

DISCRETE DYNAMICS LAB

Cellular Automata - Random Boolean Networks.

July 1995 (copyright © Andrew Wuensche 1993)

Santa Fe Institute
and The University of Sussex (COGS)
wuensch@santafe.edu
andywu@cogs.susx.ac.uk

Index.

Overview.

- A1. Introduction
 - A3. Presentation options
 - A4. Filing and Printing
 - A5. Mutations
 - A6. Statistical measures and data
 - A7. Reverse algorithms
 - A8. Sculpting attractor basins
 - A9. Hardware and software requirements
 - A10. References
-

Quick Start Instructions

1. Unzipping and running.

- 1.1. Updates to DDLab.
-

2. The user interface.

- 2.1. Input.
 - 2.2. Backtrack.
 - 2.3. Changing the graphics set-up.
-

3. Quick start examples

- 3.1 Basin of attraction fields.
 - 3.2. "Backwards" space-time patterns, and state-space matrix
 - 3.3. Basins of attraction fields for a range of network sizes
 - 3.4 A single basin of attraction.
 - 3.6. A subtree.
 - 3.7. Space-time pattern only, 1d
 - 3.8. Space-time pattern only, 2d (game of life).
 - 3.9. "Screen-saver" demo.
-

Reference Manual

4. Starting DDLab, memory, and the user interface.

- 4.1. Title bar.
 - 4.2. DDLab version, graphics and mouse status.
 - 4.3. Memory available.
 - 4.4. Virtual Memory.
 - 4.5. Prompts.
-

5. The first prompt (and graphics setup).

- 5.1. Field, or a run requiring a seed.
- 5.2. Graphics set-up.
 - 5.2.1. VGA and SVGA.

- 5.2.2. Black or white background.
- 5.3. Options.
- 5.4. Random number seed.
- 5.4.

Exit

DDLab.

6. Network size, 1d.

- 6.1. Setting range of sizes.
 - 6.2. Setting one network size.
 - 6.3. Network size limitations.
-

7. Cell scale.

8. Neighbourhood.

- 8.1. Setting neighbourhood size or mix.
 - 8.2. Setting the neighbourhood mix.
 - 8.3. Loading a neighbourhood mix file.
 - 8.4. Setting the neighbourhood mix by hand.
 - 8.5. Setting the neighbourhood mix at random.
 - 8.6. Reviewing the neighbourhood mix.
 - 8.6.1. Reviewing the neighbourhood mix in large networks.
 - 8.6.2. Jumping to a new cell index.
 - 8.6.3. Saving the neighbourhood mix.
 - 8.6.4. Other options to review and alter the neighbourhood mix.
-

9. Wiring.

- 9.1. Network geometry
 - 9.2. The pseudo-neighbourhood.
 - 9.3. Setting the wiring.
-

10. Quick wiring settings.

- 10.1. Loading the wiring scheme.
 - 10.2. Regular 1d wiring.
 - 10.3. Regular 2d wiring
 - 10.3.1. Conserving wiring memory in regular 2d networks
 - 10.3.2. Setting 2d network size.
 - 10.4. Random 1d wiring.
-

11. Special wiring.

- 11.1. Set network geometry.
 - 11.2. Set special wiring.
 - 11.3. Special wiring, regular 1d (treat as random).
 - 11.4. Special wiring, regular 2d.
 - 11.5.
 - Special wiring, random.
 - 11.5.1. Confining random wiring to a set zone.
 - 11.5.2. Self wiring.
 - 11.5.3. Distinct wiring.
 - 11.6.
 - Wiring by hand.
 - 11.6.1. Wiring by hand in large networks.
-

12. Reviewing Wiring.

- 12.1. The wiring matrix, and 1d or 2d wiring graphic.
- 12.2. The wiring matrix.
 - 12.2.2 Revise wiring, wiring matrix.
 - 12.2.3 Revise by hand.
 - 12.2.4 Randomly rewire.
 - 12.2.5 Reset wiring options.

13. Rules.

- 13.1. The rule-table.
 - 13.1.2. The totalistic code table
 - 13.1.3. Totalistic rules.
- 13.2. Rule table matrix.
- 13.3. Single rule or rule mix.
 - 13.3.1. A rule mix with just one rule.
- 13.4. Rule mix.
- 13.5. Rule mix options.
 - 13.5.1. Rule mix with no limit.
 - 13.5.2. By hand.
 - 13.5.3. Majority.
 - 13.5.4. Majority + end bits flipped.
 - 13.5.5. Random.
 - 13.5.6. Random + canalyzing.
 - 13.5.7. Setting a limited sub-set of rules for random selection.
- 13.6. Setting the selection method for a single rule (by hand).
 - 13.6.1. Methods for setting a rule.
 - 13.6.2. The rule window.
- 13.7. Setting the majority rule.
- 13.8. Setting the "game of life" rule.
- 13.9. Setting the rule in decimal.
- 13.10. Setting the rule at Random.
- 13.11. Setting the rule as bits.
 - 13.11.1. Setting bits with the mouse.
 - 13.11.2. Setting bits from the keyboard.
- 13.12. Setting the rule in hex.
- 13.13. Repeating the last rule.
- 13.14. Loading Rules.
 - 13.14.1. Single rule file encoding.
 - 13.14.2. Conflicts in neighbourhood size between network and file.
- 13.15. Rule mix selection by hand.
 - 13.15.1. Changing the cell index or selection method.
 - 13.15.2. Changing the cell index.
 - 13.15.3. Changing the rule selection method..
- 13.16. Automatic saving of last rule.
- 13.17. Transforming the rule, and setting canalyzing inputs.
 - 13.17.1. *Complimentary* transformation.
 - 13.17.2. Equivalent rule by *negative* transformation.
 - 13.17.3. Equivalent rule by *reflection* transformation.
 - 13.17.4. Equivalent rules with a greater neighbourhood, size, k .
 - 13.17.5. Setting canalyzing inputs.
 - 13.17.6. Saving the transformed rule.

14. Reviewing wiring/rules.

- 14.1. 1d or 2d wiring graphic and rule scheme.
- 14.2. Wiring graphic, 1d.
 - 14.2.1. Moving or jumping between cells, 1d wiring graphic.
- 14.3. Wiring graphic, 2d.
 - 14.3.1. Moving or jumping between cells, 2d wiring graphic.
- 14.4. Options, 1d and 2d wiring graphic.
 - 14.4.1. Revising the wiring.
 - 14.4.2. Revising the rule.
 - 14.4.3. Computing the (weighted) average λ and Z parameters.
 - 14.4.4. Learning pre-images without attractor basins.
 - 14.4.5. Printing the wiring graphic window.
 - 14.4.5. Options.

- 14.5. Revise wiring with the 1d or 2d wiring graphic.
 - 14.5.1 Changing the neighbourhood size.
 - 14.5.2 Hand rewiring.
 - 14.5.3. Random wiring.
 - 14.5.4. Random special wiring.
 - 14.5.5. Regular 1d wiring.
 - 14.5.6. Regular 2d wiring.
 - 14.6. Revise rules with the 1d or 2d wiring graphic.
 - 14.7. Rule/wiring scheme: saving, loading, or printing (to a printer or file).
 - 14.7.1. Save/load/print selection.
 - 14.7.2. Mixed rules and non-local wiring.
 - 14.7.3. One rule and non-local wiring:
 - 14.7.4. Mixed rules and local wiring,
 - 14.7.5. Printing to a printer or file.
 - 14.7.6. Wiring/rulemix file names.
 - 14.7.7. Wiring/rulemix encoding..
 - 14.7.8. Wiring/rulemix file loading conventions.
 - 14.7.9. Neighbourhood mix loading conventions.
-

15. Seed.

- 15.1. Setting the selection method for the seed.
 - 15.2. Seed bit pattern display.
 - 15.3. Rotating the seed.
 - 15.4. Setting the seed in decimal.
 - 15.5. Setting the seed at random.
 - 15.6. Setting the seed in hex.
 - 15.7. Setting the seed set as bits, drawing with the mouse.
 - 15.7.1. 1d networks (shown as 1d).
 - 15.7.2. 1d networks (shown as 2d).
 - 15.7.3. 2d networks.
 - 15.7.4. Setting/deleting bits with the mouse or keyboard.
 - 15.8. Repeating the seed..
 - 15.9. Loading a seed.
 - 15.9.1. Seed file encoding and loading conventions.
 - 15.9.2. Seed file, 1d.
 - 15.9.3. Seed file, 2d.
 - 15.10. Saving a seed.
-

16. Output parameters

- 16.1. The first output parameter prompt.
 - 16.1.1. Accept all output parameter defaults.
 - 16.1.2. Space-time patters only.
 - 16.1.3. Restore all output parameter defaults, or just layout defaults.
 - 16.1.4. Jump to layout of attractor basins.
 - 16.1.5. Jump to display of attractor basins, and data
 - 16.1.6. Jump to mutation.
-

17. Set various output parameter options.

- 17.1. State-space matrix scatter-plot.
 - 17.1.1. Display all states.
 - 17.1.2. Display attractor states only.
 - 17.1.3. Change matrix size.
 - 17.1.4. Change start colour.
- 17.2. In-degree frequency histogram.
 - 17.2.1. In-degree frequency cut-off.
 - 17.2.2. In-degree frequency window.
 - 17.2.3. Data shown.
 - 17.2.4. Options.
 - 17.2.5. Re-scaling the x -axis.

- 17.2.6. Re-scaling the y-axis.
 - 17.3. States with majority of 1s or 0s.
 - 17.4. "Screen save" demo.
 - 17.5. Graphs: G-density, Z and λ ratio.
 - 17.5.1. G-density against Z and/or λ ratio.
 - 17.5.2. Z against λ ratio.
 - 17.6. "Backwards" space-time patterns.
 - 17.7. Scroll space-time patterns.
-

18. Graphic conventions for attractor basins.

- 18.1. Network states, nodes.
 - 18.2. Attractor cycles.
 - 18.3. Transient trees.
 - 18.4. Transient tree colors.
 - 18.5. Transient trees for the states all 0s or all 1s.
 - 18.6. Subtree only.
-

19. Layout of attractor basins.

- 19.1. The layout preview.
 - 19.2. The first layout prompt.
 - 19.2.1 Reset all layout defaults.
 - 19.2.2 Reset layout for learning.
 - 19.2.3. Basin scale, attractor radius.
 - 19.3. Basin start position.
 - 19.4. Show the field as successive basins.
 - 19.5. Basin spacing for fields.
 - 19.6. Select minimum right border width.
 - 19.7. Amend the layout after each basin or tree.
 - 19.7.1. Amend the spacing and right border after each basin.
 - 19.7.2. Amending the next position (and spacing) after each basin.
 - 19.8. Amend the spacing increase for successive fields.
 - 19.10. Revise layout parameters.
-

20. Display of attractor basins, and data.

- 20.1. Compression of equivalent CA dynamics.
 - 20.1.1. Suppress compression.
 - 20.1.2. Suppress copies of trees (and subtrees)..
 - 20.2. Node display.
 - 20.2.1. Show bit pattern nodes for 1d networks in 2d.
 - 20.2.2. Alter node label size, decimal or hex.
 - 20.2.3. Alter node size, bits.
 - 20.3. Highlight one state in each attractor.
 - 20.4. Pause, and data.
 - 20.4.1. Data on sub-trees.
 - 20.4.2. Data on a subtree from a uniform state.
 - 20.4.3. Data on trees.
 - 20.4.4. Data on basins.
 - 20.4.5. Pause after each field:
 - 20.4.6. Pause after each basin or tree:
 - 20.5. Print data to printer.
 - 20.6. Save data to file.
 - 20.7. Format of printed or saved data.
 - 20.7.1. Network data.
 - 20.7.2. Basin field data (no tree data).
 - 20.7.3. Basin field data, including tree and subtree data.
 - 20.7.4. Single basin data file.
 - 20.7.5. Subtree data file.
-

21. Mutation.

- 21.1. The first mutation prompt.
- 21.2. Flip bits.
 - 21.2.1. Alter the number of bits to flip.
 - 21.2.2. Alter the percentage of bits to flip.
- 21.3. Special rule mutation.
 - 21.3.1. Mutate by the rule or code number.
 - 21.3.2. Assign a random rule or code number.
 - 21.3.3. Mutate by bitflip.
- 21.4. Mutate wiring.
 - 21.4.1. Alter the number of wires to move.
 - 21.4.2. Alter the percentage of wires to move.
- 21.5. No pause before next mutant.

22. Run backwards for subtree, or forwards for a single basin.

- 22.1. Subtree:Run forwards before running bakwards.
- 22.2. Single basin repeat check limit.
 - 22.2.1. Alter repeat check limit, or forwards only.

23. Final options: subtree, single basin or basin of attraction field. 23.1. Local wiring.

- 23.2. Non-local wiring.
- 23.3. Partial preimage stack, local wiring.
- 23.4. Partial preimage histogram, non-local wiring.
- 23.5. The exhaustive testing reverse algorithm.
 - 23.5.1. Load exhaustive pairs.
 - 23.5.2. Computing exhaustive pairs.
 - 23.5.3. Saving exhaustive pairs.

24. Final options for space-time patterns (forwards only).

- 24.1. Color cells by value or neighborhood.
- 24.2. Pause when screen full.
- 24.3. Frozen generation size.
- 24.4. Analysis.
 - 24.4.1. Pattern density.
 - 24.4.2. Lookup frequency and entropy.
- 24.5. Analysis generation size.
- 24.6. State-space matrix scatter-plot (forwards only).

25. Changing settings "on the fly" (attractor basins).

- 25.1. Interrupting attractor basins.
 - 25.1.1. Backtrack through prompts.
 - 25.1.2. Abandoning a tree and continuing with the next tree.
 - 25.1.3. Other options.
- 25.2. Key hit attractor basin options.
- 25.3. Attractor basin complete options.
- 25.4. Further attractor basin complete options.
 - 25.4.1. Rule options.
 - 25.4.2. Seed options.

26. Changing settings "on the fly" (space-time patterns only).

- 26.1. The "Interrupt Key Index" (space-time patterns only).
 - 26.2.1. *change rule table*
 - 26.1.2. *change wiring*
 - 26.1.3. *change seed/size*
 - 26.1.4. *presentation*
 - 26.1.5. *2d space-time pattern*
 - 26.1.6. *highlight frozen cells*
 - 26.1.7. *analysis*

- 26.1.8. *state-space matrix*
 - 26.1.9. *index, pause, quit*
 - 26.2. The "Interrupt Key Index", some further details
 - 26.2.1. *change rule table*
 - 26.2.2. *change wiring*
 - 26.2.3. *2d space-time pattern*
 - 26.2.4. *highlight frozen cells*
 - 26.2.5. *analysis*
 - 26.3. Pause space-time pattern.
 - 26.4. Rule details (space-time patterns only).
 - 26.5. Further space-time pattern options .
 - 26.5.1. Screen options.
 - 26.5.2. Rule options.
 - 26.5.3. Seed options.
 - 26.5.4. Other options.
 - 26.5.5. Further options, first backtrack.
-

27. Learning, forgetting, and highlighting.

- 27.1. Highlighting states.
 - 27.2. Default attractor basin layout for learning.
 - 27.3. Selecting the wiring graphic.
 - 27.4. Selecting the learn/highlight window.
 - 27.5. Select the "target" state.
 - 27.6. Select a set of states as pre-images.
 - 27.6.1. An arbitrary list of aspiring pre-images.
 - 27.6.2. Pre-images according to Hamming distance.
 - 27.6.4. Odd or even parity.
 - 27.6.5. Repeat previous selection.
 - 27.7. Review target state and aspiring pre-images.
 - 27.8. Learn, forget, or highlight only.
 - 27.9. Learning/forgetting by wire moves or bit-flips.
 - 27.9.1. Learning/forgetting by bit-flips.
 - 27.9.2. Learning/forgetting by wire-moves.
 - 27.10. Highlighting options.
 - 27.11. Learning complete.
-

28. Filing

- 28.2. Saving and Loading
- 28.3. Changing the directory
 - 28.4. List files
- 28.5. File encoding
 - 28.5.1. SCREEN (image)
 - 28.5.2. N'HOOD MIX
 - 28.5.3. WIRING ONLY, RULE SCHEME ONLY, WIRING-RULE SCHEME
 - 28.5.4. SINGLE RULE
 - 28.5.5. SEED
 - 28.5.6. DATA
 - 28.5.7. EXHAUSTIVE (pairs)

Overview.

A1. Introduction

DDLab is an interactive graphics program for research into the dynamics of finite binary networks, from Cellular Automata (CA) to random Boolean networks, including their attractor basins (for a DOS-PC platform). The program is relevant to the study of complexity, emergent phenomena, neural computation and theoretical biology, and implements the investigations presented in (Wuensche 1992-1995).

Using a flexible user interface, a network can be set up for any architecture between regular CA (1d or 2d with periodic boundary conditions) on the one hand, and random Boolean networks (disordered CA) on the other. The latter have arbitrary connections, and rules which may be different at each site. The neighbourhood (or pseudo-neighbourhood) size may be set from 1 to 9, and the network may have a mix of neighbourhood sizes.

The program iterates the network forward to display space-time patterns (mutations are possible "on the fly"), and also runs the network "backwards" to generate a pattern's (or state's) predecessors and reconstruct its branching sub-tree of all ancestor patterns until all the "garden of Eden" states, the leaves of the sub-tree, have been reached. Sub-trees, basins of attraction or the whole basin of attraction field (referred to collectively as "attractor basins") can be displayed as a directed graph or set of graphs in real time, with many presentation options. Attractor basins may be "sculpted" towards a desired configuration. Various statistical and analytical measures and numerical data are made available.

Whereas large systems sizes may be run forward to look at space-time patterns, or backwards to look at subtrees, much smaller sizes (say less than 18) are appropriate for the basin of attraction field given that state-space grows exponentially with system size. The program's upper limit for basin of attraction fields is 32; there is no limit for single basins or sub-trees. Larger sizes may be tried, but may impose unacceptable time, memory and display constraints. Sub-trees may be generated for much larger systems for rules having a low in-degree.

The network's parameters, and the graphics display and presentation options, can be very flexibly set, reviewed and altered. Changes can be made "on the fly", including mutations to rules, connections or current state. 2d networks (including the "game of life" (Conway 1982) or any mutation thereof) can be displayed as a space-time pattern in a 3d isometric projection..

Statistical measures and data (mostly presented graphically) include: λ and Z parameters, the frequency of canalizing "genes" and inputs, rule-table lookup frequency and entropy, pattern density, detailed data on sub-trees, basins and fields, garden-of-Eden density, in-degree frequency and a scatter-plot of state-space. Network parameters, states, data, and the screen image can be saved and loaded in a variety of tailor-made file formats.

Learning/forgetting algorithms allow attaching/detaching sets of states as predecessors of a given state by automatically mutating rules or changing connections. This allows "sculpting" the basin of attraction field to approach a desired scheme of hierarchical categorisation.

#A1-10 gives an overview of the program. #1-3 give some "quick start" examples. These should be tried initially to get the flavour of the DDLab. #4-28 is a detailed reference manual. For further background on the attractor basins of CA and random Boolean networks, and their implications, refer to Wuensche (1992-1995), Kauffman (1993) and other references listed in #A10.

A2. Network parameters

DDLab has a flexible user interface for setting, viewing and amending network parameters. Parameters are set by responding to a main sequence of screen prompts, and various context dependent prompt windows, all of which allow default choices and backtracking. A flashing cursor indicates the prompt. "Return" (or the left mouse button) steps forward through the prompts, "q" (or the right mouse button) backtracks, or interrupts a run.

CA rules (or totalistic codes) are set in decimal, hex, as a lookup table bit pattern (using the mouse), at random or loaded from a file. The "game of life" and the "majority" rule can also be selected. Rules may be changed into their equivalents (by reflection and negative transformations), and transformed into equivalent rules with larger neighbourhoods. The latter is useful to achieve finer mutations of interesting rules. The frequency of canalizing "genes" or inputs in a network can be set to any arbitrary proportion, useful in modelling gene regulatory networks.

Wiring is set as regular 1d or 2d for the given neighbourhood size, at random, with a "spread sheet" to set or alter individual couplings by hand, or a wiring scheme may be loaded from a file. In most cases regular 2d wiring defines a square grid on the torus, and includes the von Neumann and Moore neighbourhoods of 5 and 9 cells. However the 7 cell regular 2d neighbourhood is wired to define a triangular grid. Non standard wiring can be constrained in various ways, including confinement to a local patch of cells with a given radius.

The network parameters can be scrutinised and amended in a 1d or 2d graphic display format, or a "spread sheet" format, showing each cell's particular wiring and rule, using the arrow keys to move between cells.

An initial network state is required as a seed for a forwards or backward run (to seed a sub-tree or single basin). A basin of attraction field does not require a seed. The seed is set in 1d or 2d, in decimal, hex, as a bit pattern (using the mouse), at random, or loaded from a file. The bit pattern method is a mini drawing program, using the mouse (or keyboard) to set cell values (0,1). It is particularly useful for 2d patterns.

A3. Presentation options

Many options are provided for the presentation of attractor basins and space-time patterns. Again, many of these settings can be changed "on the fly", including the scale and colour of space-time patterns. Cells in space-time patterns are coloured according to their value (0,1) or alternatively according to their neighbourhood at the previous time step (the entry in the look-up table that determined the cell's value). A key press will toggle between the two. Alternatively, the colour presentation can be set to highlight cells that have not changed in the previous x generations, where x can be set to any value. The development of such *frozen clusters*, which may link up into *percolating walls*, is associated with canalizing "genes", and underlies Kauffman's "random Boolean network" model of gene regulatory networks (Kauffman 1993).

A space-time pattern is presented in successive vertical sweeps, or may be scrolled, which is slower. 2d networks (including the "game of life" or any mutation thereof) can be displayed as a space-time pattern in a 3d isometric projection (as if looking up at a glass lift shaft). The projection can be toggled between 3d, 2d and 1d (which shows a series of slices).

For attractor basins, options allow the selection of the basin of attraction field, a single basin (from a selected seed), or a sub-tree (also from a seed). A random seed is likely to be a garden of Eden state. For sub-trees, an option is therefore offered to run the network forward a given number of steps to a new seed before running backward. This guarantees a sub-tree with at least that number of levels.

Options (and defaults) are provided for the layout of attractor basins, their size, position, spacing, and type of node display (as a spot, in decimal, hex or a (1d or 2d) bit pattern). Regular 1d and 2d CA produce attractor basins where sub-trees and basins are equivalent by rotational symmetry. This allows "compression" of basins (by default) into non-equivalent prototypes, though compression can be suppressed. Attractor basins can be generated for a given system size, or automatically for a range of sizes. As attractor basins are generating, a reverse space-time pattern plot, showing each state in black and its predecessors (if any) in red can be displayed (and toggled on and of).

An attractor basin run can be set to pause to see data on each transient tree, each basin, or each field. Normally the run will pause before the next "mutant" attractor basin, but this pause may be turned off to create a continuous demo of new attractor basins. A "screensave" demo option shows new basins continually growing in random positions

At any time, a space-time pattern or attractor basin run can be interrupted to pause, save or print the screen image, or backtrack through options. The graphics can be set up for VGA (640x480) or SVGA (1024x768) (in the SVGA mode there is a problem with the mouse). The colour setting in both modes can be set for either a white or black background. The default setting is VGA with a white background.

A4. Filing and Printing

Network parameters and states can be saved and loaded with default file extensions (and default file names) for the following: rules, rule-schemes, wiring-schemes, wiring/rule schemes, neighbourhood mix, and network state.

The screen image is saved and loaded using a home made compressed format. To save the image in a standard format (and print to any printer), use a "stay resident screen grabber". A number a specialist screen grabbers are available, and others are part of "paint" programs. Alternatively run DDLab as a DOS application in Microsoft Windows and use their "paint" screen grabber. The screen image can be printed at any time directly from DDLab, but reliably only to the Epson MX-82 dot matrix printer. Printing to other printers is worth a try but is unpredictable (these shortcomings will be dealt with in time).

Detailed data on attractor basins can be automatically saved (also printed). A file of "exhaustive pairs", made up of each state and its successor, can be created for very fast generation of a basin of attraction field of a random Boolean

network.

A5. Mutations

Network parameters (as well as graphics presentation) can be altered "on the fly". For example during forward iteration various key presses allow mutations to rules, wiring or current state; the network size can be increased or decreased, and the graphical display of analytical data can be changed. About 60 1d complex 5-neighbour rules (with glider interactions) are provided in a file and can be set with a button press, and also a few 6 and 7 neighbour complex rules. Examples of these are presented in (Wuensche 1994b,1995).

When attractor basins are complete, a key press will regenerate the attractor basin of a mutant network. Various mutation options can be pre-set including random bit flips in rule tables, or random moves of a given number of wires. Sets of states can be specified and highlighted in the attractor basin to see how mutations affect their distribution.

A6. Statistical measures and data

Numerical, statistical and analytical data, mostly presented graphically, includes the following: A rule-table lookup frequency histogram, which can be toggled between 2d and 3d to include a time dimension. An alternative plot of the lookup frequency histogram and its entropy. A plot of pattern density. (these may be spread over a number of generations). The lambda and Z parameters. The proportion of canalizing rules and inputs. Detailed data on attractor basins including garden-of-Eden density (this data may be automatically saved). A histogram of in-degree frequency. A scatter-plot of state space (plotting the left half against the right half of each state bit string), using colour to identify different basins, or attractor cycle states. A plot of garden-of Eden density against the Z parameter. Most of these topics are explained in (Wuensche 1994b,1995).

A7. Reverse algorithms

There are three different reverse algorithms for generating the pre-images of a network state. The first is specifically for "local" 1d wiring with a homogeneous neighbourhood, such as 1d CA, though the rules may be heterogeneous. This is the most efficient thus fastest algorithm, described in (Wuensche 1992). Furthermore, compression of 1d CA attractor basins by rotation symmetry (see #20.1) speeds up the process.

