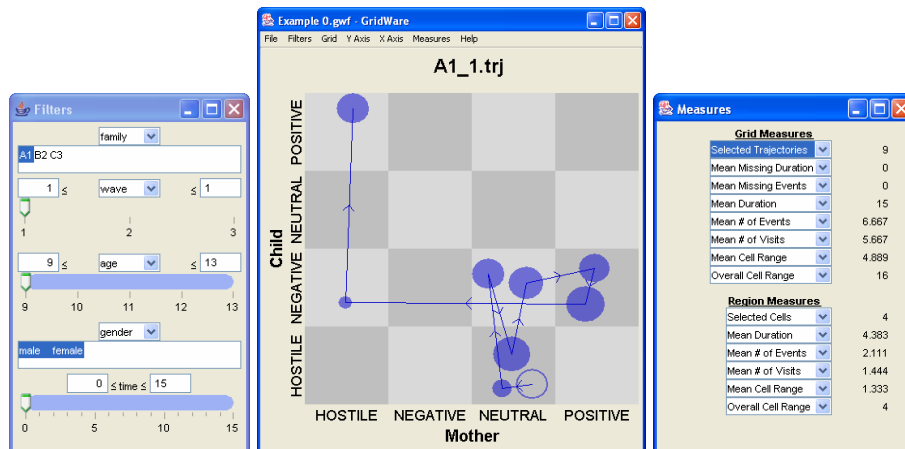# The GridWare Manual

Version 1.1

Overview

GridWare is a versatile visualization and data manipulation tool for multivariate time series of sequential (ordinal or categorical) data. This application is based on the work of Marc Lewis and colleagues who devised state space grids based on dynamic systems principles. Briefly, there are 2 basic dynamic systems concepts upon which GridWare is based. The first is that of a state space: the space of all *possible* states of a system. This corresponds to the 2 dimensional grid in GridWare. Each possible state is represented as a cell on the grid. A trajectory or sequence of states can be plotted on a state space grid to depict the states of the "system" as they change in time. The second dynamic systems concept is that of an attractor: an absorbing state which "pulls" the system from other potential states. GridWare provides the visualization and measures to identify and quantify the relative strength of attractors. While these concepts are not necessary to use the software, it is recommended that the user get familiar with these concepts before setting up data for GridWare. Appendix C provides a more detailed explanation of these concepts and some examples of state space grids used in empirical studies.

There are several new features in Version 1.1.
- Improved sliders for more versatile filtering
- New measures: Dispersion, First Entry, Last Exit and # of Missing events
- More user-friendly Variable Creation Dialogues
- New ways to add/remove filters and measures
- New save features: node colours and node positions
- Onset times can now have one of several formats
- Option to display arrows to show the direction of trajectories
- More efficient processing for faster response and better handling of large data sets

Table of Contents

i. System Requirements

The user must have Java 2 Platform, JRE Version 1.4 (or later) installed on their computer. However, it is strongly recommended that the user have Java 1.4.2 to get the best possible performance from GridWare. The Windows version can be downloaded for free from the Java web site (http://www.java.com) and the Mac version can be downloaded from www.apple.com/java. GridWare can be run on Windows XP and Mac OS X (although only Mac OS X 10.3.3 can run Java 1.4.2). The Windows version may run on other versions of Windows but the Mac version will not work with Mac OS 9 or earlier. GridWare will take about 32 MB + 1MB of RAM per 2,000 events (rows in Trajectory Files) in your dataset. So if you have 200 trajectories of about 1000 events each, you should have at least 132MB of free memory before running GridWare. Display settings should be at least 800x600, but higher resolution is preferable.

ii. Installation Instructions

Once you have downloaded the compressed file, decompress it (double-click to open) and save GridWare in a folder of your choosing. GridWare will run on data files that are located in another folder anywhere else on the computer's hard drive.

iii. Contact and Reference Information

GridWare was programmed by Alex Lamey and developed in collaboration with Marc Lewis, Isabela Granic, and Tom Hollenstein. The manual was written by Tom Hollenstein. All authors are available for consultation and technical assistance in the application of this software (see section on Technical Support for all support related issues). Any scholarly reference to this program should be to:

Lamey, A., Hollenstein, T., Lewis, M.D., & Granic, I. (2004). GridWare (Version 1.1). [Computer software]. http://www.statespacegrids.org.

When publishing or reporting results from GridWare, we request that you please cite Lewis, Lamey, & Douglas (1999) and any of the following publications if relevant:

Lewis, M. D., Lamey, A. V., & Douglas, L. (1999). A new dynamic systems method for the analysis of early socioemotional development. *Developmental Science, 2*, 457-475.

Granic, I., Hollenstein, T., Dishion, T. J., & Patterson, G. R. (2003). Longitudinal analysis of flexibility and reorganization in early adolescence: A dynamic systems study of family interactions. *Developmental Psychology, 39,* 606-617.

Granic, I. & Hollenstein, T. (2003). Dynamic systems methods for models of developmental psychopathology. *Development and Psychopathology*, *15*, 641-669.

Granic, I. & Lamey, A.V. (2002). Combining dynamic-systems and multivariate analyses to compare the mother-child interactions of externalizing subtypes. *Journal of Abnormal Child Psychology, 30*, 265-283.

Hollenstein, T., Granic, I., Stoolmiller, M., & Snyder, J. (2004). Rigidity in parent-child interactions and the development of externalizing and internalizing behavior in early childhood. *Journal of Abnormal Child Psychology, 32(6)*, 595-607

Lewis, M.D., Zimmerman, S., Hollenstein, T., & Lamey, A.V. (2004). Reorganization in coping behavior at 1 ½ years: dynamic systems and normative change. *Developmental Science,7*, 56-73.

1. Data Format

All data for a project should be saved in the same folder (the data must be within a subfolder – see below), although this does not have to be the same folder as the GridWare program. There are 2 file types necessary for input to GridWare. The first is the raw trajectory data, one file for each trajectory. The second is a single text file that is the list of the Trajectory files. This is the GridWare file (*.gwf) that specifies the variables and values for each trajectory. Before using GridWare, the user must prepare these files according to the following specifications.

IMPORTANT! The folder that contains the data (Trajectory Files) must be in the same directory as the GridWare File or Trajectories List File. This folder must be named in this way: the name of the Trajectories List File or GridWare File followed by "_trjs". For example, if the name of your GridWare File is "Cohort 1.gwf", then the data to which "Cohort 1.gwf" refers must be in a folder named "Cohort 1_trjs". Each time you save in GridWare, the Trajectory Files in the folder get overwritten to include information about the changes made and the color and position of all the nodes. Therefore, it is recommended that the user keeps a copy of the original Trajectory Files somewhere else as a backup before starting a new GridWare project.

1.1 Trajectory Files:

The structure of each Trajectory file is as follows:
- The first row contains the labels for each column
- There must be only one column for onset times, and at least 2 other columns of time series data (state variables)
- Each row is a new event. For event-based data where each new row is not an onset time but a new event, ONSET can be simply an ordered sequence of integers.
- the final row must have only a time value, but no values or spaces for any other variable. That is, the last digit of the last onset time (end time) must be the end of the file. This value represents the OFFSET of the final event
- Onset time for each event must be the first column. These times can be in one of several formats: decimal (...0000.000...), real time (hh:mm:ss), or a combination

(mm:ss.00). All onsets will be converted to seconds, however, both within the onscreen display and in any file output, including when re-saving trajectory files.

- Separation between columns **must be** tab delimited. Blank spaces are considered part of a single text label or value.
- Missing events (i.e. points where subject is inaudible or out of view) must be represented with a symbol that is consistent with the user-defined symbol set in the Preferences menu option. It is recommended that the user select a missing symbol (i.e."*" or "." or "NA") which is compatible with the statistical software that will be used with the exported measures. The default symbol is "–". See Tips and Troubleshooting for details about how to change the missing value symbol.

For example, here is a short trajectory file ("A1_1.trj") with 3 state variables:


```
Onset    variable 1      variable 2      variable 3
0.00     1      3        2
1.65     1      3        3
2.30     2      3        3
4.10     1      3        3
6.48     2      3        3
8.60     2      4        3
10.25    2      4        1
12.82    2      1        1
13.17    4      1        1
15.00
```
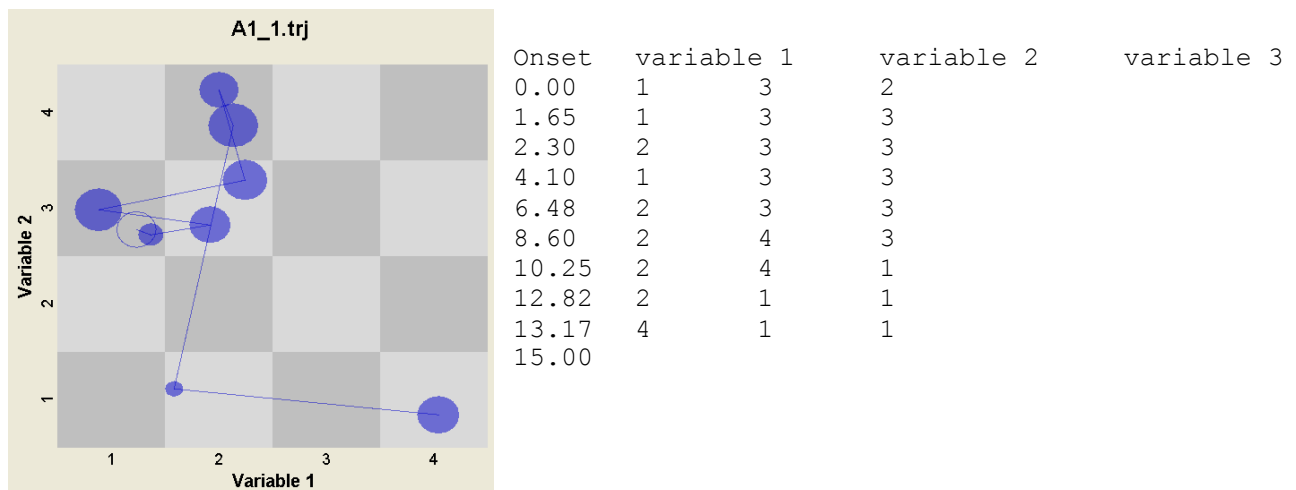
In this file, the trajectory starts in variable 1 = 1, variable 2 = 3, and variable 3 = 2. The next event is 1.65 seconds (or whatever time units these values represent) later when variable 3 switches to 3, but the other two variables remain the same as before, so the display does not change. At 2.3 seconds, variable 1 now shifts from category 1 to category 2. The trajectory ends at 15 seconds.

1.2 GridWare File (*.gwf) and the Trajectories List File

A GridWare file begins as a Trajectories List File that the user wants to put into GridWare. Before the first use, the Trajectories List File need only contain a minimum of one column of ID's and one column of Trajectory file names. This file is also tab delimited. This is the file that allows the user to specify other variables

that are associated with each trajectory. Hence, these variables are called <u>trajectory variables</u>. These variables may specify gender, the wave of the study, classifications of those on whom the trajectory is based (i.e. depressed/non-depressed, familiar peer/non-familiar peer), experimental condition, or any other grouping which distinguishes one trajectory from another.

For example, a longitudinal observational study of children's behaviour with both father and mother might have a Trajectories List File that looks something like this:

```
Family  Wave    Age     Gender  Filename
A1      1       11      male    A1_1.trj
A1      2       12      male    A1_2.trj
A1      3       13      male    A1_3.trj
B2      1       9       female  B2_1.trj
B2      2       10      female  B2_2.trj
B2      3       11      female  B2_3.trj
C3      1       10      male    C3_1.trj
C3      2       11      male    C3_2.trj
C3      3       12      male    C3_3.trj
```

In this file, Family and wave correspond to the file name. Wave designates the wave of observation for each trajectory, Age indicates the age of the child, and Gender indicates the child's gender, and the file names are the Trajectory files that contain the multivariate time series data.

Once this file is read in to GridWare and the trajectories are displayed, it is saved upon closing as a GridWare File. The saved GridWare File contains the original information provided by the user plus has all of the other specifications (variable definitions, preferences, axis labels) necessary for each subsequent use. It is possible for the user to edit or even create the full, saved version of a GridWare File from scratch. Some users may find this a better way to prepare their files. However, there are strict limitations to the format of the file (i.e. <u>tab delimiters</u>) and GridWare simply will not run if there is <u>any</u> deviation from the format (see section <u>2.1 Opening a New Project</u>).

For the above example, the saved GridWare File might look like this:

```
<GridWare>
<Config>
trajectory      categorical     family
trajectory      ordinal wave    1       3
trajectory      ordinal age     9       13
trajectory      categorical     gender
state   ordinal variable 1      1       4
state   ordinal variable 2      1       4
state   ordinal variable 3      1       4
MinReturns      2
MaxReturnTime   10
MaxReturnVisits 6
MinEventDuration        0
MinCellDuration 0
```

```
MissingValueSymbol        --
</Config>
<Trajectories>
Family  Wave    Age     Gender  Filename
A1      1       11      male    A1_1.trj
A1      2       12      male    A1_2.trj
A1      3       13      male    A1_3.trj
B2      1       9       female  B2_1.trj
B2      2       10      female  B2_2.trj
B2      3       11      female  B2_3.trj
C3      1       10      male    C3_1.trj
C3      2       11      male    C3_2.trj
C3      3       12      male    C3_3.trj
</Trajectories>
</GridWare>
```

In this example file, the lines in red text are opening and closing tags that signal the different parts of the file. The Config section (<Config> to </Config>) specifies all the variable definitions and preferences. The trajectories section (<Trajectories> to </Trajectories>) is the specification of trajectory variables and the file names of the individual trajectory files. The contents of this section are identical to the contents of the Trajectory List File which the user must create before the first use of GridWare.

The blue colored lines in the Configuration section are the variable definitions for the Trajectory Variables. The first column is always "trajectory", the second column specifies the variable format (categorical or ordinal), and the third column is the Trajectory Variable names. For categorical variables, the fourth column and beyond are the discrete categories (i.e. Family ID's) for that variable. If the user is creating the GridWare File from scratch, it is not necessary to specify these categories – GridWare will include them when the file is saved. For ordinal variables, the fourth column is the minimum value and the fifth column is the maximum value.

