JWatcher^{TM} 1.0
An Introductory User's Guide

Daniel T. Blumstein, Janice C. Daniel & Christopher S. Evans

http://www.jwatcher.ucla.edu/

Last modified: 9 January 2006

Table of Contents

1 Thank you!

Since releasing JWatcher 0.9 in November 2000, we have had thousands of downloads from all over the world. Neuroscientists, human factors experts, psychologists, and veterinarians have joined the many behavioral ecologists who use JWatcher to analyze their observations and experiments. We have taught thousands of students the fundamentals of quantifying animal behavior using JWatcher, and these labs at UCLA have been extremely popular. Educators in other institutions have used these labs to teach the process of quantifying behavior to their students. We're very excited about our new release that updates JWatcher to make it more compatible with current versions of Java, as well as adding numerous new and powerful features. Our philosophy is to provide a free event recorder and analysis program to the research community. We are developing a book "Analyzing behavior the JWatcher way" and are instituting a paid support program to help you get the most out of JWatcher. We hope that JWatcher helps you solve your behavioral analysis problems.

2 A Brief Overview of JWatcher™

JWatcher is a powerful tool for the quantitative analysis of behavior. It can be used to address any theoretical problem that requires a complex sequence of actions to be scored by a human observer. Thus, it can be used to study human or non-human behavior. JWatcher is written in the Java™ computer language (http://www.sun.com/java) and is therefore not dependent on a particular operating system. It will run on virtually any microcomputer capable of running a Java Virtual Machine™, and it has been tested on both Windows™ (http://www.microsoft.com) and Macintosh™ (http://www.apple.com) operating systems.

The simplest use of JWatcher is as an event recorder that logs the time at which keys are pressed. It was designed primarily for focal sampling with continuous recording (i.e., a single subject is observed continuously for a predetermined time). The real power of JWatcher is, however, in its analysis routines. These calculate time budgets and statistics about the duration of behavioral states, the intervals between them, and a variety of sequential analyses.

We developed JWatcher because existing event-recording programs typically force the user to make a restrictive series of decisions before beginning to observe behavior. JWatcher imposes few such constraints. It is based on a "score once, analyze many times" philosophy that we have adopted in our studies of animal behavior. For instance, a single data file can be used to calculate time budgets over both short and long periods, or to track changes in different subsets of behaviors seen. Each behavior can be analyzed both as an instantaneous event (measuring frequency), and as a state (measuring total duration). The logical relationships between behaviors can be edited in post-processing (i.e., a given behavioral state can be redefined as independent or as mutually exclusive with others). The number of states that can be tracked independently is limited only by the number of alphanumeric keys available. Users may choose to exclude or ignore certain behaviors in analyses, and categories can be combined to generate behavioral codes that were not originally scored. Batch mode analyses allow efficient processing. Results files can be easily combined for subsequent analysis in statistical programs. JWatcher can help you check inter- and intra-observer reliability. JWatcher allows you to conduct complex "conditional" analyses (i.e., to calculate the amount of time that individuals chew gum given that they are already walking). Sequential analysis algorithms fit a variety of Markovian and non-Markovian models to look for trends in the sequence of different activities. We are developing a module to allow data to be scored on a PDA capable of running the Java Operating System and then be imported into JWatcher for post-processing. And, we are developing new capabilities to permit QuickTime videos to be scored directly at different speeds.

JWatcher was designed to be used in association with common word processing, spreadsheet, and graphics software. It produces simple output files that can readily be imported by such programs. We have concentrated on developing new scoring/analysis algorithms, rather than duplicating functions in existing software.

2.1 Compatibility & Reliability

JWatcher has been comprehensively tested but there are still some quirks in the way that it behaves on different operating systems and under different JVMs (i.e., Java operating systems). Version 1.0 is compatible with the Java 2 Platform, Standard Edition (J2SE) and requires Java 1.4.2. You should only use this JVM to run JWatcher.

To our knowledge, the results are logically and numerically correct. If JWatcher behaves strangely, it is most likely a memory problem. JWatcher should be run on a machine with a minimum of 128mb free RAM. However, if you should experience memory problems, quitting and restarting the program (and quitting other open programs) should allow JWatcher to run more smoothly. JWatcher runs best on reasonably fast machines (i.e., >500 mHz).

We welcome bug reports and suggestions for new features via email to marmots@ucla.edu. We will attempt to fix bugs in future versions, but currently have no funding for future development.

Please note: This manual is the only free support available with JWatcher. We are in the process of writing a more elaborate book that will comprehensively work through examples using JWatcher, and we provide support and experimental design assistance for a fee (please contact janice@ucla.edu for details).

2.2 Installing JWatcher

You **must** have a current version of Java installed on your computer in order to run the installation program. To reduce confusion later, we suggest first downloading the appropriate JRE/MRJ.

•For Windows systems, go directly to Sun Microsystems (http://java.sun.com/j2se/1.4.2/index.jsp) and download the JRE 1.4.2.

•JWatcher 1.0 is compatible with Macintosh OS-X. For Macintosh OS-X systems, Java 1.4.2 is already installed.