Any other network architecture, with non-local wiring, will be dealt with by a slower *general* reverse algorithm described in (Wuensche 1994a). A histogram revealing the inner workings of this algorithm can be displayed. Regular 2d CA will use this general reverse algorithm; however compression algorithms will come into play to take advantage of the many rotation symmetries on the torus (unless suppressed, and not for a neighbourhood of 7 as yet, because the symmetries of a triangular grid on the torus have yet to be resolved).

Both these reverse algorithms generate the pre-images of a state directly; their speed is sensitive to neighbourhood size as well as system size. A neighbourhood of 9, such as the game of life, is slowest. A third reverse algorithm first sets up a list of "exhaustive pairs" of each state in state-space and its successor (this can be saved). The pre-images of states are generated by reference to this list. This method is efficient for basin of attraction fields, or large basins that use up a good proportion of state-space, and additionally is not sensitive to neighbourhood size (the basin of attraction field of the 4x4 "game of life" by this method takes about 30 minutes on my 66MHz 486).

The method used to generate pre-images will be chosen by default, which can be overridden. For example, a regular 1d CA can be made to use either of the two other algorithms for benchmark purposes. The time taken to generate attractor basins is displayed, and for the basin of attraction field, a progress bar indicates the proportion of states in state-space used up so far.

A8. Sculpting attractor basins

Learning and forgetting algorithms allow attaching and detaching sets of states as predecessors of a given state by automatically mutating rules or wiring couplings. This allows "sculpting" the attractor basin to approach a desired scheme of hierarchical categorisation. When an attractor basin run is complete, various "learning" windows allow a "target" state to be set, together with a number of "aspiring pre-images" (predecessors). These may be selected in various ways including by Hamming distance. Learning/forgetting algorithms will attempt to attach the aspiring pre-images to the target state, and can be set for either rule-table bit-flips or wire moves. In fact the bit-flip method cannot fail. In this way, new point and cyclic attractors can be created and sub-trees transplanted. The result of learning/forgetting, including side effects, will be apparent in the new attractor basins. The algorithms, and their implications are described in (Wuensche 1994a).

A9. Hardware and software requirements

Platform: PC-DOS 386 or higher, with VGA or SVGA graphics; maths co-processor, mouse (optional). (ideally a fast 486 with 8MB ram). The code is written in C and developed with BorlandC, but compiled with the Watcom C32

compiler (and the Rational Systems DOS extender) giving access to extended memory. The supplied Watcom file "dos4gw.exe" must be in the same directory as "ddlab.exe".

A10. References

- Conway,J.H., (1982) "What is Life?" in "Winning ways for your mathematical plays", Berlekamp,E, J.H.Conway and R.Guy, Vol.2, chap.25, Academic Press, New York.
- Gutowitz,H.A., ed., (1991) "Cellular Automata, Theory and Experiment", MIT press.
- Hopfield,J.J. (1982) "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of the National Academy of Sciences* 79:2554-2558.
- Kauffman,S.A., (1993) *The Origins of Order, Self-Organization and Selection in Evolution*, Oxford University Press.
- Langton,C.G., (1986) "Studying Artificial Life with Cellular Automata", *Physica D*, 22, 120-149.
- Langton,C.G., (1990) "Computation at the Edge of Chaos: Phase Transitions and Emergent Computation", *Physica D*, 42, 12-37.
- Wolfram,S., ed. (1986) "Theory and Application of Cellular Automata", World Scientific, 1986.
- Wuensche,A., and M.J.Lesser. (1992) "The Global Dynamics of Cellular Automata; An Atlas of Basin of Attraction Fields of One-Dimensional Cellular Automata", Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley.
- Wuensche,A., (1994a) "The Ghost in the Machine; Basin of Attraction Fields of Random Boolean Networks", in *Artificial Life III*, ed C.G.Langton, Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley.
- Wuensche,A., (1994b) "Complexity in One-D Cellular Automata; Gliders, Basins of Attraction and the Z IParameter", Santa Fe Institute Working Paper 94-04-025.
- Wuensche,A., (1994c) "The Emergence of Memory", Cognitive Science Research Paper 346, Univ. of Sussex, to appear in "Towards a Scientific Basis for Consciousness", eds. S.R.Hameroff, A.W.Kaszniak, A.C.Scott, MIT Press.
- Wuensche,A., (1995) "Gliders in One-D Cellular Automata; some new measures of dynamical behaviour", to appear in *Complexity*.
-

Quick Start Instructions

1. Unzipping and running.

"ddlab.zip" will unzip to give the following files:

"ddlab.exe"	the program
"dos4gw.exe"	the DOS extender, giving access to extended memory.
"smalle.fon", "sserife.fon"	2 text fonts
	Files with "complex" 1d CA rules which feature interacting gliders.
"glider5.r_s"	about 60 complex 5-neighbour rules.
"glider6.r_s"	about 5 complex 6-neighbour rules.
"glider7.r_s"	about 10 complex 7-neighbour rules.
"pento.eed"	the "rpentomeno" 2d pattern, which seeds interesting "game-of-life" dynamics.
"readme.txt"	a text file (Index, Introduction and Quick Start Instructions only).
"ddlabop.doc"	a word for Windows file with the full (un-illustrated) reference manual).

Make a new directory, say "ddlab", and copy these files into it. To run the program, from this directory enter "ddlab". The "DOS extender" banner will briefly appear, and the program will start. Section 2 and 3 give brief "quick start" examples. You are encouraged to try these examples first to get the flavour of DDLab. Section 4 onwards is the detailed program reference, listing all the various options and prompts with explanatory notes.

Note that DDLab may also be set up and run as a non-Windows application under Microsoft Windows.

1.1. Updates to DDLab.

Check for the latest version of DDLab available at Alife Online at the Santa Fe Institute. Download "ddlab.zip" by ftp or www as follows,

World Wide Web <http://alife.santafe.edu/alife/software/ddlab.html>

ftp download instructions <ftp://alife.santafe.edu/pub/SOFTWARE/ddlab>

Download just the file "ddlab.txt" (48k) for details about DDLab, or the complete set of files (including the un-illustrated reference manual) contained in the binary file "ddlab.zip" (545k). Unzip this file with "pkunzip".

```
%      ftp alife.santafe.edu
name:  anonymous
password: your complete email address
ftp>  cd pub/SOFTWARE/ddlab (note "SOFTWARE" is upper case)
```

```
ftp>      binary
ftp>      get ddlab.zip
ftp>      close
```

2. The user interface.

When DDLab is run, the user interface (white background) appears, with a title bar across the bottom of the screen. A series of prompts are presented to set up the network and presentation parameters. These prompts appear either in a main sequence for the most commonly needed settings, or in various windows that automatically open up. If a mouse is detected, a mouse cursor will also appear. It is used to set bits in rule-tables and network states, and for "drawing" bit patterns in 2d networks. Otherwise it is just a pointer.

2.1. Input.

The green flashing cursor prompts for input. Enter appropriate input from the keyboard (backspace or press the left mouse button to revise) and "return" (or press the left mouse button) to accept and move on to the next prompt or routine. Just entering "return" (or left mouse button) will automatically select a default.

2.2. Backtrack.

To backtrack through the prompts (or interrupt a running routine) press the "q" key (or right mouse button). When a routine is running, such as space-time patterns or attractor basins, the mouse disappears, To interrupt the running routine enter "q" (mouse presses only work when the green flashing cursor is visible, not during the running of a routine). Then backtrack to any stage in the prompt sequence with "q" (or right mouse button), eventually to exit the program and return to DOS.

2.3. Changing the graphics set-up.

The graphics set-up can be changed between a white and black background, and VGA (640 × 480 pixels) and Super VGA (1024 × 768 pixels), given the necessary graphics card and monitor. At the first prompt on the main sequence (and at other points in the program) select "g" for graphics. The "Change graphics setup" screen will appear showing screen colours. Enter "b" to toggle the background, "g" to toggle VGA and SVGA. Note that at present the mouse cursor is invisible in SVGA, though the left and right button presses work.

3 Quick start examples

The following examples will give the flavour of the program. Accept defaults with "return" (or left mouse button) or experiment with other settings.

3.1 Basin of attraction fields.

From the first prompt keep accepting defaults with "return" (or left mouse button) until "OUTPUT PARAMETERS" appears at the top centre of the screen, and a window with various options in the top right. Enter "d" (to accept all output defaults).

A new prompt window appears top right. Enter "return".

The basin of attraction field will be generated for a 1d 3-neighbour rule, system size 10. The rule (chosen at random by default) appears in a window at bottom of the screen A window in the top right shows brief data on the field once it is generated. A progress bar below this window shows the proportion of state-space as it is used up. Vertical lines on this bar indicate the states used to seed the basins.

A prompt window appears top left. Enter "return".

Another basin of attraction field is generated, a one bit mutant of the previous rule, with corresponding information. This process can continue indefinitely (it can be set on automatic).

Enter "q" to interrupt (and backtrack).

3.2. "Backwards" space-time patterns, and state-space matrix

While the attractor basins are generating, various display settings can be changed "on the fly". These are indicated on the right of the bottom title bar. (If the attractor basin generates too fast, backtrack to slightly increase the network or neighbourhood size settings).

Enter "s" to toggle the "backwards" space-time pattern on-off, and see predecessors (pre-images) being generated (on the left of the screen) as a space-time pattern. Initially the attractor states will be displayed, then each state (black) and its set of pre-images (red). Expand or contract the scale of the space-time pattern with "e" and "c".

Enter "m" to toggle the display of the state-space matrix (or scatter-plot) in the lower right corner. This reveals interesting symmetries. Different colours represent states in different basins.

Enter "q" to interrupt (and backtrack).

3.3. Basins of attraction fields for a range of network sizes

Backtrack with "q" (or right mouse button) to the start of the program. At the second prompt, select "r" for a range of network sizes.

Enter "return" repeatedly until OUTPUT PARAMETERS, then "a" to restore all defaults, then "d" to accept all defaults.

Enter "return" to start a range of CA basin of attraction fields (same rule), for network size 5 to 12.

Enter "return" for the next mutant.

Toggle the display of the "backwards" space-time pattern and the state-space matrix as described in 3.2. Enter "q" to interrupt (and backtrack).

3.4 A single basin of attraction.

Backtrack with "q" (or right mouse button) to the start of the program.

At the very first prompt enter "s" for single basin or sub-tree.

At the "Neighbourhood size..." prompt select 4.

Keep entering "return" to accept defaults for the rule, seed etc. (The last seed and rule selected are automatically saved).

At the OUTPUT PARAMETERS prompt enter "a" to restore all defaults, then "d" to accept all attractor basin defaults.

A prompt window appears "run network backwards-b, forwards (default):", enter "return". (generating a single basin requires running forward first to uncover the attractor cycle).

Enter "return" in response to further options.

A single basin for network size 14 is generated.

Enter "return" for the next mutant.

Toggle the display of the "backwards" space-time pattern and the state-space matrix as described in 3.2. Enter "q" to interrupt (and backtrack).

3.6. A subtree.

Backtrack with "q" (or right mouse button) to the start of the program.

At the very first prompt enter "s" for forwards only.

At the "network size..." prompt select 25.

At the "Neighbourhood size..." prompt select 5.

Keep entering "return" to accept defaults for the seed etc.

At the "Select SEED..." prompt enter "return" then "a" to set all cells at random.

At the OUTPUT PARAMETERS prompt enter "d" for "space-time pattern only".

At the "run network backward-b, ..." prompt enter "b".

At the "forwards before backwards?" prompt select say 3 steps forward (from the seed) before running backward (because the seed itself is likely to be a garden-of Eden state with no pre-images).

Enter "return" at the next prompt. The subtree is generated with its seed highlighted as a bit pattern. If the seed happens to be an attractor state, the whole basin will be generated with the message "subtree=basin" in the top right data window.

Enter "return" for the next mutant.

Toggle the display of the "backwards" space-time pattern and the state-space matrix as described in 3.2. Enter "q" to interrupt (and backtrack).

3.7. Space-time pattern only, 1d

Backtrack with "q" (or right mouse button) to the start of the program.

At the very first prompt enter "s" for forwards only.

At the "cell size..." prompt select 1 (for 1 pixel).

At the "network size..." prompt select 150.

At the "Neighbourhood size..." prompt select 5.

Keep entering "return" to accept defaults for the rule, seed etc.

At the OUTPUT PARAMETERS prompt enter "s" for "space-time pattern only".

Keep entering "return" to accept space-time pattern defaults.

The space-time pattern is generated on the left of the screen. Alongside is a plot of the lookup frequency of each entry in the 5-neighbour lookup-table, and the entropy of this frequency distribution. A histogram of the lookup frequency changing at each time step is shown alongside. An "interrupt key index" on the right of the screen gives a list of key presses to change settings "on the fly".

Enter "g" a few times to change the rule to different complex "glider" rules. Try other key presses to change the rule, wiring, seed, colour, analysis, scale and size.

Enter "q" to interrupt (and backtrack).

3.8 Space-time pattern only, 2d (game of life).

Backtrack with "q" (or right mouse button) to the start of the program.

At the very first prompt enter "s" for forwards only.

Enter "return" until a "wiring:..." prompt window appears top right. Enter "2" (for regular 2d).

At the next prompt window enter "return".

The next prompt window sets the size of the 2d lattice. Accept the default size 36x36 (for a faster computer increase the size to say 50x50).

At the next prompt window enter 9 (the neighbourhood size).

At the "Select n=9 rule,..." in the main sequence of prompts, enter "f" for the "game of life" (the lookup-table of this rule will be displayed as a bit pattern..

Keep entering "return" to accept defaults for the seed etc.

At the OUTPUT PARAMETERS prompt enter "s" for "space-time pattern only".

Enter "d" to accept all space-time pattern defaults.

The 2d space-time pattern is generated in the top left of the screen. An "interrupt key index" on the right of the screen gives a list of key presses to change settings "on the fly".

Enter "t" to toggle between the 2d display and a 3d isometric projection. (imagine looking up at the ceiling of a transparent lift shaft). The "game of life" may soon settle to a stable or null state. To re-seed the network enter "k" for a new (random) central block. With luck gliders will emerge. Note that colours represent the cell's rule-table lookup rather than its value, making the 3d graphics clearer (key "3" toggles this colour setting). Other key pressed include changes to the rule, wiring, seed and scale.

Enter "q" to interrupt (and backtrack).

3.9 "Screen-saver" demo.

For a continuous show of attractor basins "boiling" on the screen.

Backtrack with "q" (or right mouse button) to the start of the program.

At the very first prompt enter "s" for single basin or sub-tree.

At the "Network size..." prompt select 10. (for a faster computer select 11 or 12).

At the "Neighbourhood size..." prompt select 4.

Keep entering "return" to accept defaults for the rule, seed etc.

At the OUTPUT PARAMETERS prompt enter "a" to restore all defaults, then "return" until the "savescreen demo.." window appears and enter "s" to accept the option. Then enter "d" to accept all further defaults.

Enter "return" until the show starts (the rule picked at random will be shown in the bottom rule-window).

Toggle the display of the "backwards" space-time pattern and the state-space matrix as described in 3.2. Enter "q" to interrupt (and backtrack).

Try the screen saver with a black background (see #2.3).

Program Reference.

4. Starting DDLab, memory, and the user interface.

To start DDLab, at the DOS prompt enter "ddlab" (make sure you are in the directory containing "ddlab.exe" and the other files listed in #1). The "DOS extender" banner will briefly appear, and the program will start. To suppress this banner enter "set dos4g=quiet" at the DOS prompt before running DDLab. The following sections list all the various options and prompts with explanatory notes.

On start-up the user interface appears as described in #2.

4.1. Title bar.

A permanent title bar is displayed across the bottom of the screen, including two reminders. Key "q" (or the right mouse button) backtracks through prompts or interrupts a run. At any time when the green flashing cursor is showing, the screen image can be printed with the "Home" key (see #A4. for printer limitations).

DISCRETE DYNAMICS LAB copyright © Andrew Wuensche 1993 backtrack/interrupt-q print screen-home

Additional reminders are displayed to the right of the title bar from time to time.

4.2. DDLab version, graphics and mouse status.

A top right window shows the date of the current version of DDLab, the current graphics setting and mouse status (found or not found), and a reminder that the left mouse button is equivalent to "return", and the right mouse button is equivalent to "q" for backtracking through prompts.

DDLab, 6 June 1995, graphics=640x480, randseed=15687 (randseed changes on each run, see #5.4)
mouse found: left button=ret, right button="q"=backtrack

4.3. Memory.

4.3.1. Memory available.

A small top centre window shows the amount of RAM available in bytes (including extended memory), for example,

mem=4146056

This is frequently updated. Other information is also displayed in this window from time to time.

4.3.2. Virtual Memory.

You may create a swap file on disk to augment RAM, using Watcom's "Virtual Memory Manager" (VMM), and use more memory than your machine actually has. When RAM is not sufficient, part of the program is swapped to the disc file until needed. The combination of the swap file and available RAM is the virtual memory.

To specify virtual memory, set the DOS environment variable, "dos4gvm" by entering the following at the DOS prompt:

"set dos4gvm= [option[#value]] [option[#value]] ..."

or "set dos4gvm=1" to accept all defaults.

the VMM options and default values are listed below,

minmem	The minimum amount of RAM managed by VMM, default 512KB
maxmem	The maximum amount of RAM managed by VMM, default 4MB
swapname	The swap file name, default "dos4gvm.swp" in the root directory.
virtualsize	The size of the virtual memory space, default 16MB.

For example "set dos4gvm=maxmem#8192 virtualsize#32768 swapname#c:\ddlab\ddlab.swp" gives 32MB of virtual memory and a swap file called "ddlab.swp" in the "ddlab" directory. Note that the virtual memory available will be displayed in the memory window in DDLab, as described in #4.3.1 above.

4.4. Prompts.

The first of a series of prompts for setting network and presentation parameters is displayed, as described below. The green flashing cursor prompts for input. Enter appropriate input from the keyboard (backspace or press the right mouse button to revise) and "return" (or press the left mouse button) to accept and move on to the next prompt or routine. Just entering "return" (or left mouse button) will automatically select a default. To backtrack through the prompts enter "q" (or press the right mouse button).

5. The first prompt (and graphics setup).

The first prompt in DDLab is presented as follows,

EXIT-q, graphics set-up-g, options-o.
forward only/single basin/subtree-s, (default field):

5.1. Field, or a run requiring a seed.

An initial choice is made to generate a basin of attraction field (the default), or any other alternative requiring a seed (the initial network state) to be specified. A basin of attraction field will be seeded automatically, starting with the null state.

Select "s" for a "forward only" run (i.e. to generate just space-time patterns) or for a single basin or subtree. In this

case a seed will be required, and will be set at a later stage in the prompt sequence (see #15).

5.2. Graphics set-up.

Selecting "g" in #5 above gives a new screen showing some titles and the colour palette, and a top right window with prompts as follows,

```
graphics driver: VGA(640x480) -> SVGA(1024x768)-g,    (this may be in reverse order)
background WHITE<->BLACK-b cont-ret
```

5.2.1. VGA and SVGA.

Select "g" in #5.2 above to toggle between VGA (640 × 480) to SVGA (1024 × 768) graphics. Your system must have the necessary monitor and graphics driver. This manual assumes VGA graphics.

5.2.2. White or black background.

Select "b" in #5.2 to toggle between a black and white background. In both cases the new color palette will be shown. In SVGA a black background may give a more legible interface. When the graphics changes are complete the program reverts to the first prompt (#5). References to colors in this manual are based on a white background. Colors on a black background may be different.

5.3. Options. If "o" is selected at the first prompt (#5), a top left window presents the following prompt:

```
image: print/save/load-p/s/l
back-q, cont-ret
```

This allows a "*.nat" screen image file be loaded and printed (see filing below). When these options are complete the program reverts to the first prompt.

5.4. Random number seed.

A new random number seed is generated (at random by the computer's timer) whenever DDLab is started. If "r" is selected at the first prompt (#5) above, a specific random number seed can be set. The following prompt appears in a top right window,

```
enter random number seed 0-65535 (default random):
```

Using the same random number seed results in the same sequence of random numbers being produced by the random number generator. This allows identical multiple runs using default random settings of network parameters such as wiring, neighbourhood mix, rules and initial state. Alternatively these setting can be loaded from previously saved files.

5.4. Exit DDLab. To exit DDLab enter "q" at the first prompt (#5). An exit warning is displayed at the centre of the screen,

```
? sure you want to EXIT-q, restart-ret:
```

Enter "q" again to exit the program and return to DOS (or to Windows if DDLab was set up as a non-Windows application in Microsoft Windows).

6. Network size, 1d.

6.1. Setting range of sizes.

It is possible to generate a series of basin of attraction fields of a given rule for a range of 1d network sizes. The following main sequence prompt is presented,

```
range of network size-r, else-ret:
```

If "r" is selected the next 2 prompts are as follows:

```
Start size, default 5:
End size, default 12:
```

Take care not to select excessively large sizes (see #6.3 below).

6.2. Setting one network size.

If "r" was not selected in #6.1, the 1d network size is set with one of the following prompts (the various values are examples),

```
Network size (length) max 32, default 10:          (if "field" was selected at the first prompt, #5)
```

```
Network size (length) max to fit 160, default 14:   (if "s" was selected at the first prompt, #5)
```

For 2d networks, the size will be set in the wiring options, so prompts #6.1-2 may be ignored.

6.3. Network size limitations.

For attractor basins it is best to try out small sizes first, especially for networks other than regular 1d CA. Much larger

systems sizes may be run "forward only" for just space-time patterns, or "backwards" to generate a subtree. The maximum sizes DDLab can support are 9999 for 1d, and 255×255 for 2d, but in practice smaller sizes are preferable. A good size for 1d space-time patterns for a clear view of other features on the screen is 150-200.

A small network size n is appropriate for attractor basins as these exist in a state-space that grows exponentially (though the program's upper limit for single basins or subtrees is as above, and $n=32$ for basin of attraction fields).

As an example, the time to generate the basin of attraction field of some 1d CA rules for a range of neighbourhood, k , and system size, n , is noted below. With some notable exceptions, the table shows the time doubling with increasing n , and by a factor of about 1.6 with increasing k (on my 66MHz 486),

```
n= 12 13 14 15 16 17 18
time= 2s 4.9s 8s 16s 32s 1m4s 2m13s k=3 rule (dec) 193
time= 3.8s 7s 13s 27s 50s 1m47s 3m37s k=4 rule (hex) e9 24
time= 5.4s 9.8s 30s 37s 1m18s 2m36s 8m46s k=5 rule (hex) b7 55 d3 d9
```

A similar exercise for random Boolean networks (wiring and rules selected at random) gave the following more erratic timings, especially sensitive to k .

```
n= 12 13 14 15 16
time= 12s 21s 40s 1m13s 3m21s k=2
time= 18s 50s 1m8s 2m43s 8m39ss k=3
time= 60s 2m8s 4m33s 34m32s 87m20s k=4
```

The timing depends on n , k , the rule, and the resulting topology of attractor basins (the greater the characteristic in degree, the longer the time). In the case of random Boolean networks, the timing also depends to the particular wiring and rule mix.

The system size that is appropriate for generating an attractor basin in a reasonable time can be inferred from the above. Large sizes may be tried, but may impose unacceptable time, memory and display constraints. Single basins, and especially sub-trees, may be generated for relatively much larger systems, especially for rules having a low in-degree.

7. Cell scale.

enter cell size in pixels, default 4: (for SVGA the default is 6 pixels)

Set the scale of a cell for space-time patterns. When running "forward only" with a system size of say 150 set this to 1. The cell scale can be changed later "on the fly".

8. Neighbourhood.

8.1. Setting neighbourhood size or mix.

The neighbourhood size, or a mix of sizes is specified,

Neighbourhood size, mixed-m, or enter 1-9 (default 3):

Enter "m" for a network with a heterogeneous mix of neighbourhoods, otherwise select the new size, which becomes the new default setting. For 2d networks the mix and size will be reset in the wiring option, so this prompt may be ignored.

8.2. Setting the neighbourhood mix.

If "m" is selected for a neighbourhood mix in #8.1 above, the following prompt appears in a top right window: (otherwise continue with "Wiring" #9.). Note that settings for particular cells in a neighbourhood mix can be changed later in the "wiring graphic" options (see #14).

set neighbourhood MIX: load-l, by hand-h, at random (default):

8.3. Loading a neighbourhood mix file.

If "l" is selected in #8.2 above, filing prompt windows will allow loading of a "*.mix" file (the default filename is mymix.mix)(see "filing" below).

If the file is too big (contains excess "mix" data), as much mix data as will fit in the network will be loaded, starting from cell index 0. If on the other hand the file is too small (contains insufficient mix data), all the data will be loaded from cell index 0, the excess cell indices will be allocated a uniform background mix with the following prompt,

```
mix loaded (10) less than net size (150)(values shown are examples)
enter background mix 1-9, (default 5):
```

8.4. Setting the neighbourhood mix by hand.

If "h" is selected in #8.1, a series of prompts allow the neighbourhood size at each cell index (starting from 0) to be set. The default is a random size between 1 and 9. "revise-r" allows backtracking to the previous cell index.

set neighbourhood size (1-9) at cell 0 (revise-r):

8.5. Setting the neighbourhood at random.

If the default (random) is selected in #8.1, an initial prompt enables the neighbourhood mix be confined within upper and lower bounds before the random mix is assigned.

set nhood bounds 1-9: lower (default 1): upper (default 9):

8.6. Reviewing the neighbourhood mix.

Finally, the neighbourhood mix is displayed in a window (or a succession of windows) with an option to revise, save or jump, for example...

mixed nhoods 1-9; net size=10, start index=9 (values shown are examples)
8527664973 (the nhood mix, from cell index n-1 to 0)
mix shown from cell index 9-0
reset-r save-s jump-j cont-ret

Enter "r" to return to the prompt at #8.4 and reset, "s" to save the neighbourhood mix (#8.6.4), "j" to jump to a new cell index (see #8.6.2), or "return" to exit the neighbourhood mix setting routine..