The purple colored lines are the definitions for the state variables (in the individual Trajectory Files). The first column is always "state", the second column is the variable format (ordinal or categorical), and the third column is the state variable name. If the state variable is ordinal, a minimum and maximum value must be provided in the fourth and fifth columns, respectively. Unlike Version 1.0, Version 1.1 now saves the labels for variables in the Trajectory Files directly and any edits (i.e. giving text names to integer categories: 1 = hostile, 2 = neutral, etc.) *overwrite* the original Trajectory Files. See section on Saving a Project for more details.
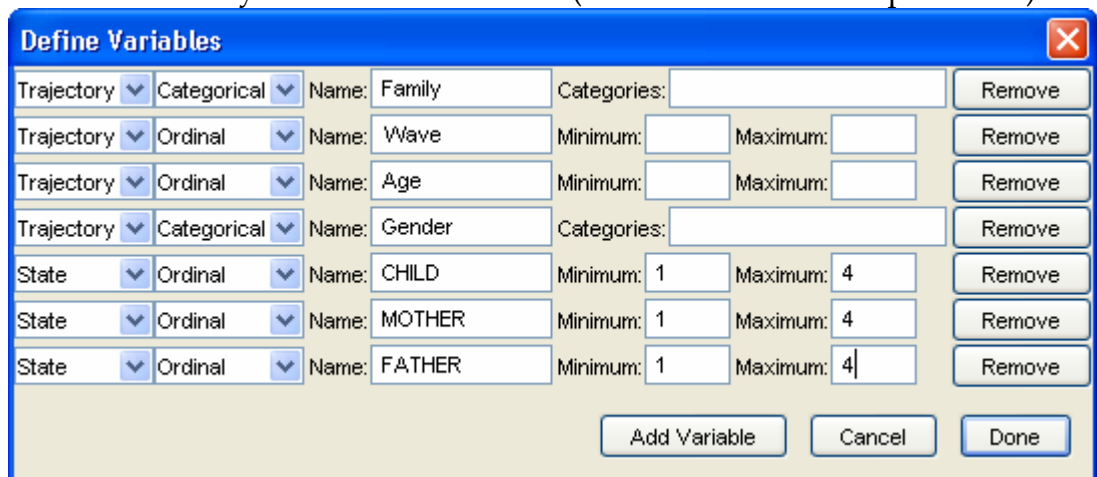
The green colored lines are the user preferences for methods of calculating measures and the missing value symbol. Preferences can be easily edited from within the program. See the section on measures for an explanation of these values.

2. Opening a Project

2.1 Opening a New Project

After setting up the Trajectory files and GridWare file (or Trajectories List File), double click on the GridWare icon to open the program (remember to put the Trajectory Files into a correctly named folder!). This will open a dialogue box for you to choose your GridWare file. If the file being opened is an existing GridWare file, go to the section on Opening an Existing Project. If the GridWare file selected is a Trajectories List File, a dialogue box appears and the user can then begin the creation of grid variables. GridWare automatically reads in the variable names from the first row of the Trajectories List File and the individual Trajectory files and inserts them in the Variable Creation Dialogue Box. There are 2 *variable types* in the first column; the Trajectory Variables are read in from the Trajectories List File and the State Variables are read in from the Trajectory files. The second column lists the variable formats (categorical or ordinal). The user can edit these two columns by using the drop down menu to change from one type or format to another. However, GridWare will allow only one change per variable in each column. If a mistake is made, the user must restart GridWare and reopen their Trajectories List File. It is not necessary for the user to enter the categories or minimum/maximum for Trajectory Variables. However, it is necessary for the user to enter the minimum and maximum for every ordinal state variable (as shown in the example below).

The user may also add or remove variables that will be used in GridWare. It is essential that users know about the Save process and how it overwrites existing files before using GridWare.

When finished defining all of the variables, click on "Done" and the three GridWare Windows will appear on the screen.

NOTE: NO TRAJECTORIES ARE DISPLAYED WHEN GRIDWARE FIRST OPENS. THE USER MUST SELECT DISPLAY OPTIONS ON THE GRID MENU BEFORE ANYTHING IS DISPLAYED.

2.2 Variable Types

Trajectory variables (columns within [GridWare File](#) or Trajectories List File):

These are variables that together uniquely identify, describe, classify, index or are otherwise associated with the trajectory or the source of the trajectory data. At minimum, the ID variable in the [GridWare File](#) must be entered here. Variables such as gender, experimental condition, and diagnosis are just a few examples of Trajectory Variables.

State variables (columns within each [trajectory file](#) except ONSET):

These are the time-series variables which define the events within a [Trajectory file](#). Missing values are possible states that will be displayed as dotted lines when plotted on the grid.

2.3 Variable Formats

There are 2 possible variable formats: categorical or ordinal. How the user identifies the format of each variable has implications for how they may be used.

Categorical: Categorical trajectory variables are alphanumeric and, although they may be sequential, they are treated as discrete, non-adjacent values or strings. GridWare displays these variables as a list from which the user may [select](#) one, some or all of the categories.

Ordinal: Ordinal variables are necessarily sequential and can contain only integer values. For trajectory variables, this allows the user to [select](#) a range of values. Only a single range may be selected; thus, unlike categorical variables, it is not possible to select 2 or more non-adjacent values.

2.4 Saving a Project

When you save a project (File menu → save), most of the information is saved to the GridWare file but some may be saved in each of the Trajectory Files. Saving will create (if the project is new) or update the GridWare file with any changes in Preferences and in the filters displayed at the time of saving. This is the [GridWare File](#) and it will be saved with the extension ".gwf".   It is a text file that can be edited by the user. However, the format must be consistent or GridWare will not work. Therefore, it is not recommend that users edit *.gwf files without great care.

There are several types of changes that will edit and overwrite the Trajectory Files. The positions and colours of the nodes are saved in the Trajectory Files as extra columns of information appended to the existing files. However, any edits of the [axis labels](#) will be saved as replacements for the original values of the state variables. For example, if Variable 1 had values of 1, 2, and 3 and the user [edits the axis labels](#) in GridWare to "Low", "Medium", and "High", saving will replace all of the values in the Trajectory Files. Thus, it is highly recommended that the user save a back-up copy of the original Trajectory Files as a protective measure.

2.5 Opening an Existing Project

To open an existing project, the user selects a previously saved (*.gwf) file from the Open File dialogue box. This file may be a new one that the user created from scratch as long as it fits all of the specifications of a GridWare File.

3. The GridWare Windows

GridWare has four windows. The largest window is the state space grid that displays the trajectories. The Data Filters window allows the user to select trajectories to display based on trajectory variables. The third window displays the measures for the whole grid and individual cells, if selected. The menu bar also has several features that control the display. The fourth window is the Run Log window that displays the messages related to the processes and functions performed in the running of GridWare as well as any error messages to aid in troubleshooting.

3.1 The Grid Window

The Grid window is the main screen for GridWare. It displays trajectories on a two-variable state space. In this window, the user can edit the axis labels and select individual cells or groups of cells. The menu bar offers features that allow the user to change the layout, select which state variables and filters are displayed, colour some or all of the trajectories, and edit the preferences for the measures.

NOTE: NO TRAJECTORIES ARE DISPLAYED WHEN GRIDWARE FIRST OPENS. THE USER MUST SELECT DISPLAY OPTIONS ON THE GRID MENU BEFORE ANYTHING IS DISPLAYED.

Missing events are depicted with a dotted line.
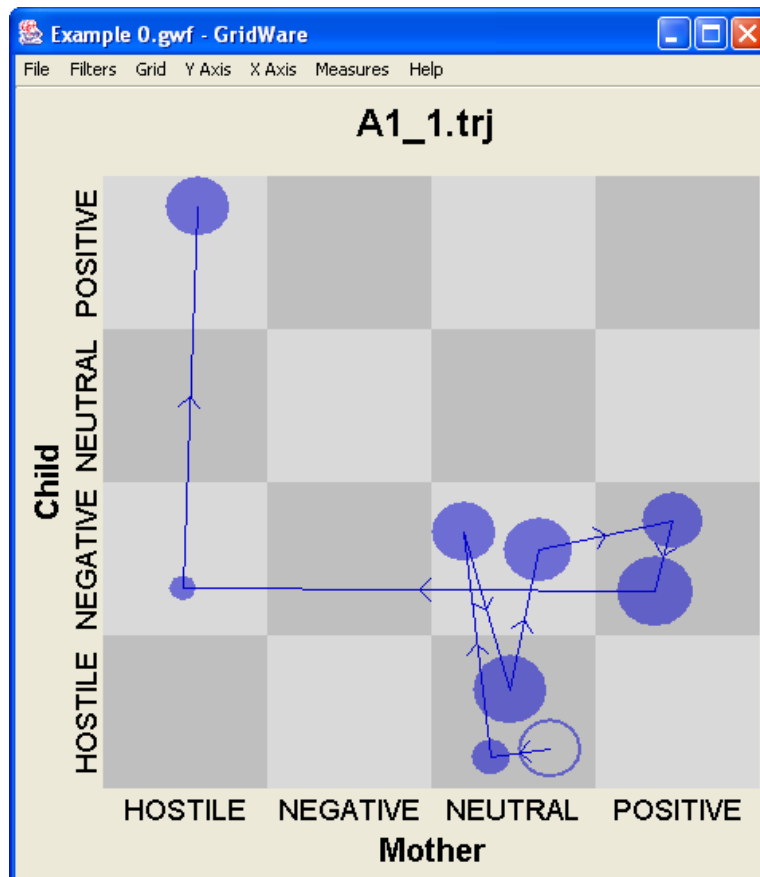
Axis labels: Press Alt (Windows) or option (Mac) and single click the x-axis labels to put into entry mode. Use tab to jump from one label to another. Enter must be pressed when finished to get out of edit mode. Labels can be entered one at a time for each state variable on the x-axis. Note, saving will change all the Trajectory Files (see Saving a Project).

Cell and Region selection: Select a single cell with a single mouse click. To form a region or cluster of cells, select multiple cells by holding down the shift key while selecting. To select a whole row or column, single click on the axis label. Subsetting of the grid in this way provides measures specific to that cell or region and can also be used to set a different colour for the trajectory in the selected region. The cells constituting a region need not be adjacent. To de-select a cell or region, single click again on the same cell or region and the yellow highlight disappears.

Back to Table of contents

Menu Options

File Menu:

Open: This opens a dialogue box to select a new GridWare file. See Opening an Existing Project.

Close: This closes the current project without saving any changes made.

Save: This saves the current project. Beware – saving may overwrite the Trajectory Files and GridWare file. See [Saving a Project](#).

Save As: This allows the user to rename the GridWare File and save changes made. However, this does not prevent the Trajectory Files from being overwritten. See section on [Saving a Project](#) for more details.

Export Images: Exports a png image of the current Grid Window display. See [Exporting Grid Images](#).

Export Measures: Opens a dialogue box for the user to choose which types of measures to export. See section on [Exporting Measures](#) for more details.

Quit: Closes GridWare.

Filters Menu*:*

Filtering is one of the central features of GridWare that gives the user a great deal of flexibility in exploring data (see section on Filters Window). However, it can also get complicated if many filters are used at once. It is highly recommended that the user keep track of which Trajectory Variables are filtering ("restricted") and which are not filtering ("open"). To assist the user, GridWare has two options on the Filters Menu.

Add Filter: Use this option to insert a filter location in the [Filters Window](#). Several filters can be displayed at one time – the maximum number of filters displayed depends on the size of the user's computer screen. To remove a filter from the Filters Window, there is an option in the filter chooser itself that will allow for removal. See the section on the Filters Window for more details about controlling which filters are displayed.

Reset Filters: Use this option to reset all Trajectory Variables to be unfiltered. Using this function can be useful when the data displayed does not meet the user's expectations – it may be that the data was being filtered. If a large amount of data is selected, and the time filter is in use as well as other filters, a filter reset may take some time.

NOTE: It is recommended that the Time Filter be reset manually before using the Reset Filters option. With large data sets resetting the time filter with the rest of the filters can take an inordinately long time.

Grid Menu*:*
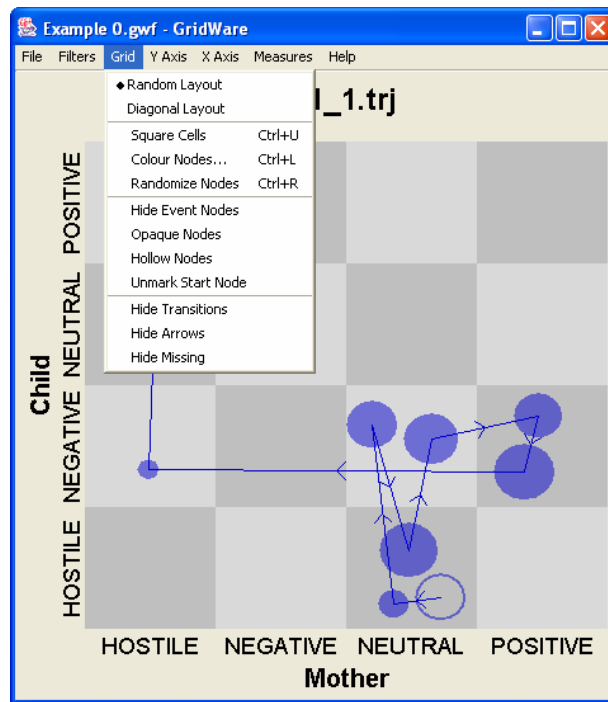
Random or Diagonal Layout: There are two options for the layout of the plot points on the grid: Random and Diagonal. The Random option makes a random arrangement of the plot points within each cell.

With Diagonal layout plotting of the first event begins in the lower left corner of the cell in which the event occurs. Each subsequent event is plotted along the diagonal from the lower left to upper right corner of whichever cell is being occupied. Thus, it is possible to detect when in the sequence a behaviour occurred within any cell: lower left at the beginning, upper right at the end of the sequence. This feature is especially useful for longer time series.

Square Cells: The grid display can become rectangular for various reasons (i.e. expanding to full screen). The square cells option resizes the grid back into a square.

Colour Nodes: Use this to change the colour of some or all of the trajectory or trajectories currently displayed. Selecting this option will bring up a colour pallet from which to choose from among 256 colours. This feature is best used in combination with variable filters or cell selection. For example, to compare two trajectories on the same grid, select one trajectory and colour it a different colour. Then, select both trajectories using the trajectory filters. Now both trajectories are displayed but each is a different colour. If the Trajectory files each contain more than two synchronized time series, there is another useful application of colour. Select a cell or region of interest and colour the trajectory in that cell a different colour. Then, switch one of the axes to a different state variable and it is possible to see the

coloured portion in the trajectory of the first selection in the grid of the second selection.

Often it is useful to colour groups of trajectories based on a filter variable. For example, it is possible to compare trajectories of subjects who scored low, medium and high on another test or measure. Select the range of the "low" values of this variable, change the colour, select the range of "medium" values, change to a second colour, and leave the high values the same as the original colour. Return the filter to the full range of values to display all trajectories in 3 colours.

It is also possible to colour trajectories based on time windows. However, the user must be aware that this will be a little rough around the edges because events will be coloured even if they only partially fall within the selected time segment.

One way to colour that does not require filtering is to colour based on a state variable that is not displayed. For example, if there are three state variables for a family interaction study (child, mother, and father), only two will be displayed at a time. The user could colour particular rows or cells of one subject's states, and then display the trajectory for the other two. In this way, the user could see, for instance, what the mother and father were doing (the grid and trajectory displayed) while the child was having a tantrum (child's tantrum cells are coloured differently, but the child's axis is not displayed).

Because colouring often requires filtering, it is important to keep track of which filters are open and which are restricted (see section on Filtering below). To return all trajectories to the same colour, select all trajectories (reset filters) and reassign a colour.

