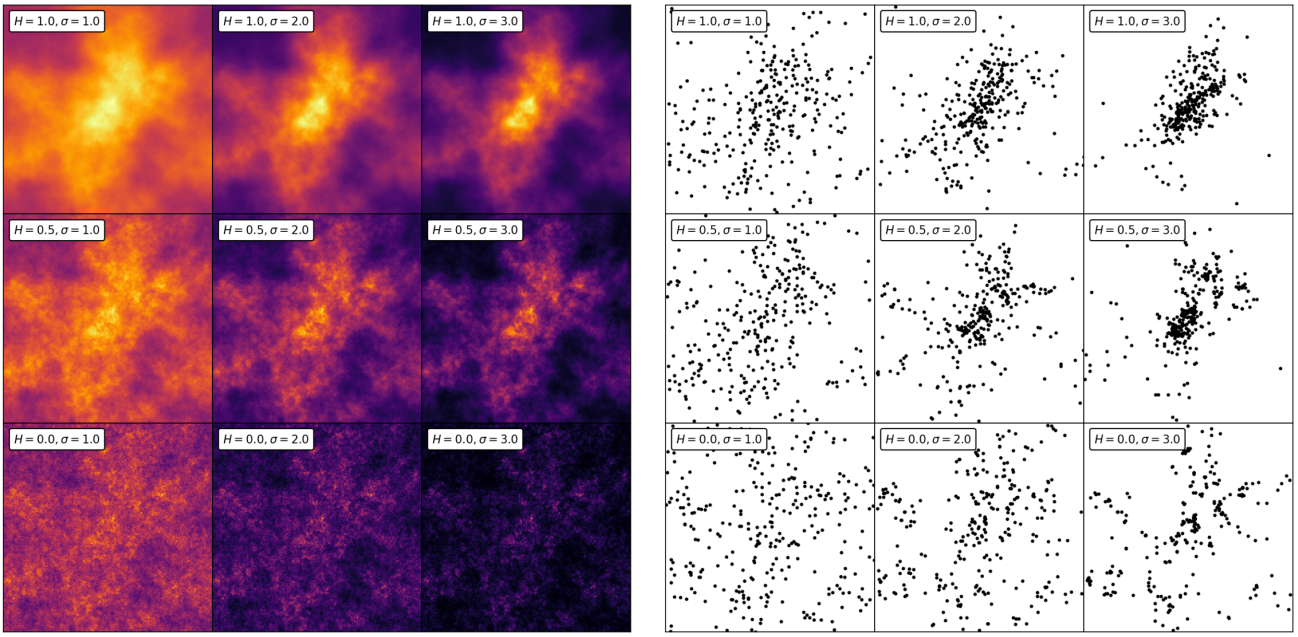

Figure 1. Left-hand panels: nine E_2 FBM fields generated using the same random seed. From the top to bottom, the rows show fields with $H = 1.0,\ 0.5,$ and 0.0. From the left to right, the columns show fields with $\sigma = 0.5\,,\ 1.0,$ and 2.0. The colour scale gives an indication of the relative surface density. Right-hand panels: nine sets of 300 points, randomly sampled from the corresponding fields on the left.

Table 1. Cluster properties and sources. The first column gives the numeric identifier used throughout this paper; the second column gives the name of the cluster; the third column gives the number of stellar objects (after multiple systems are fused into single objects); the fourth column gives the assumed distance to the cluster; the fifth column cites the source of the data.

#	Name	$N_\star$	D (pc)	Reference
1	Lupus 3	67	170	Comerón (2008)
2	IC 348	350	315	Lada et al. (2006); Muench et al. (2007)
3	ρ Oph	198	130	Bontemps et al. (2001)
4	IC 2391	200	150	Barrado y Navascués et al. (2001)
5	Cha I	234	160	Luhman (2007)
6	Taurus	335	140	Luhman et al. (2010)

The top frame of Fig. 4 shows how model clusters with different values of D or α occupy different regions on the $\bar{m}$–$\bar{s}$ plot. Here, the parameter estimates for Lupus 3, IC 348, and ρ Oph are unchanged from their respective $\mathcal{Q}$ estimates. In addition, the plot suggests that IC 2391 is similar to a BF cluster with $D \sim 2.8$. However, we find that Taurus and Cha I do not match up to *any* of the BF or RDP clusters. On visual inspection (see Fig. 2), they are clearly sub-clustered, but their $\bar{m}$–$\bar{s}$ values cannot be matched to any value of D.

The remaining two frames of Fig. 4 shows the $\bar{m}$–$\bar{s}$ values for FBM clusters. Here, unlike the BF and RDP models, the FBM clusters fill an area of the plot, which overlaps all of the observed clusters. We see that FBM clusters with $H \sim 1$ fill the same region of the plot as RDP clusters. This is unsurprising, as they both represent smoothly distributed, centrally concentrated clusters. However, clusters with $H \lesssim 1$ do not appear to occupy distinct regions of the plot. Finally, We see a very strong negative correlation between $\bar{m}$ and σ. This shows that the mean edge-length of the MST is much more sensitive to the surface-density dynamic range of a cluster than its fractal properties.

3.4 Machine learning regression

$\mathcal{Q}$ and $\bar{m}$–$\bar{s}$ plots are often used to estimate underlying parameters by visual inspection. By this, we mean that a large ensemble of $\mathcal{Q}$ or $\bar{m}$–$\bar{s}$ measurements for a known set of models are plotted; parameters are attributed to an observation based on the plot-distance from the observation's measurements to the equivalent model values. This methodology makes it difficult to quantify parameter uncertainties. Furthermore, we have shown that the BF model, which typically is used to calibrate the two methods, is unable to produce clusters with similar properties to Taurus or Cha I. The latter of these two problems may be addressed by implementing the FBM cluster model. However, the $\mathcal{Q}$ and $\bar{m}$–$\bar{s}$ methods are poor at distinguishing the underlying parameters. We address these shortcomings with a machine learning regressor wthat uses FBM clusters as training data.