8.6.1. Reviewing the neighbourhood mix in large networks.

Large networks may require several windows to display the neighbourhood mix. In this case the prompt above (#8.6) includes an option "m" to see successive windows (for example),

.....
mix shown from cell index 7011-4221 (values shown are examples)
reset-r save-s more-m jump-j cont-ret

8.6.2. Jumping to a new cell index.

Alternatively, enter "j" at #8.6 to jump to any cell index in the network, which becomes the first index in the new window. The following prompt is displayed,

.....
jump to index (9800-0): (this example for a 2d network, 99x99)

8.6.3. Saving the neighbourhood mix.

If "s" is selected in #8.1, the rule mix may be saved as a "*.mix" file (the default filename is mymix.mix, see "filing" #28).

8.6.4. Other options to review and alter the neighbourhood mix.

The neighbourhood mix may also be reviewed with the "wiring graphic" (#14) or "wiring matrix" (#12) options. The "wiring graphic" options also allows individual neighbourhood sizes to be changed.

9. Wiring.

9.1. Network geometry

Networks cells are indexed and arranged in a regular space with periodic boundary conditions as shown below (where "x" represents a cell). A 1d space forms a ring of cells. A 2d space forms a regular orthogonal grid on the torus (except when the neighbourhood is 7, in this case the grid is hexagonal, though shown as orthogonal). The diagrams below illustrate the geometry of 1d and 2d networks .

1d network, (size n)

n-1 n-2 n-3 ... 2 1 0
x x x ... x x x

2d network (size n, i x j)

(column=j-1) (column=0)
| |
cell n-1-- x x x x x x --(row i-1)
x x x x x
x x x x x
x x x x x
x x x x x
x x x x x --cell 0 (row=0)

9.2. The pseudo-neighbourhood.

The wiring options define how each "target" cell is connected to other cells with respect to its notional or "pseudo-neighbourhood" of nearest neighbours. In cellular automata (regular 1d or 2d) the actual (local) neighbourhood is identical to the pseudo-neighbourhood. In a "non-local" network the set of cells that influence a given cell (its actual "neighbourhood" to use the term loosely) may be scattered arbitrarily within the network. Each scattered cell is mapped back (or wired) to one position in the target cell's pseudo-neighbourhood. This allows a CA rule to be applied equally to regular CA and to networks with non-local connections (see "Rules" #13).

In 1d networks the pseudo-neighbourhood (size k=1-9) is centred on the target cell (odd neighbourhoods have an extra cell on the right) as shown below, where "o" represents the target cell and "x" its pseudo-neighbours. Cells in the pseudo-neighbourhood are indexed k-1, k-2,0.

```

1 2 3 4 5 6 7 8 9
o ox xox xoxx xxox xx0xxx xxxoxxx xxxoxxxx xxxoxxxx

```

In 2d networks, the pseudo-neighbourhoods (size 1-9) are defined and indexed as shown below (again, "o" represents the target cell and "x" its pseudo-neighbours).

```

1 2 3 4 5 6 7(odd) 7(ev) 8 9
o x  o  x  x  xxx  xx  xx  xxx  xxx
  x  xox xox xox xox xox xox xox xox
  o  x  x  xx  xx  xx  xx  xxx
0  1  2103 2103 543 65 65 765 876
  0  0  0  10 10 10 10 210 543

```

9.3. Setting the wiring.

The first wiring prompt is displayed in the top right of the screen, and gives options for quick wiring settings in 1d or 2d, random wiring in 1d, or loading a wiring file. Alternatively, "special" wiring allows more flexible wiring methods.

```

WIRING: special-s, load-l, random-r,
regular 2d-2, regular 1d-default:

```

A series of context dependent wiring options are presented in a top right window to set the wiring, described in #10 and #11.

10. Quick wiring settings.

10.1. Loading the wiring scheme.

If "1" is selected in #9.3 above, file prompts will allow a "*.wir" file to be loaded. Loading such a file resets the network size. (see "filing" below).

10.2. Regular 1d wiring.

If the default "regular 1d" wiring is selected in #9.3, the wiring is set up as regular 1d with periodic boundary conditions. The network may be a CA, or have mixed neighbourhoods and/or rules. The remaining wiring options will be skipped.

10.3. Regular 2d wiring

If "2" is selected in #9.3, the wiring will be set up as regular 2d with periodic boundary conditions. The network may be a 2d CA, or have mixed neighbourhoods and/or rules.

As series of prompts will appear to set the 2d network size and neighbourhood mix as shown below, after which the remaining wiring options will be skipped. Any previous network size and neighbourhood setting will be superseded:

10.3.1. Conserving wiring memory in regular 2d networks

The following option restricts the subsequent run to "forwards only" for just space-time patterns.

```

forwards only (uses less memory, slower)-y:

```

If "y" is selected. the 2d network is not wired up, saving memory that would otherwise be allocated to specify the wiring. The option is useful to run just space-time patters (as for the "game of Life") if memory is limited. The "forwards only" run will be slower as each cell's 2d neighbourhood must be redefined at each pass. For a system with 8Mb RAM selecting this option should not be necessary.

10.3.2. Setting 2d network size.

The following option sets the 2d array size, which may be rectangular as well as square.

```

2-D, enter width (default 36):  height (default 36);
run 2d space-time pattern in 3d-3:

```

The maximum size is 255x255 for space time patterns only, but a smaller size is recommended to optimise speed and memory use (see Network size limitations, #6.3).

Selecting "3" will start a "space time patterns only" run as a 3d isometric projection (but this can be toggled between 1, 2 and 3d at run time). If the intention is to generate attractor basins, the 2d size should be kept very small (i.e. 3x3 for quick results, 4x4 for slower results).

The next series of prompts sets the neighbourhood size or mix as in #8.1-7.

In general, if "basin field" was selected in #5, and the 2d network size is set bigger than 32, the "basin field" setting will be changed automatically to a single basin or spacetime pattern, with the following message,

```

size too big for basin field (max 32);
changed to single basin or space-time pattern only
continue-ret:

```

10.4. Random 1d wiring.

If "r" is selected in 9.3, a wiring scheme will be assigned at random according to the previously selected neighbourhood (and mix) settings. The next prompt allows a 1d "wiring graphic" or a "wiring matrix" to be displayed, and the wiring to be reviewed and altered in various ways (see #12).

To restrict random wiring in various ways(see #11.5), select "s" for "special" wiring" in #9.3 before selecting random wiring.

11. Setting special wiring.

Selecting "s" in #9.3. allows a greater range of methods for wiring the network. The network can be set for regular 1d, 2d or random wiring as in #10.2-4, but additional options are offered for setting up, reviewing and amending the wiring. Special wiring is necessary, for instance, to set random wiring for a 2d network.

11.1. Setting network geometry.

The first option allows the network geometry to be set as 2d (instead of the default 1d). The network may subsequently be wired up arbitrarily.

Network geometry: 1d-default, 2d-2:

11.2. Set special wiring.

The next option allows various wiring settings: regular 1d, 2d or random, or for the wiring to be set "by hand" in a wiring matrix "spread sheet". Note that 1d wiring can be set for 2d network geometry (set in #11.1), and visa versa. Once the overall wiring is set, the wiring of individual cells may be reviewed and altered in the "wiring matrix" (#12.1-2) or "wiring graphic" (#14 options).

hand wire-3,

regular 2d wiring-2, regular 1d-1, default random:

11.3. Special wiring, regular 1d (treat as random).

If "1" for regular 1d wiring is selected in #11.2 above, a subsequent option allows the regular wiring to be treated as if it were "random" or non-local, allowing the regular wiring to be altered in any way (and also for wire moves to be allowed in "learning", see #27). However, the "compression" of attractor basins (see #20.1) may be retained as long as the network retains regular CA architecture. The following prompt is presented:

treat regular 1d as random wiring-1, and compress-2:

11.4. Special wiring, regular 2d.

If regular 2d is selected in #11.2, the 2d network size is first set as described in #10.3, and the neighbourhood mix as described in #8.1-6

11.5.

Special wiring, random. If random wiring (default) is selected in #11.2, a series of prompts first allow some restrictions to the random wiring before it is set.

The following options are presented in sequence (use "q", or the right mouse button, to backtrack).

confine to local periodic zone, enter zone size:

exclude all selfwiring-2, selfwire centre wire-1:

distinct wiring (no duplication)-d:

11.5.1. Confining random wiring to a set zone.

The wiring may be confined within a periodic zone of a given diameter relative to each target cell. This diameter relates to 1d or 2d according to the network geometry setting in #11.1.

11.5.2. Self wiring.

Target cells may be forced to wire to themselves (from the *central* pseudo-neighbourhood position, see #9), or self wiring can be excluded.

11.5.3. Distinct wiring.

Wiring may be made distinct, i.e. two or more positions in the pseudo-neighbourhood not wired to the same cell. If there is insufficient space for distinct wiring within the "wiring zone", as set in #11.5.1, this setting will be ignored.

11.6.

Wiring by hand.

If "3" is selected in #11.2, a blank wiring scheme is presented in the form of a matrix or "spread sheet", which may be filled in with the wiring positions for each cell's pseudo-neighbourhood wire. The wire position of the cell's pseudo-

neighbourhood, $k-1..0$, indexes columns, the cell network position, $n-1..0$, indexes rows. The 0-0 grid (or the 0-minimum n grid if the whole matrix does not fit within one window) is in the lower right corner. Thus each grid records the coupling made by a single "wire" to an arbitrary position in the network, $n-1..0$.

Move around the spread sheet with the arrow keys. Enter the position at the flashing green cursor and complete the entry by moving to another grid with an arrow key or "return". "return" on a blank or 0 gives a random position. An entry outside the network limits will be ignored. Enter "q" to finish (any undefined grids will be set to position 0).

While the wiring matrix is being set, the following prompt and reminder will be displayed in top right window.

```
hand wire/revise: jump-j
enter wiring positions 0-144 ("return" on blank/0=random)    (depending on network size)
move-arrows, more/complete-m, quit-q:
```

Enter "m" to complete or see successive windows, "j" to jump to a new cell index (see #11.5.1), or "q" exit the wiring by hand" routine. The wiring scheme can be reviewed and revised in the same wiring matrix format (see #12.2) or as a wiring graphic in 1d or 2d (see #14).

11.9.1. Wiring by hand in large networks.

Large networks may require several windows to display the wiring matrix. In this case enter "m" to see successive windows. Alternatively, enter "j" to jump to any cell index in the network, which becomes the first index in the new window. The following prompt is displayed,

```
jump to index (9800-0):                (this example for a 2d network, 99x99)
```

12. Reviewing Wiring.

12.1. The wiring matrix, and 1d or 2d wiring graphic.

The wiring that was set in "Special Wiring" (#11) or in #10.4 may reviewed and amended as a "wiring graphic" in 1d or 2d, or as a wiring matrix ("spread sheet"). One of the following prompts is presented,

```
wiring graphic-1d-1,, wiring matrix: view/print-v:          (for 1d networks)
wiring graphic-1d-1, 2d-2, wiring matrix: view/print-v:      (for 2d networks)
```

Note that for 2d networks, both a 1d and 2d "wiring graphic" is applicable..

For a description of the 1d ad 2d wiring graphic, and rewiring options, see "Reviewing wiring/rules" #14. The "wiring matrix options" are described below.

12.2. The wiring matrix.

If "v" is selected in #12.1 above, the networks wiring will be displayed as a wiring matrix in "view" mode (it may be amended subsequently as a "spread sheet"). The wire position of the cell's pseudo-neighbourhood, $k-1..0$, indexes columns, the cell network position, $n-1..0$, indexes rows (as in #11.6). The 0-0 grid (or the 0-minimum n grid if the whole matrix does not fit within one window) is in the lower right corner. Thus each grid shows the position to which each wire is coupled.

A column to the left of the cell index (coloured green) shows the out-degree (and out degree histogram) of each cell. Depending on whether or not the matrix display can fit in just one window, the following top right prompt is displayed,

```
MORE-ret, print-p, quit-q, jump to index (9800-0):  (this example for a 2d network, 99x99)
(or)  print-p, quit-q, jump to index (24-0):          (this example for a 1d network, size 25)
```

"return" exits the routine or allows further sections of the matrix to be viewed in successive windows, "j" jumps to a new cell index as in (see #11.6)."p" prints the current matrix window. Enter "q" to stop viewing.

12.2.2 Revising wiring in the wiring matrix.

Once viewing the wiring matrix is complete, the following prompt is displayed,

```
view/print-v, randomly rewire-r,
revise by hand-h, reset wiring options-q, cont-ret:
```

12.2.3 Revising by hand.

If "h" is entered at 12.2.2 above, the wiring matrix is displayed as a "spread sheet" and may be amended as described in #11.6 (the only difference is that the initial matrix is not blank).

12.2.4 Randomly rewiring.

If "r" is entered at 12.2.2, all values are randomly reset, and the program reverts to #12.2, showing the matrix in "view" mode.

12.2.5 Resetting wiring options.

If "q" is entered at 12.2.2, the program reverts to #9.3, the initial wiring prompt.

13. Rules.

13.1. The rule-table.

A network or CA rule is defined by a lookup-table or rule-table. A neighbourhood of size k has 2^k alternative configurations. These are set out in a lookup table matrix in reverse value order. Each neighbourhood is assigned an output (0,1), and the resulting bit string defines each of the 2^{2^k} possible rules. The rules may also be expressed in decimal (for a maximum neighbourhood of 5) or in hexadecimal.

For example, the $k=3$ neighbourhoods are set out thus (with the neighbourhood index on the left),

```
7.....0 - rule-table index
2 - 11110000
1 - 11001100
0 - 10101010
-----
11000001 - = rule 193 in decimal, c1 in hex.
```

The $k=5$ neighbourhoods are set out thus, (with the neighbourhood index on the left),

```
31.....0 - rule-table index
4 - 11111111 11111111 00000000 00000000
3 - 11111111 00000000 11111111 00000000
2 - 11110000 11110000 11110000 11110000
1 - 11001100 11001100 11001100 11001100
0 - 10101010 10101010 10101010 10101010
-----
11111000 01100010 10001000 00110111- = rule fc 62 88 37 in hex
```

13.1.1. The totalistic code table

"Totalistic" rules are rules where a cell's value depends only on the sum of 1s in its neighbourhood. The totalistic "code" is defined by a lookup table of $k+1$ possible totals set out in reverse value order from k to 0. Each total is assigned an output (0,1). The resulting bit string defines each of the 2^{k+1} possible totalistic codes. DDLab automatically transforms the code into the rule-table format.

For example, the $k=5$ totals are set out thus,

```
5 4 3 2 1 0 - totals
1 1 0 0 1 0 - code 50 in dec, 32 in hex = rule 3900801302, e8 81 81 16 in hex.
```

13.1.2. Totalistic rules.

The first rule prompt (in the main sequence) allows just totalistic rules to be selected:

totalistic rule-t:

If "t" is selected, subsequent rule settings (including random settings) will be confined to the totalistic rules only. DDLab automatically transforms the code into the rule-table format.

13.2. Rule table matrix.

If there is no neighbourhood mix, a graphic representation of the rule table matrix, as defined in #13.1 above, is displayed. The size of rule table relative to the neighbourhood size, k , is as follows:

```
size of lookup table.=      k = 1 2 3 4 5 6 7 8 9
                             2^k = 2 4 8 16 32 62 128 256 512
```

13.3. Single rule or rule mix.

The network may have a single rule, or a heterogeneous rule mix. A rule mix can subsequently be set at random from the whole of rule-space (i.e. "no limit"), or provided there is no neighbourhood mix, a limited subset of rules can be specified from which the random setting is made. The rule mix (or scheme) may be loaded from a "*.r_s" file (see filing below).

Provided there is no neighbourhood mix, a top right prompt presents these options,

```
RULES: single rule (default), load rule mix-l
mix: no limit-n, or set limit up to 200:
```

The "single rule" options are described in #13.6. "Rule mix" options are described below. **13.3.1. A rule mix with just one rule.**

To set up regular CA architecture that allows later manipulations as if it has a rule mix, select a rule mix with a limited subset of rules of just 1 (i.e. enter "1" at the prompt above), and set the rule subset "by hand" in #13.5.2. This results in a "rule-mix" where all rules are the same.

13.4. Rule mix.

If a rule mix is selected in #13.3 above (with or without a limit), a series of further top right prompts are presented. The first forces rules set at random to have the "all 0s" neighbourhood output set to 0:

if random, allow any rule-a, or force all 0s->0 (default):

13.5. Rule mix options.

13.5.1. Rule mix with no limit.

If "n" (for "no limit") was selected in #13.3, a prompt is presented to set the network rules individually by the various methods described in #13.6, or to set the rule mix at random (possibly with a set proportions of canalizing rules or inputs).

select by hand-h, random (default)

random + canalizing-c:

If the network has a neighbourhood mix, an additional option is offered to set "majority" rules (optionally with end bits flipped)

select by hand-h, maj-m, +flipped-f, random (default)

random + canalizing-c:

13.5.2. By hand.

If "h" is selected in #13.5.1 above, the method of rule selection is first set with prompt #13.6 below. The various methods of "by hand" selection are described in #13.6 and #13.7-13). The rule at each cell index is set, starting from cell 0. A reminder keeps track of the cell index and selection method, and it is possible to jump to another cell index or change the selection method during the procedure.

13.5.3. Majority.

If "m" is selected, in #13.5.1, the "majority" rule will be set for each cell according to its neighbourhood size. The majority rule outputs 1 for a majority of 1s in the neighbourhood, 0 for a majority of 0s. In the case of even size neighbourhoods, and a tie between 1s and 0s, the output is set according to the value of the central neighbourhood index (see #9.2).

13.5.4. Majority + end bits flipped.

If "f" is selected in #13.5.1, "majority" rules are set as above, but the "end bits" in the rule table (i.e. the entries for the neighbourhoods "all 1s" and "all 0s") are flipped (complimented). This can lead to some interesting "bi-stable" dynamics.

13.5.5. Random.

If the default is selected, rules will be assigned at random and the program will continue with #14.

13.5.6. Random + canalizing.

If "c" is selected in #13.5.1, further prompts will be presented to set at random a percentage or number of canalizing rules (referred to as "genes"), or of canalizing inputs (a given canalizing gene may have up to k canalizing inputs). Rules with the given canalizing setting will be assigned at random and the prompts will continue with #14.

The following top right prompts are presented in sequence,

set canalizing inputs-i genes-g:

According to whether "i" or "g" is entered, the next prompt selects whether to enter the canalizing inputs as a percentage of the total inputs in the network, or as a number,

canalizing inputs, percentage-p (default number): (or genes)

If "p" is entered, the next prompt gives information about the location, number and percentage of the current canalizing genes and inputs, and prompts for the new percentage as follows,

canalizing inputs = 9/42=21.4% genes = 7/14=50.0% (values shown are examples)

set new percentage canalizing inputs, accept-ret: (or number, or genes)

After the new percentage (or number) is entered, this prompt will reappear, enter "return" to accept. Note that the new percentage (or number) may not be exactly the value requested, because of the workings of the algorithm for randomly assigning the canalizing genes and inputs. More than one try may be attempted.

13.5.7. Setting a limited sub-set of rules for random selection.

(Note that the following does not apply for a neighbourhood mix)

If a limited rule-set for random selection was specified (by entering its size between 1 and 200 at #13.3), a prompt is presented to set the rules in the rule-set individually by the various methods described in #13.6 below, or to set the rules in the rule-set at random.

limited ruleset (12), select by hand-h, random (default): (depending on the "limit" setting)

If "h" is selected, the method of rule selection is first set with prompt #13.6 below. The following reminder is

displayed,

select subset of 12 rules (depending on the "limit" setting)

The various methods of "by hand" selection are described in #13.6 and #13.7-13). The rule at each "limited rule-set" index is set, starting from index 0. A reminder keeps track of the index and selection method, and it is possible to jump to another index or change the selection method during the procedure. Once the rules in the "limited rule-set" have been set, they are assigned at random to the network, and the prompts will continue with #14.

If the default (random) is selected in #13.5.1, rules will be assigned at random and the prompts will continue with #14.

13.6. Setting the selection method for a single rule (by hand).

Various methods of setting single rules (or "by hand" selection for a rule mix) are available (see #13.6.1-16). In most cases except filing, they apply equally to setting rules or totalistic codes.

If "single rule" was selected in #13.3,
or if "by hand" was selected in #13.5.1 (for a rule mix),
or if "by hand" was selected in #13.5.7 (for a limited rule-set),

then the following main sequence prompt is displayed (the precise wording is context sensitive and may vary considerably),

Select k=5 rule, empty-e, (default-random) (k=neighbourhood size)
maj-m life-f dec-d random-r bits-b hex-h repeat-p load-l:

13.6.1. Methods for setting a rule.

The rule can be set in a variety of different ways,

"e" - reset to null (all rule table entries=0), the program then reverts to the prompt above (#13.6).
"m" - set the "majority" rule.
"f" - set to the "game of life" rule (or quasi "life", if k is less than 9 but greater than 4),
"d" - in decimal (for $k \leq 5$).
"r" - at-random (the default).
"b" - as bits in a 1d or 2d graphic array using the mouse or keyboard.
"h" - in hex (in a mini "spread sheet").
"p" - a repeat of the last rule set for the given neighbourhood.
"l" - loaded from a "*.rul" file.

Once set, the rule remains current and can be amended by resetting bits (bits-b), or resetting half bytes (hex-h). For example, after setting the "majority" rule, backtrack to #13.6 and change the "end bits" in the rule table, resulting in some interesting "bi-stable" dynamics. Resetting the rule to null ("empty-e") allows a clean slate for setting the rule as bits or in hex. The various methods are described in detail in #13.7-14 below.

13.6.2. The rule window.

When a rule has been entered, it is displayed in a lower rule window. The window shows the rule as a bit pattern (for $k \leq 6$), in decimal (for $k \leq 5$), and in hex (for all k), where k =neighbourhood size. In addition the window gives the network size, and details about the rules λ and Z parameters, and canalizing inputs. This window is updated as rules are transformed or mutated.

An example of the data in a rule window is shown below:

```
■●■■■■●●
k3-rule=(dec)186 (hex)ba (values shown are examples)
Size=10 ld=0.825 ld-r=0.75 zl=0.75 zr=0.25 Z=0.75 C=1/3 **0
```

key to data

k3-rule the neighbourhood size k , the rule number in decimal and hex.
Size the network size n .
ld the λ parameter.
ld-r the λ ratio.
zl Z left.
zr Z right
Z the Z parameter.
C= $x/3$ the proportion of canalizing inputs.
**0 shows exactly which of the k inputs are canalizing (if none this is not shown),
in this case there is 1 canalizing input, at neighbourhood index 0.

13.7. Setting the majority rule.

If "m" is selected, in #13.6 above, the "majority" rule will be set. The majority rule outputs 1 for a majority of 1s in the neighbourhood, 0 for a majority of 0s. In the case of even size neighbourhoods, and a tie between 1s and 0s, the output is set according to the value of the central neighbourhood index (see #9.2).

13.8. Setting "game of life" rule.

(applicable only to neighbourhood $k \geq 5$)

If "f" is selected in #13.6 and $k=9$, the rule for John Conway's game of "life" is set.(Conway 1982) The "life" rule is as follows, where a 1-cell is alive, and a 0-cell is dead:

- Birth:* A dead cell comes to life if its neighbourhood had exactly three live cells.
Death by overcrowding: A live cell dies if its neighbourhood has 4 or more live cells apart from itself.
Death by exposure: A live cell dies if its neighbourhood has less than 2 live cells apart from itself.

The "life" option is available where $k \geq 5$ as well as the usual $k=9$ neighbourhood, for 1d as well as 2d, and non local wiring or mixed neighbourhoods. Once selected, the "life" rule is displayed as a bit pattern, and in the rule window.

13.9. Setting the rule in decimal.

(applicable only to neighbourhood $k \leq 5$)

If "d" is selected in #13.6, the rule can be specified by its decimal equivalent. The following prompt is displayed.

n=3 rule, enter 0-255 (default random):

If just "return" is entered, or a number outside the permitted range, the selection will revert to "random" as in #13.10 below. When selected, the rule is also shown in hex and as the rule table bit pattern, and displayed the rule window.

13.10. Setting the rule set at random.

If "r" is selected in #13.6, a series of random rules are displayed in decimal (for $n \leq 5$), in hex, and as the rule table bit pattern. A rule can be accepted or further alternative random rules may be generated indefinitely.

For each alternative rule, a top right window gives lambda and Z parameter details. The rule table can also be "rotated" to the left or right as if it were a periodic 1d state. The following prompt is displayed,

(rotate-l/r, another-n, backtrack-q, accept-ret)

"n" gives another random rule, "l" and "r" rotate the rule table left and right. To accept the random rule enter "return". When selected, the rule is displayed the rule window.

13.11. Setting the rule as bits.

If "b" is selected in #13.6, a bit pattern representing the current rule table is presented. This is divided into byte size chunks (and 8 byte rows for $k \geq 7$, i.e. 2 rows for $k=7$, 4 rows for $k=8$ and 8 rows for $k=9$). 0s are coloured grey, 1s are coloured red. To provide a clean slate for setting bits, reset the rule table to null at #13.6 with "empty-e". Alternatively, use this method for fine adjustments to rule tables set by other methods.

Bits in the rule table are set to 1 or 0 with the mouse or keyboard. During the bit setting procedure a top right reminder window displays the following:

keyboard: 0/1-set bit, arrows-move, accept-ret
mouse: press-set bit, keep pressed-draw, release-move
mouse buttons: left-set 1, right-set 0
rotate-l/r, accept-ret

13.11.1. Setting bits with the mouse.

The mouse arrow cursor is confined within the bit pattern area. A left button press sets bits on (to 1), a right button press sets bit off (to 0. i.e. deletes 1s). If the left mouse button is held down a line of 1 bits is drawn, holding the right mouse button down draws 0 bits.

13.11.2. Setting bits from the keyboard.

The current bit to be updated is highlighted by a small green flashing cursor, which is moved with arrow keys. The left/right/up/down arrow keys are used to displace the cursor. The up/down arrow keys only apply for rules where $k \geq 7$, where the bits are presented in 2, 4 or 8 rows. Hit keys "1" and "0" (without "return") to set or delete bits. This automatically moves the cursor one position right (the keyboard cursor is moved when bits are set with the mouse, as well as with the arrow keys).