Download the appropriate version of JWatcher from the JWatcher site **(http://www.jwatcher.ucla.edu)** for your operating system. At this stage, we are providing a Windows XP and a Macintosh OS-X version. Once downloaded, double-click on the installation icon. By default, JWatcher is installed into your Applications directory and will create an alias on your desktop. If you are running Windows, you may have to specify the JRE that JWatcher is to run with; be sure to select **JRE 1.4.2.**

2.3 Compatibility with JWatcher 0.9 Files

Data files scored with Version 0.9 are compatible for use with all applicable Version 1.0 routines; therefore, no rescoring of existing data files is necessary (but see exception below).

Global definition and focal master files created by Version 0.9 should be opened and resaved in Version 1.0, after verifying that no unintended alterations have occurred. Note that a number of key codes previously allowed in Version 0.9 global definition and focal master files are no longer able to be defined in Version 1.0 (see section 4 – Global Definition for more details - the six key codes are \ | : = , .). If your previous global definition and focal master files contain any of these codes, we suggest that you create entirely new files in the new version. Version 0.9 data files containing these codes will be able to be analyzed, but these codes will be ignored, and thus information may be lost. If your Version 0.9 data files intentionally contained any of these key codes then you will need either to edit them or rescore.

Focal analysis master files created by Version 0.9 must be updated within the new version before use to prevent unexpected results. Open, fix any inconsistencies, and re-save. All of your original selections should be preserved *except for one case within the Event Analysis window* (see below). Files not updated may still be analyzed, but we have not extensively tested these results.

Event Analysis window: In Version 0.9, two analyses (event count and interval occurrence) were combined into one check box. In the new version, these two analyses are now separated into two independent check boxes. If you had selected the "Event Count and Interval Occurrence" box in Version 0.9, you won't get the "Event Count" box checked in the new version. Therefore, you must update this portion manually.

Combinations master files created by Version 0.9 should not be used with Version 1.0. We suggest creating entirely new combination master files with Version 1.0, and rerunning the combine key codes routine. See section 9 – Combinations.

Version 1.0 files are not compatible for use with Version 0.9.

2.4 Legal MumboJumbo

The program and manual are copyright by the authors, but JWatcher is "freeware": this version is given away with no warranty expressed or implied. Feel free to use it, teach with it, and tell others about it. We request that you tell others to download the program directly from the JWatcher web site (http://www.jwatcher.ucla.edu) to ensure they obtain the most recent version of the program and the manual. The authors and their respective institutions are not responsible for its use or misuse.

JWatcher is a registered trademark of the authors. Java, the Java Virtual Machine, and Sun, are registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Windows XP and Internet Explorer are registered trademark of the Microsoft Corporation. Macintosh, OS-X, and QuickTime are registered trademarks of Apple Computer, Inc. Any other trademarks are the property of their respective owners.

2.5 Acknowledgements

1.0 Development: For programming, we thank Nada and Jose da Viega of Convolution L.L.C. (www.convolution.ws). 1.0 development was generously supported by the U.S. National Institute of Health grant 5R21MH065226. We thank the UCLA Life Science Computing staff for help distributing JWatcher and for a number of 0.9 users for suggesting additional features.

0.9 Development: For programming, we thank Fiona Walkerden, Xuhong Li and Derek Renouf of Adaptive Arts Pty Ltd (www.adaptive-arts.com), Linda Evans for creating the JWatcher icon, and members of the Macquarie University Animal Behavior Lab for help with detailed testing. Initial JWatcher development was supported by a Macquarie University Research Infrastructure Block Grant. During initial development, Dan Blumstein was supported by an Australian Research Council Postdoctoral Fellowship, and Chris Evans by an Australian Research Council Large Grant.

3 Using JWatcher

To quantify behavior you must create several types of files. The following description is an outline only to illustrate the overall structure and logical relationships (Figures 1-3). Detailed instructions are provided in the subsequent sections, and also in a series of on-line help pages.

Note: The on-line help requires your computer to have a web browser. If the help file does not load properly, try changing your default web browser to Microsoft Internet Explorer^{TM}.

First, you must create a Global Definition File (.gdf). Global definition files are a list of all the behaviors that you might possibly be interested in scoring. The .gdf file could be regarded as an ethogram–a catalog of behavior. Additionally, you can specify modifiers that may follow certain behaviors. Modifiers can be anything that has a logical relationship with a behavior (e.g., the identity of the other individual in a dyadic social interaction, the location where a behavior was performed, or a variable component of a display). The total number of behaviors and modifiers in a .gdf file cannot exceed the number of alphanumeric characters on a keyboard.

When asking a focused behavioral question, however, you are typically interested in a sub-set of behaviors and modifiers. You must specify the duration of a focal observation, and any question with which you are to be prompted before beginning a focal observation (e.g., subject identity, date, time, etc.).

To do this, you will have to create a Focal Master File (.fmf). The focal master file first inherits a pre-existing global definition file (.gdf) to which you can add or subtract behaviors or modifiers. Additionally, the .fmf requires you to specify questions to be prompted with prior to scoring behavior, to specify a focal duration, to choose whether you want the event-recorder's clock to count up or count down, and to choose whether the end-of-file sound should be played when scoring behavior.

To score behavior, you must specify a focal master file (.fmf). The Data Capture routine allows you to score behavior and then creates a data file (.dat) which contains a log of the key presses scored to the millisecond.