A *regressor* is an analytical function or numerical procedure $F(x)$, which gives an estimate of y for a given input (or feature) x. In order to make these estimates, the regressor must first be *trained*. A simple example of a regressor is linear regression, i.e. $F(x) = m\,x + c$. Training the regressor involves taking N training data, x_i and y_i, and finding values m and c (hyperparameters[3]), which minimize a loss statistic, e.g. $L = \sum_{i=1}^{N}(F(x_i) - y_i)^2$.

A similar approach can be used to estimate the parameters of a star cluster. Here, $\boldsymbol{x}$ is a vector of statistics that are directly taken from the cluster (we define these in Section 3.4.1), and $\boldsymbol{y} = (H, \sigma)$ is the vector of underlying parameters. Here, the regressor $F(\boldsymbol{x})$ is an

[3]In most contexts, these are referred to as *parameters*. We refer to them as *hyperparameters* so that they are not confused with cluster model parameters, e.g. H and σ.

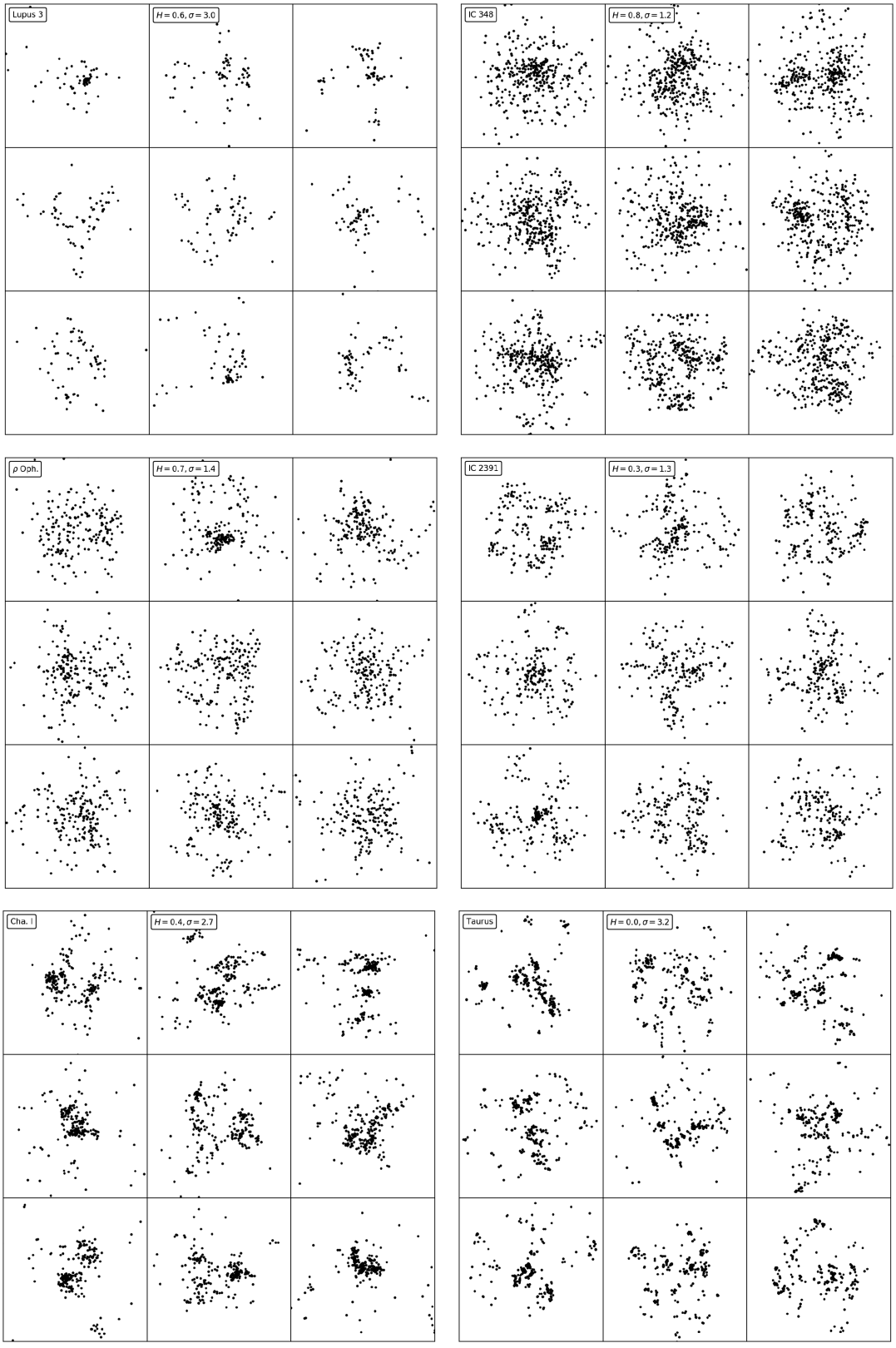

Figure 2. The positions of stars in real clusters, presented alongside those of artificial FBM clusters. In each of the six frames, the real cluster is plotted in the top left-hand corder. The remaining eight frames show different realizations of FBM clusters with the most likely estimated parameters (see Section 3.4). In all cases, the aspect ratios of the clusters have been removed (see equation 9).

