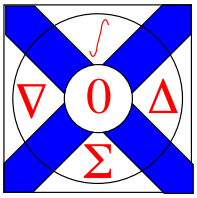


Computational Science Technical Note **CSTN-017**

Manual and Semi-Automated Classification in a Microscopic Artificial Life Model

K. A. Hawick and H. A. James and C. J. Scogings

2005

We discuss algorithms and methods for classifying the clusters of model animals that emerge from simulations of collective behaviour in artificial life models. We show how important statistical properties for understanding scaling and universal growth can be measured from complex and chaotic model systems. We describe animal clustering algorithms and the difficulties involved in automatic tracking of herds that move and change shape, orientation and size in time. We present some heuristic rules for semi-automated classification over time and some preliminary results from our study of a predator-prey multi-agent model.

Keywords: classification; clustering; chaotic and complex systems; multi-agent systems

BiBTeX reference:

```
@INPROCEEDINGS{CSTN-017,
        author = {K. A. Hawick and H. A. James and C. J. Scogings},
        title = {Manual and Semi-Automated Classification in a Microscopic Artificial
                Life Model},
        booktitle = {Proc. Int. Conf. on Computational Intelligence (CI'05)},
        year = {2005},
        pages = {135-140},
        address = {Calgary, Canada.},
        month = {4-6 July},
        publisher = {IASTED},
        keywords = {classification; clustering; chaotic and complex systems; multi-agent
                systems}
}
```


Prof Ken Hawick, Computer Science, Massey University, Albany, North Shore 102-904, Auckland, New Zealand.

Technical Report CSTN-017

Manual and Semi-Automated Classification in a Microscopic Artificial Life Model

K.A. Hawick, H.A. James and C.J. Scogings
Institute of Information & Mathematical Sciences
Massey University – Albany
North Shore 102-904, Auckland, New Zealand
email: {k.a.hawick,h.a.james,c.scogings}@massey.ac.nz

25 January 2005

ABSTRACT

We discuss algorithms and methods for classifying the clusters of model animals that emerge from simulations of collective behaviour in artificial life models. We show how important statistical properties for understanding scaling and universal growth can be measured from complex and chaotic model systems. We describe animal clustering algorithms and the difficulties involved in automatic tracking of herds that move and change shape, orientation and size in time. We present some heuristic rules for semi-automated classification over time and some preliminary results from our study of a predator-prey multi-agent model.

KEY WORDS
classification; clustering; chaotic and complex systems; multi-agent systems.

1 Introduction

The problem of identifying clusters in spatial data is a well studied one in geographical, social and physical science applications [3,4,6]. We have employed cluster classification methods in analysing the model configurations that arise in simulations of microscopic artificial life models. Our main model involves a range of different species that live, move around, reproduce and die in a two-dimensional or three-dimensional world, interacting with one another. In this paper we focus on two-dimensional "flatland" models and simplified model dynamics that derives from fixing the life-form species present, without incorporating species evolution. The model is essentially a predator-prey system where individual animals behave according to coded rules and our main interest in it is in studying the complex structures and patterns that emerge from the collective behaviours of animals.

The typical model configuration shown in figure 1 arises from a particular set of parameter choices and initial random animal placements. We are interested in the statistical averaged behaviour measured over many different starting configurations, but with the same model parameters. The model can be run on a parallel computer and many different time sequences generated (see [9]). It is useful to make movies of these sequences and by doing so we have already identified several interesting emergent behaviour patterns such as defensive spiralling and various attack formations [5]. To gain a more quantitative characterisation and hence understanding of the emergent patterns we need to be able to identify them and plot statistical measurements. One useful measure is the simple one of cluster size or radius. Others include shape-moments describing orientation of the emergent patterns from a herd of prey under attack for example. The main problem in automating the error-prone process of measuring individual patterns or clusters by eye is in deciding how to track clusters as they evolve in time. Clusters form, move, change shape and orientation, and may dwindle away over the course of a simulation run as the animals die or flee. We aim at codifying the necessary heuristics to enable an

automated software program to track and record such data and build up the necessary statistical trends over many simulated runs. Managing the parameters to control simulation input and data storage is in itself not a trivial task [8]. Tracking fickle emergent [2] clusters is even harder.