NOTE: colours can be saved. To do this, GridWare writes this information directly into each trajectory file (see section on Saving a Project). The colours displayed at the time of saving will be the colours displayed when re-opened. If this is not desired, reset the colours as described above.

Randomize Nodes: Randomizes the arrangement of plot points on the grid – only for random, but not for diagonal, layout. If a single cell or region has been selected, only the plot points in those cells are randomized. This feature can be useful to rearrange the plot points to optimize the view of the trajectory. When saving in GridWare, the positions of the nodes are also saved. This information is saved directly in each Trajectory File.

Show/Hide Event Nodes: This is a toggle to control whether the nodes are displayed. The default selection is for the nodes to be hidden. Thus, nodes are hidden when GridWare is first opened. With this option the diameter of the plot points reflects the proportion of time for that event – a

large dot depicts a long duration event and a small dot depicts a brief event. The dot area is derived from the following ratio:

$$\frac{\text{Dot area}}{\text{Cell area}} = \frac{\text{event duration}}{\text{longest trajectory duration}}$$

Opaque/Translucent Nodes: This is a toggle to control whether the nodes are opaque or translucent. When nodes overlap, the translucent option allows the user to see through the node colour so that all of the overlapping nodes can be seen. Opaque nodes are solid and do not reveal the entirety of every overlapping node. When dealing with large data sets that put a strain on computer processing and memory, opaque nodes should be used because they require less computer resources.

Hollow/Filled Nodes: This is a toggle to control whether the nodes displayed are empty or solid circles. However, the start node, if selected (see below), will be displayed in the opposite format. That is, when the Hollow Nodes option is selected, the start node will be filled; when the Filled Nodes option is selected, the start node will be hollow. When dealing with large data sets that put a strain on computer processing and memory, hollow nodes should be used because they require less computer resources.

Mark/Unmark Start Nodes: This is a toggle to control whether the start nodes are contrasted with the other nodes by being either filled or hollow. If the start nodes are unmarked, all nodes will be of the same type (filled or hollow). If the start nodes are marked, when the Hollow Nodes option is selected, the start node will be filled; when the Filled Nodes option is selected, the start node will be hollow.

If there are multiple trajectories displayed, there is a start node for each starting point for each trajectory. If the user selects a subset of the total time (i.e. the last 30 seconds), the start node depicts the *relative* start point; that is, the first event in that time window.

Show/Hide Transitions: This is a toggle to control whether the transitions between nodes are displayed. These are the lines that connect the nodes. If the Show Event Nodes option is selected along with the Hide Transitions option, only the nodes will be displayed. When dealing with large data sets that put a strain on computer processing and memory, Hide Transitions should be used because this option requires less computer resources.

Show/Hide Arrows: This is a toggle to control whether arrows are displayed on top of the transition lines. This feature places arrows at the midpoint of each line connecting two nodes to indicate the direction of the trajectory in time.

Show/Hide Missing*:* This is a toggle to control whether missing values are displayed. If the Show Missing option is selected, a dotted line connects the event node prior to the missing event(s) to the event node that follows the missing event. If the Hide Missing option is selected the pre-missing event node is connected to the post-missing event node. If the missing event is the first or last event of a trajectory, it will not be displayed at all (either as a node or a transition).

<u>**Axis Selection Menus (Y Axis and X Axis menus)**</u>:
The time series variables that form the trajectories can be selected for each axis. If the same variable is selected for both axes, the trajectory necessarily falls along the diagonal of the grid. This feature may be used for obtaining measures based on that one variable. Trajectory files may have more than 2 synchronized time series (i.e. more than two state variables), yet only 2 can be displayed at any time. However, <u>colour</u> can be used to depict a third dimension. Also note that events that are due to a change in a state variable not displayed are represented by a new plot point in the same cell. See the description of events in the <u>measures</u> section for more detail.

<u>**Measures Menu**</u>
The measures menu is for selecting which measures will be displayed in the <u>Measures Window</u>. Measures can be removed from the Measures Window by the remove option in the scroll down menu located beside each measure in the window.

Add Grid Measure*:* Opens a selection window to choose which whole-grid measure will be added to the Grid Measures section of the Measures Window.

Add Region Measure*:* Opens a selection window to choose which region measure will be added to the Region Measures section of the Measures Window.

Preferences (under the GridWare menu in Mac version): This gives the user the option to change minimum and maximum values for several calculations of measures. These include the Minimum Number of Returns, Maximum Return Time, Maximum Return Visits, Minimum Event Duration, Minimum Cell Duration, and the missing value symbol. See the section on <u>Measures</u> for more detail about these preference values.

3.2 Data Filters Window
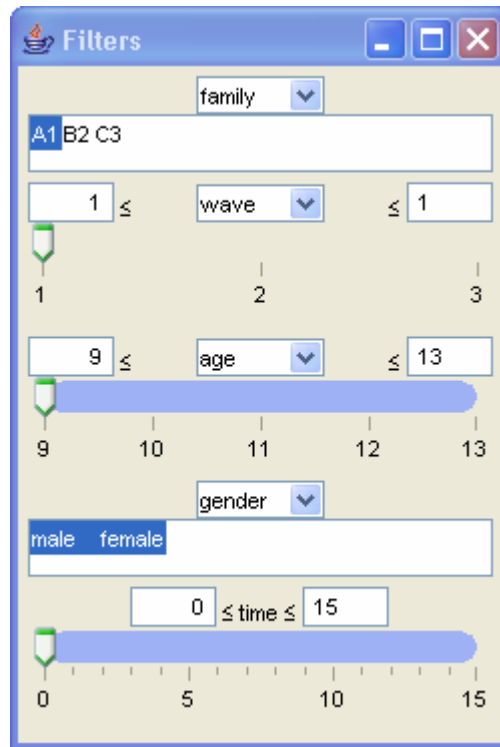
Trajectory filters*:* Each filter is made from the variables in the GridWare file. The primary variables are the ID variables that provide the identity of the data within the Trajectory file (typically a subject ID and wave for developmental studies). A drop down variable selection menu allows for the selection of each trajectory variable of interest as well as the option to remove that filter location altogether (see section on Filter Menu for an explanation of how to insert a new filter into the filter window). Several filters can be displayed at one time – the maximum number of filters displayed depends on the size of the user's computer screen.   However, if the number of trajectory variables exceeds the filter display capacity, each filter can be switched to control any trajectory variable you choose, via the drop down variable selection menu. Each filter can be used to select none, one, some or all of the trajectories in the data set. Filters have different options depending on the variable formats (categorical or ordinal). When GridWare first opens, all trajectories are displayed with no filtering in effect. The Reset Filters option in the Filter Menu allows the user to return to this original, unfiltered state of the data.

There are 2 basic types of filters: list filters and range filters, of which a special case is the time filter.

List filters: Categorical variables show up as a box with all the values present (up to 3 rows – beyond 3 rows a scroll bar is added on the right). They are arranged in the order in which they were

entered into the GridWare File and the user can select one, some or all in the same manner as other software programs. To select more than one string variable value, press the control key (command key on a Mac) while clicking the value with the mouse pointer. To select a sequence, press the shift key when selecting and all values between the first and second click will be selected.
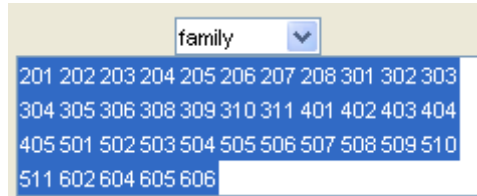

Range filters: Ordinal values can be used to select sets of data via the min and max value windows on either side of the drop down variable selection menu. The default is the minimum and maximum of that variable. The user can type in different values in these windows, press enter, and thus restrict the trajectories displayed on the grid. Underneath these boxes is a blue slider.
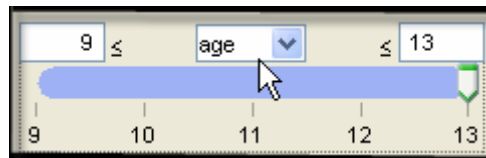

Sliders: Range and Time Filters are manipulated by means of sliders that can be used in several ways. By default, GridWare opens with all values of every Trajectory Variable and Time. Thus, the entire slider is blue when all values are selected.

- To select a single value within the range, hold down the ALT (Windows) or Option (Mac) key and click and drag the slider head from the maximum value to the minimum value. The mouse can then be used to select particular values on the slider or to move from one value to the next.
- To select a range of values, hold down the ALT (Windows) or Option (Mac) key while selecting and drag the slider head until the desired range is reached.
- To move a range window (i.e. shorter than the full range), click on the slider head and drag it with the mouse. Alternatively, click on the open slider area (not blue) next to the blue range area to move the range. This is especially useful for Time filters.
- To return to the full range, the user can either use the minimum and maximum windows to type in the values directly, or move the slider to the minimum value and then drag it to the maximum while using the Alt/Option key.

- Generally, clicking on the blue slider field moves both ends of the slider one unit towards the cursor, pressing alt while clicking in the slider field moves one end one unit closer to the cursor, and pressing shift while clicking in the slider field moves the whole slider by a step (i.e. maximum becomes minimum). The alt and shift options also work with using the arrow keys.
- Pressing the tab key will move the selected area of the filter window in a forward or downward direction. Pressing shift and tab will move the selected area of the filter window in a backward or upward direction.
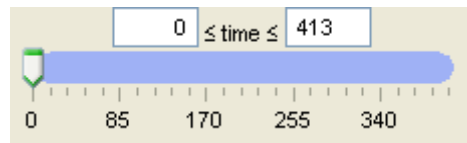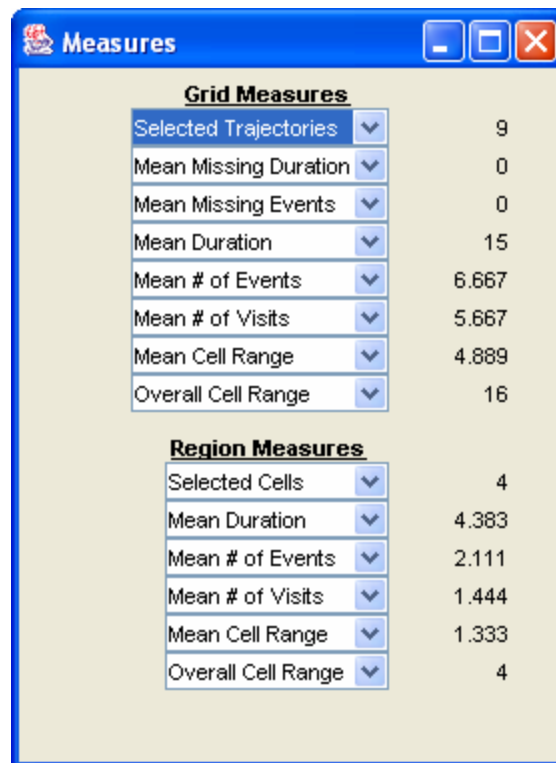

Time filter: There is always a time filter with default minimum of 0 and maximum of the length of the longest trajectory. This filter has an extra feature that allows the user to view trajectories unfolding in time. As described above, the user can set a time window by entering a new value in the maximum value window or by using the slider to select a restricted range of values. Now the trajectory or trajectories displayed are only during the time period from 0 (the first event) to the new maximum value. Note that the circle outlined in black node now represents the start of the time window regardless of when that start point occurred.
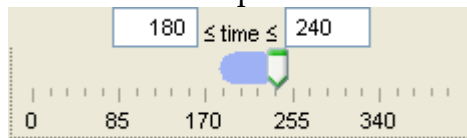
The slider can be used as a moving window control. Moving the slider to the right with the mouse increases both the maximum and minimum values by the same amount, which retains the width of the time window. The trajectory or trajectories displayed will "move" through time – plot points will expand and contract, and the points and lines will appear and disappear as the time slider is moved. For example, if the maximum of 400 seconds is changed to 60 seconds, the slider can be used to view a 60 second window as it moves through time from the window of 0 to 60 seconds up to the window of 340 to 400 seconds.

Back to

3.3 Measures Window

Measures are divided into two basic types: Whole-grid Measures which are derived from the information across all the cells, and <u>Cell or Region Measures</u> which only apply to individual cells or regions selected by the user. Thus, the values of <u>Cell or Region Measures</u> are blank or missing if no cells are selected. The user can select which measures to display in the Measures Window. In the <u>Measures Menu</u> (<u>Grid Window</u>), the user can select which type of measure to add: Whole-grid or Cell/Region. As with the <u>Filters Menu</u> option, this action will insert a measure location in the Measures Window. Each measure location has a drop down menu for the user to select which measure to display. Each drop down menu also has a Remove option for removing that measure location from the Measures Window.

Whole-grid Measures

Selected Trajectories: The number of individual trajectories currently displayed.

Mean Duration[1]: The total duration of a single trajectory or the average duration of all the trajectories currently displayed.

Mean # of Events[1]: The total number of discrete events in a single trajectory or the average number of events of all the trajectories currently displayed. An event is a distinct episode occupying a particular cell, with an onset and an offset. It corresponds to a single line in a trajectory file, and to a single node in the onscreen display. Thus, this count includes events that

repeat in the same cell. These repeated events can be due to either truly repeated events or a change in a non-axis (not displayed) state variable.

Mean # of Visits[1]: The sum of the number of discrete visits to each of the cells for a single trajectory. For multiple trajectories, this value is the average across all the trajectories displayed. A cell visit is one or more consecutive events occurring within a single cell of the currently visible grid, beginning upon a trajectory's entry into the cell, and ending upon its exit. Consecutive events in the same cell are considered one visit. This value can also be used to derive the number of transitions between cells: transitions = visits – 1.

Mean Cell Range[1,2]: For a single trajectory this is the total number of cells visited. For multiple trajectories, this is the average number of cells visited across all the trajectories.

Overall Cell Range[1,2]: The total # of cells visited by at least one trajectory across all trajectories selected. This value is equal to Mean Cell Range for a single trajectory.

Duration per Event[1]: The mean, across trajectories, of the duration of each trajectory displayed divided by its number of events.

Duration per Visit[1]: The mean, across trajectories, of the duration of each trajectory displayed divided by its number of visits.

Duration per Cell[1,2]: The mean, across trajectories, of the duration of each trajectory displayed divided by its cell range.

Dispersion: The sum of the squared proportional durations across all cells corrected for the number of cells and inverted so that values range from 0 (no dispersion at all – all behavior in one cell) to 1 (maximum dispersion). This measure is created by the formula:

$$[(n\Sigma (d_i/D)^1) – 1] / n\text{ -}1$$

Where D is the total duration, d_i is the duration in cell i and n is the total number of cells.

Mean Missing Events: The total number of missing events within the trajectories displayed.

Mean Missing Duration: The total duration of all the missing events within the trajectories displayed.

Cell or Region Measures

Selected cells*:* The number of cells selected.

Mean Duration[1]*:* Total duration in the selected cell or region or, for multiple trajectories, the average of the total durations across trajectories.

Mean # of Events[1]*:* The total number of discrete events in the selected cell or region for a single trajectory or the average number of events in the

selected cell or region across all the trajectories currently displayed. An event is a distinct episode occupying a particular cell, with an onset and an offset. It corresponds to a single line in a trajectory file, and to a single node in the onscreen display.