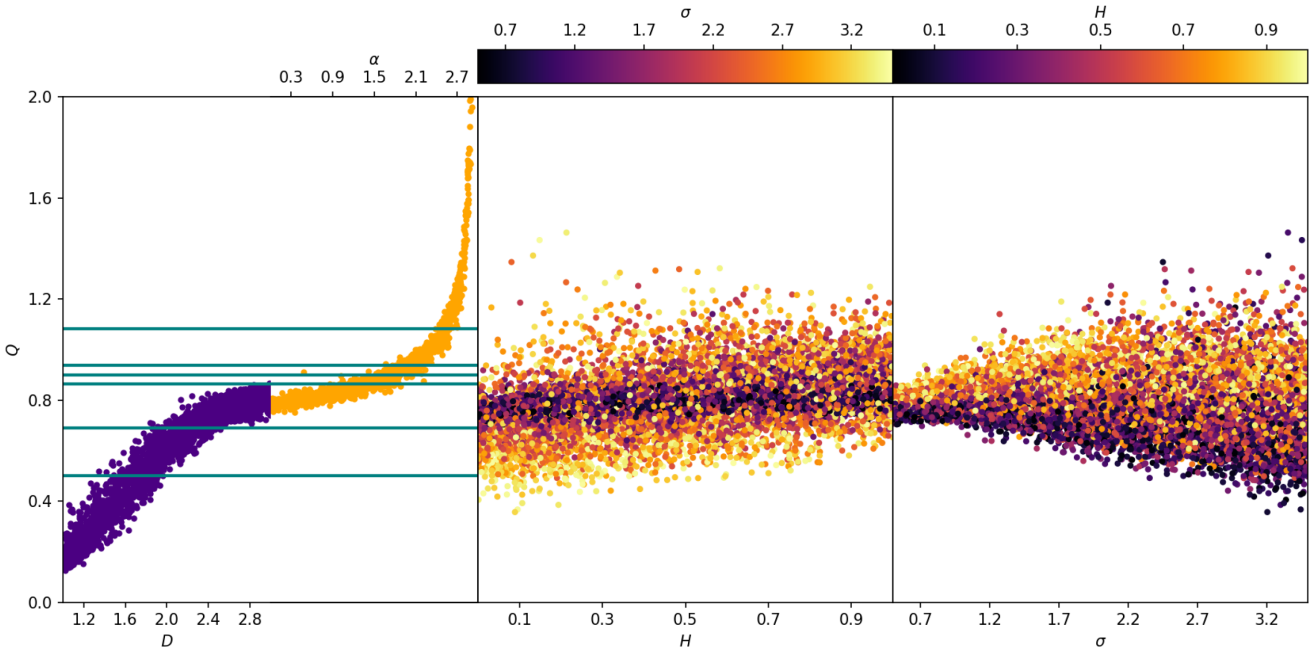

Figure 3. $\mathcal{Q}$ values plotted against the underlying parameters of artificial star clusters. Each artificial cluster contains between 300 and 1000 stars. The left-hand panel shows $\mathcal{Q}$ values for BF clusters (purple) and RDP clusters (gold). Each set of points represents 2000 clusters with random D in the interval [1,3] and α in the interval [0,3). The horizontal teal lines show the values of $\mathcal{Q}$ calculated for the clusters listed in Table 1 (the vertical order of the lines is the same as the table). The centre and right-hand panels show the $\mathcal{Q}$ values for FBM clusters as a function of H and σ. Each panel contains 5000 clusters with H in the interval [0,1] and σ in the interval [0.5,3.5]. In each case, the colour scale gives the value of the other parameter.

artificial neural network (ANN). Complex ANNs are routinely used in fields such as image analysis (e.g. Lecun et al. 1998). However, comparatively simple ANNs can be used for numerical regression problems with multiple inputs and outputs (e.g. Rafieferantsoa, Andrianomena & Davé 2018).

Details of ANN used here, along with links to the full implementation in PYTHON, are given in Appendix B. If the reader is not concerned about these technical details, they should simply note that the ANN hyperparameters are estimated from *training data*. Once trained, the ANN is applied to *test data*. This enables us to ensure we are not overfitting the training data and quantify the uncertainties of the regressor. In the following sections, we discuss the training, testing, and the results of the ANN.

3.4.1 Training

For each star cluster, we generate a set features $\boldsymbol{x}$ using its CG and MST. However, as noted by Cartwright & Whitworth (2009), the elongation of a star cluster may affect these graphs. Before we build the graphs, we *whiten* the distribution of points. This completely removes the size scale and aspect ratio from the distribution. We calculate the covariance matrix $\Sigma(\boldsymbol{r})$ for the set of stellar positions $\boldsymbol{r}$. From this, we calculate a new set of positions $\boldsymbol{r}'$, where each value $\boldsymbol{r}'_i$ has elements

$$r'_{ij} = \frac{1}{\sqrt{\lambda_j}} \boldsymbol{r}_i \cdot \hat{\boldsymbol{v}}_j \,, \quad j = 1, 2, \ldots, E \,. \tag{9}$$

Here, λ_j and $\boldsymbol{v}_j$ are, respectively, the jth eigenvalues and eigenvectors of $\Sigma(\boldsymbol{r})$. Note that $\Sigma(\boldsymbol{r}')$ is equal to the identity matrix $\boldsymbol{I}$.