Although the reported work here is restricted to classifying clusters in our simulation model data, we envisage the methods we describe will be of wider interest to scientists who have time dependent spatial data whose clusters have finite lifetimes.

In this paper we give a brief description of the sort of complex systems microscopic model we are studying in section 2. We present our main methods and ideas for manual and semi-automatic cluster classification in section 3. Section 4 outlines some of our preliminary growth scaling results, and we discuss the wider ramifications and uses of our classification approach in section 5

2 Microscopic Animal Models

The topic of Artificial Life has been quite popular in the literature recently (e.g. [12]) and many systems for studying it have been developed, including [1, 11, 14].

Our experimental model is a simple predator-prey model; the agents nominally consist of prey ("rabbits") and predators ("foxes"). Rabbits are considered to have an unlimited amount of food or "grass". In contrast our foxes only predate rabbits; they do not eat grass in the model. A two-phase randomized update method is used to evolve the system between discrete time-steps. As reported in [9], this allows us to establish a well-defined movement phase where only spatial positions are changed and a number-changing phase where animals are born or die.

Our model is based on an open system space. Animals occupy real-space coordinates in "flatland", but these are rounded to integers for analysis. Consequently in some sense the model is an automata. We have not included an exclusion principle, meaning that more than one agent may inhabit the same space cell at the same time (this is used primarily when agents produce offspring). We find in practice there is very little "agent stacking" or multiple site occupancy. In those rare occasions where there is stacking, typically only two or three agents inhabit the same space and at most ten agents do.

Animals live on an open coordinate system space. There is no "bounding box" or array of cells. Each animal stores its own coordinates. Although we do not enforce conservation of energy in the sense that grass is always available to rabbits, we do ensure transactional semantics to ensure no rabbit is eaten more than once.

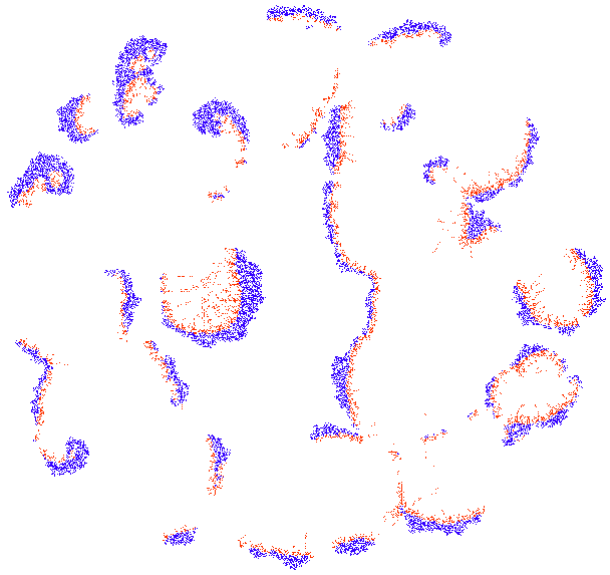
We have implemented a pseudo-sexual breeding process which requires less book-keeping than a more realistic paired breeding process. This breeding process consists of each animal that desires to breed producing a random number and checking if this number is less than *half* the current birth rate. If this is the case, a new animal is produced with the same location as the parent (thus stacking the animals). If this is not the case, no new animal is produced but breeding has (unsuccessfully) taken place for purposes of fulfilling the rule and thus no further rules are invoked in the current time step.

Animals have a list of rules which they apply in order to compute their behaviour at each time step in the model's evolution. The rabbits' first rule is to attempt to preserve their life by fleeing as soon as a fox moves into a space perceived by the rabbit as 'close'. Another important rule essentially embodies a "flocking" behaviour: when not immediately hungry both predators and prey prefer the company of other animals for the purposes of reproduction. We believe it is the combination of these two agent rules that have most directly contributed to the spiral phenomenon seen in the model configuration. A detailed discussion of the rules governing our system can be found in [9]. We discuss our earlier artificial life prototypes in [7] and the frameworks required to manage the large amounts of experimental data produced in such simulations in [8].