For basic analysis (Figure 1), before analyzing a data file (a .dat file), you must create a Focal Analysis Master File (a .faf file). The focal analysis master file is the "guts" of JWatcher and specifies which behaviors are to be treated as states and which are to be treated as events. It also allows you to "time window" a single longer focal observation and summarize behavior in each time window. It allows you to specify which behaviors are mutually exclusive, to exclude or modify certain behaviors, and a variety of other specific functions. Once a .faf file is created, JWatcher can analyze a single data file or it can batch process all data files contained in a folder. JWatcher results files (.res) are comma delimited text files that are opened easily in spreadsheet programs for subsequent post-processing and analysis. (See section 10.1 for difference between cd.res and tr.res files.)

Sometimes, you may wish to score behavior in some detail and later combine detailed key codes into a single behavior type. For instance, a subject may "look" in a variety of postures—it can stand and look, crouch and look, or rear-up and look. JWatcher contains a routine to combine these different key codes into a single key code–look. To do so, a Combinations Master File (.cmf) must be created. Once a .cmf is created, JWatcher can combine the codes in either a single file, or it can batch process many files. The files resulting from combination (c.dat files) are analyzed by a modified .fmf and a new .faf. Both of these files are based on the new key codes defined in the .cmf.

Once you have analyzed a number of focal data files, you will probably want to combine your results files into one summary file and calculate some summary statistics. JWatcher's Summarize Results algorithms will help you with this. Importantly, this will also create a file that can be exported into other spreadsheet or statistical analysis programs for subsequent analyses. Note that only .cd.res files are able to be summarized.

One more important tool allows you to calculate inter- or intra-observer reliability (Figure 2). JWatcher's reliability routine enables you to compare the sequence of two data files, and is located in the Analysis menu. Reliability result files (rel.res) cannot be summarized with the Summarize Results routine.

For sequential analysis (Figure 3), you must first create a global definition file (.gdf), a focal master file (.fmf), and then score data files (.dat) with the data capture routine. Sequential analysis functions independently from basic analysis, and does not require a focal analysis master file (.faf). The General tab enables you to specify which behaviors are to be modified or ignored, and several other settings. Unlike basic analysis, sequential analysis assumes that all key codes represent a simple sequence, and does not take into account the times that these codes were logged. There can be no simultaneously occurring behaviors for this type of analysis, so data files must be scored accordingly. Note that the General tab specifications apply to all Sequential Analysis tabs except for the Runs Test. The Runs Test examines whether a sequence of dichotomous behaviors is random, and requires that data files are comprised of no more than two behaviors. Sequential analysis results files (.run.res, .lag.res, .seq.res, .mar.res, .com.res) cannot be summarized with the Summarize Results routine.

Figure 1. A taxonomy of JWatcher files and their relationships for basic analysis.