The rule can be rotated as in #13.10. To accept the rule table bit pattern, enter "return". When selected, the rule is also displayed in decimal (for $n \leq 5$) and in hex, and in the rule window.

13.12. Setting the rule in hex.

If "h" is selected in #13.6, the hex expression of the current rule is displayed. The hex character to be updated is highlighted by a green flashing cursor, which can be moved with the arrow keys. The left/right/up/down arrow keys are used to displace the cursor. The up/down arrow keys only apply for rules where $k \geq 8$, where the hex digits are presented

in 2 or 4 rows. To overwrite, enter a hex character (1-9, a-f, without "return") from the keyboard. This automatically moves the cursor one position to the right.

A top right reminder window displays the following,

```
enter hex, arrows-move
rotate-l/r, accept-ret
```

The rule can be rotated as in #13.10. To accept the hex expression of the rule table bit pattern, enter "return". Once selected, the rule is also displayed in decimal (for $n \leq 5$) and as the rule table bit pattern, and in the rule window.

13.13. Repeating the last rule.

If "p" is selected in #13.6, the last rule that was set (for a given neighbourhood size, k) is repeated by automatically loading a "last[k].rul" file. The file was automatically saved when the last rule was set, and named according to neighbourhood size, k (for example, "last5.rul" for $k=5$). In the case of a rule mix where a sequence of rules was entered by hand, the previously entered rule is repeated. In the case of a neighbourhood mix and a previous rule with a different neighbourhood, a "last[k] rule" file is loaded (for a totalistic code a random code is assigned). The repeated rule is displayed in decimal (for $n \leq 5$), in hex and as the rule table bit pattern, and in the rule window.

13.14. Loading Rules.

(not applicable to totalistic code)

If "l" is selected in #13.6, top right filing windows are presented allowing a rule to be loaded from a "*.rul" file, the default is "myrule.rul" (see filing).

13.14.1. Single rule file encoding.

The binary file defining the rule is encoded as follows:

Byte 0 = k (neighbourhood size), the rest of the rule table (from 0 to $n-1$) is set as bits in successive bytes .

13.14.2. Conflicts in neighbourhood size between network and file.

For a single rule network, if the file is for a different neighbourhood size k , the file will be accepted (in the main prompt sequence only, not when resetting rules at a later stage). The k setting of the network will be automatically altered with a top left message, for example,

```
rule loaded, nhood changed from 5 to 9, cont-ret: (values shown are examples)
```

Once loaded, the rule is displayed in the rule window.

For a network with a rule mix (or when resetting a single rule at a later stage), a file with a different k will not be accepted, with a top left message, for example,

```
not loaded, tried nhood 9, this nhood=3, cont-ret: (values shown are examples)
```

13.15. Rule mix selection by hand.

If a rule mix is being set by hand, the "selection method" prompt above (#13.6) is repeated for each cell index (or limited rule-set index), with the following context dependent top right reminder, and options to jump to a new cell index or change the default rule selection method.

```
index 3 (k=5 rulemix), default selection=r (values shown are examples)
```

13.15.1. Changing the cell index or selection method.

If the index is greater than 0 and "q" is entered at the #13.6 prompt, the following top right prompt will allow the current cell index or selection default to be changed.

```
change: index8-i, selection-s, default start again: (values shown are examples)
```

13.15.2. Changing the cell index.

If "i" is selected in #13.15.1 above, the following prompt allows a jump to a new cell index.

```
this index=3, enter new index 0-15: (values shown are examples)
```

13.15.3. Changing the rule selection method..

If "s" is selected in 13.15.1 above, the following prompt allows the rule selection method to be changed (the precise wording is context sensitive and may vary considerably).

```
index=5 (k=9), default selection=r, to change enter (values shown are examples)
life-f random-r bits-b hex-h repeat-p load-l:
```

13.16. Automatic saving of last rule.

In the case of a single rule, the once will be automatically saved with a default name (last[k].rul depending on the neighbourhood size. For example, if the $k=5$ the file name will be "last5.rul".

13.17. Transforming the rule, and setting canalizing inputs.

In the case of a single rule, when the rule has been set, it may be complimented, transformed in a number of ways to an equivalent rule, or canalizing inputs may be set. The (changed) rule may be saved as a "*.rul" file (see filling #28) and is

displayed in the rule window.

The following top right prompt is presented.

```
transform rule: comp-c, neg-n, ref-r canalyzing-C (values shown are examples)
equivalent with greater nhood (4-9), save-s: ("greater nhood" if k<9)
```

The rule transformations and changes are as follows::

13.17.1. *Complimentary transformation.*

Enter "c" to transform the rule into its "compliment" (by flipping all entries in the rule table).

13.17.2. *Equivalent rule by the negative transformation.*

Enter "n" to transform the rule into its "negative" equivalent rule (by swapping the outputs of complimentary neighbourhood pairs in the rule table, then transforming the resultant rule table into its compliment). A negative rule and the start rule have equivalent dynamics, but with all state configurations in the space-time pattern complimented.

13.17.3. *Equivalent rule by the reflection transformation.*

Enter "n" to transform the rule into its equivalent rule by "reflection" (by swapping the outputs of pairs of asymmetric reflected neighbourhoods in the rule table). A reflected rule and the start rule have equivalent, but reflected dynamics.

Note that a negative transformation followed by a reflection transformation produces another equivalent rule, making 4 in all, though for certain rules the transformations result in identity.

13.17.4. *Equivalent rules with a greater neighbourhood size, k.*

A rule with a given neighbourhood size k has equivalent rules with any value of greater k . If $k < 9$, enter the greater value of k to transform the rule. For instance a $k=3$ rule may be transformed to a $k=5$ or $k=9$ rule. Although the larger k rule's dynamics is identical, mutations of the larger k rule can be finer grained.

13.17.5. *Setting canalyzing inputs.*

Enter "C" to set a given number of "canalyzing" inputs either explicitly or at random. The following top right prompt is presented.

```
set canalyzing inputs: explicitly-e, number at random-n
```

Enter "n" to set a given number of inputs as canalyzing at random. The following appears, enter the required number,

```
canalyzing inputs=1, set new number (0-3): ("wire" values shown are examples)
```

Enter "e" at #13.17.5 to set the canalyzing inputs explicitly, the following prompts appear in sequence,

```
wire (0-3): input value (1,0): output value(1,0): (values shown are examples)
```

Enter the neighbourhood index (wire) to be set as canalyzing, and the input and output canalyzing values (0 or 1).

13.17.6. *Saving the transformed rule.*

Enter "s" at #13.17 to save the the transformed rule (see filing). The transformed rule is displayed in the rule window.

14. Reviewing wiring/rules.

14.1. 1d or 2d wiring graphic and rule scheme.

Once the wiring is set with "Special wiring" in #11.1-5 (or in #10.4), or the rule scheme has been set in #13.1-17 above, the network's architecture may be reviewed and amended as a "wiring graphic" in 1d or 2d. (The wiring may alternatively be revised and amended as a "wiring matrix" described in #12.1-5).

The following prompt is presented,

```
learn/show/revise rulemix + wiring -1: (for 1d networks)
```

```
learn/show/revise rulemix + wiring: 1d-1 2d-2: (for 2d networks)
```

Note that for 2d networks, both a 1d and 2d "wiring graphic" is applicable.

14.2. Wiring graphic, 1d.

If "1" is selected in #14.1 above, a window opens up showing (graphically and numerically) how each cell's pseudo-neighbourhood in the 1d array is wired up, one cell at a time, to cells at the previous time step i.e., from time-

step t to $t-1$. Cells at $t-1$ are shown in a row above cells at t , and the out degree of each cell is indicated by the depth of the cells at $t-1$.

The following top right reminder and options are displayed.

```
show cell wiring/rule, use arrows to move, jump-j
one cell with right/left arrow, 10 cells with up/down arrow (values shown are examples)
options-o, print box-p, revise wiring-w, revise/transform rule-r/t ("r/t" only if a rule scheme was set)
show avZ-z, learn/highlight-l, cont-ret: ("z" only if mixed rule network set)
```

The options described below.

14.2.1. Moving or jumping between cells, 1d wiring graphic.

To move between cells, use the arrow keys. The left and right arrow keys move by one cell, the up and down arrow keys move by one tenth the network size left and right, for sizes of 20 and over. Alternatively, enter "j" to jump to a specific cell index. The following prompt is displayed,

```
jump to index (54-0) (this example for a 1d network, size 55)
```

14.3. Wiring graphic, 2d.

If "2" is selected in #14.1, the 2d wiring graphic is presented in a window showing (graphically and numerically) how each cell's pseudo-neighbourhood in the 2d array is wired up, one cell at a time. The target cell is coloured red, its pseudo-neighbourhood yellow. If the "actual neighbourhood" cells are distinct from "pseudo-neighbourhood" cells, the couplings between them are shown and the "actual neighbourhood" cells are coloured green.

To move between cells, use the arrow keys. The following top right reminder and options are displayed.

```
show cell wiring/rule, use arrows to move,;
options-o print box-p revise wiring-w rule-r ("r" only if a rule scheme was set)
show avZ-z, learn/highlight-l, cont-ret: ("z" only if mixed rule network set)
```

The options are described below.

14.3.1. Moving or jumping between cells, 2d wiring graphic.

To move between cells, use the arrow keys. Alternatively, enter "j" to jump to a specific 2d cell index ($i \times j$). The following prompt is displayed,

```
jump to index i x j
enter i (35-0): enter j (35-0): (this example for a 2d network, 36 x 36)
```

14.4. Options, 1d and 2d wiring graphic.

14.4.1. Revising the wiring.

(does not apply to 1d regular wiring).

If "w" is entered in #14.2 or #14.3, the wiring for individual cells in the network may be revised, as well as the neighbourhood size for mixed neighbourhoods. The "revise wiring" options are described further in #14.5, below.

14.4.2. Revising the rule.

(applies only once the rule scheme has been set in #13)

If "r" is entered in #14.2 or #14.3, the rule for individual cells may be reviewed or changed. For a single rule network any revision applies to the whole network. The rule relative to each cell is displayed in a lower rule window. The "revise rule" options are described further in #14.6, below.

14.4.3. Computing the (weighed) average λ and Z parameters.

(applies only once the rule scheme has been set in #13, and for a rule mix, with or without random wiring). If "z" is entered at #14.2 or #14.3, the program calculates the averages, and weighted averages, of the λ , λ ratio, and Z parameter values of the rules making up the rule mix. This data is displayed at the top of the wiring graphic window. For random wiring and/or mixed neighbourhoods, the weighted averages take account of the influence each cell has on the network according to its proportion of the network's out-wires. In networks with regular wiring and no rule mix, the weighted average equals the average.

14.4.4. Learning pre-images without attractor basins.

Entering "l" at #14.2 or #14.3 starts the learning/forgetting routine for attaching/detaching sets of states as pre-images of a target state. The network's wiring and/or rule scheme is automatically amended to achieve the required transitions between states. The learning routine is described in detail in #27.

Learning is also invoked in the context of "sculpting attractor basins" (see #27). In this case, the results and side affect of learning are displayed when attractor basins are generated, but this places a limit on the size of the network (see #6.3).

Invoking the routine at this point in the program, or when running networks "forwards only", allows much larger networks to "learn".

14.4.5. Printing the wiring graphic window.

If "p" is entered at #14.2 or #14.3, a printout is made of just the window showing the wiring scheme (see printer limitations #A4).

14.4.5. Options.

Entering "o" at #14.2 or #14.3 gives options to "print/save/load" the screen image as described in (see #5.3).

14.5. Revise wiring with the 1d or 2d wiring graphic.

If "w" is selected in #14.2 or #14.3, the wiring for individual cells may be revised. Note that this does not apply if the wiring was set as "regular 1d" in #10.2. To allow regular 1d wiring to be revised it must be set in "special wiring" (#11.3) and the option set to treat local 1d wiring as non-local wiring.

The following context dependent secondary window is presented in the lower right corner of the screen.

REWIRE cell=33 move-arrows jump-j cont-ret	(values shown are examples)
change k=8 nhood-n	(if neighbourhood mix was set)
hand-h random-r special-rand-s reg 1d-1 reg 2d-2:	(reg 2d only for a 2d network)

While this prompt is displayed, the arrow keys may be used to move between cells, or "j" may be entered to jump to a new cell index (as in #14.2.1 and #14.3.1).

Once the cell's wiring has been revised by one on the various methods, the new wiring is immediately displayed, the out-degree and other data is updated, and the program reverts to this prompt (#14.5).

Note that once the rule scheme has been set (see #13), the rule relative to the individual cell is displayed in a lower rule window.

14.5.1 Changing the neighbourhood size.

If the network was set to have a neighbourhood mix, "n" may be selected in #14.5 above to change the neighbourhood size of a cell. The following additional prompt appears,

set nhood size (1-9);

The new neighbourhood will preserve as much as possible of the old neighbourhood wiring. If the new neighbourhood is bigger, the wiring at the extra pseudo-neighbourhood indices is set as regular 1d or 2d with periodic boundary conditions.

If a rule scheme has been set (see #13), the existing rule table (or part of it if the new neighbourhood size is smaller) will be preserved Any excess rule table entries are set to 0.

14.5.2 Hand rewiring.

If "h" is selected in #14.5, individual wires can be rewired in a way similar to that described in #11.6. The pseudo-neighbourhood wiring connections are displayed as a 1d "spread sheet" with k entries. The following reminder is displayed,

hand rewire;
use arrows or "return" to move
enter wiring position 0-(n-1), q to complete

14.5.3. Random wiring.

If "r" is selected in #14.5, the cell's wiring is randomly reset.

14.5.4. Random special wiring.

If "s" is selected in 14.5, a series of prompts first allow some restrictions to the random wiring before it is set. The following options are presented in sequence (use "q", or the right mouse button, to backtrack).

confine to local periodic zone, enter zone size:
exclude all selfwiring-2, selfwire centre wire-1:
distinct wiring (no duplication)-d:

These option are described in #11.5.1-3. The cell's wiring is randomly reset within the set restrictions.

14.5.5. Regular 1d wiring.

If "1" is selected in #14.5, the cell's wiring is reset as regular 1d with periodic boundary conditions.

14.5.6. Regular 2d wiring.

If "2" is selected in #14.5, the cell's wiring will be set up as regular 2d with periodic boundary conditions.

14.6. Revise rules with the 1d or 2d wiring graphic.

If "r" is selected in #14.2 or #14.3, the rule for each individual cell may be revised. Note that this only applies to a mixed rule network, not to a "single rule" set in #13.3. However, a rule mix where all the rule are the same can be set up as described in #13.3.1. A secondary window is presented in the lower right corner of the screen, with rule selection prompts. The various methods for revising and re-selecting rules are described in #13.6 and #13.6.1-16. Once complete, the new rule is displayed in the rule window, and the program returns to the prompt at #14.2 or #14.3, which allows moves to other cells. To change a rule to one with a different neighbourhood size k (in a rule mix), first change k in the rewiring window (#14.5.1), then change the rule.

14.7. Rule/wiring scheme: saving, loading, or printing (to a printer or file).

(not applicable to regular 1d CA)

14.7.1. Save/load/print selection.

Following the wiring graphic and rule scheme prompt (#14.1), the rule/wiring scheme may be saved, or printed to a printer or file. The rule scheme and wiring scheme may be treated in isolation, or combined.

Depending on the type of network architecture, one of the prompts below (#14.7.2-4) is displayed in a top right window. If the save or load option, or printing to a file, is selected, top right filing windows appear to select the filename

etc (see filing). The file types are described in #14.7.6.

14.7.2. Mixed rules and non-local wiring.

For a network with mixed rules and non-local wiring the following prompt is presented,

```
just wiring: print-pw, load-lw, save-sw,
just rules: print-pr, load-lr, save-sr,
rulemix+wiring: print-pb, load-lb, save-sb:
```

Enter two key strokes; firstly "p", "l" or "s" to print, save or load, followed by "w", "r" or "b" (the default) for wiring, rule or the rule/wiring scheme combined.

14.7.3. One rule and non-local wiring.

For a network with just one rule but non-local wiring the following prompt is presented,

```
wiring: print-p, load-l, save-s:
```

Enter "p", "l" or "s" to print, save or load the wiring scheme.

14.7.4. Mixed rules and local wiring.

For a network with mixed rules but local 1d wiring (and no rulemix) the following prompt is presented,

```
rulemix: print-p, load-l, save-s:
```

Enter "p", "l" or "s" to print, save or load the rule scheme.

14.7.5. Printing to a printer or file.

If "p" is selected in #14.7.2-4 above, the following prompt allows the data to be printed to a printer or to a "*.dat" file, or to both simultaneously. Note the printer limitations described in the introduction (#4). As the printing is ascii text only, these are likely to be less severe.

```
print network data: to printer-p, file-f both-b:
```

Before printing to a file, top right windows will prompt for the file name (default "net_data.dat", see filing #28). The format of the ascii network data file (and the decoding of the data) is shown in the examples in below.

Example of a network with mixed rules and non-local wiring, but homogeneous neighbourhood size.

data

k=3, n=10

cell. wiring(2-0) rule(hex) ld ld-r Z C

9. 7 3 1, ec 0.625 0.75 0.625 1/3 *1*

8. 0 7 2, ee 0.75 0.5 0.5 2/3 *10

7. 8 9 5, 5a 0.5 1 1 0/3

6. 7 8 0, ee 0.75 0.5 0.5 2/3 *10

5. 0 6 2, 74 0.5 1 0.5 0/3

4. 5 9 1, e8 0.5 1 0.5 0/3

3. 6 3 1, 9a 0.5 1 1 0/3

2. 8 4 6, 46 0.375 0.75 0.75 0/3

1. 3 7 4, 4c 0.375 0.75 0.625 1/3 *1*

0. 1 0 7, ba 0.625 0.75 0.75 1/3 **0

average:lda-r=0.8 Z=0.675

weighted average:lda-r=0.8 Z=0.708333

decode

neighbourhood size k , network size n

remainder of data order (see below)

cell index, wiring, rule in hex, rule parameters

average network parameters

average network parameters

weighted according to the proportion of cell outputs.

key to data order: cell. wiring(2-0) rule(hex) ld ld-r Z C

cell the cell index.

wiring the wiring scheme of the pseudo-neighbourhood (see #9.2).

(2-0) remainder of the pseudo-neighbourhood index, in this case 2,1,0.

rule the rule in hex.

ld the λ parameter.

ld-r the λ ratio.

Z the Z parameter.

$x/3$ the proportion of canalizing inputs.

1 shows exactly which of the k inputs are canalizing (if none this is not shown),

i.e. *1* - input 1 is canalizing, 2** - input 2, *10 - both inputs 1 and 0.

Example of a network with mixed rules and non-local wiring, and mixed neighbourhood size.

data

k=mixed(9-1), n=10

cell. k, wiring rule(hex) ld ld-r Z C

decode

range of neighbourhood size k , network size n

remainder of data order (see below)

9. 3,----- 6 7 3, c0 0.25 0.5 0.5 2/3 21* cell index, k, wiring, rule in hex, parameters
8. 1,----- 2, 00 0 0 0 1/1
7. 9, 3 2 4 9 0 0 2 0 3, b1d58087978f6d4ff62350e883511474b1d58087978f6d4ff62350e883511474879136a1a62c647ebb37f
53f31709b91879136a1a62c647ebb37f53f31709b919d36dd660a91b2335a4b35258b0eae369d36dd660a
91b2335a4b35258b0eae36e03b54a573f028ad759ad1f959d32426e03b54a573f028ad759ad1f959d3242 6 0.498 0.996 0.596 0/9
6. 7, - - 1 1 7 2 7 1 4, 6dae8d684b01c66b0b5ba0e68b3bf2de6dae8d684b01c66b0b5ba0e68b3bf2de 0.516 0.969 0.561 0/7
5. 5, ---- 7 1 8 5 9, 70596206 0.375 0.75 0.625 0/5
4. 2, ----- 7 7, 00 0 0 0 2/2 10
3. 8, - 5 9 7 6 2 5 5 1, 4f16fbf92059a943d13dea4ceea1a7024f16fbf92059a943d13dea4ceea1a7027cbd307e50d33c8c8ab0b
5f84bf1e3a47cbd307e50d33c8c8ab0b5f84bf1e3a4 0.512 0.977 0.562 0/8
2. 6, --- 1 8 6 9 2 7, 6ae885c5eb5aafd4 0.547 0.906 0.678 0/6
1. 4, ----- 4 7 4 6, 5e9c 0.562 0.875 0.719 0/4
0. 3, ----- 2 8 0, 84 0.25 0.5 0.5 1/3 *1*
average:lda-r=0.647266 Z=0.473972 network parameters
weighted average:lda-r=0.715902 Z=0.51767 average network parameters, weighted according
to the proportion of cell outputs.

key to data order: cell. k, wiring rule(hex) ld ld-r Z C

cell the cell index.
k the neighbourhood size k.
wiring the wiring scheme of the pseudo-neighbourhood (see #9.2).
rule the rule in hex, the length of the hex string depends on k.
ld ld-r Z C same as the first example above.

14.7.6. Wiring/rulemix file names.

The binary files defining the wiring scheme, rule scheme or wiring/rule scheme combined are named as follows (and encoded as described in #4.7.7-8 below),

Wiring scheme:	extension ".w_s",	default file name "my_wso.w_s".
Rule scheme:	extension ".r_s",	default filename "my_rso.r_s"
Combined wiring-rule scheme:	extension ".wrs",	default filename "my_wrs.wrs"

14.7.7. Wiring/rulemix encoding..

The encoding for all three file types above (#14.7.6) starts with 5 bytes as follows:

byte 0 = neighbourhood size, k (or the maximum k for a neighbourhood mix).
byte 1,2 = network size, n.
byte 3,4 = i, j for 2d networks (for 1d these bytes are set to null)

The encoding continues as follows, for each successive cell index from 0-(n-1), where k = the maximum neighbourhood:

A. Pseudo-neighbourhood wiring, indexed 0-k,	if $n \leq 256$, k bytes, if $n > 256$, $2 \times k$ bytes
B. Rule:	2^k bytes (minimum 1 byte)
Combined wiring-rule scheme:	A followed by B.

14.7.8. Wiring/rulemix file loading conventions.

When loading wiring/rulemix files, the (maximum) neighbourhood size, k in file must equal k in the defined network, and the file network size, n, must not be bigger than the defined network size. If this occurs, the file to be loaded is rejected is rejected with either or both of these message ("Return" to continue),

net nhoud=9 not equal to 5:

net size=100 < 150:

The file network size can, however, be smaller than the defined network size.

14.7.9. Neighbourhood mix loading conventions.

The specification of a neighbourhood mix should be saved as a separate "*.mix" file (see #8.6.3). To load the mixed neighborhood rule scheme (or combined wiring/rule scheme) first load the correct "*.mix" file (#8.3), then load the rule scheme (or combined wiring/rule scheme) file. If there is a conflict between these files as to the size of neighbourhoods, errors will occur.

15. Seed.

15.1. Setting the selection method for the seed.

If "s" for "forward only, single basin or subtree" (i.e. not a basin field) was selected at the first prompt (#5.1), an initial network state, or seed, needs to be specified. Various methods of setting the seed are available. A lower left window is displayed with the prompt below, which may vary according to context,

Select SEED, empty-e fill-f (default-dec) ("default-dec" if $n \leq 32$, otherwise "default-random")
random-r 1d-b 2d-B hex-h repeat-p load-l: (for a 2d network "2d-bits-b")

The seed can be set in a variety of different ways:

"e" -reset all cells to 0 null, the program then reverts to the prompt above (#15.1).

"f" -reset all cells to 1, the program then reverts to the prompt above (#15.1).

"d" - in decimal (for the network size, $n \leq 32$).

"r" - at-random.

"b" - as bits in a 1d or 2d graphic array, drawing with mouse.

"B" - as bits for a 1d network shown in 2d (1 networks only).

"h" - in hex (in a mini "spread sheet")

"p" - a repeat of the last seed.

"l" - loaded from a "*.eed" file.

Once set, the seed remains current and can be amended by resetting bits (1d-b, 2d-B), resetting half bytes (hex-h) or loading a seed file (*.eed), which may be a small patch (1d or 2d) put onto the current seed. Resetting the seed to "empty-e" or "full-f" allows a clean slate for setting (drawing) 1 or 0 bits, or amending the seed as above. The various methods of setting and amending the seed are described in #15.2-8 below (they are similar to the rule setting methods #13.6-13).

15.2. Seed bit pattern display.

The display of seed bit patterns during the selection procedure is in 1d or 2d depending on the network geometry setting in #11.1. However, the bit pattern of a 1d network can also be shown in 2d.

Networks cells are indexed and arranged in a regular space with periodic boundary conditions as described in "network geometry" #9.1. A 1d space forms a ring of cells. A 2d space forms a regular square grid on the torus.

0s are coloured grey, 1s are coloured red. Normally a 1d bit pattern is displayed in a row, or successive rows for a large array, and broken up into byte size chunks, whereas a 2d bit pattern is shown as a rectangular or square block. The exceptions to this are that a 1d pattern is not broken into bytes in the "setting bits" (drawing) option (#15.7), and the 2d pattern is broken into bytes after being set in hex. The scale of the bit pattern graphic varies according to the size of the network.

If the bit pattern of a 1d network is selected in 2d, the 1d pattern is set out in successive rows starting with the maximum cell index in the top left hand corner (see also #20.2 for attractor basin node display of 1d networks in 2d).

15.3. Rotating the seed.

The various methods for setting the seed allow it to be "rotated" left/right for 1d, left/right/up/down for 2d. This is done from the keyboard with the "l/r/u/d" keys. The bit pattern is displaced within the ring of cells in 1d, or within the torus in 2d. and this change shows up immediately in the bit pattern, hex or decimal representation of the seed.

15.4. Setting the seed in decimal.

For network size, $n \leq 32$, if "d" (or default) is selected in #15.1, the seed can be specified by its decimal equivalent. The following prompt is presented,

decimal seed, enter 0-16363 (default-random): (example for $n=14$)

If just "return" is entered at this prompt, or a number outside the permitted range, the selection will revert to "random" as (#15.5) below. When selected, the seed is also shown as a bit pattern (in 1d or 2d) and in hex.