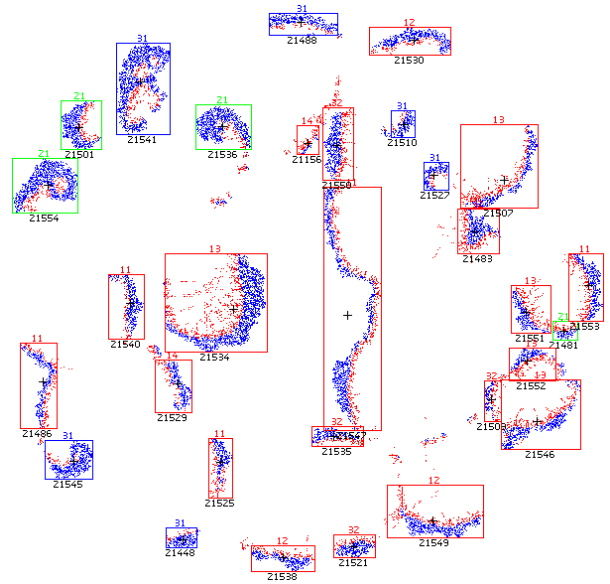
Figure 1a) shows a typical snapshot of our main model. In it predator animals are shown pursuing the prey, while the prey are fleeing the predators. When not otherwise engaged, both predators and prey cluster together. Interesting phenomena emerge when clusters ("blobs") of animals are predated. This leads to the emergence of the defensive spirals, reported in [5], that often rotate around an axis.

Figure 1b) shows the same system state when it has been processed by the algorithms described in this paper: groups of animals are classified by their group shapes and are able to be tracked across successive steps in the simulation's history.

These models are formulated in terms of quite simple rules, yet they display a rich emergent set of features and behaviours. Classifying the patterns is the key to gaining a quantitative understanding of the model's universal properties.



a)



b)

Figure 1: a) Rendering of a system state in our artificial life "flatland" model. Red (light grey) predators are pursuing blue (dark grey) prey. b) Output of our analysis program. Clusters of animals are identified and given a cluster number (shown beneath each cluster). A rectangular bounding box is drawn around the members of the cluster, coloured according to how the cluster is classified and a classification ID is printed above each cluster: classification IDs starting with '1' indicate a narrow front of animals, those starting with '2' indicate a spiral formation and those starting with '3' indicate a 'blob' of animals. A small cross-hair identifies the centre of mass for each cluster.

3 Classification methodology

Through observing the evolution of groups of animals in our model we have been able to manually categorise most of the structures exhibited by long runs of the simulation. In order to automate this process we had to make some simplifying assumptions. One of these assumptions was that we could convert the animals, which live in a "real number (x,y) space" into an integer lattice-based space. Converting the representation of our animals into this integer spaced lattice eases the production of images (necessarily pixel-based) and also simplifies the process of clustering related animals.

As described above, the animal model has been adapted to generate integer x-y coordinates for every animal at each time step. This produces a sequence of files called the step files. The `MakeClusters` program reads the step files and recognises the clusters within each Step file. The algorithm for the `MakeClusters` program is as follows:

1. Clusters are formed using the Eardley [13] sweeping algorithm. Each cluster is given a unique ID and clusters are sorted into ID order.