Mean # of Visits[1]**:** The total number of discrete events in the selected cell or region for a single trajectory or the average number of events in the selected cell or region across all the trajectories currently displayed. A region visit is one or more consecutive events occurring within the selected region, beginning upon a trajectory's entry into the region, and ending upon its exit.

Mean Cell range[1][2]**:** The average number of cells visited. For single cells, this value will be 0 or 1 depending on whether that cell was visited. If more than one cell has been selected, this is the average range in that region.

Overall Cell Range[1][2]**:** The total # of cells within the region visited by at least one trajectory across all trajectories selected. This value is equal to Mean Cell Range for a single trajectory.

Duration per Event[1]**:** The mean, across trajectories, of the duration within the region of each trajectory in the group divided by its number of events within the region.

Duration per Visit[1]**:** The mean, across trajectories, of the duration within the region of each trajectory in the group divided by its number of visits within the region.

Duration per Cell[1][2]**:** The mean, across trajectories, of the duration within the region of each trajectory in the group divided by its cell range within the region.

Mean Return Time: "Return time" is the latency to return to the selected cell or region and is a measure of attractor strength. Return time is the average duration of the intervals between visits to the selected cell or region. Low return times indicate a strong attractor; high return times indicate a weak or non-existent attractor. There are 2 settings for return time in the Preferences menu option. The first allows the user to set the criteria of the minimum number of returns to the selected cell or region. The lowest possible value is 1 since a single visit necessarily cannot return. The default minimum is 2 returns. The second value is the maximum return time. This is the maximum latency of a return and all returns that exceed this maximum are truncated to equal the maximum value. For example, with the default maximum of 10, all returns greater than 10 are changed to 10 and these are the values used to calculate the average. Thus, the average latency would be close to 10 for a weak attractor and closer to zero for a strong attractor. See Appendix C for more detail on the concept of return time.

Mean time to return for a trajectory is calculated as follows: A region return is defined as any sequence of events outside the region beginning

with an exit from the region in question and ending with a re-entry. The number of region returns is equal to the number of region visits minus one. Returns with a duration greater than Maximum Return Time are considered to have a duration of Maximum Return Time for the purpose of this measure. The event sequence from the final region exit in each trajectory until the end of that trajectory is considered a 'pseudo-return': it is included in this mean (truncated by Maximum Return Time if applicable) if and only if it has a duration greater than the mean of the trajectory's return times as calculated from proper returns alone. Otherwise, it does not contribute. The Mean Region Return Time for a trajectory group is simply the mean, across the trajectories, of each trajectory's return time. Trajectories containing fewer than Minimum Number of Returns (including the final pseudo-return if applicable), however, do not contribute to the Mean Region Return Time. Events within a region of duration less than Minimum Event Duration are not taken as sufficient to terminate a return to the region, but can contribute to the return times of other regions.

Mean Return Visits: The total number of discrete visits to any of the other cells outside the selected region before returning, averaged across trajectories. There is also a Preference value that can be set for this feature. Mean visits to return for a trajectory is calculated as follows: A region return is defined as any sequence of events outside the region beginning with an exit from the region in question and ending with a re-entry. The number of region returns is equal to the number of region visits minus one. Returns comprising a number of cell visits greater than Maximum Return Visits are considered to have a length of Maximum Return Visits for the purpose of this measure. The event sequence from the final region exit in each trajectory until the end of that trajectory is considered a 'pseudo-return': it is included in this mean (truncated by Maximum Return Visits if applicable) if and only if it has a length greater than the mean of the trajectory's return visits as calculated from proper returns alone. Otherwise, it does not contribute. The Mean Region Return Visits for a trajectory group is simply the mean, across the trajectories, of each trajectory's return visits. Trajectories containing fewer than Minimum Number of Returns (including the final pseudo-return if applicable), however, do not contribute to the Mean Region Return Visits. Events within the region of duration less than Minimum Event Duration are not taken as sufficient to terminate a return to the region, but can contribute to the return visits of other regions.

Dispersion: The sum of the squared proportional durations across all cells in a region (not in a single cell) corrected for the number of cells and inverted so that values range from 0 (no dispersion at all – all behavior in one cell) to 1 (maximum dispersion). This measure is created by the formula:

$$[(n\Sigma (d_i/D)^1) - 1] / n - 1$$

Where D is the total duration, d_i is the duration in cell i and n is the total number of cells.

First Entry: The mean, across trajectories, of the time until the first entry into the cell or region. Trajectories that never enter the cell or region are given the maximum value, which is the total duration of that trajectory. Thus, First entry is never a missing value. First Entry and Last Exit should be used with caution. One good rule of thumb is that if the mean First Entry is a larger value than the Last Exit, the number of visits to that region are low and generalizations made about that region would be based on a small portion of trajectories.

Last Exit: The mean, across trajectories, of the time of the last transition out of the cell or region. Trajectories that never enter the cell or region are given a value of missing. First Entry and Last Exit should be used with caution. One good rule of thumb is that if the mean First Entry is a larger value than the Last Exit, the number of visits to that region is low and generalizations made about that region would be based on a small portion of trajectories.

Back to

3.4 Run Log Window

The Run Log window displays the Java messages related to the processes and functions performed in the running of GridWare. The contents are the same as the gw_run_log.txt that is automatically created in the same directory as the GridWare program each time GridWare is opened. When GridWare opens and runs normally, it is not necessary to consult this window. However, if there is an error or problem loading in the Trajectory Files, then the user should consult the Run Log window as an aid to troubleshooting the problem. The language in the window is not entirely user friendly, but most of the time even a basic user can discern the nature of the problem. The user can close the window and GridWare will run normally. However, it is not recommended as once it is closed the window cannot be reopened. The information will be saved to the gw_run_log.txt file upon closing GridWare.
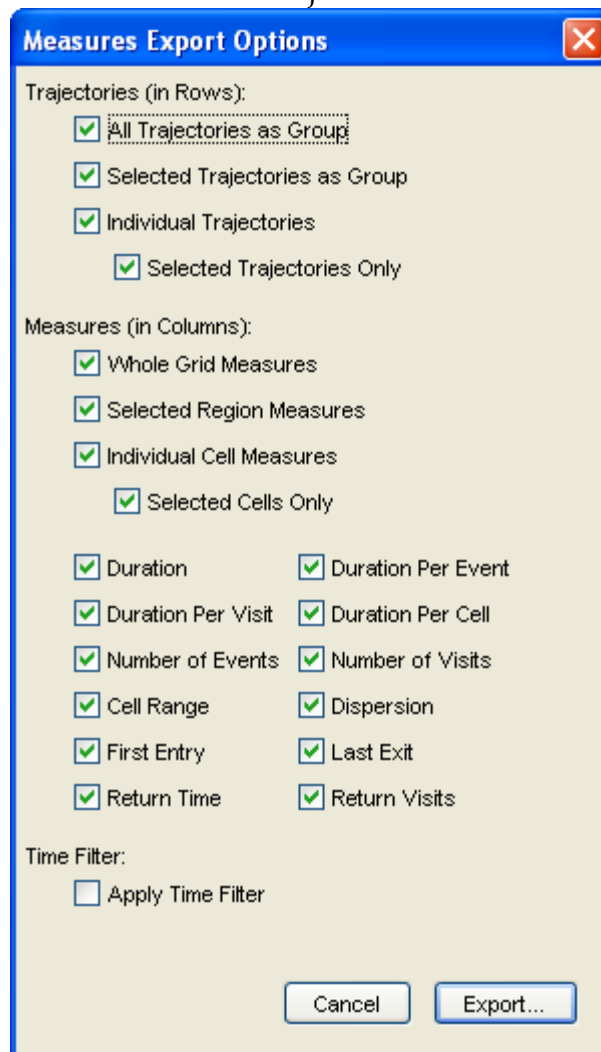
4. Exporting

4.1 Exporting Grid Images

The current screen image can be exported as a *.png file (File → Export Image). This file can then be edited with image software or inserted into other documents and then printed. Future versions of GridWare will give the user more options when saving a grid image.

4.2 Exporting Measures

      The measures displayed for the currently selected trajectories and cells can be exported as a tab-delimited text file (File → Export Measures). If the user doesn't specify a .txt extension, it will be appended automatically to the file name. The Measures Export Options dialogue box allows the user to select which measures and what groupings to export. The box is separated into options for which trajectories to export. These will be exported as the rows or cases. The bottom selections are the measures to be exported. By default all options are selected except for Apply Time Filter. All selections are optional. All output contains the trajectory variables and the number of trajectories within each case.

Trajectories (in rows): The cases that will be exported

All Trajectories as a Group: This will provide one row of output that is the average of all the trajectories in the data set.

Selected Trajectories as a Group: This will provide one row of output that is the average of all the selected trajectories.

Individual Trajectories: This will provide one row of output for each individual trajectory.

> *Selected Trajectories Only:* This limits the individual trajectories output to just the currently selected trajectories.

Measures (in columns): The variables that will be exported

These are the [Whole Grid Measures](#), [Selected Region Measures](#), and [Individual Cell Measures](#) (with option to choose only the selected cells) options available in the Measures Export Dialogue box. Note that when selecting which measures to export that some measures only pertain to one type of measure (i.e. Whole-grid or Individual Cell). Variable names are in parentheses.

Whole Grid Measures:

Number of Grid Cells (gridCells): Just the number of cells in the 2-D grid. This does not depend on the trajectories, and thus will be the same for every case.

Mean Grid Duration[1] (gridDuration): The mean, across trajectories, of the total duration of each trajectory in the group. When this option is selected, the Mean Missing Duration value is also exported.

Mean Number of Grid Events[1] (gridEvents): The mean, across trajectories, of the number of events in each trajectory in the group. An event is a distinct episode occupying a particular cell, with an onset and an offset. It corresponds to a single line in a trajectory file, and to a single node in the onscreen display. When this option is selected, the Mean Missing Events value is also exported.

Mean Number of Grid Visits[1] (gridVisits): The mean, across trajectories, of the number of cell visits in each trajectory in the group. A cell visit is one or more consecutive events occurring within a single cell of the currently visible grid, beginning upon a trajectory's entry into the cell, and ending upon its exit. For individual trajectories, this value can also be used to derive the number of transitions between cells: transitions = visits – 1.

Mean Grid Cell Range [1] [2](gridRangeM): The mean, across trajectories, of the number of unique cells visited by each trajectory in the group.

Overall Grid Cell Range[1] [2] (gridRangeO): The number of unique cells visited by at least one trajectory in the group. This value will equal the Mean Grid Cell Range for an individual trajectory.

Mean Grid Duration Per Event[1] (gridDurPerEvent): The mean, across trajectories, of the duration of each trajectory in the group divided by its number of events.

Mean Grid Duration Per Visit [1] (gridDurPerVisit): The mean, across trajectories, of the duration of each trajectory in the group divided by its number of cell visits.

Mean Grid Duration Per Cell[1] [2] (*gridDurPerCell*): The mean, across trajectories, of the duration of each trajectory in the group divided by its cell range.

Dispersion (Dispersion): The sum of the squared proportional durations across all cells corrected for the number of cells and inverted so that values range from 0 (no dispersion at all – all behavior in one cell) to 1 (maximum dispersion). This measure is created by the formula:

$$[(n\Sigma (d_i/D)^1) - 1] / n - 1$$

Where D is the total duration, d_i is the duration in cell i and n is the total number of cells.

Mean Missing Events (MissingEvents): The total number of missing events within the region for the trajectories displayed.

Mean Missing Duration (MissingDuration): The total duration of all the missing events within the region for the trajectories displayed.

Selected Region Measures:

Number of Region Cells (regCells): The number of cells in the selected region. This does not depend on the trajectories, and thus will be the same for every case.

Mean Region Duration[1] (*regDuration*): The mean, across trajectories, of the duration within the region of each trajectory in the group.

Mean Number of Region Events[1] (*regEvents*): The mean, across trajectories, of the number of events within the region of each trajectory in the group. An event is a distinct episode occupying a particular cell, with an onset and an offset. It corresponds to a single line in a trajectory file, and to a single node in the onscreen display.

Mean Number of Region Visits[1] (*regVisits*): The mean, across trajectories, of the number of visits within the region of each trajectory in the group. A region visit is one or more consecutive events occurring within the selected region, beginning upon a trajectory's entry into the region, and ending upon its exit.

Mean Region Cell Range[1] [2] (*regRangeM*): The mean, across trajectories, of the number of cells within the region visited by each trajectory in the group.

Overall Region Cell Range[1] [2] (*regRangeO):* The number of cells within the region visited by at least one trajectory in the group.

Mean Region Duration Per Event[1] (*regDurPerEvent):* The mean, across trajectories, of the duration within the region of each trajectory in the group divided by its number of events within the region.

Mean Region Duration Per Visit[1] (regDurPerVisit): The mean, across trajectories, of the duration within the region of each trajectory in the group divided by its number of visits within the region.

Mean Region Duration Per Cell[1] [2] (regDurPerCell): The mean, across trajectories, of the duration within the region of each trajectory in the group divided by its cell range within the region.

Mean Region Return Time (regReturnTime): Mean time to return to the region for each trajectory in the group, summed across trajectories, and divided by the number of trajectories in the group. Mean time to return for a trajectory is calculated as follows: A region return is defined as any sequence of events outside the region beginning with an exit from the region in question and ending with a re-entry. The number of region returns is equal to the number of region visits minus one. Returns with a duration greater than Maximum Return Time are considered to have a duration of Maximum Return Time for the purpose of this measure. The event sequence from the final region exit in each trajectory until the end of that trajectory is considered a 'pseudo-return': it is included in this mean (truncated by Maximum Return Time if applicable) if and only if it has a duration greater than the mean of the trajectory's return times as calculated from proper returns alone. Otherwise, it does not contribute. The Mean Region Return Time for a trajectory group is simply the mean, across the trajectories, of each trajectory's return time. Trajectories containing fewer than Minimum Number of Returns (including the final pseudo-return if applicable), however, do not contribute to the Mean Region Return Time. Events within a region of duration less than Minimum Event Duration are not taken as sufficient to terminate a return to the region, but can contribute to the return times of other regions.

Mean Region Return Visits (regReturnVisits): Each trajectory in the group's mean number of visits to return to the region, summed across trajectories, and divided by the number of trajectories in the group. Mean visits to return for a trajectory is calculated as follows: A region return is defined as any sequence of events outside the region beginning with an exit from the region in question and ending with a re-entry. The number of region returns is equal to the number of region visits minus one. Returns comprising a number of cell visits greater than Maximum Return Visits are considered to have a length of Maximum Return Visits for the purpose of this measure. The event sequence from the final region exit in each trajectory until the end of that trajectory is considered a 'pseudo-return': it is included in this mean (truncated by Maximum Return Visits if applicable) if and only if it has a length greater than the mean of the trajectory's return visits as calculated from proper returns alone. Otherwise, it does not contribute. The Mean Region Return Visits for a trajectory group is simply the mean, across the

trajectories, of each trajectory's return visits. Trajectories containing fewer than [Minimum Number of Returns](#) (including the final pseudo-return if applicable), however, do not contribute to the Mean Region Return Visits. Events within the region of duration less than [Minimum Event Duration](#) are not taken as sufficient to terminate a return to the region, but can contribute to the return visits of other regions.