For a set of N graph edges l, we define the mean edge-length $\mu(l)$ and the nth central moment $M_n(l)$ as

$$\mu(l) = \frac{A}{N} \sum_{i=1}^{N} l \,,$$

$$M_n(l) = \left(\frac{A}{N} \sum_{i=1}^{N} \left[l_i - \frac{\mu(l)}{A} \right]^N \right)^{\frac{1}{N}} \,,$$

$$A = \begin{cases} \frac{N}{(N+1)^{(E-1)/E}} & \text{for MST,} \\ 1 & \text{for CG.} \end{cases}$$

We construct $\boldsymbol{x}$ using the mean and the second, third, and fourth central moments of the MST and CG edge-lengths. Note that the second, third, and fourth central moments are related to the variance, skewness, and kurtosis. We do not need to normalize these features to a length-scale as we have already whitened the distribution of points.

We perform two analyses: one with E_2 FBM clusters and another with E_3. For each analysis, we train three regressors with different ranges of $N_\star$. There are two reasons for this. First, it is useful to quantify parameters uncertainties as a coarse function of $N_\star$. Secondly, the MST normalization (see equation 10) is technically only valid for the limit $N_\star \to \infty$ (Steele 1988). Splitting the analysis into different $N_\star$ bins helps to isolate any biases which may occur as a function of $N_\star$. For each regressor, we generate 10^4 training clusters with randomly sampled values of $N_\star$, H, and σ. The ranges and distributions of these parameters are given in Table 2.

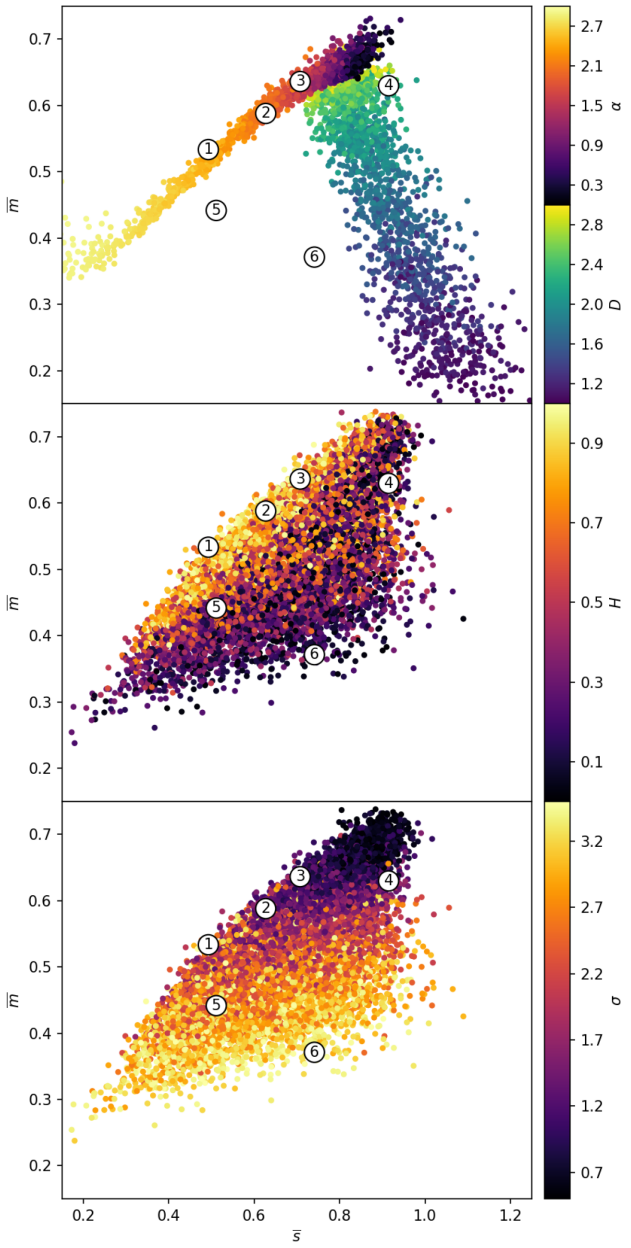

Figure 4. Plots of $\bar{m}$ versus $\bar{s}$ for the BF and RDP clusters (top panel) and the FBM clusters (middle and lower panel). The colour scale gives the values of the underlying cluster parameters. The points are generated from the same clusters as Fig. 3. The numbered points give $\bar{m}$–$\bar{s}$ values for the clusters given in Table 1. In all cases, the value of R (see equation 8) is set to the square root of the area of the cluster convex hull.

Table 2. The range and distributions of parameters used to train the ANN regressor. The first column gives the parameter; the second column gives the type of distribution; the third column gives the interval. Note that σ has a different ranges for E_2 and E_3.

Parameter	Distribution	Interval
$N_\star$	log-uniform	[32,99], [100,315], [316,999]
H	uniform	[0,1]
σ	uniform	[0.5,3.5] (E_2), [0.5,4.5] (E_3)

3.4.2 *Testing and results*

We test each trained ANN by generating an additional 5×10^3 artificial clusters. These are randomly generated the same way as the training clusters, but with different random seeds. Fig. 5 shows the estimated parameters of E_2 test clusters as a function of their underlying true parameters. From these plots, we see that the parameters can be estimated with a useful degree of accuracy for clusters with $N_\star > 100$. We can approximate the uncertainties as the root-mean-squared errors, $\Delta H = \sqrt{\langle (H_{\rm est} - H)^2 \rangle}$ and $\Delta\sigma = \sqrt{\langle (\sigma_{\rm est} - \sigma)^2 \rangle}$. Here, the est subscripts denote estimated parameters; the terms with no subscripts denote underlying parameters. For both the E_2 and E_3 cases, $\Delta H \approx \pm 0.2$. For the E_2 case, $\Delta\sigma$ varies from ± 0.3 to ± 0.5. For E_3, $\Delta\sigma$ varies between ± 0.5 and ± 0.7. The magnitudes of the uncertainties decrease as $N_\star$ increases (we give values for E_2 clusters to two significant figures in Fig. 5). We also find that the H and σ uncertainties are correlated, i.e. there is some degeneracy in the expression of the two parameters. Here, high σ can make a smooth distribution (determined by H) appear rougher, and *vice versa*. We note that this uncertainty approximation may underestimate the error on $H_{\rm est}$ when $N_\star < 100$. Here, the correlation between $H_{\rm est}$ and H is visibly less tight than the other cases.