2. The following characteristics are calculated for each cluster:

   count number of animals in the cluster

   minx, maxx minimum and maximum x values in the cluster

   miny, maxy minimum and maximum y values in the cluster

   cx, cy average x and y coordinates (i.e. the cluster centre of gravity)

   deminrad minimum distance from some animal to (cx, cy) (often zero)

   maxrad maximum distance from some animal to (cx, cy)

   avgrad the average distance for all animals to (cx, cy)

3. To classify each cluster we repeat the following steps:

   (a) For all clusters:

```
dx = maxx - minx; dy = maxy - miny;
if (dy >= 1.8 * dx)
  cluster is a vert wave front
  (code 11)
if (dx >= 1.8 * dy)
  cluster is a horiz wave front
  (code 12)
```

(b) For all clusters not assigned code 11 or 12: use (cx, cy) to divide the cluster into quarters and count animals in each quarter. quarters are checked to see what percentage of animals are contained within them (e.g. if 120 animals in the cluster then a quarter containing 40 would contain 33%.)

```
if (one quarter >= 28% and
    other three >= 18%) then
  cluster is a spiral
  (code 21)
if (only one quarter <= 10%) then
  cluster is a diag front
  (code 13)
if (two or more quarters <= 10%) then
  cluster is a v.thin diag front
  (code 14)
```

(c) For all clusters not assigned code 21: (i.e. blobs can override wave fronts)

```
calculate inverse density of cluster:
  inverse = area / count
if (inverse < 3.2) then
  count the rabbits in cluster and
  if (rabbits >= 80%) then
     cluster is a rabbit blob
     (code 31)
  else cluster is a general blob
     (code 32)
```

(d) For all clusters not yet assigned a code: cluster is undefined (code 0)

4. The Cluster files are created. There is one Cluster file for every Step file.

All clusters are stored in the cluster file but only clusters with $count >= 100$ are studied further. The `ConnectClusters` program is used to connect clusters through a series of time steps. It creates a single trace file for the selected series. Every cluster X in time step i is matched up with cluster Y in time step $i+1$ such that the centre of X is closer to the centre of Y than to any other cluster in time step $i+1$. The ID of Y then becomes the nextID of X.

Figure 2 illustrates this basic problem we encounter in automatically classifying clusters of simulated animals as they move, breed and die in the model space of our artificial life simulations. The figure shows three time frames where we suppose four clusters of animals can be initially identified. After some simulated time animal clusters might have grown or re-oriented themselves spatially. We need to be able to re-identify a previously identified cluster as the same cluster even if it

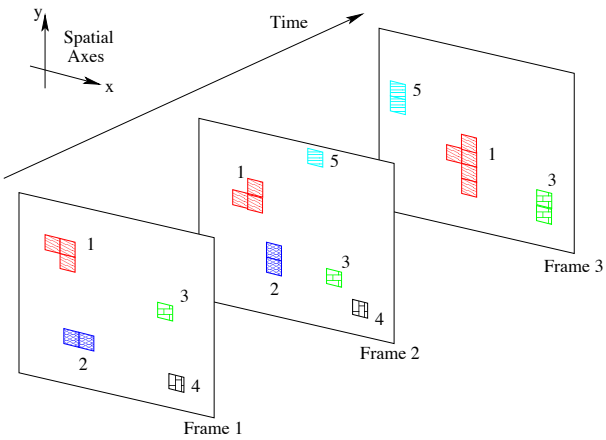


Figure 2: Three time frames showing animals moving spatially, clustering together and some new ones being created. Cluster 1 and 2 in the first frame become cluster 1 in the final frame. Animals and clusters need to be uniquely identified in order to be tracked. Various closeness and movement heuristics are needed to decide which cluster is which in subsequent time frames. Heuristics are not necessarily time-symmetric.

is in a different position, is oriented differently or has grown or shrunk slightly. Our algorithm is able to identify cluster 1 in all three frames as being the same cluster, despite the fact that it has moved and re-oriented itself. Likewise the addition (birth) of new animals in frames 2 and 3, and their coalescence into a cluster needs to be recognised by the classifier. Identifying every animal entity (primitive) in the data set by a unique integer allows us to use various conventions such as labelling (uniquely) each cluster by the smallest integer label of any animal in that cluster. The cluster matching process works extremely well except for the following three cases:

1. A cluster disappears (or becomes too small to be of interest). This can be identified by finding two clusters that have the same nextID in the following time step. The cluster closest to nextID is retained in the trace and the other cluster is declared to be dissipated.