15.5. Setting the seed at random.

If "r" is selected at #15.1, a seed, or a segment only, can be set at random. The following prompt is displayed, depending on whether the network is 1d or 2d:

central segment only enter size (default 20), all-a (example for a 1d network, $n=100$)

central 2d block only enter diameter (default 7), all-a: (example for a 2d network 36×36)

The default segment size is 1/5 of the 1d network size, and 1/5 of the 2d network width, i . Enter "a" for a random bit pattern across the whole network. A seed is selected at random and displayed in decimal (for $n \leq 32$), in hex (for $n \leq 256$), and as the rule table bit pattern (in 1d or 2d). The seed can be accepted or an alternative random seed may be generated.

The seed can be rotated as in #15.3. To accept the seed bit pattern, enter "return". During the random setting procedure the following prompt is displayed, depending on whether the network is 1d or 2d.

(rotate-l/r, another-n, backtrack-q, accept-ret) (for 1d, "rotate-l/r/u/d" for 2d)

Enter "n" for another random rule, "l", "r", "u" or "d" to rotate.

15.6. Setting the seed in hex.

If "h" is selected in #15.1, the hex expression of the current seed (which may first be reset to all 0s or all 1s) is displayed. The hex digit at the green flashing cursor can be overwritten from the keyboard; and the cursor will automatically move

to the next digit. The left/right (and up/down) arrow keys are used to displace the cursor. The up/down arrow keys apply for seeds where $n \geq 256$, where the hex digits are presented in successive rows.

The seed can be rotated as in #15.3. To accept the seed bit pattern, enter "return". During the bit setting procedure a top right reminder displays the following, depending on whether the network is 1d or 2d.

enter hex, arrows-move
rotate-l/r, accept-ret (for 1d, "rotate-l/r/u/d" for 2d)

To rotate enter "l", "r", "u" or "d". Once selected in hex, the rule is also displayed in decimal (for $n \leq 5$) and as the rule table bit pattern, broken into byte size chunks.

15.7. Setting the seed as bits, drawing with the mouse.

15.7.1. 1d networks (shown as 1d).

If "b" is selected at #15.1, and the network is 1d, the following prompts allow just a segment of the seed to be defined,

enter segment to define as bitstring, max/default=99 (example for a network size, $n=99$)

Enter the size of the segment, or accept the default; the whole network. If a segment is selected, a further prompt positions the segment within the network,

enter start position from the left 0-88, default centred: (for segment size=10)

Enter the start index of the segment. Note the above does not apply to 2d networks. A bit pattern representing the current seed (or segment) is displayed.

15.7.2. 1d networks (shown as 2d).

If "B" is selected at #15.1, the 1d network is represented in 2d as a square or rectangular bit pattern, $i \times j$. The default i (the number of columns) is set automatically as the highest factor of the network size n . The following prompt gives the current $i \times j$ setting, to revise enter a new i value,

now 15 x 15, reset i-axis : (values shown are examples)

The 1d network is broken up into successive rows starting with the maximum cell index in the top left hand corner. If i is not a factor of n , some cells will be missing from the bottom row.

If this option is set, the default node display for attractor basins will be set identically (see #20.2 and #19.2.2).

15.7.3. 2d networks.

If "b" is selected at #15.1, and the network is 2d, a rectangular or square bit pattern, $i \times j$, is displayed.

15.7.4. Setting/deleting bits with the mouse or keyboard.

The scale of the bit pattern (1d or 2d) varies according to the size of the network. 0s are coloured grey, 1s are coloured red. The mouse arrow cursor is confined within the bit pattern area. A left button press sets bits on (to 1), a right button press sets bit off (to 0). i.e. deletes 1s). If the left mouse button is kept down a line of 1 bits is drawn, keeping the right mouse button down draws 0 bits.

Bits may also be set from the keyboard. The current bit to be updated is highlighted by a small green flashing cursor, which moves to the right when 1 and 0 are entered to set or delete bits. This cursor is moved with the arrow keys as well as with the mouse. The index of the current bit is given in a top centre "memory window", giving the absolute index i.e. $n-1$ to 0, and also the $i \times j$ co-ordinates.

The seed can be rotated as in #15.3. To accept the seed bit pattern, enter "return". During the bit setting procedure a top right reminder displays the following,

keyboard: 0/1-set bit, arrows-move, accept-ret
mouse: press-set bit, keep pressed-draw, release-move
mouse buttons: left-set 1, right-set 0
rotate-l/r/u/d, accept-ret (or "rotate-l/r" for 1d where $n < 64$)

When selected, the seed is also displayed in hex and also decimal for $n \leq 32$.

15.8. Repeating the seed.

If "p" is selected in #15.1, the current seed is repeated.

15.9. Loading a seed.

If "l" is selected in #15.1, top right windows allow a seed be loaded from a "*.eed" file, the default filename is "myseed.eed" (see filing).

15.9.1. Seed file encoding and loading conventions.

The binary file defining the seed is encoded as follows:

Bytes 0,1 = n , bytes 2, 3 = i, j for 2d (for 1d these bytes are set to null), the rest of the seed is set as bits from 0 to $n-1$. Excess bits in the last byte are set to 0.

The following describes the consequences if the network size in the file to be loaded differs from the actual

network size, or if its geometry conflicts. If a 2d file is loaded to a 1d array, its 2d characteristics will be ignored, If a 1d file is loaded to a 2d array, if the file is bigger, as many bits as will fit are loaded.

15.9.2. Seed file, 1d.

If the 1d file array size is smaller than the actual network (whether 1d or 2d), the smaller file can be positioned somewhere on the larger array, the default is centrally. In this example, a file for a network size=20 is being loaded for a network size=100. The following prompt is presented,

1d: array length=100, file: file length=20, enter start position(values shown are examples)
(default 40, max 80):

Enter the lower index start position to locate the filed bit pattern. If the file network size is bigger than the network size, n , the beginning section of the larger file (from cell index 0) is loaded to fit the small network,

15.9.3. Seed file, 2d.

If the file array $i \times j$ is smaller than the network $i \times j$, the smaller file array can be positioned somewhere on the larger array, the default is centrally. The following prompt is presented,

2d: i=36, j=36, file: i=6, j=6, enter start coords (values shown are examples)
(default 15, 15, max 30,30) i: j:

Enter the lower left 2d index, i followed by the index j to locate the filed bit pattern. If the file network size $i \times j$ is bigger than the network $i \times j$, the centre of the bigger file array will be put onto the smaller network.

15.10. Saving a seed.

Once set, top right filing windows allow the seed to be saved to a "*.eed" file (the default filename is "myseed.eed", see filing #28).

16. Output parameters

The output parameter prompts define how attractor basins are displayed, and also various analytical settings, printing and/or saving data, and mutation settings. The prompts are skipped by entering "s" to run space-time patterns forward only (if a basin of attraction field was *not* selected), or by entering "d" to accept defaults. Entering "d" can be done at any stage to accept remaining defaults and skip remaining options. Once set, new values generally become the output parameter defaults, but can be reset to the original values. All the output parameter options may be looked at in turn, but they are divided into four categories to allow jumping directly to the category where options need to be set. The categories are: *various*, *layout*, *display/data* and *mutation*.

The output parameter prompts are also skipped if the "forward only" option for 2d regular CA was selected in the wiring prompt #10.3.1. The following reminder is displayed in a top right window,

2d CA (36 x 36), forward only, return to continue: (values shown are examples)

16.1. The first output parameter prompt.

"OUTPUT PARAMETERS" appears in a top centre memory window, and an initial prompt is presented in a top right window. If a basin of attraction field was specified, the first line in this window is as follows:

accept all basin defaults-d
...

If *forward only*, *single basin* or *subtree* (thus also a seed) was specified (i.e. not the entire field), the first line allows a decision to be made at this stage to run the network forwards only to see just space-time patterns, or alternatively to generate a basin of attraction or subtree.

accept all basin defaults-d, space-time patterns only-s
restore defaults: all-a, layout only-L
revise from: layout-l, display/data-p, mutation-m
revise from start-ret

16.1.1. Accept all output parameter defaults.

Enter "d" to accept all output parameter defaults, and skip further output parameter prompts. This can be done at any stage. The next prompt for a basin field will be #23, otherwise #22.

16.1.2. Space-time patterns only.

Enter "s" to run the network forward only and see just space-time patterns. This skips further output parameter prompts. The next prompt sequence is described in #24.

16.1.3. Restore all output parameter defaults, or just layout defaults.

Once set, new values generally become the output parameter defaults. Enter "a" to restore all original defaults, or "L" to restore the original defaults for just the layout of attractor basins.

16.1.4. Jump to layout of attractor basins.

Enter "l" to jump directly to a sequence of prompts to set the scale, position and spacing of attractor basins (see #19).

16.1.5. Jump to display of attractor basins, and data

Enter "p" to jump directly to a sequence of prompts to make changes to the state transition graph display, pause to show data at various degrees of detail, and options to print and/or save this data (see #20).

16.1.6. Jump to mutation.

When the drawing of attractor basins is complete for a given network, another may be generated with the same setting, but with a mutation to the network's rule, or to its rule scheme or wiring scheme, or a new random network may be set.

Enter "m" to jump directly to set the type of mutation to be applied to the next network (see #21).

17. Set various output parameter options.

If "return" is entered at #16.1. A number of miscellaneous options are presented in sequence in a top right window (described below). At any point enter "q" (or "q" more than once) to backtrack to the first output parameter prompt. A reminder is shown in a top centre window that entering "d" will accept all remaining output parameter defaults and skip further output parameter prompts.

17.1. State-space matrix scatter-plot.

The state-space matrix (or scatter-plot) plots each state in state-space on a 2d grid in the lower right corner of the screen, plotting the left half against the right half of each state bit string. The x axis represents the left $n/2$ bits, the y axis represents the right $n/2$ bits. If n is odd, the extra bit is included on the left, and the grid is a flat rectangle, otherwise the grid is square. The scale of the matrix is set automatically according to n , though this scale can be changed.

For a single basin, attractor cycle states and transient states are shown in different colours. For a basin of attraction field, each basin is assigned a different color (cycling through 15 colours). The start color for the colour cycle may be reset to produce different color schemes.

VARIOUS: state-space matrix: all-states-m, attractor only-a
show and change: matrix size-s, start color-c:

The space-time matrix can be toggled on/off while basin are being drawn. The following reminder will appear in the bottom title bar:

matrix-m

Enter "m" to show/hide the state-space matrix. Note that the state-space matrix can also be set when showing space-time patterns (forward) only. This option is described in #24.6.

17.1.1. Display all states.

Enter "m" at #17.1 to display the matrix, including all states in the subtree, single basin or field. For a single basin, attractor cycle states and transient states are shown in different colours. For a basin of attraction field, each basin is assigned a different colour (cycling through 15 colours).

17.1.2. Display attractor states only.

Enter "a" at #17.1 to display the matrix, showing just attractor states in the single basin or field. For a basin of attraction field, each basin is assigned a different colour (cycling through 15 colours).

17.1.3. Change matrix size.

Enter "s" at #17.1 to change the matrix size from the default. The following prompt is presented,

change matrix size: % of 1/2 screen(default 62.70%):

Enter the new size as a percentage of half the screen width. The new setting may be rounded up or down to give integer grid sizes for the matrix.

17.1.4. Change start colour.

Enter "c" to change the start colour in the colour cycle of 15 colours, to produce different colour schemes. The following prompt is presented,

change start color (now 10): enter 1-15:

17.2. In-degree frequency histogram.

DDLab may keep a record of the frequency of different in-degrees (the number of pre-images or predecessors) that occur in a subtree, basin or field. If this option is selected, the in-degree information is displayed as a histogram when the attractor basin is finished (whether complete or interrupted). In the case of a subtree, the in-degree histogram may also be displayed during a temporary interrupt (see #25.1). The following prompt is presented,

show pre-image frequency histogram-y

If "y" is entered, the histogram setting remains for all further attractor basins. It may be deactivated above, by

restoring all "output parameter" defaults (see #16.1.3), or by backtracking through the main prompt sequence beyond #8.

17.2.1. In-degree frequency cut-off.

If "y" is entered above, the following prompt allows setting an upper threshold (or cut-off) to the in-degree, in order to conserve memory. Any cut-off in the range of 50 to 5000 may be set, the default is 200, or the current setting. The occurrence of in-degrees equal to or above the cut-off value will be added to the cut-off frequency "bin" in the histogram.

select cut-off in-degree bin (50-5000, default 200):

17.2.2. In-degree frequency window.

The in-degree histogram will be displayed in a window in the lower section of the screen. The x -axis represents the range of in-degrees, from in-degree zero (garden-of Eden states) upward. In-degrees equal to the cut-off value and above are combined and represented by the last frequency column. The y -axis represents the number of states having a given in-degree. The cut-off value column (if included) is colored black. The garden-of-Eden column is colored red, the other columns yellow.

Initially the scales of the x and y axis are automatically set according to the network size n , and the range of in-degrees encountered. However, once the histogram is drawn the x and y axis can be separately re-scaled for a clearer view of the spread of frequencies or to amplify smaller in-degrees (see #. 17.2.5-6).

An example of the data included in the window, and the re-scaling prompts, are shown below for the basin of attraction field of $k=3$ rule (dec) 193, $n=16$.

```
total att/nodes/edges=285/65536/65536          (for a subtree "subtree: nodes/edges....")
max in-degree=90, gE=37542
(o-options prtbox-p quit-q)
re-scale x/y axis: new x max (25-90).....new y max: ("new x max..." only if
                                                    the max in-degree found is greater than 25)
```

The example below is for a subtree $k=5$ rule (hex) 6c 1e 53 a8, $n=55$, from a random seed (hex 3f 14 b6 2d 42 f4 45) but running forward 100 time-steps before running backward. The in-degree cut-off was set at 1000. The subtree was stopped during its 6th backwards level.

```
subtree: nodes/edges=52195/93552
max in-degree=411, gE=42464
(o-options prtbox-p quit-q)
re-scale x/y axis: new x max (25-411).....new y max: ("new x max..." only if
                                                    the max in-degree found is greater than 25)
```

Enter new values to re-scale the x and/or y axis, or select other options (see #172.4-6).

17.2.3. Data shown.

The data included above signifies the following,

att	The number of attractor states (does not apply to subtrees)
nodes	The number of nodes whose pre-images have been computed
edges	The total number of edges, including edges to nodes that have not been computed. For a complete basin (or field), nodes=edges. For a complete subtree, nodes=edges+1. For an incomplete attractor basin, (i.e. if interrupted), nodes<edges.
max in-degree	The largest in-degree found. The maximum equals the cut-off limit.
gE	The frequency of garden-of-Eden states (in-degree 0), usually the greatest frequency.

17.2.4. Options.

Hit the "home" key to print the full screen, or "p" to print just the in-degree histogram window (note printer limitations #A4). Enter "o" for further options (see #25.3.) including saving the screen image to a file. Enter "q" to quit the in-degree histogram window and continue.

17.2.5. Re-scaling the x -axis.

The x -axis is initially scaled to show the spread of in-degrees from 0 to the maximum found (or to the cut-off value). If the cut-off value is shown, "+" is added at the end of the x -axis indicating that in-degrees equal to or greater than the cut-off value occurred.

If the maximum in-degree found is greater than 25, the x -axis can be re-scaled by entering a new maximum in-degree. Note that if there are excess in-degrees (off the scale) their total frequency will be shown as a black column on top of the maximum in-degree column, and "+" will be added to the x -axis.

17.2.6. Re-scaling the y -axis.

The y -axis is initially scaled to contain the highest in-degree frequency column in the histogram (usually column 0, garden-of-Eden states). It may be re-scaled to any value, for instance to show smaller in-degree frequencies at a larger scale.

17.3. States with majority of 1s or 0s.

If the pre-image frequency histogram option was selected (#17.2 above), the following prompt is presented:

show/display states with majority of: 1s-1, 0s-0:

Enter "1" (or "0") to highlight states in the attractor basin which have a majority of 1s (or 0s). The highlighted states are displayed as small yellow squares with a black border. This option is included to test the performance of a favourite exercise in "evolutionary computation", where CA rules are evolved to discriminate between states with a majority of 0s and 1s. This entails evolving two point attractors, the all 0 state attracting the majority-0 states, and the all-1 state attracting the majority-1 states. Within the pre-image frequency histogram window, the number and percentage of "exceptions" to either of these conditions is displayed.

17.4. "Screen save" demo.

If a single basin or subtree was selected, an option is presented for a continuous show where basins or subtrees are generated randomly in overlapping bubbles on the screen. The following prompt is presented,

screensave demo -s:

Enter "s" to start the demo. The backwards space time patterns and state-space matrix may also be set, or toggled "on the fly". (see #17.1. and #17.6.)

17.5. Graphs: G-density, Z and λ ratio.

The density of garden-of-Eden states in a subtree, basin or field (G-density) is a measure of the "convergence" of network dynamics, which in turn relates to the quality of typical space-time patterns, *ordered-glider-chaotic*. The λ parameter measures the density of 1s in a CA rule table, whereas Z measures the probability that the *next cell* in a candidate partial pre-image is determined. Both of these related parameters predict convergence and thus dynamics to a varying degree, though Z is more focused. The λ ratio is the (density $\times 2$) of 1s or 0s in the rule table, whichever is less. It normalises the λ parameter to a value between 0 and 1 for direct comparison with Z.

Graphs of G-density against the Z and/or λ ratio may be plotted for a predefined set of CA rules in order to test these relationships. Graphs of Z against λ ratio may be plotted to relate the two parameters. The following prompt is presented,

G-density graph: Z-parameter-1, and Lambda ratio-2

Z-parameter - Lambda ratio graph -L:

17.5.1. G-density against Z and/or λ ratio.

Enter "1" for a graph of G-density against Z, and "2" for a graph of G-density against Z and λ ratio simultaneously. The Z graph will be drawn in the lower right of the screen, the λ ratio graph beside it on the left. A subtree, basin or field for each of the set of rules specified will be drawn (at a small scale by default) in a window in the upper part of the screen. When complete the network's G-density (y-axis) is plotted against the rule's Z and/or λ ratio, x-axis.

If the rules specified are for a neighbourhood $K \leq 3$, or totalistic, the complete set of non-equivalent rules "clusters" will be plotted (rules in a cluster have the same G-density, Z and λ), subject to limiting the range of Z described below. Otherwise the following prompt is presented,

G-density graph: number of rules (default 1250):

Enter the number of rules to be plotted or accept the default. The rules will be selected at random, but with a bias to include an even distribution over values of Z. The following prompts is presented to limit the range of Z, firstly the lower limit, then the upper. Note that low values of Z should be avoided except for small network sizes, because in-degree (and G-density) will be very high. Very high in degree can exhaust computer memory or take a long time to compute.

set min Z (default 0.2):

set max Z (default 1):

17.5.2. Z against λ ratio.

If "L" is selected in #17.5 above a graph of Z (x-axis) against λ ratio is plotted. Both of these measures are calculated just from the rule table. The graph is plotted in the lower right of the screen. If the rules specified are for a neighbourhood $K \leq 3$, or totalistic, the complete set of (optionally non-equivalent) rules will be plotted. For totalistic rules the following prompt is presented:

show only equivs-e all-a (default clusters):

Enter "a" to show all totalistic rules (expressed as codes, see #13.1.1), "e" to show equivalent rules only. The default is to show a representative rule from each rule cluster. There are at most 2 equivalent totalistic rules, the start rule and its negative (the reflection of a totalistic rule = identity), and at most 4 rules in a cluster (the compliments of both

equivalents). The cluster has the same Z and λ . A further prompt allows a pause to note the equivalents or members of the cluster.

pause to show codes-p:

If the rules specified are not for a neighbourhood $K \leq 3$, or totalistic, the following prompt allows the required number of rules to be set,

lambda-Z graph, number of rule (default 2200):

The rules will be selected at random, but with a bias to include an even distribution over values of λ .

17.6. "Backwards" space-time patterns.

At the same time that attractor basins are being drawn, the "backwards" space-time pattern may be displayed on the left of the screen. For basins of attraction, this will show an initial run forward, representing a transient and the attractor cycle itself (blue and grey representing 1s and 0s). Thereafter the space-time pattern shows the information used to draw each transient tree in turn, starting from the "root" of the tree on the attractor, and back through successive levels in the tree. First the pre-images of the root are computed (the pre-image *on* the cycle is omitted), comprising level 1 in the tree. Then the pre-images of each state in level 1 are computed, comprising level 2 in the tree. This process continues through an arbitrary number of levels until all garden-of-Eden states (those without pre-images) are reached. Then the next tree is computed (note that for CA, "compression" allows equivalent trees to be simultaneously computed).

The "backwards" space-time patterns shows each state (black and grey representing 1s and 0s) and its set of pre-images if any (red and grey representing 1s and 0s). The scale depends on the setting in #7 in the main prompt sequence, but this may be changed "on the fly" (see below).

For a basin of attraction field the initial default is *not* to show the backward space-time patterns, and the following prompt is presented, enter "y" to show,

SHOW space-time patterns-y:

For a single basin or subtree, the initial default is to *show* the backward space-time patterns, and the following prompt is presented, enter "n" *not* to show,

NO space-time patterns-n:

Note that "backwards" space-time patterns can be toggled on/off while attractor basins are being drawn, and the scale can also be changed. The following reminder will appear in the bottom title bar:

STP:togg-s exp-e contr-c

Enter "s" to show/hide space-time patterns, "e" to expand the scale, "c" to contract the scale.

17.7. Scroll space-time patterns.

By default, space-time patterns (both forward and backward) are presented in successive vertical sweeps, When the pattern reaches the bottom of the screen it continues from the top. Space-time patterns may alternatively be set to scroll, though this considerably reduces the speed of the program. The following prompt is presented if space-time patterns were set,

SCROLL space-time pattern-s:

Enter "s" to scroll.

18. Graphic conventions for attractor basins.

Attractor basins are represented by state transition graphs, where nodes representing network states are linked by directed arcs representing state transitions. Attractor basin topology consists of transient trees rooted on attractor cycles. The attractor cycle may have an arbitrary period, including a period of just one, a point attractor cycling to itself. Transient trees may consist of just one node outside the attractor, or a number of nodes with arbitrary in-degree (but one out degree) at each node. Nodes with zero in-degree, the leaves of the trees, are the so called garden-of-Eden states. The graphic conventions for drawing attractor basins and subtrees are described below.

18.1. Network states, nodes.

Network states are usually represented by circular nodes, but may also be shown as a bitstring pattern in 1d or 2d, or as the decimal or hex value of the state (see #20.2 and #19.2.2).

18.2. Attractor cycles.

Where 3 or more states make up an attractor cycle, this is represented as a regular polygon with nodes at the vertices. The direction of time is clockwise. The "last" vertex is shown due east, so the first vertex at which the cycle is entered is one node clockwise from due east. A point attractor is shown as a node (positioned north-east) on a circle, indicating that

the node cycles to itself. A two state attractor is shown as two nodes on a circle (positioned south-east and north-west), where the south-east node is the "last" node. The diameter of attractor cycles increases asymptotically with system size n , up to the maximum selected, which also determines the scale of the entire basin.

18.3. Transient trees.

The pre-images or predecessors (if any) of the "last" attractor state are computed first. The pre-image fan angle is set according to the in-degree and increases asymptotically to a maximum value with increasing in-degree. In general the centre line of the fan angle is radial to the attractor cycle centre point, but for fans rooted on the attractor, the fan angle is tilted slightly to indicate that the direction of time around the cycle is clockwise. The pre-image fan is computed and drawn for each node at each successive level in the tree. The endpoints of the fan where the nodes are drawn are positioned on notional concentric circles around the attractor cycle corresponding to successive levels in the tree. The distance between each successive concentric circle, and thus between tree levels away from the root, decreases asymptotically for 90 levels, thereafter remains constant. The tree will grow away from the attractor, backwards in time. Once the tree is complete with all its garden-of-Eden states reached, the next tree (anticlockwise) is computed. (note that with *compression* on, equivalent trees will be computed and drawn simultaneously).

18.4. Transient tree colors.

Four colour are used to draw transition arcs. The arcs in one pre-image fan are drawn in the same colour (except for the uniform states, see below). If the attractor period is 5 or less, successive fans are given a different colour so that a given transient tree may have a mix of colours. For attractor periods of 6 or more, all fans in the same tree are given the same colour, and the colour is changed for the next non-equivalent tree. Note that equivalent trees will be coloured identically.

Conventions for node colors are described in #20.2.

18.5. Transient trees for the states all 0s or all 1s.

For CA with *compression* set, a special algorithm is employed to speed up computation of the subtree of the "uniform" states consisting of all 0s or all 1s, when these are on the attractor (it turns out this must be a 1 or 2 state attractor). If a given state is a pre-image of a uniform state, then the given state's rotation equivalents must also be pre-images, and may be computed simultaneously. The first level of the uniform state's tree is organised into groups of equivalents (shown in different colours). The subtree or each representative state in each group at this first level is computed in turn. Each subtree will be completed before the next is started. Successive pre-image fans in each non-equivalent subtree are given a different colour so that a given subtree may have a mix of colours. However equivalent subtrees will be computed and drawn simultaneously, and will be coloured identically.

18.6. Subtree only.

A subtree *only* may be selected, running backward from a given state, or from a state a specified number of steps forward from a given state. In this case the first pre-image fan is evenly spread around a notional circle with a diameter equal to the current maximum attractor diameter. Successive pre-image fans (assigned different colours) are computed and drawn for each node at each successive level in the subtree.

19. Layout of attractor basins.

Basins of attraction are drawn either singly, or in groups in the case of basin fields or a range of network size n . Groups of basins are drawn in successive rows starting from a top left position.

Before drawing starts, the following layout parameters may be altered (or defaults accepted): their scale, position, horizontal and vertical spacing, spacing increase in case of a range of network size n , and the minimum right border before the next row starts. Some layout parameters may be varied after each basin is drawn. A field may be displayed as a successive single basins as well as in a row. The default layout values depend on previous selections. Once revised, the new values generally become defaults, but may be reset to the original values.