Dispersion (Dispersion): The sum of the squared proportional durations across all cells corrected for the number of cells and inverted so that values range from 0 (no dispersion at all – all behavior in one cell) to 1 (maximum dispersion). This measure is created by the formula:

$$[(n\Sigma (d_i/D)^1) – 1] / n\text{ -}1$$

Where D is the total duration, d_i is the duration in cell i and n is the total number of cells.

First Entry (regFirstEntry): The mean, across trajectories, of the time until the first entry into the cell or region. Trajectories that never enter the cell or region are given the maximum value, which is the total duration of that trajectory. Thus, First entry is never a missing value. First Entry and Last Exit should be used with caution. One good rule of thumb is that if the mean First Entry is a larger value than the Last Exit, the number of visits to that region are low and generalizations made about that region would be based on a small portion of trajectories.

Last Exit (regLastExit) The mean, across trajectories, of the time of the last transition out of the cell or region. Trajectories that never enter the cell or region are given a value of missing. First Entry and Last Exit should be used with caution. One good rule of thumb is that if the mean First Entry is a larger value than the Last Exit, the number of visits to that region are low and generalizations made about that region would be based on a small portion of trajectories.

Back to [Table of contents](#)

Individual Cell Measures (All or Selected Only):

Note that the numbers in the headings for cell measure columns do not necessarily correspond to the ordinal values of the state variables, but only to their rank order. The lowest value will be represented by '1', and greater values by successive numbers. Thus, the lower left corner of the grid is cell x1y1.

Mean Cell Duration[1] (x#y#Duration): The mean, across trajectories, of the total duration within cell x#y# for each trajectory in the group. When this option is selected, the Mean Missing Duration value is also exported.

Mean Number of Cell Events[1] (x#y#Events): The mean, across trajectories, of the number of events within cell x#y# for each trajectory in the group. An event is a distinct episode occupying a particular cell, with an onset and an offset. It corresponds to a single line in a trajectory file, and to a single node in the onscreen display. When this option is selected, the Mean Missing Events value is also exported.

Mean Number of Cell Visits[1] (x#y#Visits): The mean, across trajectories, of the number of cell visits within cell x#y# for each trajectory in the group. A cell visit is one or more consecutive events occurring within the cell, beginning upon a trajectory's entry into the cell, and ending upon its exit.

Mean Cell Duration Per Event[1] (x#y#DurPerEvent): The mean, across trajectories, of the duration in cell x#y# for each trajectory in the group divided by its number of events in cell x#y#.

Mean Cell Duration Per Visit[1] (x#y#DurPerVisit): The mean, across trajectories, of the duration in cell x#y# for each trajectory in the group divided by its number of visits to cell x#y#.

Mean Cell Return Time (x#y#ReturnTime): Each trajectory in the group's mean time to return to cell x#y#, summed across trajectories, and divided by the number of trajectories in the group. Mean time to return for a trajectory is calculated as follows: A cell return is defined as any sequence of events outside the cell beginning with an exit from cell x#y# and ending with a re-entry. The number of cell returns is equal to the number of cell visits minus one. Returns with a duration greater than [Maximum Return Time](#) are considered to have a duration of [Maximum Return Time](#) for the purpose of this measure. The event sequence from the final cell exit in each trajectory until the end of that trajectory is considered a 'pseudo-return': it is included in this mean (truncated by [Maximum Return Time](#) if applicable) if and only if it has a duration greater than the mean of the trajectory's cell return times as calculated from proper returns alone. Otherwise, it does not contribute. The Mean Cell Return Time for a trajectory group is simply the mean, across the trajectories, of each trajectory's cell return time. Trajectories containing fewer than [Minimum Number of Returns](#) (including the final pseudo-return if applicable), however, do not contribute to the Mean Cell Return Time. Events within cell x#y# of duration less than [Minimum Event Duration](#) are not taken as sufficient to terminate a return to the cell, but can contribute to the return times of other cells.

Mean Cell Return Visits (x#y#ReturnVisits): The mean number of visits to return to cell x#y# for each trajectory in the group, summed across trajectories, and divided by the number of trajectories in the group. Mean visits to return for a trajectory is calculated as follows: A cell return is defined as any sequence of events outside the cell beginning with an exit from cell x#y# and ending with a re-entry. The number of cell returns is

equal to the number of cell visits minus one. Returns comprising a number of cell visits greater than [Maximum Return Visits](#) are considered to have a length of [Maximum Return Visits](#) for the purpose of this measure. The event sequence from the final cell exit in each trajectory until the end of that trajectory is considered a 'pseudo-return': it is included in this mean (truncated by [Maximum Return Visits](#) if applicable) if and only if it has a length greater than the mean of the trajectory's cell return visits as calculated from proper returns alone. Otherwise, it does not contribute. The Mean Cell Return Visits for a trajectory group is simply the mean, across the trajectories, of each trajectory's cell return visits. Trajectories containing fewer than [Minimum Number of Returns](#) (including the final pseudo-return if applicable), however, do not contribute to the Mean Cell Return Visits. Events within cell x#y# of duration less than [Minimum Event Duration](#) are not taken as sufficient to terminate a return to the cell, but can contribute to the return visits of other cells.

First Entry (x#y#FirstEntry): The mean, across trajectories, of the time until the first entry into the cell or region. Trajectories that never enter the cell or region are given the maximum value, which is the total duration of that trajectory. Thus, First entry is never a missing value. First Entry and Last Exit should be used with caution. One good rule of thumb is that if the mean First Entry is a larger value than the Last Exit, the number of visits to that region are low and generalizations made about that region would be based on a small portion of trajectories.

Last Exit (x#y#LastExit): The mean, across trajectories, of the time of the last transition out of the cell or region. Trajectories that never enter the cell or region are given a value of missing. First Entry and Last Exit should be used with caution. One good rule of thumb is that if the mean First Entry is a larger value than the Last Exit, the number of visits to that region are low and generalizations made about that region would be based on a small portion of trajectories.

Apply Time Filter:

This option allows the user to only export the values that pertain to the reduced time window selected by the [time filter](#). All selected measures will be exported based on this reduced time frame as well as 2 more variables: minimum and maximum time values for the time window selected.

5. Tab-delimited Text Files

The files exported from GridWare are tab-delimited. That is, each value in a row is separated by a tab, rather than a space or a comma. When viewing these files as text, this will make the column labels and column values not line up and may be visually confusing. However, this format allows for the easiest integration with other spreadsheet software. The user can cut and paste or import the exported measures text file into most statistical packages for further analysis.

This is also the reason why we recommend using tab-delimited text files for the raw trajectory data and the GridWare file. This is the easiest format to transfer files from other software. It also allows for spaces to be considered text characters for easier variable naming (i.e. Father Affect instead of FAff or FatherAffect or some other less readable format).

6. Tips and Troubleshooting

Performance:
- GridWare can be very slow in opening, displaying or saving large datasets. Be patient, and refrain from clicking while GridWare is processing a prior command.
- Avoid displaying many trajectories at once unless necessary, as this will slow down the display considerably. For faster display performance, switch to opaque and/or hollow nodes, and hide nodes or transitions. When you first open your data, all filters will be fully open, and so all trajectories will be selected. Using the filters to select a small number of trajectories before displaying nodes and transitions will avoid a long delay.
- Particularly on Windows, if GridWare does not quit properly, it may remain running in the background, occupying a considerable amount of memory. Press Control-Alt-Delete once to get to the Task Manager, and look in your list of processes for any called javaw.exe or GridWare.exe. Select "End Process" for each.
- Responsiveness to the 'Reset Filters' command can very slow if you are displaying many trajectories, and the time filter is not fully open. It may appear that GridWare has frozen. Be patient.

Opening and Closing:
- If you can't get GridWare to open your data by using a trajectories list file and the variable creation dialog, try manually creating a '.gwf' file by referring to the manual and example data.

- In the variable creation dialog, the variable type and format choosers seem to work just once. If you try to switch them a second time they won't work, and you'll have to remove that variable and add a new one.
- If you want to open a second dataset using GridWare, it is safer to quit the program and restart it rather than just using the 'Open' command. Otherwise you may find GridWare runs out of memory prematurely.
- GridWare asks you if you want to save after any savable change has been made. This includes a trajectories list import, changes to preferences, axis categories, colour, or randomized node positions.

Data Management:
- Keep a back-up of your data. When you first save your data from within GridWare, give your gwf file a different name from your trajectories list file, or else your original trj files will be overwritten. GridWare overwrites trj files when and only when it overwrites the corresponding gwf file, so to avoid overwriting anything, just enter a new name for your gwf file when saving.

Data Manipulation and General Operation:
- Instead of clicking with the mouse, you can use TAB and SHIFT-TAB to move the 'focus' within the filters window -- that is, to change which element of the interface will respond to input. Moving the focus to the lower or upper bound field of a range filter makes the slider thumb appear for that bound. You can then use the mouse to move that thumb, or the keyboard (provided you shift focus to the slider first -- again, by using the mouse or TAB).
- You can recode your state variables by renaming the categories along the x-axis (GridWare won't allow you to recode two values into one in this fashion, however). Changing the missing value symbol in preferences will replace this symbol in your trj files as well upon saving.
- One must press enter after typing in a new upper or lower bound in the range filters.
- When renaming x-axis categories, one must press enter to save the changes. But sometimes pressing enter yields no response. If this happens, try selecting a different category field with the mouse and pressing enter again.
- Excel, SPSS, and similar software may be limited in the number of columns of data they can import. If you export every measure for every cell of your grid, you can end up with many hundreds of columns. If you find you can't import this much data, try selecting only some cells, or only some measures, to export.

- Many measures have customizable parameters which determine how they are calculated. It is important that you understand how the values of these parameters (set in the Preferences window) affect the results of these calculations. See the 'Measures' section of the manual for more information.
- Before closing a dataset or quitting, if any savable change has been made since opening or saving the dataset, GridWare will prompt you to save. These changes include a trajectories list import, changes to preferences, axis categories, colour, or randomized node positions.

Display:
- If running GridWare with a low screen resolution (eg. some lcd projectors are only capable of 800 x 600 pixels) minimize the number of filters and measures displayed and close the run log window, and arrange the other windows to allow for a large grid window. You can also minimize windows into the taskbar or dock to temporarily get them out of the way.
- Randomizing nodes one cell at a time will optimize your layout more effectively than trying to do the whole grid at once.
- Colour by a hidden state variable to add a third dimension to your grid.
- If the grid title disappears (sometimes as a result of switching axis variables), make a change in the filters, then change them back.
- If one of the axes is partially obscured, or the trajectories come slightly out of alignment with the grid, resize the grid slightly.
- If the min and max fields in your ordinal filters are 'squished', widen the filters window.
- You may close the run log window at any time after start-up to get it out of the way. You won't be able to get it back, however, so check gw_run_log.txt for error messages if something goes wrong.

Missing Data:
GridWare's default symbol for missing values in both input and output is '--'. If your data uses a different symbol (eg. '*'), you will need to manually create a gwf file, instead of importing a trajectories list file. After the variable definition lines, but before the '</config>' line, add the following line:

MissingValueSymbol   *

Note that GridWare will use this symbol to represent missing values when it saves your trajectory files, and also when exporting measures. If you want the exported measures to use a different symbol from that used in your original trj files, change the Missing Value Symbol within GridWare Preferences while the program is running. Again, keep in mind that this will also affect the trj files saved by GridWare.

Technical Support:
- Before contacting tech support for GridWare, please go to the GridWare website to check for known issues and bugs.
- To contact technical support, go to http://www.statespacegrids.org/
- When requesting help, send us your gwf file and a couple trj files to look at, as well as the file 'gw_run_log.txt'.

Appendix A: Simple example used in manual (Example 1)

These data were generated to show a generic format for GridWare. There are 3 state variables and only a few trajectory variables (ID, wave, age, and gender). The 3 state variables could be any time series variable. One option is to think of these as the same measure for three different people, such as child, mother, and father. Another option is to think of these as 3 variables for one individual. For example, variable 1 could be the degree of a toddler's engagement with a frustrating toy, variable 2 could be the degree of engagement with mother, and variable 3 could be the level of distress (for a similar example see Lewis, Zimmerman, Hollenstein, & Lamey, 2004).

It is highly recommended that new users open this file in GridWare and use it to explore some of the features. Below are some suggestions that highlight key features:

- In the Example 1 folder there is a trajectories list file (Example 1 trajectories list file.txt). Open this to practice using the Variable Creation Dialogue.
- Select family A1 with the mouse by single-clicking on it. Now the trajectories for all three waves of A1 are displayed. To limit to a single trajectory for one wave, use the ALT/Option key while clicking and dragging the slider head from 3 to 1.
- Select Variable 3 for both the Y Axis and the X Axis. Select the 1/1 cell in the lower left corner. Colour the events in that cell red. Now return to Variable 1 on the X Axis and Variable 2 on the Y Axis. The events that are colored red are those that correspond to a value of 1 on Variable 3. (Go to colour dialogue box to reset all to blue)
- Select a single trajectory (i.e. A1 wave 1) and then change the maximum value on the Time Slider from 15 to 3 and hit return. This will restrict the trajectory

displayed to a 3 second window. Click once on the slider itself and drag it to move the trajectory through time. You can change your trajectory selection to replace or add new trajectories to see how these change over time. To return to the view of the entire duration, enter 0 and 15 as minimum and maximum, respectively.

- Save a back-up of the Example 1 data in another folder. After changing the axis labels and/or colouring some of the nodes, save the Example 1 data as "Example 1 new.gwf". Look at the trajectory files and GridWare file to observe the changes made from the save procedure. Close GridWare and reopen the "Example 1 new.gwf" file.

Appendix B: Example 2

These data simulate a test-retest design of parent child interactions for 4 families across 4 waves. The sessions are 2 weeks apart, the waves are once a year, and each session is 900 seconds (15 minutes) long. The state variables are the affective states of the parent and child (Negative Engagement, Negative Disengagement, Neutral, Positive Engagement) and a third variable that identifies the task that the dyad is engaged in (Off-task, playing games, planning a fun activity together, academic task, and problem solving discussion). The trajectory variables are child gender, parent gender, child's age in months, child's externalizing t-score on the Child Behavior Checklist, child's internalizing t-score on the Child Behavior Checklist, and parent's depression score from the Beck Depression Inventory.

Appendix C: State Space, Attractors and Dynamic Stability

Dynamic, self-organizing systems share several key properties. One key feature of open systems is that, although theoretically they have the potential to exhibit an enormous number of behavioural patterns, they tend to stabilize in a limited range of these possibilities. Stable patterns emerge through feedback among many lower order (more basic) system elements; these emergent patterns are referred to as *attractors* in DS terminology. In real time, attractors may be understood as absorbing states that "pull" the system from other potential states. Behaviour moves in a trajectory across the state space toward these attractors. Over developmental time, attractors represent recurrent patterns that have stabilized and are increasingly predictable. As noted by Thelen and Smith (1994), all developmental acquisitions can be described as attractor patterns that emerge over weeks, months, or years.