2. Several clusters *merge* to become a single new cluster. The difference between the animal counts in each cluster and its nextID cluster is calculated. If the difference (increase) is greater than 100 animals then a *merge* may have taken place. A box is placed around the nextID cluster and a search is made for any cluster declared *dissi-*

pated (above) such that the borders of the *dissipated* cluster would fit within the box. If such a cluster is found, it has its status changed from *dissipated* to *merging*. Several *merging* clusters may be found in this way.

3. A cluster *splits* into several smaller clusters. The difference between animal counts in the nextID cluster and each cluster is calculated. (This is the opposite difference to that used in merging above). If the difference (decrease) is greater than 100 animals then a *split* may have taken place. A box is placed around the cluster and a search is made for any clusters in the next time step that have not been assigned as the nextID for some cluster and would fit within the box. If such a cluster is found, it becomes another nextID for the cluster in question.

A single trace file is produced to show the connections between clusters. For each time step all clusters are listed in the file with the nextID(s) and the *merge* or *split* or *dissipate* status of the cluster.

Statistically our problem is to be able to infer properties for typical clusters average over many different simulation random starting conditions and therefore with different detailed clusters, but with the same controlling parameters.

We see this tool as being far more useful, in a general sense, than simply for this artificial life application. Through the course of our other research we have generated many two-, three- and many-dimensional datasets that are in need of cluster processing. This programme, when modified to load different simulation configuration types, will aid our understanding of data with both global and local clustering characteristics.

4 Cluster Growth Results

In this section we describe and discuss some of the preliminary data that our analysis toolkit has been able to identify. Two of the fundamental questions of this work are: is there a law that governs the growth of populations within each of the major categories of populations we have observed, and also how far do clusters of populations move over time?

In the early time steps of our simulation most of the clusters of populations are of less than 100 animals, making their classification quite difficult with our current system.

Figure 3 shows the motion trails for all clusters of animals (populations greater than 100) in our sample

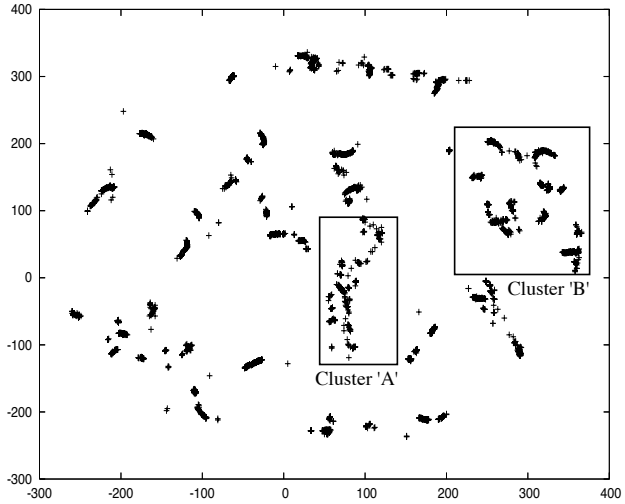


Figure 3: Motion trails of clusters (size larger than 100 animals) over the last 100 steps of the simulation. Points represent the centres of mass for each identified cluster in each time step. This image corresponds with the images in figure 1. Axis labels are included to give a sense of the amount that cluster moves in time. Cluster 'A' is shown in figure 1 and the growth and decay of cluster 'B' in detail in figure 4.

sequence. While these motion trails only represent a sample of the last 100 time-steps, it is quite interesting to observe that the centre of mass of most clusters observed in figures 1a) and b) hardly move at all. In fact, our program calculates that the average movement of the centre of mass of the 28 largest clusters is a distance of approximately 2.8 spatial units.