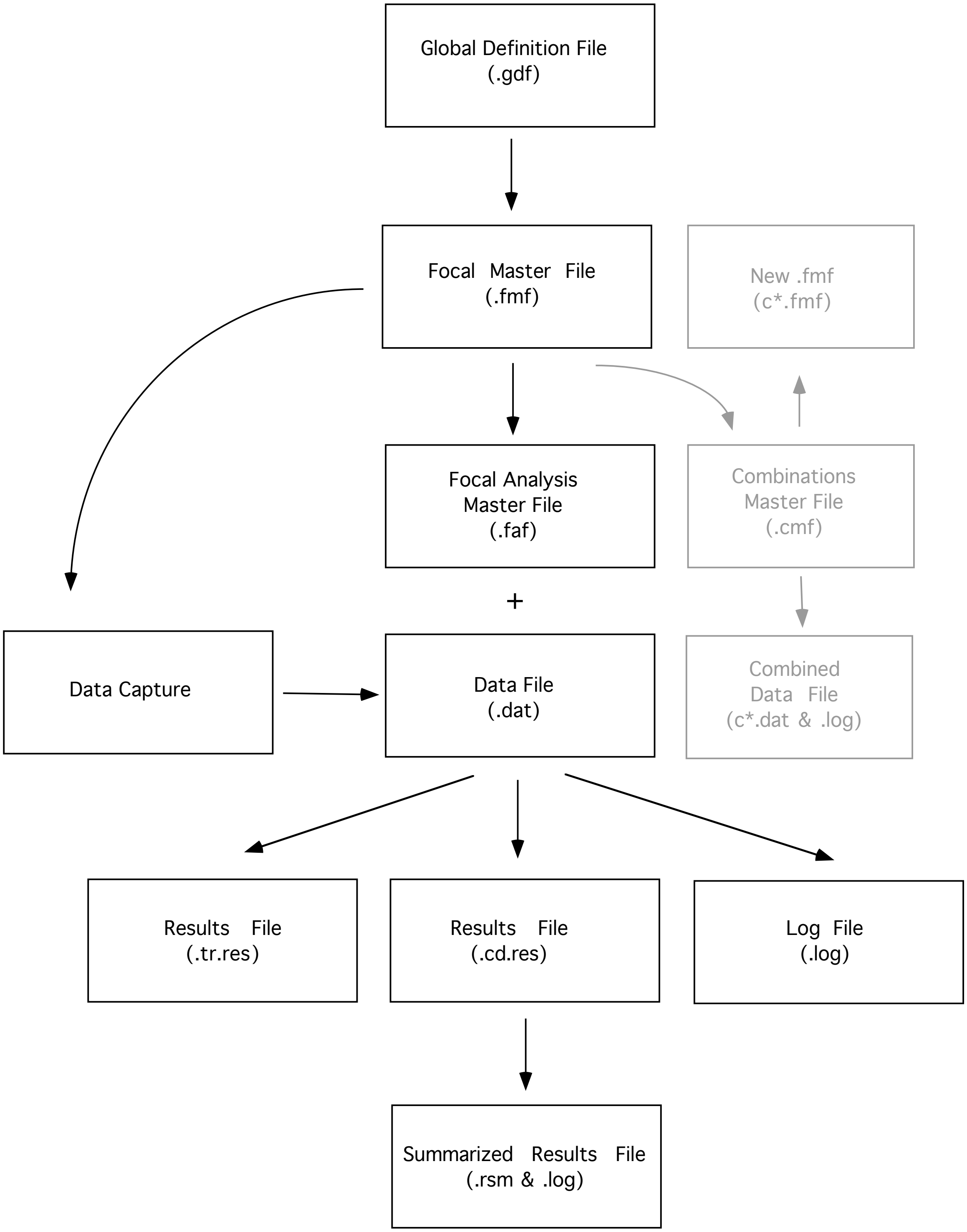


Figure 2. A taxonomy of JWatcher files and their relationships for reliability analysis.

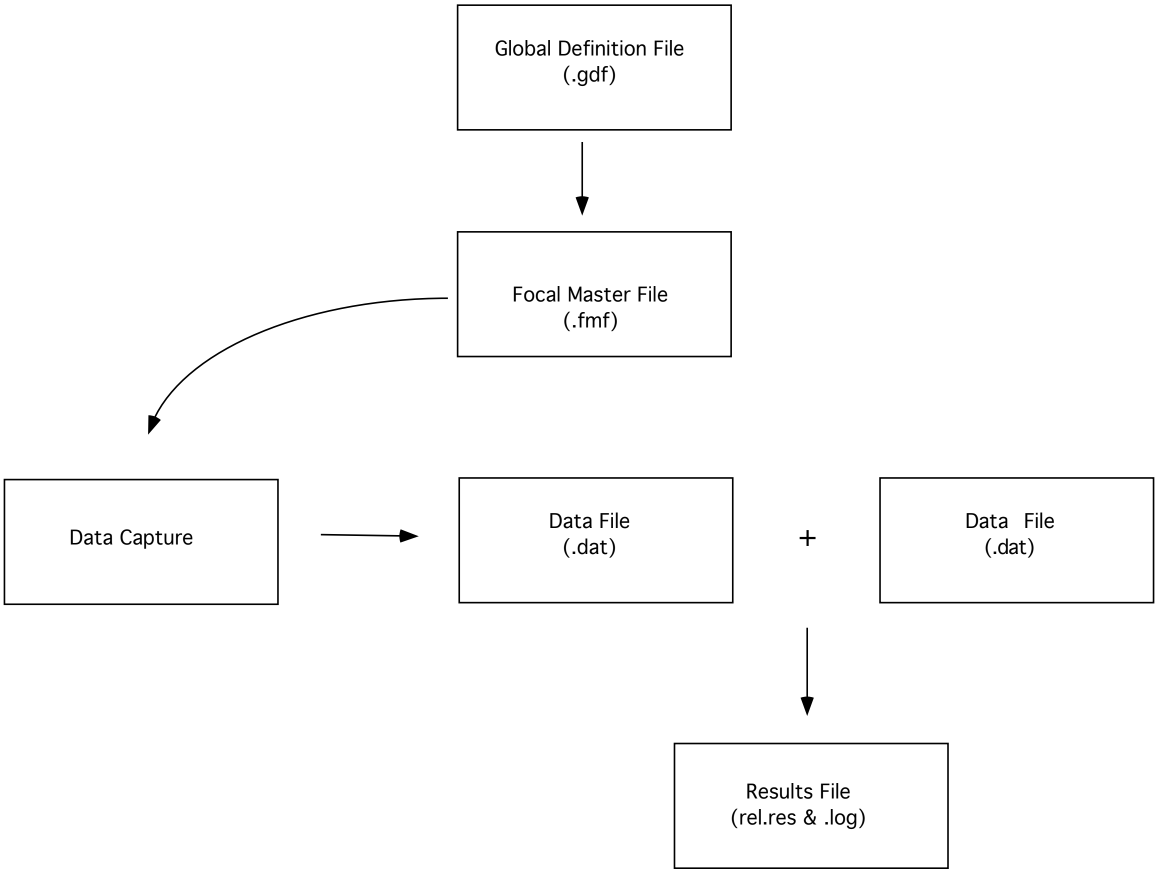
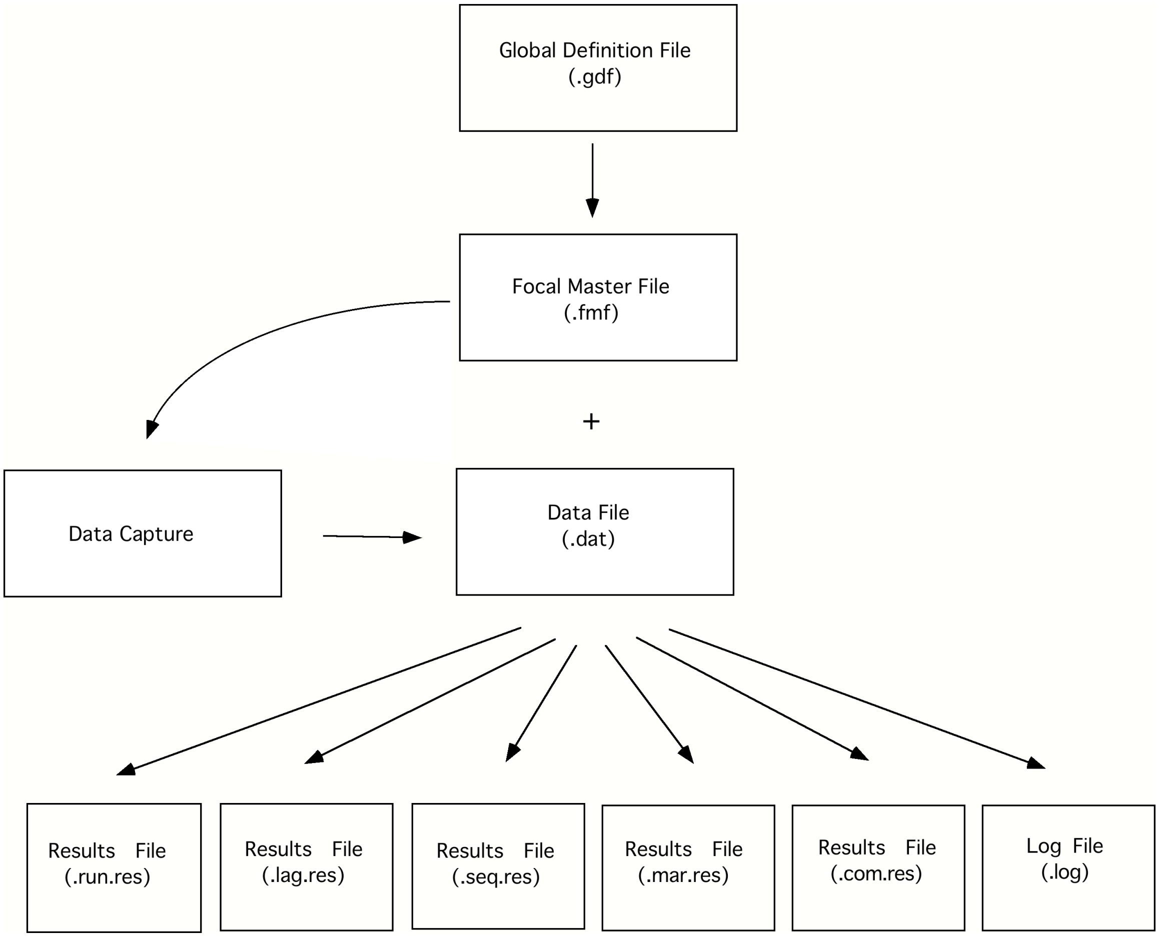