Table 3 shows the parameter estimates for the observed star clusters. We find that for these six cases, H appears invariant with respect to E, whereas the E_3 values of σ are approximately one and a half times greater than the E_2 values. We also include approximate D and α values estimated using $\bar{m}$–$\bar{s}$ plots for comparison. Fig. 6 shows a plot of σ against H for the E_2 analysis. Here, we see that Taurus, Cha I, and IC 2391 have similar levels of fractal structure to each other (determined by H), but are distinguished by different surface-density dynamic ranges (determined by σ). IC 348 and ρ Oph are indistinguishable from one another, each with a smooth structure and a low level of surface-density variation. Lupus 3 has a high amount of surface-density variation, but the relatively low number of stars makes it difficult to estimate the uncertainty on H.

4 DISCUSSION

4.1 Comparison of methods

We have shown that the BF star cluster model struggles to reproduce the observed features of substructured clusters. This is because, as identified by JWL17, the BF model *only* produces clusters with very high surface-density variances. Therefore, we strongly suggest that the model should be retired from star cluster analysis. The FBM model presented here overcomes this problem. FBM clusters have independent parameters that control the amount of fractal clustering and set the global level of surface-density variation. In addition, clusters with $H \sim 1$ fulfil the same role as centrally concentrated RDP clusters. This removes the need for using two unrelated models (i.e. BF and RDP) in the same analysis.

We find that $\mathcal{Q}$ and/or $\bar{m}$–$\bar{s}$ plot analyses are poorly suited to FBM clusters. Furthermore, they do not present robust uncertainties. This limits their efficacy, and we suggest that they should no longer be used as is. However, these analyses can be reformulated using modern machine learning techniques. Here, we present an ANN that makes robust estimates of star cluster parameters and their uncertainties. We note that alternatives to this method also exist. For example, JWL17 use principle component analysis to reduce a large range of observables to two principle features.

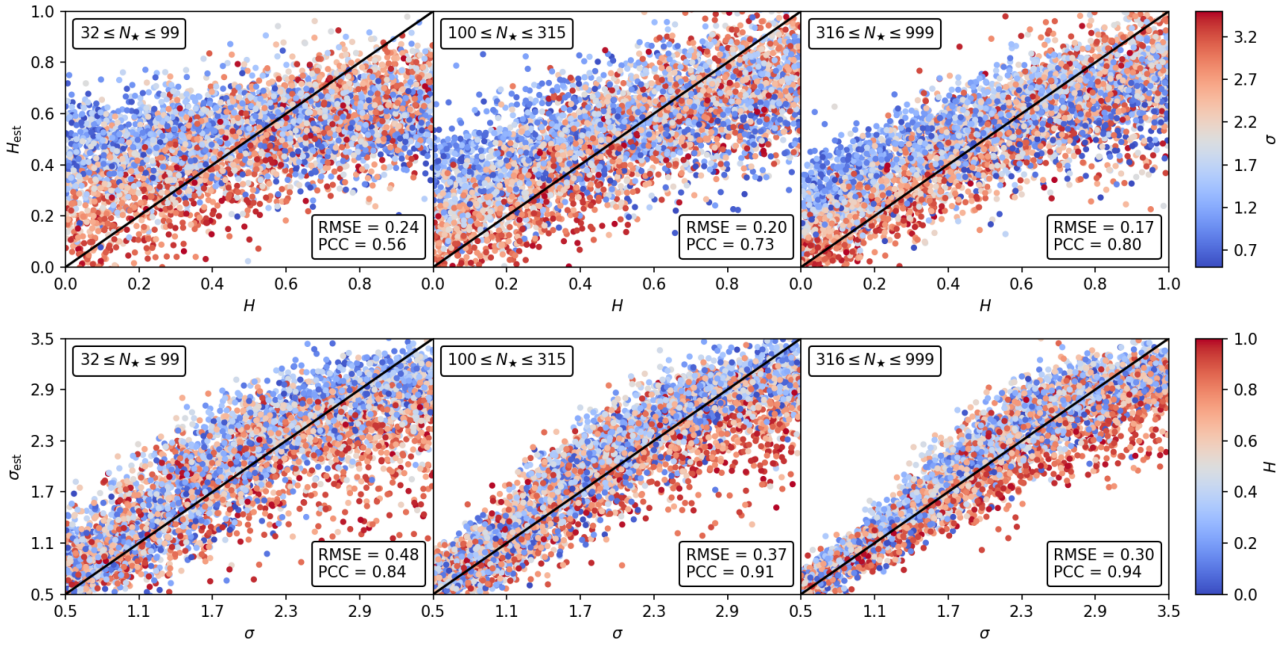

Figure 5. Test data parameter estimates as a function of the underlying parameters. The top row shows the ANN's ability to predict H. The bottom row shows ANN's ability to predict σ. The colour scale gives the value of the other parameter. The range of $N_\star$ is indicated in the top left-hand corner of each plot. The solid black line shows the hypothetical performance of a perfect estimator. We also give the root-mean-squared error and Pearson's correlation coefficient for each plot in the bottom right-hand corner.

Table 3. Parameter values estimated for observed star clusters. The first and second columns give the identifier and name of the cluster. The third and fourth columns give the E_2 FBM parameters, inferred using the ANN. The fifth and sixth columns give the same values for E_3. The seventh and eighth columns give the approximate D or α values, estimated using $\bar{m}$–$\bar{s}$ plots.