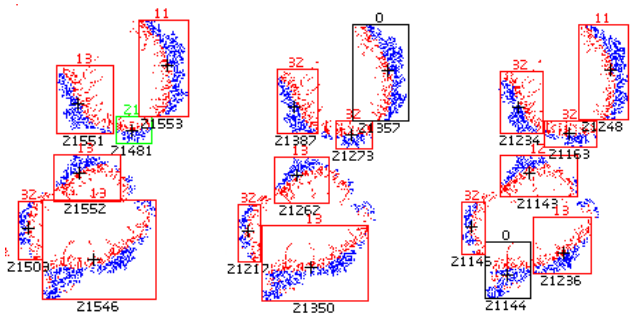


Figure 4: Sequential analysis of animal states show how clusters form, disperse and rotate. For example, cluster 21546 can be seen dispersing into clusters 2144 and 21236, and cluster 212481, originally identified as a spiral, decays into a 'clump' with ID 21163. Finally, cluster 21551, identified as a narrow front of predators and prey, degenerates into what is classified as a clump in cluster 21234.

Figure 4 shows a sequence of three different animal state files at steps 998, 1000 and 1002 of the sequence under consideration. From left to right it can be seen that the large cluster at the bottom of the figure is breaking into smaller clusters, while the cluster at the top left of the diagram is re-classified due to movement of animals within its structure.

Figure 5 shows the populations of four different types of cluster identified by our system. The top curve represents a rabbit clump, the next top represents the cluster population size for a typical spiral. We are attempting to understand the temporal growth of these features in terms of scaling analysis [10]. A simple linear fit gives a slope of 18 for the clump, 15 for the spiral. These formations seem to be more robust against predation by predators through introducing more layers of prey between the 'average' rabbit and the foxes.

Horizontal and vertical wavefronts, by their nature, are thinner structures formed by the spreading out of the rabbits when attacked by foxes. In some cases this is caused by a small group of foxes breaking through a line of rabbits, dissuading them from forming a rabbit clump, but instead causing them to run 'down the line' to escape predators. This can cause a long thin line of

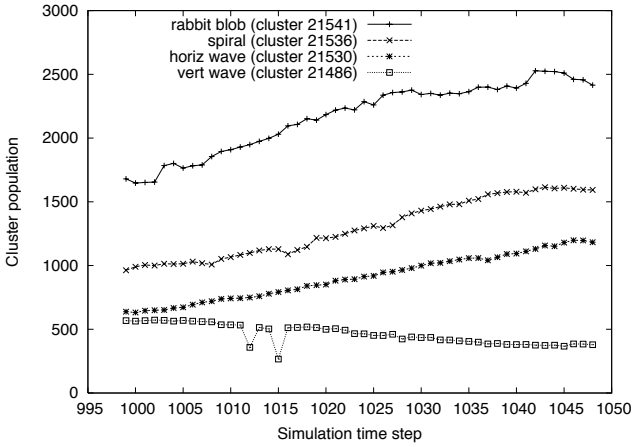


Figure 5: Population sizes of representative clusters identified in figure 1b) against simulation time. As a general rule, rabbit clumps and spirals tend to increase in population, while the horizontal and vertical wavefronts are more susceptible to erosion by predation.

rabbits to emerge and the foxes to spread out even more to catch them. Sometimes our system re-classifies a spiral that has flattened out as a horizontal, vertical or diagonal wave if the majority of the spiral's population shifts outside the critical threshold.

The curve representing the spiral formation also features two troughs. These drops in population are caused by the population moving in such a way to cause the cluster analysis program to mistakenly re-classify the cluster as two smaller clusters, before re-classification as a spiral.

5 Summary and Conclusions

Our method is useful because it allows us to approach the understanding of our complex simulation using bulk (or *derived*) properties: we are able to abstract over individuals in the population and look at general behaviour without getting overwhelmed by details. In contrast, the method also allows us to focus in on any interesting features and further identify those participants in interesting phenomena. We are currently developing an extension to this toolkit to allow us to track and highlight individuals which exhibit certain behaviours in movie sequences of this data.

Our toolkit allows for the semi-automated classification of bulk features in our model, which is far more

reliable (and faster) than manual processing. However, we have found the quality of classification results to be quite sensitive to the heuristics that we use to distinguish between classifications.