Figure 3. A taxonomy of JWatcher files and their relationships for sequential analysis.



4 Global Definition

Global Definition is a comprehensive ethogram you will create for a given species or project. An ethogram is a catalog of behaviors and modifiers specified, typically, at the beginning of a study.

A global definition file (.gdf) is the first step towards creating a focal master file (.fmf).

First, click [New] to create a new .gdf or [Open] to modify an existing one.

Behaviors
Use the [Add row] button to add rows. Specify a single alphanumeric key code for each behavior, define it, and describe it.

Note: JWatcher is case-sensitive: upper and lower case key codes are treated separately. Numbers and most punctuation key codes are permissible, but the space bar, backslash (\), colon (:), period (.), equals (=), given (|), comma (,), return, tab, delete and other function keys cannot be defined.

Use the [Delete row] button to remove rows.

Modifiers
Use the [Add row] button to add rows. Specify a single alphanumeric key code for each modifier, define it, and describe it.

Use the [Delete row] button to remove rows.

Type in descriptive supplementary information in the bottom text box. Do not use the return key.

Use the [Save] button to save your .gdf file when complete.

Note that key codes may be entered either as behaviors or modifiers, but not both.

For a description on how modified behaviors are treated during analysis, see section 8 (basic analysis), section 10.2 (reliability analysis), and section 12 (sequential analysis).

5 Focal Master

Focal Master creates a set of specifications (the .fmf file) used when recording data using Data Capture. It defines a set of questions the user will be asked prior to capturing the data. The answers to these questions (e.g., identity of focal subject, date, time of day, habitat, etc.) will be added to the header of the resulting data file (.dat), and will also be reported in any results files generated when analyzing the data file. The focal master file also contains the duration of a focal observation. It is best used AFTER creating a global definition file, although this is not required. A .fmf file is also a pre-requisite for analyzing data as it is required to create a "focal analysis master file" (.faf) and, if needed, to later create "combination master files" (.cmf).

5.1 Focal Master > Define Codes

Use the [New] button to open a previously created .gdf file and import the existing behaviors and modifiers. Use [Open] to open an already existing .fmf. Clicking either button will take you into a navigation window.

If you have imported a .gdf file, there should be a series of already defined behaviors and modifiers. Using the [Add row] or [Delete row] buttons, edit this list to include just those behaviors and modifiers you are interested in quantifying for your current project.