As recurring stable forms, attractors are often represented topographically as valleys on a dynamic landscape (see Figure 1). The deeper and wider the attractor, the

more likely it is that behaviour falls into it and remains there, and the more resistant it is to small changes in the environment. *Repellors*, or states that the system tends to avoid or be repelled from, are represented as hills on this landscape – states in which the system cannot rest. As the system develops, a unique *state space*, defined as a model of all possible states a system can attain, is configured by several attractors and repellors. Critically, living systems are characterized by *multistability* (Kelso, 1995); that is, their state space (i.e., behavioural repertoire) includes several co-existing attractors. Contextual constraints probabilistically guide behaviour toward a particular attractor at any given moment in time.

State Space Grids (SSGs).

DS theorists use the concept of a *state space* to represent the range of behavioural habits, or attractors, for a given system. In real time, behaviour is conceptualized as moving along a trajectory on this hypothetical landscape, being pulled towards certain attractors and freed from others (see Figure 1). Based on these abstract formalizations, Lewis, Lamey and Douglas (1999) developed a graphical approach that utilizes observational data and quantifies these data according to two ordinal variables that define the state space for any particular system. Lewis and his colleagues have primarily studied intraindividual attractor patterns that emerge and change in the early years of life (e.g., Lewis et al., 1999; Lewis et al., 2004). The grids they originally developed utilized two ordinal variables (e.g., degree of engagement and intensity of distress) that tapped individual infants' socioemotional habits across several developmental epochs.

State space grids (SSGs) have also been developed to represent dyadic behaviour (e.g., parent-child interactions, peer relations; Granic & Lamey, 2002; Granic et al., 2003). The dyad's trajectory (i.e., the sequence of behavioural states) is plotted as it proceeds in real time on a grid representing all possible behavioural combinations. Much like a scatter plot, one dyad member's (e.g., parent) coded behaviour is plotted on the x-axis and the other member's (e.g., child) behaviour is plotted on the y-axis. What is different, however, is that each x-y coordinate represents a moment (e.g., second, event, bin) in time, rather than a case in a group of cases. Two behaviour streams are segmented into time-points or events, and simultaneous values are plotted. Thus, in dyadic grids, each point on the grid represents a two-event sequence or a simultaneously coded parent-child event (i.e., a dyadic state). A trajectory is drawn through the successive dyadic points in the temporal sequence they were observed. Any time the behaviour changes a line is drawn connecting the new point and the previous point. For example, a hypothetical trajectory representing 10 seconds of coded behaviour is presented in Figure 2. The sequence depicted begins with 2 seconds in negative engagement/negative engagement[3], 2 seconds in negative engagement/neutral, 3 seconds in neutral/neutral, 1 second in neutral/negative engagement, and 2 seconds negative engagement/negative engagement. Note that the points are also plotted in sequence along a diagonal such that the first state is plotted in the lower left of the first cell and the final state is plotted in the upper right of the final cell. Plotting the points along the diagonal allows researchers to identify when in the session a behavioural event occurred.

The SSG methodology can tap both the content of interactions (e.g., mutual hostillity, permissive parenting) and the structure of these interactions. When we refer to structure, we mean the patterning of behaviour, regardless of content, such as the relative flexibility versus rigidity of interaction patterns. With SSGs, we are able to examine whether behaviour clusters in very few or many states (i.e., cells), or regions (i.e., a subset of cells) of the state space. We can also track how long the trajectory remains in some cells but not others, and how quickly it returns to particular cells. If a dyadic trajectory remains in a small number of cells, and makes very few transitions between cells, this system may be thought of as rigid, inflexible, or stuck. In contrast, a trajectory that moves around to many cells in the state space grid and makes frequent changes between these cells may indicate a highly flexible, or variable, system. We can identify attractors as those cells to which behaviour is drawn repeatedly, in which it rests over extended periods, and/or to which it returns quickly. Moreover, as discussed in the following sections, a range of variables that capture the relative stability of particular attractors may be derived from SSGs and these values can be tested statistically for changes in real and developmental time.

A major advantage of SSGs is that they provide an intuitively appealing way to view complex, interactional behaviour; thus they are first and foremost a useful tool for exploratory analysis. A recent study in which we examined the heterogeneity of family interactions with aggressive children may help illustrate this point (Granic & Lamey, 2002). SSGs were used to explore differences in the parent-child interactions of 'pure' externalizing children (EXT) and children comorbid (MIXED) for externalizing and internalizing problems. This study is useful not only for demonstrating how the grids work, but also to demonstrate design innovations based on DS principles that are useful with or without SSGs, in this case, a systematic perturbation.

Parents and clinically referred children discussed a problem for 4 minutes and then tried to "wrap up and end on a good note" in response to a signal (the perturbation) within the next two minutes. The perturbation was intended to increase the emotional pressure on the dyad, triggering a reorganization of their behavioural system. We hypothesized that, as a function of differences in the underlying structure of their relationships, EXT and MIXED dyads would be differentially sensitive to the perturbation and would reorganize to different parts of the state space. Prior to the perturbation, however, we expected dyads' interactions to look relatively similar. Separate grids were constructed for the pre- and post-perturbation interaction sessions. For this study, the lines (trajectories) are less important to notice than the points which show clustering in particular cells. Figure 3 shows an example of an interaction between a pure externalizing child and his parent, pre- and post perturbation. As exemplified in these grids, EXT dyads tended to go to the permissive region (child hostile – parent neutral/positive) of the state space grid, as well as other regions (i.e., mutual neutrality and negativity), before the perturbation. After the perturbation, EXT dyads tended to remain and stabilize in the permissive region. Figure 4 represents the interaction of a MIXED dyad. Similar to EXT dyads, the MIXED dyads occupied the permissive region, as well as other areas, before the

perturbation. But in contrast with the EXT group, MIXED dyads tended to move toward the mutual hostility, or mutual negativity, region of the state space grid after the perturbation. Granic and Lamey (2002) concluded that the perturbation was a critical design innovation that provided the means by which clinical subtypes could be differentiated.

Another contribution of this study was a more general one – the use of SSGs, with their rich case-by-case temporal narratives, provided a technique to further parse interaction processes that have been previously assumed to represent one coherent pattern. In this case, the coercive process (Patterson, 1982; Patterson, Reid, & Dishion, 1992; Snyder, et al., 1994) was shown to constitute two separate microsocial patterns – two separate attractors on a state space (Granic & Patterson, 2001). Moreover, the conditions under which dyads would be drawn towards one region or the other were found to differ for subtypes. The use of SSGs to uncover heterogeneous processes may be relevant to a variety of phenomena in developmental psychopathology including variability in the real-time unfolding of attachment patterns (cf. Coleman & Watson, 2000), in bullying interactions on the playground (cf. Pepler, Craig, & O'Connell, 1999) and in parent-adolescent interactions during puberty (Granic, Dishion et al., 2003; Granic, Hollenstein et al., 2003).

State Space Grid analysis: Real-time measures

SSG patterns can be quantified and used as variables for statistical analyses. Variables that capture the temporal and spatial patterning of behaviour have been developed for time-based (e.g., second-by-second coding) as well as event-based (e.g., conversational turns) data (Lewis et al., 1999). The same variables may be used regardless of whether the researcher is using the grids to map individual or dyadic behavioural trajectories. We have provided a list of some of these variables but, depending on the research question, additional ones may be created. Once these parameters have been computed, different types of attractors may be identified and the relative stability of these dynamic states can be measured and subsequently compared in a variety of ways (Lewis et al., 1999, Lewis, Zimmerman, Hollenstein, & Lamey, 2004).

In general, long durations and/or frequent recurrences of behaviour in a particular cell or region suggest an attractor on the state space, and these hypothetical attractors can be compared and tested within individuals across development as well as between individuals. Moreover, parameters describing the stability or variability of behaviour across the state space can be calculated for each grid, allowing global, structural comparisons over time, populations, or individuals. Below, as we list the parameters that can be derived from the grids, we refer to Figure 5 for examples. These grids were taken from a study conducted by Lewis and colleagues (1999) that examined the socio-emotional coping patterns of infants, and changes in those patterns over a hypothesized stage transition. Originally, these grids were representations of intra-individual behaviour plotted according to two ordinal variables, but we have left out the axes labels because they can just as easily represent dyadic behaviour and our intention is to provide a generic description that can be adapted to a variety of observational data. It should also

be noted that unlike the grids in the Granic and Lamey study which used event-based data (conversational turns as observational units), time-based data is plotted in Figure 5; larger dots in these plots represent longer durations.

The following are parameters that may be computed for each cell in the grid: (1) Raw density: cumulative duration (or number of hits) per cell; (2) Proportional density: density divided by total episode duration or total number of events; (3) Perseverance 1: mean duration (or mean number of consecutive hits) per cell; (4) Perseverance 2: longest duration (or longest series of consecutive hits) per cell; (5) Return time: latency to return to a cell following an event in that cell. This can be measured in units of time, number of events, or number of unique cells visited en route. For example, in Figure 5, grid B shows a high raw density in cells 2/2 (again, cell labels follow the x/y convention) and 2/3 and a very low raw density in cell 1/3. In grid C, cell 3/1 shows a high value for perseverance 1 (each time behaviour goes to that cell, it tends to stay there for some time) whereas cell 1/1 in that same grid shows a low perseverance value. Finally, grid B shows a very low return time for cell 2/2 (every time behaviour leaves that cell, it returns in approximately one turn), but a high return time for grid A, cell 1/0.

There are also several parameters and summary values that can be computed for the entire grid (rather than cell by cell): (1) Dispersion: total number of cells visited (with or without controlling for total time or total number of events); (2) Fluctuation: number of transitions between cells (with or without controlling for total time or events). Note that fluctuation may be high even though dispersion is low, providing an additional useful parameter (see grid B); (3) Stability 1: average of either mean or maximum cell duration values (or events per cell) across all cells. Note that high values indicate overall stability or stickiness of behavioural states; (4) Stability 2: mean return time (in time or event units) across all cells. Note that low values indicate overall stability or resilience of behavioural states; (5) Peak stability: maximum single duration (in time or event units) out of all the cells occupied. Returning to Figure 5, grids A and C show high dispersion, compared to grid B, and grid A shows a low stability 1 value, compared to grid B.

Researchers may choose to calculate one or a few of these whole-grid parameters, depending on what makes most sense in any particular study. Another option is to combine two or more of the whole-grid parameters into one summary construct score or latent variable that represents "flexibility/rigidity" of interaction patterns (e.g., Hollenstein, Granic, Stoolmiller, & Snyder, 2004; Granic, 2003). For example, to create a "flexibility/rigidity" construct, the SSG parameters of interest are transferred into z-scores and a reliability analysis is conducted to ensure that the chosen parameters are sufficiently correlated and reliably hang together (measured by Cronbach's standardized alphas and Pearson correlations). Any parameters that do not meet appropriate criteria can be dropped and the z-scores from the remaining reliable parameters can be averaged into one "flexibility/rigidity" construct score. This potentially more robust score can then be used in any kind of appropriate statistical analyses (e.g., growth curve analyses, SEM, regression analyses, ANOVAs).

In addition to examining the patterns across the entire grid, developmental psychopathologists are often interested in the relative stability of one or just a few behavioural patterns or attractors (e.g., depressed mother-infant mutual gaze, coercive interactions). Lewis and colleagues (1999, in press) have developed a number of quantitative strategies for identifying attractors on a state space grid. Using the measures previously listed (density, perseverance, return time), attractors can be defined as the cell or group of cells highest in these values. Once attractor cells are identified, the computed parameters for those cells serve to characterize the strength, endurance, and stability of the attractor for comparison purposes. These comparisons are particularly powerful when they are conducted across developmental time.

State space grid analysis: Developmental measures

After computing the parameters that are most relevant for a particular research question, statistical techniques (most of which are quite familiar to developmental psychopathologists) may be applied. We recommend using these statistical procedures in such a way that maintains the integrity of the individual (or dyadic) case (e.g., curve estimation procedures, cluster analysis). However, multivariate analyses including ANOVAs, regressions, and structural equation modeling (SEM) can just as easily be run on the grid variables.

One example of a developmental state space grid analysis comes from a recently completed study by Lewis, Zimmerman, Hollenstein, & Lamey (2004). These researchers were examining a developmental transition in late infancy that was hypothesized to exhibit the properties of a phase transition (i.e., increased variability in real-time patterns, a break-down of old attractors and the emergence of new ones over developmental time). The general hypothesis that guided this study and similar DS-inspired studies that focus on phase transitions is presented in Figure 6. Infants were videotaped in frustrating situations on 12 monthly visits before, during, and after a hypothesized transition point at 18-20 months. SSGs were constructed for each episode and grid-to-grid differences were compared over age. As predicted, grid-to-grid differences were greater during the transitional period than before or after, indicating a developmental reorganization of behavioural responses to negative emotion. Also, new attractors appeared more frequently during the period of transition than at other ages.

Lewis and colleagues provide two techniques to measure <u>within-subject</u> differences among SSGs. The first is called *inter-grid distance* score (IDS) and is computed in the following steps. Grid-to-grid Euclidian distance scores yield a global metric of the difference in behavioural landscapes from month to month, based on the sum of squared differences across all cells. For each grid cell, the difference in duration values over two consecutive months is calculated, then squared, and then these values are summed for all cells. Next, the square root of this sum is taken as the distance score between the two grids.

A second developmental analysis they explored was a cluster analysis technique to look at changes over time. The first step is to categorize the grids by entering all grids for all months for every child into a k-means cluster analysis. The most parsimonious cluster

solution is chosen (based on pre-set criteria). The cluster score for each grid is then recorded. Visual inspection of the grids is recommended at this point to ensure that the same cluster scores look alike topographically, having similar duration values for many of the same cells. Developmental continuity would thus be indicated by a sequence of months (2 or more) with the same cluster score, and developmental variability would be indicated by month-to-month change in cluster membership.

Another method of analyzing changes in SSG patterns over developmental time comes from a recent study that examined changes in the structure of family interactions during the early adolescent transition period (Granic, Dishion et al., 2003; Granic, Hollenstein, et al., 2003). Following Lewis and colleagues' stage transition hypothesis, we examined whether early adolescence (i.e., around puberty; age 13-14 for boys) constitutes a phase transition (a period of reorganization) marked by a peak in the variability of family interactions; before and after this period, interactions were expected to be stable (refer back to Figure 6 for schematic representation of hypothesis). Longitudinal observational data were collected in five waves prior to, during, and after the transition period. One hundred forty-nine parents and boys were observed problem-solving at 9-10 years old, and every two years thereafter until they were 17-18 years old. Based on this data, state space grids were constructed for all families across all waves. An example of a parent-child dyad across 5 longitudinal waves is shown in Figure 7. Consistent with our hypothesis, this was a characteristic pattern for the sample: the sequence of state space grids showed that behaviour became more variable (i.e. occupied more cells and moved around the grid more frequently) at the third wave when the boys were in early adolescence (age 13-14 years old). Before and after this period, dyadic behaviour looked more stable and less flexible; fewer cells were occupied and there were fewer changes between cells. Two parameters indexing the variability of the interactions (fluctuation and stability) were derived from these grids. Repeated measures ANOVAs on these variables revealed a significant quadratic effect. To ensure that these results were not just significant on the group level, but also characterized the majority of families in the sample (and, thus, to be more faithful to systems-oriented, case-based assumptions), the wave at which flexibility peaked (when fluctuation was highest) was recorded. Results revealed that the majority of families showed a significant peak in variability in the middle wave – and hardly any families peaked in the first or last wave.