By far the most complex part of this toolkit is attaining continuity of cluster ID between successive timesteps, especially in the presence of clusters that spread out too much in one time step (dissipating the cluster) and then regroup in the next (this re-forming the cluster). Also difficult is the problem of deciding which cluster gets to retain the old ID when it splits into two smaller clusters. This problem is compounded by our desire to be able to treat time as bi-directional and step backward as well as forward in order to investigate interesting phenomena in terms of their statistical entropy.

Our statistical results look promising in that we are starting to understand the correlated behaviours between different artificial animal species in our agent models.

We are currently extending this toolkit in order to apply it to the results of a computational physics simulation; we anticipate it will be incredibly useful in the bulk processing of this data.

Acknowledgements

The authors gratefully acknowledge the contribution of Helix supercomputer time by Massey University and the Allan Wilson Centre in the production of the data reported in this paper.

References

[1] Christoph Adami. Introduction to Artificial Life. Springer-Verlag, 1998. ISBN 0-387-94646-2.

[2] Peter Cariani. Emergence and Artificial Life. In J.D.Farmer C.G.Langton, C.Taylor and S.Rasmussen, editors, Artificial Life II, SFI Studies in the Sciences of Complexity, pp 775–797. Addison Wesley, 1991. ISBN 0-201-52571-2.

[3] A.D. Gordon, "A Review of Hierarchical Classification," Journal of the Royal Statistical Society, Series A (General) **150**(2): 119-137, 1987.

[4] A.D. Gordon, Hierarchical Classification. Clustering and Classification in "Clustering and Classification, " P. Arabie, L.J. Hubert and G.D. Soete (Eds), World Scientific Publications, River Edge, NJ, pp 62–122, 1996.

[5] K.A. Hawick, C.J. Scogings and H.A. James. "Defensive Spiral Emergence in a Predator-Prey Model," in Proc. Complexity 2004, Cairns, Australia, Dec. 2004, pp 662–674, Editors: Russel Stonier and Qinglong Han and Wei Li.

[6] A.K. Jain and R.C. Dubes, "Algorithms for Clustering Data", Prentice Hall, Englewood Cliffs, NJ, 1988.

[7] H.A. James, C. Scogings and K.A. Hawick. "A Framework and Simulation Engine for Studying Artificial Life," Research Letters in the Information and Mathematical Sciences ISSN 1175-2777, Information and Mathematical Sciences, Massey University, Albany, North Shore 102-904, Auckland, New Zealand, May 2004. CSTN-007.

[8] H.A. James and K.A. Hawick. "Distributed Scientific Simulation Data Management," To appear in Research Letters in the Information and Mathematical Sciences ISSN 1175-2777, Information and Mathematical Sciences, Massey University, Albany, North Shore 102-904, Auckland, New Zealand, May 2004. CSTN-008.

[9] H.A. James, C.J. Scogings and K.A. Hawick. "Parallel Synchronisation Issues in Simulating Artificial Life," in Proc. 16th IASTED Int. Conf. on Parallel and Distributed Computing and Systems (PDCS), pp, 815–820, Boston, November 2004.

[10] Leo P. Kadanoff. *Statistical Physics Statics, Dynamics and Renormalization*. World Scientific ISBN 981-02-3764-2

[11] S. Levy, *Artificial Life The Quest for a New Creation*. Penguin. ISBN 0-14-023105-6, 1992.

[12] Jean-Arcady Meyer and Stewart W. Wilson, editors. From animals to animats : proceedings of the First International Conference on Simulation of Adaptive Behavior, volume 1, Paris, France, 1990. MIT Press, Cambridge, Mass.

[13] W.H. Press and B.P. Flannery and S.A. Teukolsky and W.T. Vetterling, Chapter 8 "Sorting" in Numerical Recipes in C, pp 329-346, Cambridge University Press, 1988.

[14] T.S. Ray. An approach to the synthesis of life. Artificial Life II, Santa Fe Institute Studies in the Sciences of Complexity, xi:371-408, 1991.