#	Cluster	H (E_2)	σ (E_2)	H (E_3)	σ (E_3)	D (C09)	α (C09)
1	Lupus 3	0.6 ± 0.2	3.0 ± 0.5	0.6 ± 0.2	3.8 ± 0.7	–	∼2.5
2	IC 348	0.8 ± 0.2	1.2 ± 0.3	0.7 ± 0.2	1.6 ± 0.5	–	∼2.1
3	ρ Oph	0.7 ± 0.2	1.4 ± 0.4	0.7 ± 0.2	2.1 ± 0.6	–	∼1.8
4	IC 2391	0.3 ± 0.2	1.3 ± 0.4	0.3 ± 0.2	2.0 ± 0.6	∼2.8	–
5	Cha I	0.4 ± 0.2	2.7 ± 0.4	0.2 ± 0.3	3.7 ± 0.6	–	–
6	Taurus	0.0 ± 0.2	3.2 ± 0.3	0.0 ± 0.2	4.7 ± 0.5	–	–

4.2 Note on fractal dimension

We have, where possible, avoided discussing these results in terms of fractal dimension. This is because D does not uniquely describe a structure. For example, BF clusters with $D \sim 3$ have a roughly uniform distribution of stars. Decreasing D increases the level of substructure in the distribution. Conversely, FBM clusters with $D \sim 3$ may be very substructured. Decreasing D tends to fuse the sub-clumps together until the cluster is composed of one or two coherent objects. We, therefore, suggest caution when using the term 'fractal dimension' in scientific statements. There appears to be little relation between the value of D and the subjective *clumpiness* expressed in different models.

4.3 Caveats

While the FBM model has several advantages over the BF model, there are some caveats we must address. The FBM fields generated in Section 2.1 fill a periodic box with no true centre. Here, we shift the field's periodic centre of mass to the centre of the box. This generally places high-density structures (if present) in the centre of the box and low-density regions around the edges. However, we acknowledge that this choice is arbitrary. Also, in some instances, the outline of the cluster can appear square (the effect is most pronounced when σ is low; see Fig. 2). We could address this by culling the distribution into a sphere, but this would arbitrarily remove stars from the edges of the distribution.

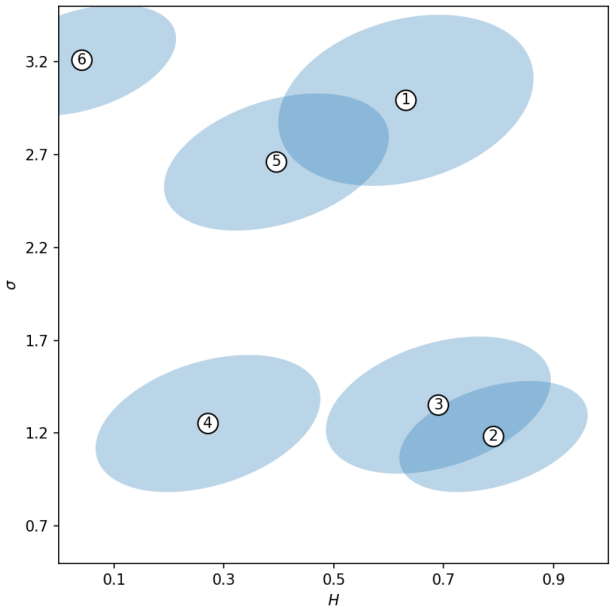

Figure 6. A scatter plot of the E_2 estimates of H and σ for the clusters listed in Table 1. The ellipses show the 1σ uncertainties, calculated from the root-mean-squared errors and their covariance.

We have demonstrated that the ANN regressor performs well at classifying and differentiating stars clusters. However, we note that there is an infinitude of measurable features for any given cluster. We have experimented with a large number of features from different graphs (e.g. centile-based statistics, features from the Delaunay triangulation). We have found through testing that the features presented here are adequate. Adding further features to the ANN only yields very minor improvements to its estimation accuracy.

4.4 Future work

The values of H and σ are useful for categorizing star clusters by their morphology. However, in order to infer physical meaning from these measures, we need to apply them to simulations. We hypothesize that, for a sub-virial substructured cluster, σ should increase as the cluster collapses under its own gravity. Meanwhile, H should increase as the collapse erases the cluster's substructure. Conversely, for a superviral equivalent of the same initial cluster, σ should decrease over time as the cluster expands. It is not clear how H will behave during this process. We will test these hypotheses by applying this analysis to an ensemble of N-body cluster simulations with various initial states.

The ANN method also provides a convenient way to compare the structure of the molecular clouds with that of star clusters. For example, Elia et al. (2014) find that molecular clouds in the galactic plane typically have $H \lesssim 0.4$, suggesting that they are similar in structure to Taurus, or Cha I. Previous numerical work has attempted to compare molecular cloud gas structure with that of embedded clusters (e.g. Lomax et al. 2011; Parker & Dale 2015). However, this was performed using $\mathcal{Q}$ analysis, which we have shown is unreliable. We will revisit this work and analyse the FBM properties of the stars and gas in molecular cloud simulations.

5 SUMMARY AND CONCLUSIONS

We present an artificial star cluster model, based on FBM. The structure of these clusters is controlled by two parameters: the drift exponent H, which controls the degree of fractal structure, and the standard deviation σ of the log-surface/volume density. The model is able to produce artificial clusters with a wide range of structural morphologies, similar to those of Lupus 3, IC 348, ρ Oph, IC 2391, Cha I, and Taurus. This contrasts with the BF model – used in $\mathcal{Q}$ analysis – which has a single parameter, D. Here, D is notionally a fractal dimension. However, changing its value simultaneously alters the degree of fractal structure and the amount of surface-density variation. Because these two properties are linked, the BF model is unable to reproduce naturally substructured clusters, like Cha I and Taurus. We note that Jaffa et al. (2017) add extra parameters to the BF model in order to address this problem. Their model has a similar level of complexity as the FBM model, and can be viewed as an alternative to the work presented here.