Use [Save] to save the file with the existing name and [Save As] to save the file with a new name. The [Save] and [Save As] buttons are only selectable if you have made any changes.

Note: Making changes in the .fmf will not modify the .gdf from which it was created.

When all behaviors and modifiers have been defined, click on Test Details to move to the next window.

5.2 Focal Master > Test Details

Define the observation session in hours, minutes, and seconds. Note, it is sufficient to fill in only one box (e.g., "1" in the minutes box will create a 60 second focal). If no session duration is specified, data capture will continue until stopped.

Select "count up" or "count down" to specify whether you want the timer in Data Capture to be increasing to the time limit, or decreasing towards 0.

Select "sound on" or "sound off" to specify whether you would like JWatcher to play a sound signifying the end of a focal session.

Add any notes to the text box.

5.3 Focal Master > Questions

Define up to 6 questions the user will be prompted with prior to capturing data. Do not use the return key. For example, you might want to note the identity of the focal subject, the time of day, the weather, etc.

6 Data Capture

Data Capture is JWatcher's event-recording program. Use this to record the onset of behavioral states (e.g., the onset of a bout of foraging) or the occurrence of behavioral events (e.g., the occurrence of an alarm call). Data capture logs any and all definable key presses, whether or not they are defined in the .fmf file.

6.1 Data Capture > Data File

In this window, you will have to both name your data file and select an .fmf used for data capture.

In data file field, select the path where the data file will be stored by clicking on the file icon. If you do not select a pathway, the data file will be stored in the default location—in the Application folder where JWatcher is installed. Selecting the file icon will open a navigation window. Using the pull-down directory in the navigation window, specify the location where you want to store the data file. Name the data file and click the [Open] box. If you select an existing .dat file you will overwrite it.

In the focal master field, select a focal master file (.fmf) that you want to use while scoring behavior by clicking on the file icon.

WARNING: Data files scored without an associated .fmf file cannot be analyzed with Sequential Analysis. Therefore, you must specify a .fmf file in this case. For basic analysis, however, scoring with a .fmf file is not an absolute requirement.

When you have named your data file and selected a .fmf file, click [Next].

Note: The first time you open JWatcher after a shut-down it will automatically revert to the default pathway. Once you select a path, JWatcher will save that path (until changed) to facilitate subsequent scoring sessions. However, this path will be lost when JWatcher is closed.

6.2 Data Capture > Questions and Answers

In this window, answer the questions that have been specified in the .fmf file. Do not use the return key.

When finished, click [Next].

Answers should be saved from one data file to next (but this is not always the case).

6.3 Data Capture > Capture Data

Using the mouse, click [Start] to begin scoring behavior (or depress alt-s). The session will either time out at the end, or you can click [Stop] to end an observation session before the pre-determined time (or depress alt-t or tab-space bar).

Alternatively, you can move the cursor around using the tab key. Ensure that the "focus" is on one of the duration boxes (i.e., there is a blinking cursor) and tab until the [Start] button is highlighted. Depressing the space bar will now start the scoring process.

To temporarily pause during an observation session, click [Pause] or depress the space bar (or alt-p). The clock will stop running until data capture is resumed. To resume, either click [Pause] again or depress space bar (or alt-p). Stopping an observation session while data capture is paused will terminate the session at the time of the last pause.

If no session duration is specified, the data can be recorded until stopped.

You can select whether you want the clock to count up to the pre-determined focal duration or count down. You can also turn the sound signifying the end of the focal on or off.

The View [Behaviors] button places a list of behaviors and their key codes into a separate window. Doing so may facilitate scoring.

The View [Modifiers] button places a list of modifiers and their key codes into a separate window.

When finished, the file is automatically saved to the location specified. Click [New] to score the next focal session.

6.4 Note on "initializing" a focal

Some event recorders require the user to initialize behavioral states before scoring behavior. We elected to allow the user to begin capturing data immediately. Therefore, to initialize a behavioral state, the user has two options:

1) Quickly hit a key after starting JWatcher. The time delay between starting JWatcher and hitting the first key may be unimportant if the focal duration is relatively long.

2) In cases where the time delay may influence later analyses, use the Edit Data function (see section 7) to edit the data set to begin at time = 0.

7 Edit Data

Edit Data allows you to modify the keystrokes and times created in Data Capture. Answers entered during Data Capture to focal master file questions may also be modified here.

Use the [Open] button to open a data file (data files have the .dat suffix).

Use the [Insert Row] button to add rows, the [Delete Row] button to delete rows, and the [Save] or [Save As] buttons to save changes.

Notes: Type or edit the time in either the "Elapsed" column or the "hh:mm:ss:ms" column. Changes to one column will automatically update in the other. If the .fmf is either in the same directory as the data file, or is in the same location as it was when the data file was scored, then typing a recognized behavior or modifier key code in the key pressed column will result in the appropriate behavior or modifier being typed in the "Behavior/*Modifier" column. Ensure that new entries are chronological. Non-chronological entries will be treated as errors during analysis and no results files will be produced. When modifying answers in the Answers window, do not use the return key.

8 Focal Analysis Master

Focal Analysis Master creates a focal analysis master file (.faf) which specifies how data files (.dat) will be analyzed and specifies the results to be reported.