From these examples, it should be clear that any statistical tool that has been developed to measure growth or change over time can be combined with SSG analysis. The important difference in these variables (compared to questionnaire data, for instance) is that they capture temporal patterns as they unfold over time. The importance of tapping these dynamic processes in terms of explanatory power cannot be understated, but developmental psychopathologists are also interested in predicting outcomes. Although some systems scholars have argued that prediction should not be a goal for developmental scientists (much like prediction is not a goal for biologists), others in the field still consider it critical.

We recently completed a study in which one of our objectives was to predict antisocial behaviour from dynamic systems measures of parent-child interactions. The study combined descriptive developmental profiles, SSG analysis, and more traditional multivariate statistical procedures (Hollenstein, Granic, Stoolmiller, & Snyder, 2004). The main goal of this study was to examine the connection between rigidity in parent-child interactions and the early development of antisocial behaviour. We began the study by conceptualizing psychopathology as "diminished flexibility and constrictions in the affective, cognitive, and behavioural correlates of adaptational patterns" (Overton & Horowitz, 1991, p. 3). Clinical researchers have long viewed psychopathology as over-learned, automatized cognitive, affective, and behavioural patterns that are impervious to changes in the environment and interfere with an individual's ability to function socially (e.g., Cicchetti & Cohen, 1995b; Mahoney, 1991). This view of psychopathology as rigidified patterns of interaction has been theoretically extended to the family system (Minuchin, 1974) and coercive family processes specifically (Patterson, 1982), but empirical studies have not explicitly focused on rigid parent-child interactions and the emergence of child psychopathology.

In this study (Hollenstein, et al., 2004), high risk children in kindergarten (n=240) and their parents were observed for two hours engaging in a variety of tasks (e.g., playing a game, problem-solving) that were expected to elicit a range of affect for most families. SSGs were constructed from the coded observational data and several whole-grid measures of rigidity were derived from the grids. A rigidity construct was created, using methods previously described. As hypothesized, results showed that parent- and teacher-reported antisocial behaviour, measured concurrently and 18 months later, were positively correlated with rigidity scores. Children rated high on antisocial behaviour were significantly more likely to exhibit more rigid parent-child interaction patterns than their non-antisocial counterparts. Regression analyses also indicated that rigidity scores in the fall of kindergarten predicted growth over 18 months in parent-reported antisocial behaviour after controlling for concurrent levels.

We recently extended these results by examining the developmental growth profiles in both externalizing and internalizing behaviour for this same sample (Hollenstein, et al., 2004). As shown in Figure 8, children's growth profiles in externalizing behaviour across 4 waves of data collection (over 24 months) showed 4 distinct types of trajectories: growers, desistors, stable high and stable low. Figure 9 shows the means on the rigidity construct for the four growth trajectory groups. Results indicated that the growers and stable high groups scored significantly higher on parent-child rigidity scores than the stable low and desistors. This was an exciting finding for us because of the potential diagnostic implications. Returning to Figure 8, the growers and desistors began approximately at the same level of antisocial behaviour. What distinguished the group of children who became increasingly more antisocial over time from those children who became less so over time was how rigid the parent-child interactions were at the first wave.

Although the SSG method is clearly still in its early stages of development, we are encouraged by its potential. One of the important advantages to this technique is its inherent flexibility. At the very least, it is a visual, exploratory tool to develop and refine hypotheses. At best, this methodology provides a source from which novel, temporal-based, process-level predictors may be tapped and used to strengthen current models of normal and atypical development. Researchers are not limited to using continuous time-series, as is the case with many other DS methods. Categorical and ordinal data are also appropriate for this type of analysis. Also, the grids are malleable in that they can represent systemic behaviour on the individual as well as dyadic level. In addition to the examples mentioned above, changes in peer, romantic couples, and sibling interactions, for example, can easily be tracked using SSGs. In fact, apart from the difficulties in visually representing the data, the variables derived from the grids can be extended past the two-dimensions on which we have focused. For instance, triadic family interactions or family interactions with siblings as well as parents, can be measured for attractor strengths, fluctuations, and so on. Along similar lines, peer interactions with three or more children on the playground can be analyzed with this strategy.
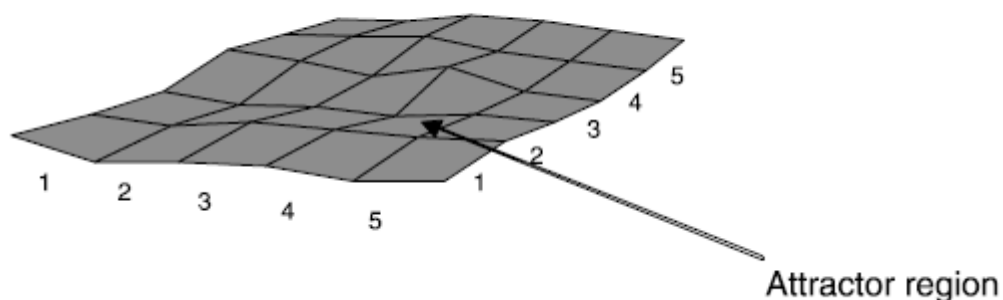
Figure 1. Hypothetical landscape of attractors.
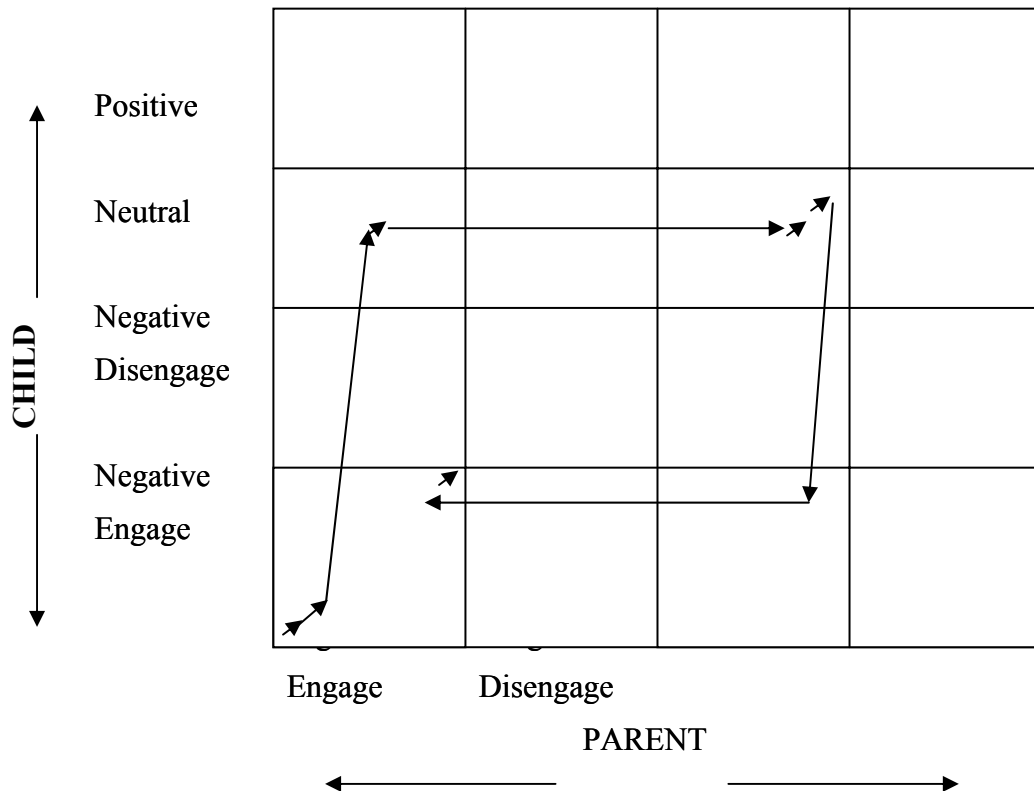
Figure 2. Example of a state space grid with a hypothetical trajectory representing 10 seconds of coded behaviour, one arrow head per second. Plotting begins in the lower left part of the cell and moves in a diagonal as each second is plotted, ending in the upper right.
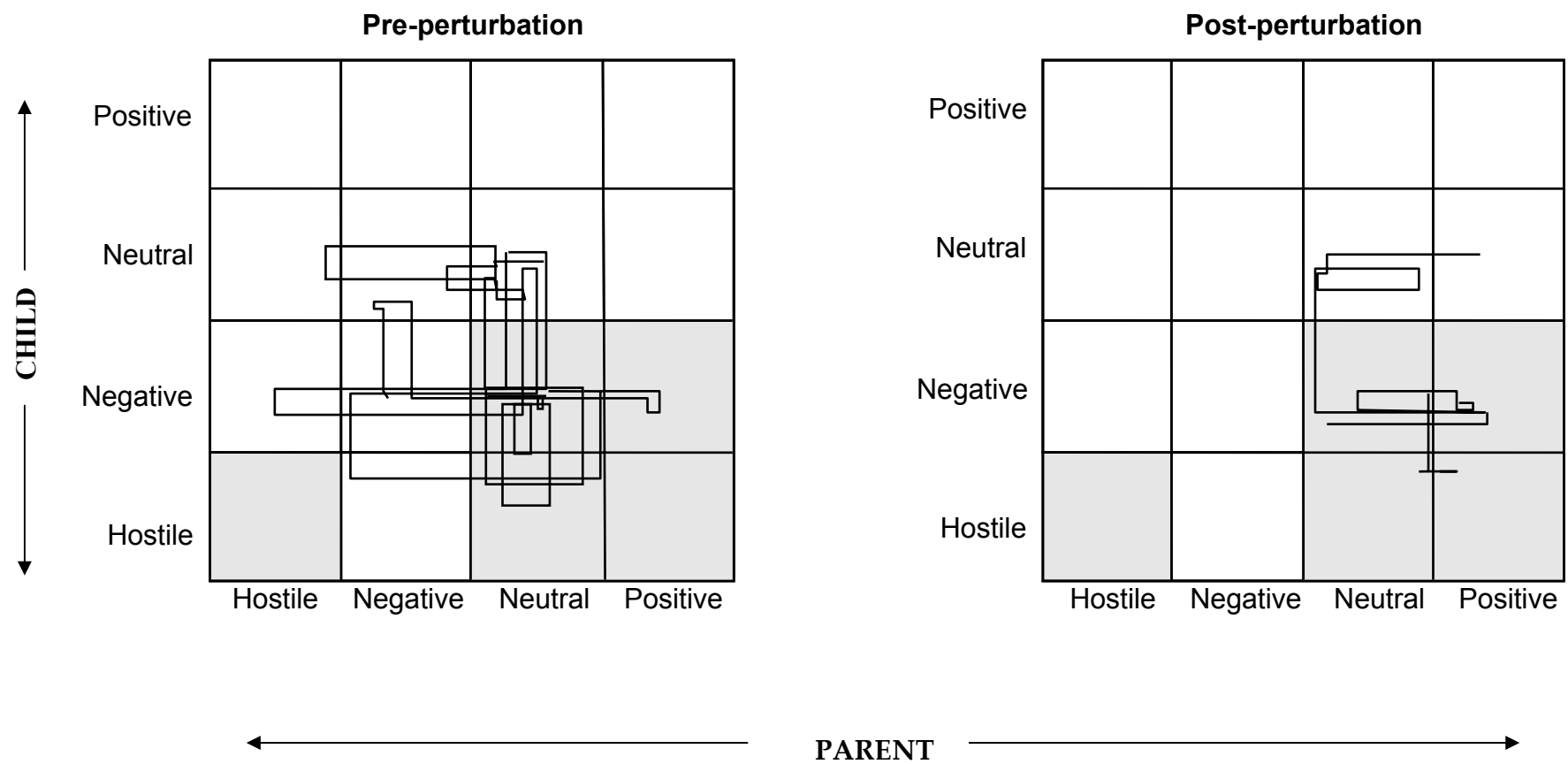
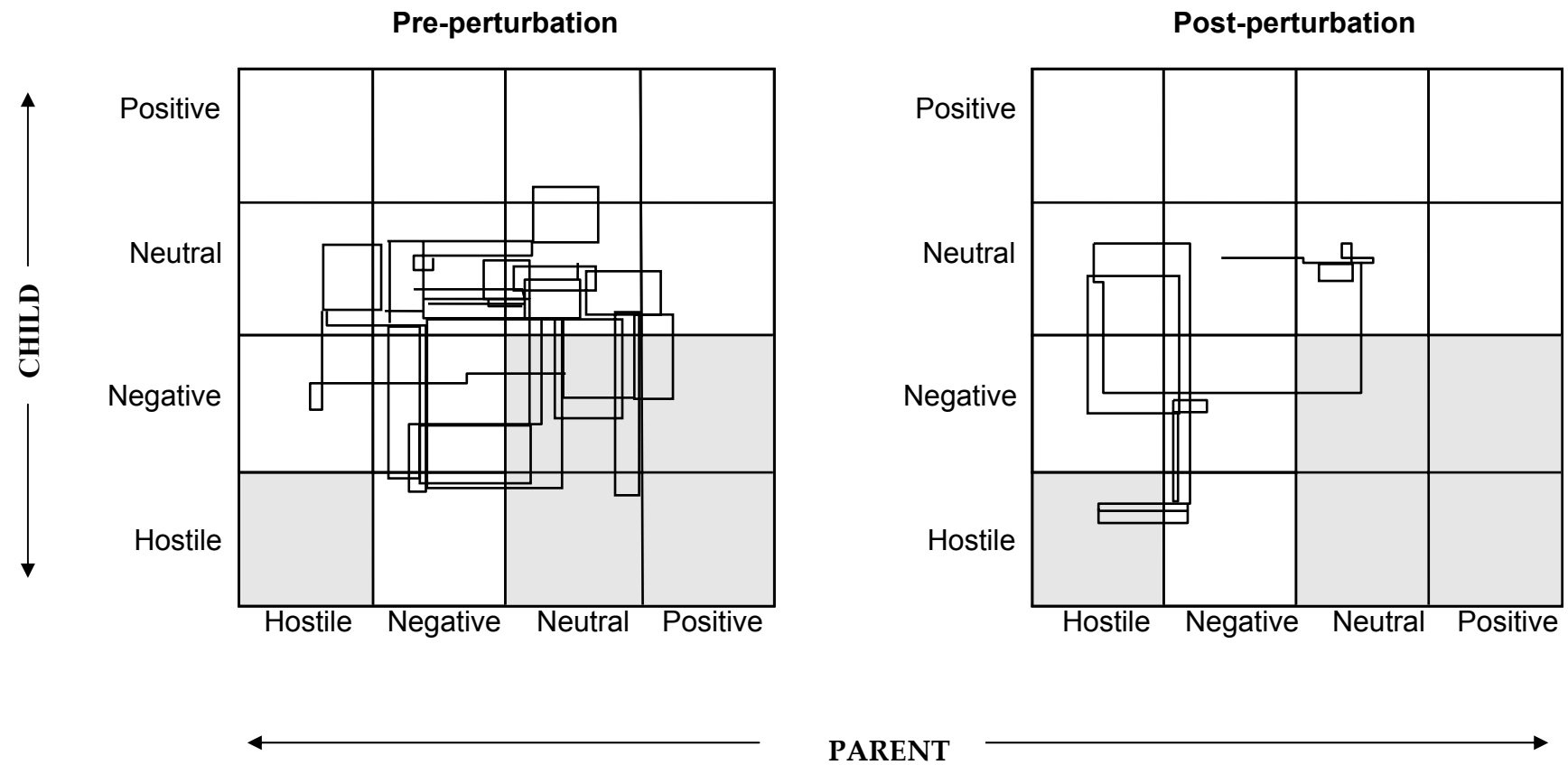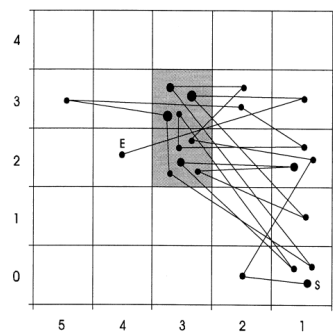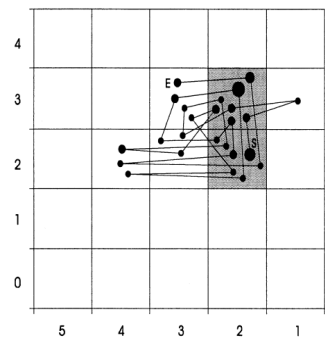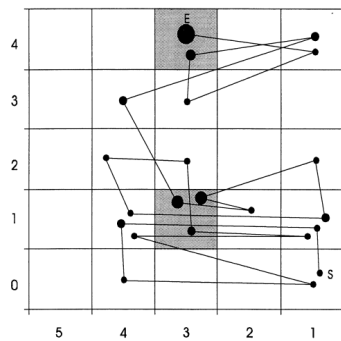Figure 3. Pre- and post-perturbation state space grids for an EXT dyad.