If "l" is selected at the first "output parameter" prompt (#16.1), or if the output parameter prompts are viewed in sequence, a layout preview for attractor basins will appear, together with the first layout prompt described below (#19.2).

19.1. The layout preview.

The layout preview indicates the layout, spacing and scale of attractor basins as they would appear on the screen according to the current default settings. This is shown in a window representing the screen at 1/4 size, in the top left quadrant of the screen. Polar co-ordinates are used to scale and position basins, and these are indicated. The x and y axis divide the screen with the x,y co-ordinates 0,0 at the centre. At the left of the screen $x = -100$, at the right $x = +100$. At the top of the screen $y = +100$, at the bottom $y = -100$.

The layout preview represents the basins that might be drawn by concentric circles, where the inner circle is the maximum attractor diameter. Successive concentric circles indicate successive levels. The first basin representation shows the first 90 levels where the distance between successive levels decreases asymptotically (thereafter the distance between levels remains constant). In the first basin every 20th circle is highlighted, as well the attractor circle itself and the first 3 levels. Other basin representations show the attractor circle itself and the first 3 levels only.

If a single basin or subtree was selected, only one basin is shown, unless a range of system size n was also selected, in which case 8 basins are shown. For a basin of attraction field, 30 basins are shown. and for a range of fields, 8 sets of

basins are shown, the first with 5 and the last with 12 basins.

19.2. The first layout prompt.

If "l" is selected at the first "output parameter" prompt (#16.1), or if the output parameter prompts are viewed in sequence, a layout preview for attractor basins will appear (described above, #19.1), together with the following first layout prompt:

```
LAYOUT: reset defaults: all-r, reset for learning-l
select att rad, % of 1/2 screen, x-axis (default 12) (the default depends on previous selections)
```

19.2.1 Reset all layout defaults.

If "r" is entered at the first layout prompt, the original default setting will be restored. These default settings depend on previous, non layout, selections. The revised layout preview will be displayed.

19.2.2 Reset layout for learning.

Enter "l" at #19.2 to choose a default layout appropriate for "learning" (described in #27). A prompt to set the node display (similar to the following) is presented (see also #20.2):

```
nodes: none-n 2d-B 1d-b hex-h, dec-c, spot-s (default-B):
("dec-c" if  $n \leq 32$ , "2d-bits-b" for a 2d network )
```

The attractor basin nodes can be displayed in a variety of different ways,

"s" - as small circles or spots.
"c" - in decimal (for the network size, $n \leq 32$).
"h" - in hex
"b" - as bits in a 1d or 2d graphic array, drawing with mouse.
"B" - (for 1 networks only) as bits shown in 2d .

If "b" is selected, the bit pattern will be displayed in 1d or 2d depending on the dimensions of the network. If "B" is selected the 1d network is represented in 2d as a square or rectangular bit pattern, $i \times j$. The default i (the number of columns) is set automatically as the highest factor of the network size n , but the $i \times j$ setting may be changed as described in the "seed" prompt #15.7.2.

Selecting the default layout for learning is a short cut, further layout and presentation options will be skipped. The full range of node display options may be set at #20.2).

19.2.3. Basin scale, attractor radius.

The first layout prompt (#19.2) gives the current attractor radius scaled according to the x -axis. In the example above "default 12" represents 12% of 1/2 of the screen width. To alter the default basin radius, and the scale of the state transition graphics, enter a new radius. (decimals, or a radius smaller than 1 are allowed).

19.3. Basin start position.

To alter the default position of the first basin, the following prompt is presented:

```
start position, % offset from screen centre:
x (default 80):      y (default 50):      (values shown are examples)
```

Enter the new polar co-ordinates, first x , then y . Remember that the screen centre is at 0,0 and the edges of the screen are as follows, left $x = -100$, right $x = +100$, top $y = +100$, bottom $y = -100$. A position outside the screen limits is legal.

19.4. Show the field as successive basins.

If a basin of attraction field was selected (but not for a range of n), a further option is offered in the #19.3 prompt window, to display each basin in the field singly as successive single basins at the start position. The following prompt is presented,

```
show field as successive basins-s:
```

Enter "s" to select successive basins. Data on each basin is also displayed, and "return" will show the next basin. If this option is selected further basin field layout options are skipped.

19.5. Basin spacing for fields.

To alter the default spacing between basins in a basin field, the following prompt is presented:

```
basin spacing, % of 1/2 screen
x spacing (default 30):      y (default 30):      (values shown are examples)
```

Enter the new spacing according to the polar co-ordinate scale, first x , then y . Remember that the screen is 200 units wide and 200 units high.

19.6. Select minimum right border width.

A further option is offered in the #19.5 prompt window, to select the minimum right border width. This is entered as a percentage of the current x -spacing. If the distance between the centre of the next basin and the right border is less than this setting, the basin will be drawn at the start of the next row. Negative values for the right border are legal. The following prompt is presented,

select right border as % of x spacing (default 50.0): (values shown are examples)

19.7. Amend the layout after each basin or tree.

If one of various "pause" options is selected in #20.4.6 (below), the spacing and/or position of the next basin, can be amended after each basin, tree or subtree is drawn. At the pause data will be displayed in a top right window, and a prompt in the top centre "memory" window will show the following:

options-o layout-a:

"o" gives various options described in #25.3. "a" gives the following top right prompt,

amend next pos-p, or just spacing-ret:

19.7.1. Amend the spacing and right border after each basin.

If "return" is entered to amend just the spacing (and right border), the following prompt is presented (as in #19.5 and #19.6 above):

basin spacing, % of 1/2 screen

x spacing (default 30): y (default 30):

select right border as % of x spacing (default 50.0): (values shown are examples)

Enter the new spacing according to the polar co-ordinate scale, first x , then y . Remember that the screen is 200 units wide and 200 units high. Enter the new the new right border as a percentage of the x -spacing. The new values become the defaults until amended again.

19.7.2. Amending the next position (and spacing) after each basin.

If "p" is entered at #19.7 to amend the next position (as well as the spacing), the following prompt is presented (as in #19.3):

(start= -80,50) amend next pos, % offset from screen centre

x (default -20.7): y (default 30) (values shown are examples)

Enter the next position's polar co-ordinates, first x , then y . Remember that the screen centre is at 0,0 and the edges of the screen are as follows, left $x= -100$, right $x= +100$, top $y= +100$, bottom $y= -100$. A position outside the screen limits is legal. The original start position is shown as a reference. After the new position is set the x co-ordinate becomes the new default x co-ordinate of the original start position, its y co-ordinate is not affected.

When this option is complete, the spacing option #17.7.1 is presented.

19.8. Amend the spacing increase for successive fields.

If a range of fields (for a range of network size n) was selected, the spacing between successive fields is set to increase by a default percentage. This is generally required because the size of basins, and thus the size of their state transition graph, tends to increase with greater n . The following prompt is presented to alter this setting:

select % increase in spacing for successive network size,

x (default 10): y (default 10) (values shown are examples)

Enter the new percentage increase, first for x , the y . This becomes the new default.

19.10. Revise layout parameters.

Finally, the layout preview showing the current settings (as amended) is displayed, with the following prompt to revise or accept:

layout: revise-q, accept-ret:

20. Display of attractor basins, and data.

This sequence of prompts allows changes to the state transition graph display, pauses to show data at various degrees of detail, and options to print and/or save this data.

If "p" is selected at the first "output parameter" prompt (#16.1), or if the output parameter prompts are viewed in sequence, the first display/data prompt will appear for regular CA as in #20.1.1, otherwise as in #20.2 below.

20.1. Compression of equivalent CA dynamics.

If the network is a regular 1d or 2d CA (i.e. with homogeneous regular wiring and rule) rotational symmetries of the periodic array result in equivalent dynamics (Wuensche 1992). A given network state (periodic pattern) A is embedded in a particular past and future made up of other connected network states. If the given state is translated around its circle (1d) or torus (2d) by an arbitrary number of moves, and transformed to state B , the states in B 's past and future must be identical transformations of the states in A 's past and future. The two sets of states are equivalent, and their connection topology is identical.

A complication arises because some states are made up of repeating pattern segments, for example 011,011,011. If the repeating segment size= s , there will be only s equivalent states irrespective of the network size n . whereas with no repeating segments (always the case if n is prime), the number of equivalents is n .

Note also that states with a given repeating segment size s , cannot be upstream of a state with greater s . In an attractor cycle the value of s for all the states must be equal. The uniform states (all 0s and all 1s) with the shortest repeating segment size ($s=1$) are often powerful attractors.

For regular 1d and 2d CA, "compression" algorithms automatically come into play (unless suppressed, see #20.1.1) to compute equivalent basins, transient trees, and sub-trees (from the uniform states), from each prototype. This considerably speeding up the program. For equivalent basins, only the prototype is shown. Equivalent trees and subtrees are shown, but may be suppressed to varying degrees (see 20.1.2)

20.1.1. Suppress compression.

The first prompt for regular 1d or 2d CA allows compression (#20.1) to be suppressed. The prompt gives a reminder of the network dimension, the value of the neighbourhood k , and network size n , and appears as follows,

```
DISPLAY/DATA: regular 1-d CA, nhood=3, size=10:          (values shown are examples)
suppress compression in basins and fields-s:
```

Enter "s" to suppress the compression algorithms.

20.1.2. Suppress copies of trees (and subtrees).

If compression remains set, a further option is offered to suppress copies of transient trees, and subtrees from the two uniform states (all 0s and all 1s). These may be suppressed to varying degrees. The following prompt is presented.

```
suppress copies of trees (& subtrees from all 0s, 1s)
all-3, show g-of-E nodes-2, reduce g-of-E nodes-1:
```

Enter "3" to suppress all copies of equivalent trees or subtrees. This results in a clearer picture of crowded basins. The other options suppress all but the garden-of-Eden states in the equivalent trees or subtrees, showing just a "shadow". Enter "2" for a bold shadow, "1" for a subdued shadow where each g-of-E state is shown as one pixel.

20.2. Node display.

The nodes in attractor basins may be displayed in a variety of ways: symbolically as small circles or spots. explicitly as bit patterns in 1d or 2d depending on the dimensions of the network, though the nodes of a 1d network can also be shown in 2d, or as the hex or decimal value of the pattern. Alternatively, transient nodes need not be shown. A prompt to set the node display (similar to the following) is presented (see also #19.2.2). The default depends on previous selections,

```
nodes: none-n 2d-B 1d-b hex-h, dec-c, spot-s (default-B):
("dec-c" if  $n \leq 32$ , "2d-bits-b" for a 2d network)
```

Note that for non-regular CA, the prompt #20.1 is omitted, and this becomes the first prompt in the display/data sequence, preceded by the title "DISPLAY/DATA".

Select the node display as follows,

- "s" - as small circles or spots. One of 4 colors are set to contrast with transient arc colors (see #18.4), except garden-of-Eden states which are colored white..
- "n" - not to show any nodes in transients, only spots in attractor cycles.
- "c" - in decimal (for the network size, $n \leq 32$).
- "h" - in hex
- "b" - as bits in a 1d or 2d graphic array. Colors of 1s are set to one of 4 colors to contrast with transient arcs colors (see #18.4), except for states on attractor cycles without a transient tree which are colored black. The color of 0s are light grey.
- "B" - (for a 1d networks only) as bits shown in 2d. Colors of 1s as above, 0s white.

If nodes are highlighted in the learning routine (see #27.1), colors of 1s are black unless highlighted. In general the default node display will correspond to the seed selection method selected at #15.1.

20.2.1. Show bit pattern nodes for 1d networks in 2d.

If "b" is selected at #20.2 above, the bit pattern will be displayed in 1d or 2d depending on the dimensions of the network. If "B" is selected, a 1d network is represented in 2d as a square or rectangular bit pattern, $i \times j$. The default i (the number of columns) is set automatically as the highest factor of the network size n . (as for a seed in #15.7.2). The

following prompt gives the current $i \times j$ setting, to revise enter a new i value,

now 15 x 15, reset i-axis : (values shown are examples)

The 1d network is broken up into successive rows starting with the maximum cell index in the top left hand corner. If i is not a factor of n , some cells will be missing from the bottom row.

20.2.2. Alter node label size, decimal or hex.

(if "nodes" are set as dec or hex at #20.2)

The default node label size in decimal or hex is set automatically depending on the "basin scale" (#19.3). This default can be altered by a given factor. The following prompt is presented,

alter default node label size, enter factor:

Enter the factor by which the node label size is to be increased or reduced.

20.2.3. Alter node size, bits.

(if "nodes" are set as bits at #20.2, i.e. "b" or "B")

The default cell cell size was set in #7. This default can be altered by a given factor. The following prompt is presented,

alter cell size, now 4 pixels: (Value shown is an example)

Enter the new cell size in pixels. Note that the cell size is the same as for "backwards" space-time patterns (#17.6) and can be changed "on the fly" (see #25.2).

20.3. Highlight one state in each attractor.

To highlight one state as a bit pattern in each attractor, enter "b" at the following prompt:

highlight one state in each attractor as bit pattern-b:

The state highlighted is the last attractor state reached in the initial forward run, and is positioned due east on the attractor cycle,

20.4. Pause, and data.

A pause may be specified during the drawing of attractor basins to show data on each tree (or subtree from the two uniform states), on each basin in a field, or each field in a range of fields. Data on each field is usually displayed when the field is complete, or on a single basin or subtree when complete, including the time taken to generate. The data is displayed in a top right window. Examples of the types of data displayed, and the decoding of the data, are shown below. More detailed data information may be printed or saved to a file (see below #20.5-7). At the pause a prompt as described in #19.7 allows amending the layout and other options.

20.4.1. Data on sub-trees.

If generating just one sub-tree from an arbitrary state, on completion the following data will be displayed. These examples are for the $k=5$ rule (hex) aa 55 66 a1.

The first example below is for a large network where $n=150$. A random state was iterated forward by 2 steps, and the subtree was generated from the resultant state. This is necessary because a random state is likely to be a garden-of-Eden state. For such a large network ($n>56$) the seed is not shown. If $n \leq 160$ the seed is shown when data is printed or saved (see #20.5-7).

```
subtree=516
g=422 gd=0.857 ml=4 mp=91 51.520 sec
```

The example below show a much smaller network where a random seed was iterated forward by 5 steps before generating the subtree. However, the state reached (the root of the subtree) was on the attractor, so the whole basin was generated. This is indicated by "subtree=basin". DDLab keeps track of a repeat to prevent the subtree running backwards for ever. The resulting transition graph is drawn as a subtree (see #18.6) but is made up of all the states in the basin.

```
subtree=basin=1582 seed(hex)=29 a6
g=1064 gd=0.673 ml=48 mp=27 15.490 sec
```

subtree	the size of the subtree.
seed(hex)	the state at the root of the subtree
seg-root(hex)	the state at the root of the sub-tree, in hex.
g	the number of garden-of-Eden states in the subtree.
gd	the density of garden-of-Eden states in the subtree.
ml	the maximum number of levels in the subtree from the attractor.
mp	the maximum in-degree found in the subtree
sec	the time taken to generate the subtree

20.4.2. Data on a subtree from a uniform state.

An example of data on a subtree from a uniform state, $k=3$ rule 96 (decimal), $n=12$:

```

seg-fan=98 (11 segs) seg-tree 1, no=12 size=264
seg-root(hex)=00 02 g=199 gd=0.754 ml=8 mp=49

segfan      the total number of pre-images of the uniform state.
(x seg)     made up of x segments or groups of rotation equivalent states.
seg tree    the subtree type index.
no          the number of subtrees of this type.
size        the size of the subtree (including the root state).
seg-root(hex) the state at the root of the sub-tree, in hex.
g           the number of garden-of-Eden states in the subtree.
gd          the density of garden-of-Eden states in the subtree.
ml          the maximum number of levels in the subtree from the attractor.
mp          the maximum in-degree found in the subtree

```

20.4.3. Data on trees.

An example of data on a tree in a basin, $k=3$ rule 193 (decimal), $n=12$:

```

tree=3, no=3 size=263 (tree types=3 att=9)
root(hex)=0c c7 g=127 gd=0.483 ml=25 mp=13

tree        the subtree type index.
no          the number of subtrees of this type..
size        the size of the tree (including the root).
tree types total number of non-equivalent tree types,
att         attractor period.
root(hex)   the state at the root of the sub-tree, in hex.
g           the number of garden-of-Eden states in the tree.
gd          the density of garden-of-Eden states in the tree.
ml          the maximum number of levels in the tree from the attractor.
mp          the maximum in-degree found in the tree

```

20.4.4. Data on basins.

An example of data on a basin, $k=3$ rule 193 (decimal), $n=12$, the same basin as for the tree above (#20.4.3).

```

basin 2, equiv basins=4 att(hex)=0c 7c
period=9 size=831 81.2% g=402 gd=0.484 ml=35 mp=13

basin 2     the basin in question is the prototype of the 2nd type.
equiv basins=4 the number of equivalent basins of this type.
att(hex)=0c 7c the "first" attractor state.
period=9    attractor period.
size=831    the size of the basin.
81.2%      the percentage of state-space made up by this basin and its equivalents.
g=402      the number of garden-of-Eden states in the basin.
gd=0.484   the density of garden-of-Eden states in the basin.
ml=35      the maximum number of levels in the basin from the attractor.
mp=13      the maximum in-degree found in the basin

```

20.4.5. Pause after each field.

If basin of attraction field for range of network size was specified in #6.1, the following prompt displayed,

```
pause after each field-y, don't pause-default:
```

Enter "y" to pause and see the data for each successive field, otherwise only the data in the last field of the sequence will be shown.

20.4.6. Pause after each basin or tree.

The following prompt is displayed in the case of a single tree or basin (including a range) or a field (or for a range of fields if the option in #20.4.5 above was selected),

```
pause for data: tree-2, basin-1, none-ret:
```

Enter "1" to pause and see the data for each successive basin, and "2" to also pause and see data on each successive tree (or subtree).

20.5. Print data to printer.

The data as described in #20.4 may be printed to a printer as the attractor basins are drawn. Note the printer limitations described in #A4. As the printing is ascii text only, these are likely to be less severe. The following prompt is presented to initiate printing,

```
print data to printer
```

basin data only-1, and tree data-2

Enter "1" or "2" to select the level of data to be printed. The format for printing the data (and the decoding of the data) is shown in the examples in #20.7 below.

20.6. Save data to file.

The data as described in #20.4 may be saved to a "*.dat" file as the attractor basins are drawn. The following prompt is presented:

```
print data to file
basin data only-1, and tree data-2
```

Enter "1" or "2" to select the level of data to be saved. If saving data is selected, further prompts will appear to set the file name etc. (see filing #28). The format of the ascii data file (and the decoding of the data) is shown in the examples in #20.7 below.

20.7. Format of printed or saved data.

Examples of data (saved to an ascii file or printed) for a basin field (with and without tree information), for a single basin, and for a subtree, are shown and decoded below.

20.7.1. Network data.

The following network data is given first, including the neighbourhood size, rule number, network size and rule parameters. This example relates to the examples in #20.7.2-3 below.

```
k3-rule=(dec)193 (hex)c1      neighbourhood size k, rule number in decimal and hex.
Size=12 ld=0.375 ld-r=0.75    network size, lambda and lambda ratio rule parameters.
zl=0.625 zr=0.75 Z=0.75 C=0/3  Z left, Z right and the Z parameter, canalizing inputs.
```

Note that for networks which are not regular CA, with mixed rules and/or mixed wiring, or mixed neighbourhoods, the network parameters are not given at this stage. They may be printed or saved as an ascii file at the prompt described in #14.7.5.

20.7.2. Basin field data (no tree data).

An example of the data for a basin of attraction field for the network in #20.7.1 above, without tree data,

data

decode

ty. at no (p) s % g gd ml mp reminder of basin data order (see key below).

1. 0fff 1 (1) 34 0.83% 29 0.853 3 29 data for each basin type in the order drawn.

2. 0c7c 4 (9) 831 81.2% 402 0.484 35 13

3. 08e0 2 (18) 312 15.2% 138 0.442 10 6

4. 0041 2 (9) 51 2.49% 27 0.529 4 7

5. 0111 2 (2) 6 0.293% 2 0.333 2 2

basins types=5 total basins=11 total number of basin types and basins in the field

field=4096 gE=1971 density=0.481 field = size of state-space, gE = total garden of Eden states, gE density

key to basin data order: ty. at no (p) s % g gd ml mp

ty basin type numbered in the order drawn.

at the "last" attractor state in hex.

no number of equivalent basins of this type.

(p) attractor period.

s total states in basin.

% percentage of state-space made up of this basin type.

g number of garden-of Eden states in the basin.

gd the density of garden-of-Eden states in the basin.

ml the maximum number of tree levels in the basin from the attractor.

mp the maximum in-degree found in the basin

20.7.3. Basin field data, including tree and subtree data.

An example of the data for the same a basin of attraction field as above (#20.7.1-2), but including tree and subtree data, which precedes the basin data summary. Note that the subtree data is given for each subtree type rooted on the pre-images of a uniform attractor (see #20.4.2).

data

decode

ty. at no (p) s % g gd ml mp reminder of basin data order (see key #20.7.1 above).

tree. root no s g gd ml mp reminder of tree (or subtree) data order (see key below)

seg-fan=29 (4 segs) uniform state total in-degree and number of sub-tree types

1. 0492 3 2 1 0.5 3 1 data for each subtree type in basin 1.

2. 04aa 12 1 1 1 2 0

3. 052a 12 1 1 1 2 0

4. 0aaa 2 1 1 1 2 0

1. 0fff 1 (1) 34 0.83% 29 0.853 3 29 data, basin type 1.
 1. 0d01 3 13 7 0.538 5 4 data for each tree type in basin 2.
 2. 0c53 3 1 0 0 0 0
 3. 0cc7 3 263 127 0.483 35 13

2. 0c7c 4 (9) 831 81.2% 402 0.484 35 13 data, basin type 2.
 1. 0a09 6 2 1 0.5 1 2 data for each tree type in basin 3.
 2. 089b 6 1 0 0 0 0
 3. 0238 6 49 22 0.449 10 6

3. 08e0 2 (18) 312 15.2% 138 0.442 10 6 data, basin type 3.
 1. 04d3 3 1 0 0 0 0 data for each tree type in basin 4.
 2. 01c7 3 3 1 0.333 2 2
 3. 0410 3 13 8 0.615 4 7

4. 0041 2 (9) 51 2.49% 27 0.529 4 7 data, basin type 4.
 1. 0444 2 3 1 0.333 2 2 data for each tree type in basin 5.

5. 0111 2 (2) 6 0.293% 2 0.333 2 2 data, basin type 4.

basins types=5 total basins=11 total number of basin types and basins in the field
 field=4096 gE=1971 density=0.481 field = size of state-space, gE = total garden of Eden states, gE density

key to tree (and subtree) data order: tree.root no s g gd ml mp

tree (or subtree) type numbered in the order drawn.

root the state at the root of the tree (or subtree), in hex.

no the number of equivalent trees (or subtrees) of this type.

s the size of the tree (or subtree) including the root.

g number of garden-of Eden states in the tree (or subtree).

gd the density of garden-of-Eden states in the tree (or subtree).

ml the maximum number of levels in the tree (or subtree) from the attractor.

mp the maximum in-degree found in the tree (or subtree).

20.7.4. Single basin data file.

An example of the data for a single basin without tree data. The data order is the same as in #20.7.2 starting with the attractor state.

no tree data: single basin. 0c7c 4 (9) 831 81.2% 402 0.484 35 13

with tree data:

1. 0d01 3 13 7 0.538 5 4

2. 0c53 3 1 0 0 0 0

3. 0cc7 3 263 127 0.483 35 13

single basin. 0c7c 4 (9) 831 81.2% 402 0.484 35 13

decode

data, tree type 1.

data, tree type 2.

data, tree type 3.

see #20.7.2.

20.7.5. Subtree data file.

An example of the data for a subtree. The rule and network size is the same as in #20.4.1 for a network size of 150. Note that the seed state is only shown for $n \leq 160$.

data

decode

k5-rule=(dec)2857723553 (hex)aa5566a1

network data as in #20.7.1.

Size=150 ld=0.469 ld-r=0.938

zl=0.938 zr=0.672 Z=0.9375 C=0/5

below, subtree size, seed (root of subtree) in hex.

subtree=516 seed(hex)=355ab2b23e87c4ea9ffb0992ae3ca6a3355ab2b23e87c4ea9ffb0992ae3ca6a3

g=442 gd=0.857 ml=4 mp=91

as in #20.7.3.

21. Mutation.

When the drawing of attractor basins is complete for a given network, another may be generated with the same setting, but with a mutation to the network's rule, or to its rule scheme or wiring scheme, or a new random network may be set. If "m" is selected at the first "output parameter" prompt (#16.1), or if the output parameter prompts are viewed in sequence, the first mutation prompt is presented as described below.

21.1. The first mutation prompt.

MUTATION: for next basin/s

wiring-w, rule: special-s, set bits-b, single bit (default):

Enter "return" to mutate a rule (or the rule scheme for a mixed rule network) by flipping one bit at random. Other mutation options are described below.

21.2. Flip bits.

If "b" is selected at the first mutation prompt (#21.1) above, the default number of bits to be flipped (at random) may be altered. The original default is to flip one bit. Successive mutations are cumulative. The following option depends on whether the network has one homogeneous rule, or a rule mix. In this example, the neighbourhood size $k=5$, and the network size $n=10$.

In a network with one homogeneous rule the random bitflips apply to one rule only. The maximum number of bits that may be flipped equals the size of the rule table. The following prompt is presented:

bits to flip=1/32=12%: all-a number-n percentage-p (values shown are examples)

In a mixed rule (or neighbourhood) network, the total number of bits specifying the rule scheme is the sum of the bits in n rule tables. This number is shown in the prompt below for a $k=5$, $n=10$ network, and is the basis for percentage setting. Mutations are made by picking a random bit of a random cell's rule table for each bit flip.

bits to flip=1/320=0.312%: all-a number-n percentage-p (values shown are examples)

If "a" is entered, a new rule scheme is assigned at each mutation.