JWatcher's "score once, analyze many times" philosophy is embedded in the focal analysis master file. A single data file can be analyzed differently by changing definitions in a focal analysis master file. You may wish to modify time bins, subtract or ignore certain behaviors, re-define mutually-exclusive behaviors, modify how incomplete time bins are handled, or explore the differences between state and event analyses.

However, **logical errors when selecting non-default options can generate meaningless results.** Be VERY careful setting up focal analysis master files. We suggest paying particular attention to the logical relationships specified in the States tab. This is particularly important when analyzing conditional states or conditional events.

Use [New] button to select a focal master file (.fmf file) that contains the key codes to be analyzed.

Use [Open] button to open a pre-existing focal analysis master file (.faf file).

Note: Ensure that you use the same .fmf to score behavior and to analyze behavior. The analysis routine will later check whether the same .fmf was used to capture data and to create the .faf for analysis. If not, JWatcher will inform you that different .fmf files were used. However, analysis will proceed even if the .fmf files do not match.

Once you have selected analysis options, save changes using [Save], or create a new file name using [Save As].

NOTE: **the first time that you save a .faf file after the program is opened may take a very long time**. Simply wait and the file will eventually be saved. This problem should not occur for subsequent saves. It is due to the .faf validation process described below, and is more likely to occur when using a computer with limited RAM.

Every time that you save a faf, the program will check for logical inconsistencies among your selections made within the file. For example, out-of-sight may not be subtracted lest meaningless statistics be generated. If such inconsistencies exist, a pop-up window will list the problems and propose solutions. You may either accept the proposed changes or cancel the save.

Known issue on PC: You might be asked if you want to save changes, even when no changes were made to your file.

[Summary] opens a text window containing all the specifications in the current focal analysis master file (.faf). Select the text using a mouse, and "control+c" to copy the contents of the text window. Contents can now be pasted into a word processor and printed.

The section under the heading "warnings" in the summary will list some potential problems that may arise during analysis.

8.1 Focal Analysis Master > Modifiers

Specify behaviors that may have modifiers following them. For instance, aggression ("a") may be directed to different individuals (e.g., "1", "2", or "3"). The default is that no modifiers follow any behaviors.

To view the list of modifiers associated with this file, click on the [Summary] button. Or, open your focal master file in the Focal Master tab.

Note: Only modifier key codes that immediately follow a behavior specified as having modifiers will be reported in the results files. Any modifier key codes that do not follow a modified behavior will be recorded in the .dat file, but will be ignored during analysis.

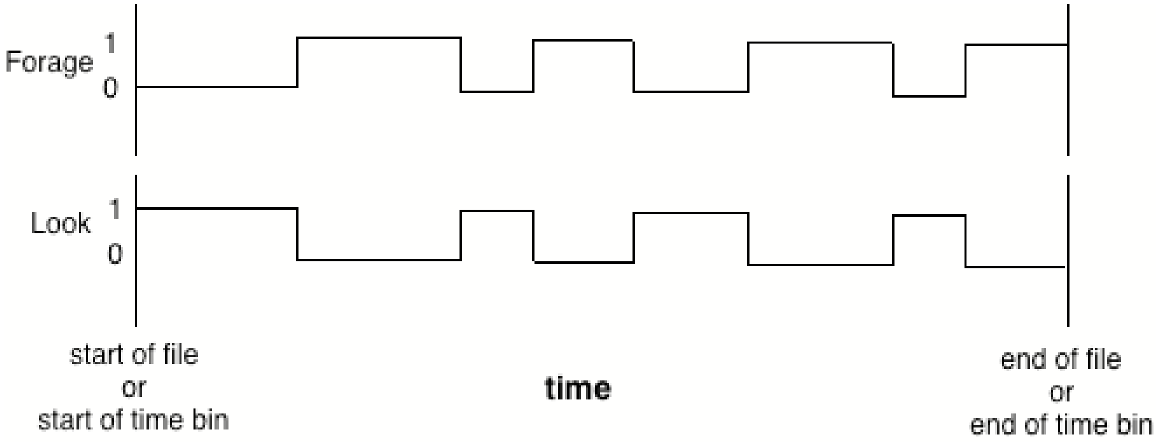
When modifiers are specified to follow a behavior, results will be reported both for each behavior/modifier combination separately and for the behavior in its entirety. For example, if count is requested in the above example, a separate count would be reported for each of "a1", "a2", and "a3". The count for "a" would also be reported, where count "a" = count "a1" + count "a2" + count "a3" + count "a"(no modifier follows).

8.2 Focal Analysis Master > States

States are behaviors that occur over time. For instance, a subject may forage for 2 minutes. The onset of the bout of foraging could be noted by hitting "f" in data capture. Some behaviors may be mutually-incompatible with foraging (e.g., "l"–locomotion), whereas others may not be (e.g., "c"–chewing). Mutually incompatible behaviors signal the termination of a state and are used to calculate durations of states. Thus, a bout of foraging begins when "f" is hit and ends when a mutually-incompatible behavior key code is hit–in this case "l". This window allows the user to specify precisely which behaviors are mutually-exclusive with each other.

The default is that all behaviors are mutually exclusive; the onset of one behavior turns off another behavior (Figure 4). Various combinations of mutually and non-mutually exclusive codes may be created by selecting and deselecting boxes within the matrix. However, codes are always mutually exclusive with themselves.