Figure 4. Pre- and post-perturbation state space grids for a MIXED dyad.

Figure 5. Examples of three state space grids from Lewis, Lamey, and Douglas, 1999.

Developmental profile

Phase Transition

Real time portraits
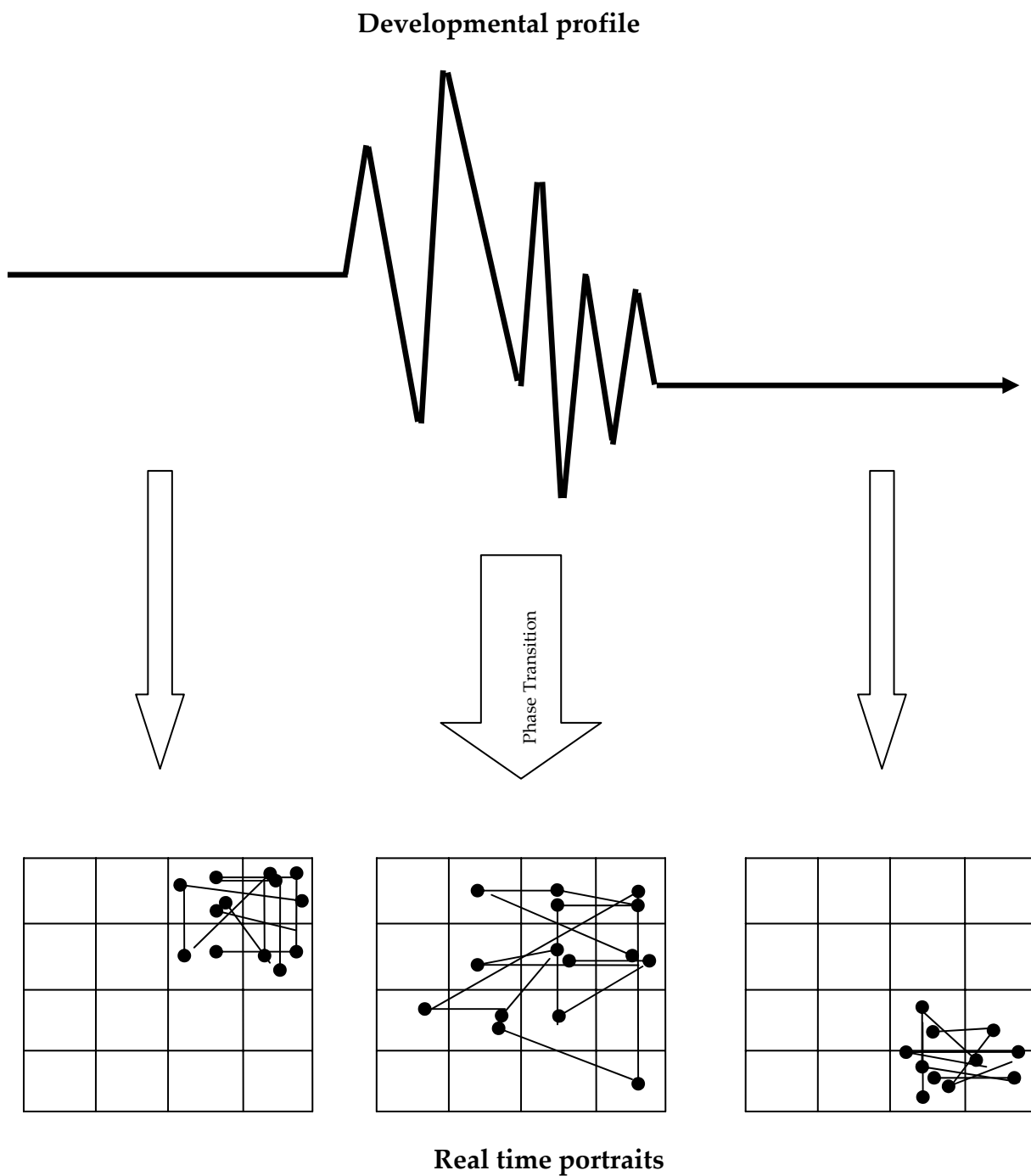
Figure 6. Model of a developmental phase transition; developmental phase transitions are periods of fluctuation in developmental time and increased variability in real time.
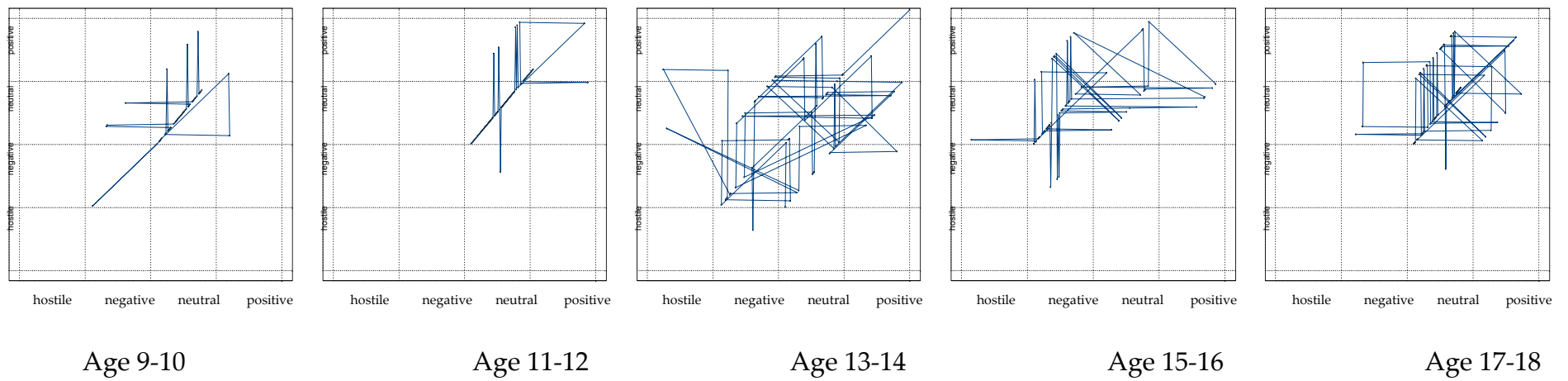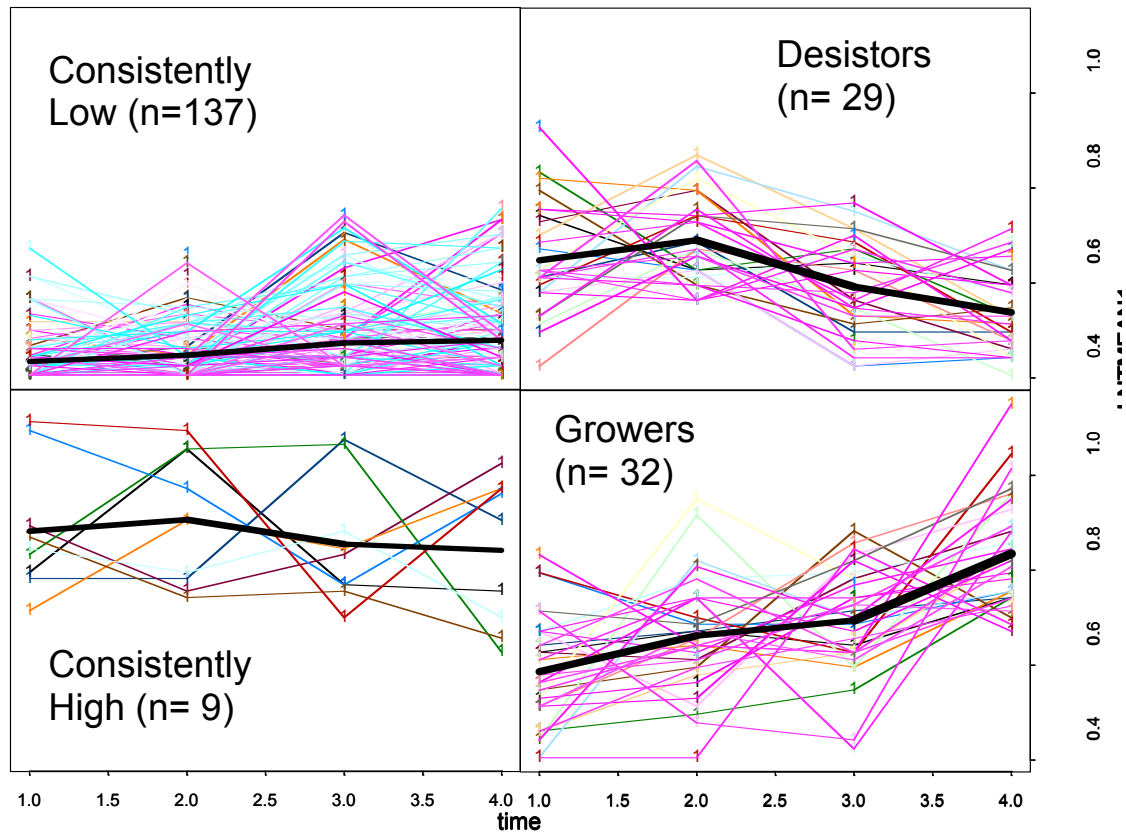
Figure 7. Sample state space grids for one family across five longitudinal waves (Granic, Hollenstein, et al., 2003).
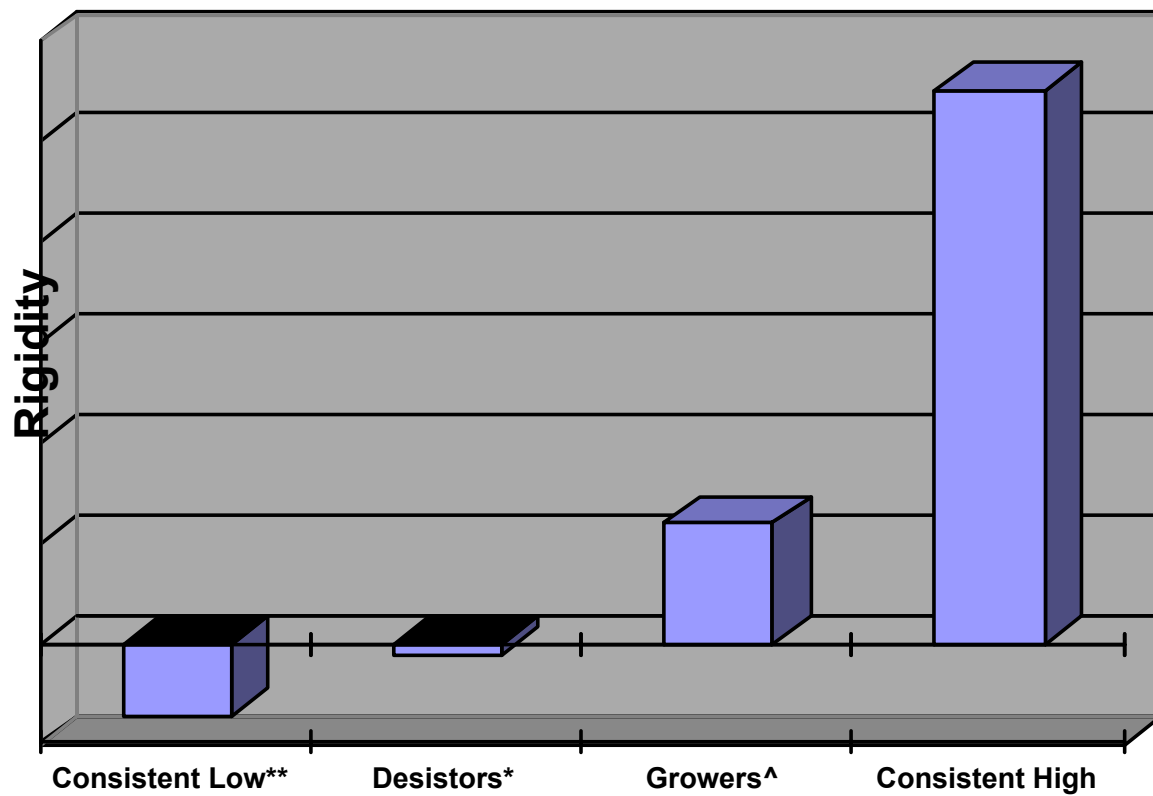
Figure 8. Clusters of antisocial trajectories.

Figure 9. Means scores on rigidity construct by antisocial

trajectory cluster.

Footnotes:

1. Events of duration less than Minimum Event Duration do not contribute to these measures.
2. Cells with cumulative duration less than Minimum Cell Duration do not contribute to these measures.
3. Note that the labeling of cells follows the x/y convention such that the first half of the label is the parent's category and the second half of the label is the child's category.