21.2.1. Alter the number of bits to flip.

If "n" is entered above, the following further prompt is presented:

number of bits:

Enter the new number of bits to flip, which becomes the default.

21.2.2. Alter the percentage of bits to flip.

If "p" is entered at #21.2, the following further prompt is presented:

percentage of total bits:

Enter the new percentage of bits to flip. The resultant number becomes the default.

21.3. Special rule mutation.

If "s", for special rule mutation, is selected at the first mutation prompt (#21.1), various other rule mutation options are offered. A prompt similar to the following is presented,

specify rule mutation

codeno-5 randcode-4 ruleno-3 randrule-2, (5,4 for totalistic only, 3 for $k \leq 5$ only)

bitflip from left: cumulative-1 single-0 (default 2) (not for a neighbourhood mix)

21.3.1. Mutate by the rule or code number.

For $k \leq 5$, if "3" is entered above the rule number is decreased by 1 at each mutation (if "5" is entered, the totalistic code number is decreased by 1). For a rule or code mix (or neighbourhood mix), each rule or code in the network is decreased by 1, after reaching 0 the rule or code number restarts at its maximum value. For a homogenous rule network, when the rule or code reaches 0 the mutation sequence stops.

21.3.2. Assign a random rule or code number.

If "2" is entered at #21.3, a random rule is assigned at each mutation. If "4" is entered, a random totalistic code is assigned.

21.3.3. Mutate by bitflip.

If "1" or "0" is entered at #21.3 above, at each mutation, successive bits in the rule table will be flipped, from left to right. The difference is that for "1" (cumulative) the bits flipped previously will remain, whereas for "0" (single) the previous bits will be restored to their former state. The "single" bitflip results in a set of rules 1 Hamming distance from the start rule, the cumulative bitflip results in a steady increase in the hamming distance. For a rule mix, the mutation will apply to each rule.

The start and end position of the bitflips can be altered from the default at the extreme ends of the rule table. The following prompt is presented:

specify start - end position of bitflip (countdown)

start position (default 31): end position (default 0) (this example for $k=5$)

Enter the new start position, followed by the new end position. Once the end position is reached, the next mutation will revert to the start.

21.4. Mutate wiring.

If "w" is selected at the first mutation prompt (#21.1), the following initial wire move option is presented. In this example, the neighbourhood size $k=5$ and the network size $n=10$, so the total number of wires is 30. In a mixed neighbourhood network the total number of wires equals the sum of the n neighbourhoods.

The original default is to move just one wire. This can be amended to move more wires. The wire mutations are cumulative. The wire moves are made by picking a random wire of a random cell for each wire move. Note that a regular

CA will be re-assigned as a randomly wired network (non-local) if wiring mutation is chosen.

wires to move=1/50=2%: all-a number-n percentage-p: (values shown are examples)

If "a" is entered, a new random wiring scheme is assigned at each mutation.

21.4.1. Alter the number of wires to move.

If "n" is entered above, the following further prompt is presented:

number of wires:

Enter the new number of wires to move, which becomes the default.

21.4.2. Alter the percentage of wires to move.

If "p" is entered above, the following further prompt is presented:

percentage of total wires:

Enter the new percentage of wires to move. The resultant number becomes the default.

21.5. No pause before next mutant.

Enter "y" at the following option to cancel the pause after the attractor basin is complete. The successive mutants will be shown continuously. To interrupt enter "q".

dont pause after mutant graphics-y:

22. Run backwards for subtree, or forwards for a single basin.

If a single basin or subtree was selected in #5.1 (and a seed selected in #15), an option allows running *backwards* to generate a subtree, or running *forwards* to determine the attractor cycle and generate a single basin of attraction, or alternatively to run forwards only for just space-time patterns.

run network: backwards for subtree-b, forwards for basin (default):

22.1. Subtree: Run forwards before running backwards.

If "b" is entered for a subtree at #22 above, an option is presented to first run the network forward from the seed state by a specified number of steps before running backwards. The new seed is the state reached on the forward run, and becomes the root of the subtree. Running forwards before running backwards is usually necessary because a seed selected at random is very likely to be a garden-of-Eden state; a state with no pre-images thus no subtree.

forwards before backwards?

how many steps (default 0, max 90): (the max number of steps depends on the context)

Enter the number of forward steps, or accept the default 0.

22.2. Single basin repeat check limit.

Enter "return" is entered at #22 to select a single basin. The network must first be run forward from the initial seed to find the attractor. States at each successive step are held in a temporary memory and checked for a repeat with each new state. A repeat would define the attractor cycle. It is possible to limit the number of steps memorised and checked from the initial seed. This may be useful for some rules (such as $k=3$, (dec) rules 60,150) which are known to have long attractor cycles with no transients, or just short transients. The default maximum number of steps to be memorised is 10,000, but this can be amended. If zero is set, the network will iterate forwards only, and the forward only prompts will be activated (see #24).

If the dynamics is interrupted (with "q") while the network is still iterating forward looking for the attractor cycle, the following prompt is presented,

still looking for attractor cycle, 4379 iterations (values shown are examples)

exit-q, cont-ret:

22.2.1. Alter repeat check limit, or forwards only.

The following prompt is presented to alter the default repeat check limit, or run forwards only..

default repeat check limit=10,000 steps

draw basin-ret (or enter revised limit)

space-time patterns only (no basin)-s:

Enter the new repeat check limit. If "0" or "s" is entered, the network will iterate forwards only, and the forward

only prompts will be activated (see #24).

23. Final options: subtree, single basin or basin of attraction field. Before drawing attractor basins, a final prompt is presented giving a reminder of the following,

The type of attractor basin; subtree, single basin or field. i.e. subtree, single basin, basin field.
The network size, or start and finish sizes for a range of sizes. i.e. L=14, L=5-12.
The wiring; whether *local* 1d, or *non-local* (including regular 2d). i.e. (local), (non-local).

Local and non local wiring use different reverse algorithms to generate pre-images. The non-local algorithm is considerably slower. Some inner workings of these algorithm may be observed by viewing the partial pre-image stack (for local wiring), or the partial pre-image stack in the form of a histogram (for non-local wiring, with an optional pause). These options are described below, and typical examples of the final prompt are shown, .

23.1. Local wiring.

The following prompt is presented for a network with *local* 1d wiring , though the network may have mixed rules. This network will use the local wiring reverse algorithm.

subtree (local) L=14, exhaustive testing-e (for a subtree with local wiring
view partial pre-image stack-1: values shown are examples)

23.2. Non-local wiring.

The following prompt is presented for a network with *non-local* (or 2d) wiring, which uses the *non-local* wiring reverse algorithm (this example for a range of network sizes).

basin field (non-local) L=5-12, exhaustive testing-e (for a basin field with non-local wiring
view partial pre-image stack-1 fan-pause-2:

23.3. Partial pre-image stack, local wiring.

Enter "1" at the final prompt (#23.1-2), to see how the partial pre-image stack varies in size using the local wiring reverse algorithm . This is shown as a horizontal bar near the bottom of the screen. The length of the bar (in pixels) indicates the varying size of the partial pre-image stack.

23.4. Partial pre-image histogram, non-local wiring.

Enter "1" or "2" at the final prompt (#23.2) for an inner view of the workings of the non-local reverse algorithm. A histogram will appear in the lower half of the screen showing the size of the partial pre-image stack for each successive cell in the network starting from the left. When the attractor basin is complete (or interrupted) final histogram gives the average over all histograms to date.

If "1" is entered, the histogram changes continually, and the following to right prompt appears.

partial pre-image histogram: start pause-p:

If "2" is entered at #23.2 (or "p" is entered above), there will be a pause after each pre-image fan (but not for garden-of Eden states), and the following top right prompt will appear,

abandon pause-p, see next-default:

Generally successive values in the histogram tend to increase up to a middle zone, then decrease. If the value reduces to zero, the state is a garden-of-Eden state (the histogram will not pause). The value of the last green column represents the final set of pre-images. However these pre-images may contain "wildcard" cells due to cells in the network that have no outwires. An additional red column on top of the last green column indicates additional pre-images that result from filling in these wildcards.

23.5. The exhaustive testing reverse algorithm.

Exhaustive testing provides an alternative algorithm for generating pre-images. The algorithm works by running the network forward by one step for each state in state-space, and keeping the 2^n pairs of states (each state and its successor, exhaustive pairs) in memory. The list of exhaustive pairs can also be saved as a "*.exh" file. The pre-images of any state are found by scanning this list.

This method is efficient for basin of attraction fields, or large basins that use up a good proportion of state-space, and additionally is not sensitive to neighbourhood size, k . However the method is highly sensitive to increasing network size n .

If "e" is entered at the final prompt (#23.1-2), the following prompt is presented,

exhaustive pairs: load-l, compute-ret, and save-s:

23.5.1. Load exhaustive pairs.

Enter "l" to load a previously saved list of exhaustive pairs (an "*.exh" file). The filing prompts are described in #28. The network size n , and n for the file to be loaded, must be equal, otherwise the file is not accepted and the following prompt is displayed.

wrong file, continue-return:

In addition, the rule or wiring setup must be identical between the network and the exhaustive pairs file (no automatic check is made), because attractor cycles are established by running forward in the normal way.

23.5.2. Computing exhaustive pairs.

Enter "return" at #23.5 to compute the exhaustive pairs (or "s" to compute and save). A horizontal red bar in the top right corner of the screen will show the proportion of state-space completed, with the following message below,

setting up exhaustive pairs, interrupt-q:

If "q" is entered above, the following prompt is presented,

setting up exhaustive pairs 56% complete, stop-q, options-o, cont-ret: (values shown are examples)

Enter "q" to abandon the exhaustive pairs computation, "o" for the "options" prompts (see #25.3) or return to continue.

23.5.3. Saving exhaustive pairs.

Enter "s" at #23.5 to compute and then save the exhaustive pairs. After the exhaustive pairs are complete (see #23.5.2 above), the following prompt is presented,

SAVE exhaustive pair-s:

If "s" is entered above, prompts for saving the exhaustive pairs as an "*.exh" file are presented (see filing #28).

24. Final options for space-time patterns (forwards only).

If "forwards only" was selected (at the first output parameter prompt #16.1, or #22.2.1) to display just space-time patterns, further prompts are presented in a top right window to set colors, presentation of 2d patterns, and select various ways of analysing the patterns. These setting may subsequently be altered "on the fly" while the space-time patterns are generating.

Note that as in "output parameter options" (#17), a reminder is shown in a top centre window that entering "d" will accept all remaining defaults for the final space-time pattern options, and skip further options.

24.1. Color cells by value or neighbourhood.

By default each cells is colored according to the rule table index that was "looked up" to determine the cells value, in other words the cell's neighbourhood at the previous time-step. These colors are assigned from a palette of 16. Alternatively the cell's may be colored by value, 0-grey and 1-blue. The following prompt is presented in a top right window,

FORWARD ONLY: color cells by vale-v, (default by nhood):

24.2. Pause when screen full.

1d (and possibly 2d) space-time patterns are presented in successive horizontal rows starting at the top of the screen (on the left). 2d network may be given a similar time dimension, see #26.1.5, 26.2.3). When the pattern reaches the bottom of the screen, the iteration may be set to pause, otherwise the pattern is deleted and continues from the top. The following prompt is presented,

PAUSE on space-time screen full-p:

24.3. Frozen generation size.

The space-time pattern presentation may be reset at a later stage to highlight "frozen" cells that have not changed in the previous x time-steps. The number of such "frozen" generations may be reset from the initial default of 20. The new setting becomes the default. The following prompt is presented,

enter new 'frozen' generation size (now 20):

24.4. Analysis.

(does not apply for 2d networks, or if "scrolling" was set in #17.7)

1d space-time patterns may be analysed in two ways: by pattern density (the density of 1s at each time-step), or (for

homogeneous neighbourhoods only) by the look-up frequency (and the entropy of the look-up frequency). These measures are displayed as a graph alongside the space-time pattern. The lookup frequency is simultaneously displayed as a histogram, and may be expanded into a 3d graph with an extra "time" axis.

In general, the following prompt is presented,

```
analysis: none-0, pattern density-1,  
lookup frequency/entropy: graph-3, histogram (default):
```

For a network with mixed neighbourhoods, the following prompt is presented,

```
MIXED NHOODS  
analysis: none-0, pattern density (default),
```

For no analysis enter "0".

24.4.1. Pattern density.

Enter "1" at #24.4 above to show the pattern density. The horizontal scale represents the density of 1s at each time-step (or for the last x time-steps, see #24.5 below). The vertical time axis corresponds to each time-step as drawn.

24.4.2. Lookup frequency and entropy.

(for networks with homogeneous neighbourhoods only)

Enter "return" at #24.4 above to show the lookup frequency at each time-step (or for the last x time-steps, see #24.5 below) and its entropy.

The horizontal scale represents the lookup frequency from 0 to the maximum lookup $n \times g$, where n is the size of the network, and g the number of generations included. The vertical time axis corresponds to each time-step, alongside which 2^k points are plotted, representing the lookup frequency of each rule-table entry, and colored accordingly. A point representing the entropy of this distribution is plotted on the same graph.

A separate histogram of the lookup frequency (on its side) is shown to the right, with 2^k columns representing the lookup frequency of each rule-table entry, and colored accordingly. If "3" is entered at #2.4, the histogram will expand into a 3d graph with an extra "time" axis, shown as an axonometric projection.

24.5. Analysis generation size.

(if "0" was *not* entered at #24.4)

The analysis of space-time patterns in #24.4.1-2 may be set to include x previous time-steps (in addition to the present time-step in the case of pattern density). The number of generations included may be reset, the initial default is one. The new setting becomes the default. The following prompt is presented,

```
enter new 'analysis' generation size (now 1):
```

24.6. State-space matrix scatter-plot (forwards only).

The state-space matrix (or scatter-plot) plots each state in the space-time pattern on a 2d grid near the bottom of the screen, plotting the left half against the right half of each state bit string (as in #17.1). The x axis represents the left $n/2$ bits, the y axis represents the right $n/2$ bits. If n is odd, the extra bit is included on the left, and the grid is a flat rectangle, otherwise the grid is square. The resolution of the matrix is set automatically according to n .

The following prompt is presented,

```
show state-space matrix-m:
```

25. Changing settings "on the fly" (attractor basins).

While attractor basins are being drawn according to the presentation, layout and other settings, some of these settings can be reset, either immediately by a key press, or by presenting further prompts. Many network parameters may also be reset without the need to backtrack to the early prompts with "q".

25.1. Interrupting attractor basins.

Enter "q" to interrupt attractor basins. If the interrupt occurred between successive pre-images within one pre-image fan (as opposed to between successive pre-image fans) the following message is initially displayed (top right),

```
early exit - in pre-image fan
```

One of the following prompts is then displayed, depending on whether the attractor basin is a subtree, single basin or a basin of attraction field,

```
next tree/basin-n, options-o, stop field-q, cont-ret: (for a basin of attraction field)
```

```
next tree-n, options-o, stop basin-q, cont-ret: (for a single basin)
```

options-o, stop subtree-q, cont-ret: (for a subtree)

If the in-degree histogram was set in #17.2 a further option is offered to view the histogram as computed so far, for example,

inhist-h, options-o, stop subtree-q, cont-ret: (for a subtree)

Enter "h" to view the in-degree histogram (see #17.2.2-6). Options "q", "o" and "n" are described below (#25.1.1-3). After these various options are complete, the program generally returns to the prompt above (#25.1). Enter "return" to continue the attractor basin from the point of interrupt.

25.1.1. Backtrack through prompts.

Enter "q" at #25.1 to abandon the attractor basin. The data on the incomplete attractor basin is shown (#20.7), and the following options are presented in a top left window (the same as when the drawing of attractor basins is complete).

graphic-g, image: print/save/load-p/s/l, options-o
toggle: single-field-t STP-P
network: view/revise/learn-n, back-q, cont-ret:

These options are described in #25.3 below.

The data on an interrupted basin of attraction field will be preceded by the words "EARLY EXIT" to indicate that the data applies to an incomplete field.

25.1.2. Abandoning a tree and continuing with the next tree.

If "n" is entered in #25.1 above, the tree that is being generated will be abandoned, and the next tree started (or subtree from the uniform states). For a subtree only, or if no more subtrees remain, the options in #25.1.1 are displayed.

Note that abandoning a subtree before it is complete will result in errors in attractor basin data (#20.7). For a basin of attraction field, the bar indicating the state-space used so far is likely to go off its scale, with a message such as "ERROR excess states=19124". This indicates that more states were computed than the size of state-space.

25.1.3. Other options.

If "o" is entered at #25.1 above, the options in #25.3 will be displayed. When the use of these options is complete, the attractor basins will continue from the point at which they were interrupted. Note that if changes to the network parameters are made at this point, errors in data and attractor basin structure are likely as in #25.1.1 above.

25.2. Key hit attractor basin options.

A number of key hits allow changing the presentation of attractor basins "on the fly". A reminder of some of these are shown in the bottom title window. The commands (apart from "q") only activate between pre-image fans, so the key may need to be hit more than once.

backtrack/interrupt-q matrix-m STP:togg-s exp-e contr-c

The following key hits will affect changes as follows,

- "s" to toggle the "backwards" space-time pattern on and off (see #17.6).
- "e" to expand the scale of the space-time pattern.
- "c" to contract the scale of the space-time pattern.
- "m" to toggle the state-space matrix scatter-plot on and off (see #17.1).
- "p" to toggle the pause on and off when showing the partial pre-image histogram for the non-local wiring reverse algorithm (see #23.4).

25.3. Attractor basin complete options.

When the attractor basin is complete, or if it was abandoned by interrupting in #25.1 (or options are selected at the pause prompt, #19.7), the following options are presented in a top left window.

graphic-g, image: print/save/load-p/s/l, options-o
toggle: single-field-t STP-P Seed-S ("S" for single basins or subtrees only)
network: view/revise/learn-n, back-q, cont-ret:

Enter "return" to generate the next "mutant" attractor basin (see #21), "q" to backtrack, or select other options to do the following, (enter "o" for a range of further rules or seed options). Once the options are complete, the program generally returns to the prompt at #25.3 above.

- "g" to change the graphics setup between VGA(640x480) and SVGA(1024x768), or to change the background color between white and black (see #5.2).
- "p" to print the graphics screen to a printer (see #A4. for printer limitations).
- "s" to save the graphics screen to a "*.nat" file (see "filing" #28).
- "l" to load a graphics image file to the screen (see "filing" #28).
- "t" to toggle between a single basin and the basin of attraction field. The prompt sequence will revert to selecting the

- seed (#15) or the first presentation parameters prompt (#16.1). Note that the "layout" defaults may need to be reset.
- "P" to toggle between showing and not showing "backwards" space-time patterns (see #17.6).
 - "S" to reset the seed, for single basins or subtrees only. A central lower window opens up, with prompts as in #15.
 - "n" to see and revise the network parameters by invoking the "Reviewing wiring/rules" routine described in #14. The "learning/forgetting" routine for attaching/detaching sets of states as pre-images of a target state may also be invoked, described in #27).
 - "o" to open up a range of further options described #25.4 below.

25.4. Further attractor basin complete options.

If "o" is entered at #25.3 above, a top right window displays a range of further options as follows,

```
rule: save-s load-l revise-v comp-c neg-n ref-r canalyzing-C
bitflip/flipback-1, flip ends-w/b, Z:higher/lower-Z/z ("1" and "Z/z" for single rule networks only)
seed: rev-e, rand/block-4/k, sing pos/neg-5/6, save/load-S/L ("Seed:" for single basins
goback-q, ret-cont: or subtrees only)
```

Select options to do the following,

25.4.1. Rule options.

```
rule: save-s load-l revise-v comp-c neg-n ref-r canalyzing-C
bitflip/flipback-1/2, flip ends-w/b, Z:higher/lower-Z/z ("2" and "Z/z" for single rule networks only)
```

- "s" to save a single rule, or the rule at cell index 0 for a mixed rule network.
- "l" to load a single rule, or the rule at cell index 0 for a mixed rule network.
- "v" to revise a single rule, or the rule at the specified cell index for a mixed rule network, where the following prompt is displayed below the window at #26.5.

enter cell index, 9-0:

A window opens at the lower right corner of the screen to reset the rule, with prompts as in #13.6-14.

- "c" transform a single rule (or all rules in a rule mix) to its *compliment*, as in #13.17.1.
- "n" transform a single rule (or all rules in a rule mix) to its *negative*, as in #13.17.2.
- "r" transform a single rule (or all rules in a rule mix) to its *reflection*, as in #13.17.3.
- "C" sets canalyzing inputs or genes in a single rule, or in a rule mix. A window opens in the lower right corner of the screen, and prompts are presented as described in #13.17.5.
- "1" flip a random bit in a single rule, or a random bit flip in each rule for a rule mix.
- "2" restore the random bit flips in reverse order (applies to single rule networks only).
- "w" flip the output of the all 1s neighbourhood in a single rule, or for all rules in a rule mix.
- "b" flip the output of the all 0s neighbourhood in a single rule, or for all rules in a rule mix.
- "Z" mutate rule to force the Z parameter higher (applies to single rule networks only)
- "z" mutate rule to force the Z parameter higher (applies to single rule networks only)

25.4.2. Seed options.

(for a single basin or subtree only)

```
seed: rev-e, rand/block-4/k, sing pos/neg-5/6, save/load-S/L
```

- "e" to reset the seed, for single basins or subtrees only. A central lower window opens up, with prompts as in #15.
- "4" to reset the seed at random, as in #15.5
- "k" to reset the seed with a random central block of cells, the remainder set to 0, as in #15.5
- "5" to reset the seed a positive singleton, a central cell set to 1 with the remainder set to 0.
- "6" to reset the seed a negative singleton, a central cell set to 0 with the remainder set to 1.
- "S" to save the seed to a "*.eed" file, see filing #28.
- "L" to load a new seed from a "*.eed" file, see filing #28.

26. Changing settings "on the fly" (space-time patterns only).

While space-time patterns are being drawn, various settings can be reset, either immediately by a key press, or by further prompts, without the need to backtrack to the early prompts with "q".

26.1. The "Interrupt Key Index" (space-time patterns only).

An "Interrupt Key Index" window is usually displayed on the right of the screen while space-time patterns are drawn as a reminder of the various "on the fly" options, with subheadings (in red) and the key hit options as shown below. This window can be suppressed. Some options depend on the current settings. The options as they appear in the window, with some brief explanations, are shown below. Note * indicates that a further explanation is provided in #26.3.

 INTERRUPT KEY INDEX

26.1.1. change rule table	
r.. random rule	change/mutate the rule table, (or tables for a mixed rule network)
1/2.. rand bitflip/restore	assign a new random rule (all rules in a mixed network)
Z/z.. force Z higher/lower	mutate rule by 1 random bit, (for each rule in a mixed network)
w/W.. toggle/force all 1s	mutate rule to change the Z parameter (single rule network only)
b/B.. toggle/force all 0s	change the "all 1s" bit, or force to 1 (for all rules in a mixed network)
8.. toggle RULEMIX/single	change the "all 0s" bit, or force to 0 (for all rules in a mixed network)
g.. load glider rule	or SINGLE/rulemix (mixed rule network only)*
	(only if the neighbourhood $k=5, 6$ or 7 at cell index 0)*
26.1.2. change wiring	
m.. move 1 wire	or a given number of wires, a local network is redefined as non-local*
M.. revise wiring moves	to change the default number of wires to be moved above*
7.. toggle NONLOCAL/local	or LOCAL/nonlocal (for a nonlocal network only)*
26.1.3. change seed/size	change/mutate the current network state
4/k.. rand seed/block	reseed the network with a random seed or random block (see #15.5).
R.. rand bitflip	mutate the current network by 1 random bit.
5/6.. singleton pos/neg	reseed the network with a central 1 surrounded by 0s, or visa versa.
i/d.. inc/decrease 1 cell	increase or decrease the network size by one cell (not for a 3d network)
26.1.4. presentation	
3.. toggle color setting	toggle between cell colors according to the neighbourhood "looked up"
	cell or nhood value and the actual cell value (see #24.1)
C.. toggle background color	if cells are colored by value, toggle the 0 color (background - grey).
S.. toggle sound	primitive sound can be produced by the network.
e/c.. expand/contr scale	expand or contract the cell scale, thus the scale of space-time patterns.
	(2d network only)
26.1.5. 2d space-time pattern	
t/T toggle 3d-2d/2d-1d	to show a <i>time</i> dimension, 3d = isometric projection, 1d = slices*
p.. draw plane	(in the 3d isometric projection)*
26.1.6. highlight frozen cells	
h.. toggle CELL-semi-frozen	or SEMI-frozen-cell or FROZEN-cell-semi*
H.. generations (now 20)	to change default number of generations defined as frozen above*
26.1.7. analysis	(if analysis not suppressed, and not for 2d)
s.. toggle LOOKUPFREQ-eeg	or EEG-lookupfreq (see #24.4.1-2)*
L.. toggle HISTOGRAM-graph	or GRAPH-histogram (not for mixed neighbourhoods)*
G.. generations (now 1)	to change default number of generations above, see #24.5*
26.1.8. state-space matrix	
y/Y.. reset/toggle on-off	to reset an empty matrix, or show/not show the matrix (see #24.6)
26.1.9. index, pause, quit	
x.. toggle index display	to show/not show this "INTERRUPT KEY INDEX".
f.. toggle end pause (off)	or (on) to pause/not pause at the foot of the screen.
q.. pause or quit	to pause and see further options (#26.3), or backtrack.

26.2. The "Interrupt Key Index", some further details .

Further explanations of some of the "Interrupt Key Index" options (marked with an * in #26.1) are given below.

26.2.1. change rule table	change/mutate the rule table, (or tables for a mixed rule network)
8.. toggle RULEMIX/single	or SINGLE/rulemix (mixed rule network only)*
	A mixed rule network can be toggled between mixed and single. When single, the rule at cell index 0 applies to all cells
	g.. load glider rule (only if the neighbourhood $k=5, 6$ or 7 at cell index 0)*

A collection of rules that produce coherent interacting structures (known as *gliders*) are provided in separate files with DDLab. There are currently 60 $k=5$ rules, 5 $k=6$ rules and 10 $k=7$ rules. Provided that the neighbourhood size of the single rule network k is 5, 6 or 7, hitting key "g" changes the current rule to a randomly chosen glider rule from the appropriate file. Loading glider rules applies to 1d networks, though the command will also work with 2d networks.