$\mathcal{Q}$ analysis and $\bar{m}$–$\bar{s}$ plots are not well suited to estimating FBM cluster parameters. We present an ANN regressor that can reliably estimate the parameter values and their uncertainties. Future work will involve using ANNs to measure how the structural properties of N-body cluster simulations evolve over time. Furthermore, FBM analysis is well suited to studying the structure of the ISM. This means we can use the method to directly compare the structure of gas and stars in star-forming complexes.

ACKNOWLEDGEMENTS

We thank the reviewer, Simon Goodwin, for his constructive comments. OL and APW gratefully acknowledge the support of a consolidated grant (ST/K00926/1) from the UK Science and Technology Facilities Council (STFC). MLB gratefully acknowledges the support of a CDT in data intensive science (ST/P006779/1) from the UK STFC.

REFERENCES

André P. et al., 2010, A&A, 518, L102
Barrado y Navascués D., Stauffer J. R., Briceño C., Patten B., Hambly N. C., Adams J. D., 2001, ApJS, 134, 103
Bate M. R., Bonnell I. A., 2005, MNRAS, 356, 1201
Bontemps S. et al., 2001, A&A, 372, 173
Cartwright A., 2009, MNRAS, 400, 1427
Cartwright A., Whitworth A. P., 2004, MNRAS, 348, 589 (CW04)
Cartwright A., Whitworth A. P., 2009, MNRAS, 392, 341
Comerón F., 2008, in Reipurth B., ed., Handbook of Star Forming Regions, Vol. II. ASP Monograph Publications, p. 295
Elia D. et al., 2014, ApJ, 788, 3
Elmegreen B. G., Falgarone E., 1996, ApJ, 471, 816
Falgarone E., Phillips T. G., Walker C. K., 1991, ApJ, 378, 186
Federrath C., Klessen R. S., 2012, ApJ, 761, 156
Brown A. G. A., Vallenari A., Prusti T., de Bruijne J. H. J., Babusiaux C., Bailer-Jones C. A. L., Gaia Collaboration, 2018, preprint (arXiv: 1804.09365)
Gaia Collaboration et al., 2016, A&A, 595, A1
Goodwin S. P., Whitworth A. P., 2004, A&A, 413, 929
Gouliermis D. A., Hony S., Klessen R. S., 2014, MNRAS, 439, 3775
Griffin M. J. et al., 2010, A&A, 518, L3
Jaffa S. E., Whitworth A. P., Lomax O., 2017, MNRAS, 466, 1082 (JWL17)
King R. R., Parker R. J., Patience J., Goodwin S. P., 2012, MNRAS, 421, 2025
Lada C. J. et al., 2006, ApJ, 131, 1574
Lecun Y., Bottou L., Bengio Y., Haffner P., 1998, Proc. IEEE, 86, 2278
Lomax O., Whitworth A. P., Cartwright A., 2011, MNRAS, 412, 627
Luhman K. L., 2007, ApJS, 173, 104
Luhman K. L., Allen P. R., Espaillat C., Hartmann L., Calvet N., 2010, ApJS, 186, 111
Molinari S. et al., 2010, A&A, 518, L100
Motte F., Andre P., Neri R., 1998, A&A, 336, 150
Muench A. A., Lada C. J., Luhman K. L., Muzerolle J., Young E., 2007, ApJ, 134, 411
Parker R. J., 2018, MNRAS, 476, 617
Parker R. J., Dale J. E., 2015, MNRAS, 451, 3664
Parker R. J., Wright N. J., Goodwin S. P., Meyer M. R., 2014, MNRAS, 438, 620
Pedregosa F. et al., 2011, J. Mach. Learn. Res., 12, 2825
Peitgen H.-O., Saupe D., 1988, The Science of Fractal Images. Springer-Verlag. Berlin
Poglitsch A. et al., 2010, A&A, 518, L2
Rafieferantsoa M., Andrianomena S., Davé R., 2018, MNRAS, 479, 4509
Schmeja S., Klessen R. S., 2006, A&A, 449, 151
Schneider N. et al., 2013, ApJ, 766, L17
Smith R. J., Glover S. C. O., Klessen R. S., Fuller G. A., 2016, MNRAS, 455, 3640
Steele J. M., 1988, Ann. Probab., 16, 1767
Stutzki J., Bensch F., Heithausen A., Ossenkopf V., Zielinsky M., 1998, A&A, 336, 697
Williams J. P., Blitz L., McKee C. F., 2000, in Mannings, V., Boss, A. P., Russell, S. S., eds, Protostars and Planets IV, University of Arizona Press, Tucson, p. 97

APPENDIX A: RANDOMLY SAMPLING VARIATES FROM 3 OR 2 DIMENSIONAL DISTRIBUTIONS

We can draw random coordinates (X, Y, Z) from any gridded three-dimensional distribution $p(x, y, z)$ using random variates $\mathcal{U}_x$, $\mathcal{U}_y$, and $\mathcal{U}_z$, drawn from the uniform distribution in the interval [0,1]. First, we calculate the cumulative distribution of $p(x, y, z)$ along the x-axis,

$$P(x) = \int_{x_{\rm MIN}}^{x} p(x)\,{\rm d}x\,,$$
$$p(x) = \int_{z_{\rm MIN}}^{z_{\rm MAX}} \int_{y_{\rm MIN}}^{y_{\rm MAX}} p(x, y, z)\,{\rm d}y\,{\rm d}z\,. \quad \text{(A1)}$$