For a mixed rule or neighbourhood network, provided that k for cell index 0 is 5, 6 or 7, hitting key "g" will load a glider rule at cell index 0. This rule can be applied to the whole network by toggling RULMIX/single with key "8" above.

26.2.2. change wiring

m.. move 1 wire	or a given number of wires, a local network is redefined as non-local*
	A default number of wires (the starting default=1), randomly chosen, are recoupled at random. If the current network has <i>local</i> CA wiring, it will be redefined as a <i>non-local</i> (randomly wired) network.
M.. revise wiring moves	to change the default number of wires to be moved above*

Hit key "M" to change the number of wires to move above. A top right prompt will appear, giving the current default as a number and percentage (values below are examples). First select the method for resetting (or enter "a" for all wires), then enter the new number or percentage.

wire to move=1/750=0.133%: all-a number-n percentage-p:
percentage of total wires: (or number of total wires:)

- 7.. toggle NONLOCAL/local or LOCAL/nonlocal (for a nonlocal network only)
Key "7" toggles a non-local network between dynamics according to its non-local wiring, and dynamics as if it were wired *locally* (as a CA). The non local wiring remains intact.

26.2.3. 2d space-time pattern (2d network only)

- t/T toggle 3d-2d/2d-1d to show a *time* dimension, 3d = isometric projection, 1d = slices*
The 2d space-time pattern is initially displayed as a normal 2d grid at the top right of the screen. Key "t" will toggle the presentation between this and an isometric projection (as if looking *up* at a transparent lift shaft), where each successive 2d grid is placed *below* the existing stack of 2d grids. This provides a time dimension as with 1d space-time patterns, is best seen when cells are colored according to the neighbourhood.

Key "T" will toggle the presentation between a 2d display and a "1d" space-time pattern made up of "slices". The rows of the 2d pattern are shown side by side, the top row on the left
p.. draw plane (in 3d a isometric projection)*

Key "p" will draw a translucent plane in the 3d isometric at the current time-step to highlight the isometric geometry. This also occurs if a new seed is assigned to the network.

26.2.4. highlight frozen cells

- h.. toggle CELL-semi-frozen or SEMI-frozen-cell or FROZEN-cell-semi
This option allows "frozen cells", that have not changed over the last *x* time-steps, to be highlighted. The initial default is *x*=20, but this can be reset (see below). Key "h" toggles between the following
CELL cell colors by value.
SEMI frozen cells colored green, the rest by value.
FROZEN frozen cells colored green, the rest the background color.

H.. generations (now 20) (to change default number of generations defined as frozen above)

The default number of generations (or time-steps) for a cell to be considered "frozen" may be reset. If key "H" is hit, the following top right prompt appears,

enter new 'frozen' generation size (now 20):

Enter the new size which becomes the default.

26.2.5. analysis (if analysis not suppressed, and not for 2d)

- s.. toggle LOOKUPFREQ-eeg or EEG-lookupfreq (see #24.4.1-2)*
Key "s" toggles the analysis of 1d space-time patterns between the two methods of analysis (see #24.4).
LOOKUPFREQ. (see #24.4.2, for homogeneous neighbourhood networks only) The lookup frequency, entropy plots, and the lookup frequency histogram or 3d graph. This is calculated on the basis of the last *x* time-steps (generations). The initial default is *x*=1, but this can be reset (see "G" below, or #24.5).
EEG. (see #24.4.1) The pattern density, the density of 1s at each time-step (analogous to the electroencephalogram (EEG) of the mean excitatory state of a patch of neurons). This is calculated on the basis of the last *x* time-steps (generations) including the current time-step. The initial default is *x*=1, but this can be reset (see "G" below, or #24.5).

L.. toggle HISTOGRAM-graph or GRAPH-histogram (not for mixed neighbourhoods)*

The lookup frequency histogram may be extended into a 3d graph with an extra "time" axis. Key "L" toggles between the two methods of presentation.

G.. generations (now 1) to change default number of generations above, see #24.5*

The default number of generations (or time-steps) for both the lookup frequency and pattern density analysis may be reset. If key "G" is hit, the following top right prompt appears,

enter new 'analysis' generation size (now 1):

Enter the new size which becomes the default.

26.3. Pause space-time pattern.

If "q" is hit at the "Interrupt Key Index" options (#26.1.9), or if *pause* is set (#24.2), when the space-time pattern reaches the foot of the screen the program will pause. Two top right windows are displayed. The first window (for single rule networks only) shows details about the current single rule (as in #13.6.2) and is described in #26.4 below. The second

window presents a number of further options, described in 26.5.

26.4. Rule details (space-time patterns only).

(for single rule networks only)

On pausing the space-time pattern, a top right window shows rule details as follows: the rule as a bit pattern (for all k), in decimal (for $k \leq 5$), and in hex (for all k), where k =neighbourhood size. In addition the window gives the network size, and details about the rule's lambda and Z parameters, and canalizing inputs. This window is updated as rules are transformed or mutated.

An example of the data in a rule window for a $k=5$ rule is shown below:

```
..... 1828671743 6c ff 50 ff
k5-rule ld=0.688 ld-r=0.625 zl=0.344 zr=0.531 Z=0.53125
C=1/3 *3*** Size=10
```

key to data

1828671743 the rule number in decimal (for $k \leq 5$ only)
6c ff 50 ff the rule number in hex
k5-rule the neighbourhood size k .
ld the λ parameter.
ld-r the λ ratio.
zl Z left.
zr Z right
Z the Z parameter.
C= $x/3$ the proportion of canalizing inputs.
*3*** shows exactly which of the k inputs are canalizing (if none this is not shown),
in this case there is 1 canalizing input, at neighbourhood index 3.
Size network size n .

26.5. Further space-time pattern options.

On pausing the space-time pattern, a top right window displays a range of further options (some similar to the options in #25.4),

```
prt scrn with/out/justbox-p/P/B save scrn-V
rule: save-s load-l revise-v comp-c neg-n ref-r canalizing-C
bitflip/flipback-1/2, flip ends-w/b, Z:higher/lower-Z/z ("2" and "Z/z" for single rule networks only)
seed: rev-e, rand/block-4/k, sing pos/neg-5/6, save/load-S/L
toggle wiring-7, start at top-t, goback-q, ret-cont: ("7" for a non-local network only)
```

These options are categorised as: "screen", "rule", "seed" and "other", and described below.

26.5.1. Screen options.

```
prt scrn with/out/justbox-p/P/B save scrn-V
```

"p" to print the graphics screen to a printer (see #A4. for printer limitations) including the rule window #26.4, but not the prompt window (#26.5) above .

"P" to print the graphics screen to a printer (see #A4. for printer limitations) *excluding* the rule window #26.4 and the prompt window (#26.5) above.

"B" to print just the rule window, #26.4.

"V" "s" save the graphics screen to a "*.nat" file (see "filing" #28).

26.5.2. Rule options.

```
rule: save-s load-l revise-v comp-c neg-n ref-r canalizing-C
bitflip/flipback-1/2, flip ends-w/b, Z:higher/lower-Z/z ("2" and "Z/z" for single rule networks only)
```

"s" to save a single rule, or the rule at cell index 0 for a mixed rule network.

"l" to load a single rule, or the rule at cell index 0 for a mixed rule network.

"v" to revise a single rule, or the rule at the specified cell index for a mixed rule network. For a mixed rule network the following prompt is displayed below the window at #26.5.

```
enter cell index, 9-0:
```

A window opens at the lower right corner of the screen to reset the rule, with prompts as in #13.6-14.

"c" transform a single rule (or all rules in a rule mix) to its *compliment*, as in #13.17.1.

"n" transform a single rule (or all rules in a rule mix) to its *negative*, as in #13.17.2.

"r" transform a single rule (or all rules in a rule mix) to its *reflection*, as in #13.17.3.

"C" sets canalizing inputs or genes in a single rule, or in a rule mix. A window opens in the lower right corner of the screen, and prompts are presented as described in #13.17.5.

- "1" flip a random bit in a single rule, or a random bit flip in each rule for a rule mix.
- "2" restore the random bit flips in reverse order (applies to single rule networks only).
- "w" flip the output of the all 1s neighbourhood in a single rule (for all rules in a rule mix).
- "b" flip the output of the all 0s neighbourhood in a single rule (for all rules in a rule mix).
- "Z" mutate rule to force the Z parameter higher (applies to single rule networks only)
- "z" mutate rule to force the Z parameter lower (applies to single rule networks only)

26.5.3. Seed options.

seed: rev-e, rand/block-4/k, sing pos/neg-5/6, save/load-S/L

- "e" to reset the seed. A central lower window opens up, with prompts as in #15.
- "4" to reset the seed at random, as in #15.5
- "k" to reset the seed a random central block of cells, with the remainder set to 0, as in #15.5
- "5" to reset the seed a positive singleton, a central cell set to 1 with the remainder set to 0.
- "6" to reset the seed a negative singleton, a central cell set to 0 with the remainder set to 1.
- "S" to save the seed to a "*.eed" file, see #15.10 and filing #28.
- "L" to load a new seed from a "*.eed" file, see filing #28.

26.5.4. Other options.

toggle wiring-7, start at top-t, goback-q, ret-cont: ("7" for a non-local network only)

- "7" to toggles a non-local network between dynamics according to its non-local wiring, and dynamic as if it were wired *locally* (as a CA). The non local wiring remains intact.
- "t" to restart space-time patterns from the top of the screen.
- "q" to backtrack through the prompt sequence.

26.5.5. Further options, first backtrack.

If "q" to bactrack is entered at #26.5.4 above, the following options are presented in a top left window (similar to #25.3).

graphic-g, image:print/save/load-p/s/l, options-o

Seed-S

network: view/revise/learn-n, back-q, cont-ret:

- "g" to change the graphics setup between VGA(640x480) and SVGA(1024x768), or change the background color between white and black (see #5.2).
- "p" to print the graphics screen to a printer (see #A4. for printer limitations).
- "s" to save the graphics screen to a "*.nat" file (see "filing" #28).
- "l" to load a graphics image file to the screen (see "filing" #28).
- "S" to reset the seed. A central lower window opens up, with prompts as in #15.
- "n" to see and revise the network parameters by invoke the "Reviewing wiring/rules" routine described in #14. The "learning/forgetting" routine for attaching/detaching sets of states as pre-images of a target state may also be invoked, described in #27).
- "o" to return to the "Further space-time pattern options" prompt (#26.5).
- "q" to backtrack.
- "return" to continue the space-time pattern.

27. Learning, forgetting, and highlighting.

The global dynamics of a network, as represented by its basin of attraction field, sums up its repertoire of behaviour, where the root of any subtree categorises all states in the subtree. This hierarchical categorisation implicit in a given network may be seen as its "content-addressable" memory in the sense of Hopfield (1982), but with the added notion that every subtree root (not just point attractors) provides a memory category, .

For networks with mixed wiring and/or rules, DDLab allows one or more states to be added or deleted as pre-images of a given state. Adding or deleting pre-images is analogous to learning and forgetting. The network architecture is automatically revised to produce the required change either by moving wires or mutating rules, using learning algorithms analogous to back propagation.

It turns out that the rule mutation algorithm is bound to learn a list of aspiring pre-images without forgetting the state's pre-existing pre-images, though pre-images not on the list may also be learnt. There is just one specific rule scheme mutation, which is bound to succeed. By contrast there are many alternative wire move mutations that may achieve the desired result, though success is not guaranteed, and pre-existing pre-images may be forgotten.

When applying both rule and wire learning algorithms, side effects are bound to occur elsewhere in the basin of

attraction field. In small networks, the revised attractor basin may be re-drawn to show the results and side affects of learning. The attractor basin may be progressively *sculpted* (and the network adapted) to produce a desired scheme of hierarchical categorisation. In general, forgetting pre-images causes less severe side effects than learning, as minimal changes to the network architecture are required.

Note that learning by wire moves requires a network with "random" or non-local wiring (see #11.3 to treat local wiring as if it were random). Mixed neighbourhoods are considered to always have non-local wiring in this context. Learning by rule mutation requires a network with mixed rules (see #13.3.1 to set up a rule mix where all the rules are the same).

27.1. Highlighting states.

The nodes corresponding to a list of states may be highlighted in attractor basins (see #27.8.1, #27.10). This highlights the results of learning and forgetting, but is also useful for seeing how a particular set of states is distributed, and how the distribution changes when network parameters are altered. Highlighted nodes are shown with an outer border. The target node is shown in red and aspiring pre-images in blue. For highlighted nodes shown as bit patterns, these are the colors of 1s, otherwise 1s are shown black. Nodes other than highlighted nodes may be suppressed entirely. The highlighting feature may be used for regular 1d CA.

27.2. Default attractor basin layout for learning.

When using the learning algorithms for "sculpting" the attractor basins of small networks, a default set of appropriate layout parameters is available. At the first "output parameter" prompt #16.1 select "1" to jump directly to the layout prompts (#19). At the first layout prompt (#19.2) enter "1" to choose the default layout for learning, and set the type of node display (see #19.2.2).

27.3. Selecting the wiring graphic.

Whenever the network is reviewed as a 1d or 2d wiring graphic (#14), learning or highlighting may be implemented. The following prompt is presented, depending on whether the network was set up as 1d or 2d (Note that for 2d networks, both a 1d and 2d "wiring graphic" is applicable).

```
learn/show/revise rulemix + wiring -1:          (for 1d networks)
```

```
learn/show/revise rulemix + wiring: 1d-1 2d-2:  (for 2d networks)
```

This prompt may be reached from various points in DDLab, including the following,

1. After the network architecture is specified (see #14 and #14.4.4).
2. After attractor basins are complete (see #25.3) or interrupted, (see #25.1.1).
3. If "forwards only" space-time patterns are interrupted, and "q" is entered to backtrack (see #26.5.5).

In the case of 2 and 3 above, a top left prompt presents various options including the following,

```
network: view/revise/learn-n
```

Enter "n" to review the network and implement learning or highlighting..

27.4. Selecting the learn/highlight window.

If the wiring graphic is selected by entering "1" or "2" (for 1d or 2d) in #27.3 above, the wiring graphic appears in the lower half of the screen, together with a top right prompt described fully in #14. This example of the prompt is for a 1d network,

```
show cell wiring/rule, use arrows to move, jump-j
one cell with right/left arrow, 10 cells with up/down arrow   (values shown are examples)
options-o, print box-p, revise wiring-w, revise/transform rule-r/t   ("r/t" only if a rule scheme was set)
show avZ-z, learn/highlight-l, cont-ret:                        ("z" only if mixed rule network set)
```

Enter "1" to open the learn/highlight window. Note that direct changes to the wiring and rules may be made directly from this prompt, described in detail in #14.4 and #14.6). These options may be used in conjunction with automatic learning/forgetting.

27.5. Select the "target" state.

If "1" is entered at #25.6 above, the wiring graphic window is replaced by a series of "learn/forget/highlight" windows. The initial prompts allow a "target" state to be selected. Prompts are presented in two separate windows, one above the other. The top window indicates that the target state is to be set,

```
learn/forget/highlight: set target state:
```

The lower window presents the first of a series of prompts to select the target state,

```
Select target state, empty-e fill-f (default-dec)   ("default-dec" if  $n \leq 32$ , otherwise "default-random")
random-r 1d-b 2d-B hex-h repeat-p load-l:
```

These prompts are the same as for selecting a seed. They are the fully described in #15.1-10.

27.6. Select a set of states as pre-images.

Once the target state is selected, one or more states may be specified as the "aspiring pre-images" of the target state. These will be added to the target state if learning, or detached if forgetting. Alternatively the states may just be just highlighted.

The set of states may be an arbitrary list or a range of decimal equivalents (if $n \leq 32$), or may be based on Hamming distance, parity, or the previous list may be repeated. The following lower left prompt is presented,

```
pre-images: as before-b parity-p range-r hamm-h1/2 number-(default 1): ("range-r" if  $n \leq 32$ )
```

Enter a number to specify the size of an arbitrary list of aspiring pre-images, or make alternative selections as follows,

```
"b" - to repeat the previous list.
"p" - to select states based on odd or even parity.
"r" - to specify a range of decimal equivalents, i.e. from 10 to 50 (if  $n \leq 32$ ).
"h1" - to select states based on a hamming distance of 1 from the target state.
"h2" - to select states based on a hamming distance of 2 from the target state.
```

Once set, the target state and list of aspiring pre-images are reviewed in a window (see #27.7).

27.6.1. An arbitrary list of aspiring pre-images.

If a number x is entered at #27.6, prompts will be presented to set each of the x aspiring pre-images, for example,

```
Select aspiring pre-image 1, empty-e fill-f (default-dec) (for aspiring pre-image 1,2,3,...x)
random-r 1d-b 2d-B hex-h repeat-p load-l: ("default-dec" if  $n \leq 32$ , otherwise "default-random")
```

These prompts are the same as for selecting a seed. They are the fully described in #15.1-10.

27.6.2. Pre-images according to Hamming distance.

If "h1" or "h2" is entered at #27.6, the aspiring pre-images will be automatically selected according to Hamming distance from the target state. "h1" selects the n states that differ by 1 bit from the target state. "h2" selects the larger set of states that differ by 2 bits from the target state (h2). The following reminder will appear at the start of the next prompt (#27.7),

27.6.3. Range of decimal equivalents.

If "r" is entered at #27.6, a decimal equivalent range of aspiring pre-images will be automatically set between two selected values (the maximum decimal equivalent that can be set is $2^{32}-1$). The following sequence of prompts is presented to set the start and end decimal equivalents, and the "step" or increment size (default 1),

```
range of pre-images (default 1-10, max 1024), start: end: step: (example for  $n=10$ )
```

27.6.4. Odd or even parity.

Odd and even parity signifies that the total of 1s in a bit string is an odd or even number (zero has even parity). If "p" is entered at #27.6, the aspiring pre-images will be automatically specified as those states with the given parity relating (at most) to the first 20 cells (where the remaining cells have value 0). The following prompt is presented,

```
enter odd-1, or even-(default):
```

Select odd or even parity. A further prompt allows the parity to relate to part only of the system size n , where $n \leq 20$, or part only of the first 20 cells.

```
enter part system size (default 10): (for a system size where  $n=10$ )
```

If a size x less than the system size n (or less than 20 if $n > 20$) is entered, states with the given parity relating only to cell indices 0 to x (and where cell indices $x+1$ to n are equal to 0) will be selected.

27.6.5. Repeat previous selection.

Enter "b" at #27.6 to repeat the previous selection of aspiring pre-images.

27.7. Review target state and aspiring pre-images.

(applies only if the maximum decimal equivalent of the pre-image is smaller than $2^{32}-1$)

Once the target state and list of aspiring pre-images have been set in # 27.5-6 above, the decimal equivalents of the selected states will be displayed in a new window. In information on range, parity, or Hamming distance selections is also displayed, for example,

```
target state=229: number of pre-images=10, 1-bit mutants, max state=228, continue-ret:
228,231,225,237,245,197,165,101,485,741, (example for Hamming distance=1,  $n=10$ )
```

```
target state=229: number of pre-images=10, range 1-10 step 1, continue-ret:
1,2,3,4,5,6,7,8,9,10 (example for a range of pre-images)
```

target state=229: number of pre-images=19, parity=even, max state=30, continue-ret:
30,29,27,24,23,20,18,17,15,12,10,9,6,5,3,0, (example for even parity, part system size=5)

If there are more states than will fit in the window, the following prompt allows more to be displayed, or to abandon the display and continue,

more-m continue-ret:

27.8. Learn, forget, or highlight only.

Once the target state and aspiring pre-image(s) have been set in #27.5-6 and reviewed in #27.7 above, a prompt is presented to learn or forget the aspiring pre-images, i.e. add/remove them from the existing pre-image fan of the target state. When the attractor basin is redrawn according to the altered network, the target and aspiring pre-image states will be highlighted (see #27.1 and 27.**). An alternative to learning or forgetting is to just highlight these states, without altering the network.

The following prompt is presented,

highlight only-h, FORGET-f, LEARN-default:

Note that for a 1d regular CA "highlight" is the only option available. See #11.3 to treat local wiring as if it were random, and #13.3.1 to set up a rule mix where all the rules are the same.

27.8.1. Highlighting only.

If "h" is entered at #27.7 above the program continues directly to the highlighting options (see #27.10).

27.9. Learning/forgetting by wire moves or bit-flips.

If "learn" or "forget" is selected at #27.8 above, a prompt is presented to proceed by moving wires or flipping bits in rule tables,

learn by wire-moves -w, bit-flips -default: (or forget.....)

27.9.1. Learning/forgetting by bit-flips.

The bit-flip learning algorithm (Wuensche 1992a) changes particular bits in rule tables to make the target state the actual successor of each aspiring pre-image, instead of its original successor under the original network parameters.

The procedure must succeed. Pre-existing (or just learnt) pre-images can not be "forgotten", i.e. detached as pre-images of the target state, by learning more states as pre-images. However other states not on the list of aspiring pre-images may also be learnt, and side effects will occur elsewhere in the attractor basin.

To "forget" a pre-image just one of a set of bit-flips is required. This is chosen from the set at random. As fewer network changes are needed there will be fewer side effects.

If "bit-flips" is selected at #27.9, the required changes to the network will rapidly be made and the following prompt

more-m

Enter "m" to repeat the procedure from #27.5.

27.9.2. Learning/forgetting by wire-moves.

The wire-move learning algorithm (Wuensche 1992a) moves a single wire at each cell position where there is a mismatch between the actual successor cell and the relevant cell in the target state. There is a choice of wires to select from the pseudo-neighbourhood, and a choice of new coupling positions, some of which will correct the mismatch. The aspiring pre-image order, the pseudo-neighbourhood order, and the coupling order, are all shuffled at random, and the first successful move is selected (if it exists).

Pre-existing (or just learnt) pre-images may be "forgotten", i.e. detached as pre-images of the target state. If "just learnt" states are forgotten when learning the next pre-image, the next wire/coupling alternative in the shuffled order will be tried. If none succeed the algorithm continues with the next aspiring pre-image. On completing the learning pass, the states learnt and not learnt are listed by their decimal equivalents (for $n \leq 32$ only), with the following prompt,

206,947, ←no (examples are Hamming distance 1 from 462)

1468,460,458,454,478,494,398,334, ←ok

re-learn by wire-moves -w, bit-flips -b: (if all ok, examples are Hamming distance 1 from 462)

or if all are "ok",

success - aspiring pre-images learnt

continue-ret:

"ok" indicates learnt states, "no" not learnt. An option is offered to re-learn again by wire moves, or by bit-flips (which always succeeds). If "w" is selected the last "no" state will be learnt first, followed by the set of shuffled "ok" states, and the remaining shuffled "no" states. On completing the pass the resultant listing and prompts are repeated.

Note that for a large system with a large list of aspiring pre-images, a wire-move learning pass may take an unpredictably long time.

To "forget" a pre-image just one appropriate wire move is required. Forgetting is generally much easier and quicker than learning. As fewer network changes are needed there will be fewer side effects. Note that in the forgetting procedure "ok" indicates states still attached to the target state, i.e. not forgotten, and "no" indicates states successfully forgotten, for example if all are "no" the following prompt is presented below the listed states,

```
success - aspiring pre-images forgotten
continue-ret:
```

If "return" is entered at the prompts above, the following prompt is presented,

```
more-m
```

Enter "m" to repeat the procedure from #27.5, otherwise the network as it stands is accepted.

27.10. Highlighting options.

(applies to attractor basins, not when running space-time patterns forwards only)

Once learning is complete, or the "highlighting only" option was selected at #27.8.1, the following option is presented to specify highlighting the target state and the set of aspiring pre-images as nodes in the attractor basin. (see #27.1, and note that the particular form of highlighting depends on the node type selected in #20.2).

```
highlighting: none-n, suppress all other nodes-s:
```

Enter "return" to highlight selected nodes and show other nodes normally according to the node type selected in #20.2. Enter "s" to suppress the display of nodes other than highlighted nodes. Enter "n" not to highlight but show all nodes as normal.

27.11. Learning complete.

Once the option at #27.10 above is selected (remember that "q" allows backtracking), the wiring graphic and associated prompts (#27.4) will reappear. Enter return to exit the wiring graphic. Top right prompts will appear to load/save/print the network parameters as described in #14.7.2.

Enter return to revert to the top left prompt #25.3 (for attractor basins) or #25.5.5 (for forwards only space-time patterns). Enter return to redraw the attractor basin (or restart space-time patterns forwards only) with the new (learnt) parameters. Attractor basins will be drawn with the specified highlighting.

From this window (#25.3 or #25.5.5) the learning procedure can be restarted, see #27.3.

28. Filing

28.1. File types, default extensions and filenames

DDLlab allows saving and loading of a number of file types with default extensions. Default filenames are also provided. The file types, and default extensions and filenames are listed below.

<u>Type</u>	<u>extension</u>	<u>Default filename(without extension)</u>
SCREEN (image)	.nat	myimage
N'HOOD MIX,	.mix	my_mix
WIRING ONLY	.w_s	my_wso
RULE SCHEME ONLY	.r_s	my_rso
WIRING-RULE SCHEME	.wrs	my_wrs
SINGLE RULE	.rul	myrule.rul
		last1, last2, last3, ... last9 (i.e. last[k], the last rule is automatically saved)
SEED	.eed	myseed
DATA	.dat	my_data (Field/Basin/Subtree ascii file)
EXHAUSTIVE (pairs)	.exh	my_exhau

28.2. Saving and Loading

In general when saving or loading a file the following prompt appear in a top right window:

```
SAVE [or LOAD] [type] - filename (no ext) .[extension] will be added
(change drive or dir-d) [current path]
list-?, quit-q, default [ filename]
```

For example, if you are running DDLab from the directory "c:\ddlab" and loading a screen image the prompt will appear as follows,