Here, the MIN and MAX subscripts denote the extreme coordinate values of the cartesian grid. Integrals are computed using the trapezium rule. Next, we numerically invert $P(x)$ to find X using the relationship

$$\frac{P(X)}{P(x_{\rm MAX})} = \mathcal{U}_x\,. \quad \text{(A2)}$$

In order to get Y, we calculate the cumulative distribution along the y-axis, given X,

$$P(y|X) = \int_{y_{\rm MIN}}^{y} p(y|X)\,{\rm d}y\,,$$
$$p(y|X) = \int_{z_{\rm MIN}}^{z_{\rm MAX}} p(X, y, z)\,{\rm d}z\,. \quad \text{(A3)}$$

In practice, we pre-compute $P(y|x)$ for all gridded values of x. $P(y|X)$ is then found by linearly interpolating $P(y|x)$ over the two x-values either side of X. The Y-coordinate can found by inverting

$$\frac{P(Y|X)}{P(y_{\rm MAX}|X)} = \mathcal{U}_y\,. \quad \text{(A4)}$$

Finally, we get the Z coordinate by calculating the cumulative distribution along the z-axis, given X and Y,

$$P(z|X, Y) = \int_{z_{\rm MIN}}^{z} p(z|X, Y)\,{\rm d}z\,, \quad \text{(A5)}$$

and inverting

$$\frac{P(Z|X, Y)}{P(z_{\rm MAX}|X, Y)} = \mathcal{U}_z\,. \quad \text{(A6)}$$

Again, we pre-compute $P(z|x, y)$ for all combinations of x and y, and bi-linearly interpolate $P(z|x, y)$ over the four (x, y) values surrounding (X, Y). As before, the Z-coordinate is found by inverting

$$\frac{P(Z|X, Y)}{P(z_{\rm MAX}|X, Y)} = \mathcal{U}_z\,. \quad \text{(A7)}$$

This method can also be performed on a two-dimensional distribution, $p(x, y)$. Here, we simply repeat the same steps (disregarding any integrals over the z-axis) until we have obtained X and Y.

APPENDIX B: ARTIFICIAL NEURAL NETWORK

An ANN can be thought of as a collection of artificial neurons. Each neuron takes an input $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ and outputs $z = f(b + \boldsymbol{w} \cdot \boldsymbol{x})$. Here, b is a bias value, $\boldsymbol{w}$ is a vector of m weights, and $f(t)$ is an activation function. The activation function is usually chosen to vary smoothly over a limited range, e.g. $f(t) = \tanh(t)$ or $f(t) = 1/[1 + \exp(-t)]$. A collection of n neurons can be grouped together to form a layer. Here, the weights are represented by an $m \times n$ matrix $\boldsymbol{W}$, and the biases by a vector $\boldsymbol{b}$ with length n. The ensemble of neurons has an output $\boldsymbol{z} = f(\boldsymbol{b} + \boldsymbol{W}\boldsymbol{x})$.[4]

For this analysis, we set up a three-layer ANN using the MLPREGRESSOR class in the SCIKIT-LEARN library (Pedregosa et al. 2011).[5] The structure of the ANN is as follows:

$$\begin{aligned} &\text{Layer 1, } m \text{ elements}: \quad \boldsymbol{x}\,; \\ &\text{Layer 2, } n \text{ elements}: \quad \boldsymbol{z} = \tanh(\boldsymbol{b}_1 + \boldsymbol{W}_1\,\boldsymbol{x})\,; \\ &\text{Layer 3, } p \text{ elements}: \quad \boldsymbol{y} = \boldsymbol{b}_2 + \boldsymbol{W}_2\,\boldsymbol{z}\,. \end{aligned} \quad \text{(B1)}$$

The first layer is the input vector of features $\boldsymbol{x}$. This is composed of the measurable properties of a star cluster. The second layer $\boldsymbol{z}$ is determined by a bias vector $\boldsymbol{b}_1$ and the weight matrix $\boldsymbol{W}_1$. The third and final layer is the output $\boldsymbol{y}$. This is composed of the underlying cluster parameters that we are trying to estimate. The values are determined by a second bias vector $\boldsymbol{b}_2$ and weight matrix $\boldsymbol{W}_2$. Note that no activation function is used to calculate the final layer; this is so $\boldsymbol{y}$ is not confined to a limited range. The number of neurons in the second layer is arbitrary; here, we find $n = 40$ provides the most accurate results (more complicated ANN regressors may contain multiple hidden layers). For simplicity, we refer to the ANN mapping of $\boldsymbol{x}$ to $\boldsymbol{y}$ as $\boldsymbol{y} = F(\boldsymbol{x})$. The ANN is trained by taking $N_{\rm train}$ training clusters, with known $\boldsymbol{y}$, and finding values of $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, $\boldsymbol{b}_1$, and $\boldsymbol{b}_2$, which minimize $\langle(F(\boldsymbol{x}) - \boldsymbol{y})^2\rangle$. This is performed by the class using gradient-descent techniques.

This paper has been typeset from a TEX/LATEX file prepared by the author.

[4]Note that here $f(t)$ is a scalar function with a scalar argument. For the same function, we define $f(\boldsymbol{t}) \equiv (f(t_1), f(t_2), \ldots)$.

[5]The hyperparameters of the class, including the number of layers and neurons per layer, are tuned using GRIDSEARCHCV cross-validation tool. The full implementation can be found at github.com/odlomax/clusterfrac.