Figure 4. Traces of two mutually-exclusive behaviors.



It is possible, and in some cases desirable, to hit the same key successively (Figure 5). For instance, if you are interested in studying bouts of vigilance behavior, a bout begins every time a subject moves its head. In this case, because codes are mutually exclusive with themselves, the duration of each bout is measured as the time in between each key stoke. The interval between bouts is 0. The number of events of vigilance is the number (N) of keystrokes. The average duration of each bout of vigilance is the total duration divided by N-1 keystrokes.

Figure 5. Details about event and state analyses.

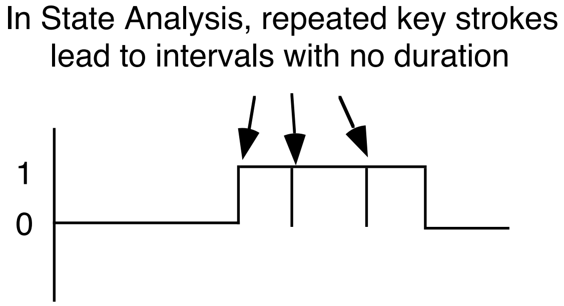
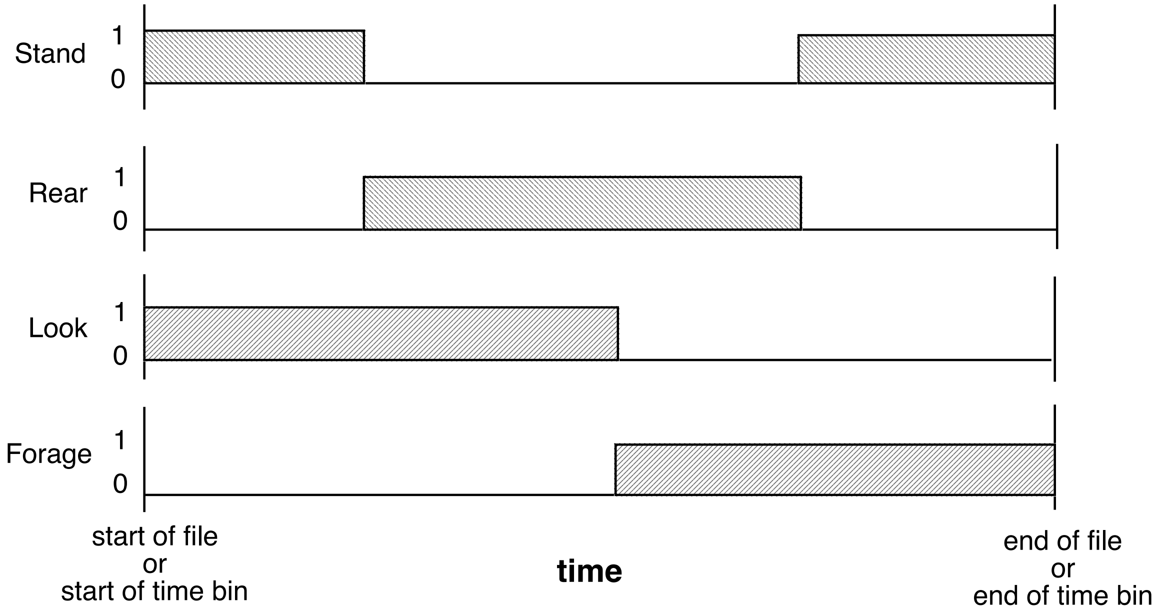

In State Analysis, repeated key strokes
lead to intervals with no duration
1
0

Figure 6. Traces of two sets of mutually exclusive behaviors that are non-mutually exclusive with each other.



In this example (Figure 6), stand and rear are mutually exclusive, and look and forage are mutually exclusive, but stand and rear may occur simultaneously with look or forage. Conditional state analysis may be used to generate statistics such as the conditional proportion of time (for example, the time that a subject is standing given that it is looking), or the number of bouts, total time, average bout duration, standard deviation of bouts, or proportion of time that two or more non-mutually exclusive states are simultaneously on (for example, the time that the subject is simultaneously standing and looking).

8.3 Focal Analysis Master > Exclusions

The "subtract" function subtracts the total time for a specified state from the total duration of a time bin when calculating time budgets. For example, you might wish to calculate the time allocated to running as a function of total locomotion. In this case behaviors other than those involved in locomotion could be subtracted. The proportion of time and proportion of time in sight statistics reported for each behavior in the results will be adjusted accordingly.

Subtract only influences duration statistics, but not interval statistics.

Subtract does not affect the logical relationships with other behaviors. Time is removed, but subtracted behaviors still terminate bouts of mutually-exclusive behaviors.

Note: Subtracting a behavior will generate different results than specifying that same behavior code as representing "out-of-sight" (see 8.5 for more information on out-of-sight and its effects). Why? Scoring out-of-sight means that you are unsure of what is going on. This has consequences for both interval and duration statistics. In contrast, when you subtract a behavior you know what has occurred; subtraction simply removes that time from the denominator when calculating time budgets.

"Ignore" literally ignores selected behaviors when calculating states. Ignored behaviors, therefore, do not turn off mutually exclusive behaviors; it is as if the key code was never pressed.

Note: This can potentially cause nonsensical output if some behaviors in a related set are ignored, while others are retained for analysis.

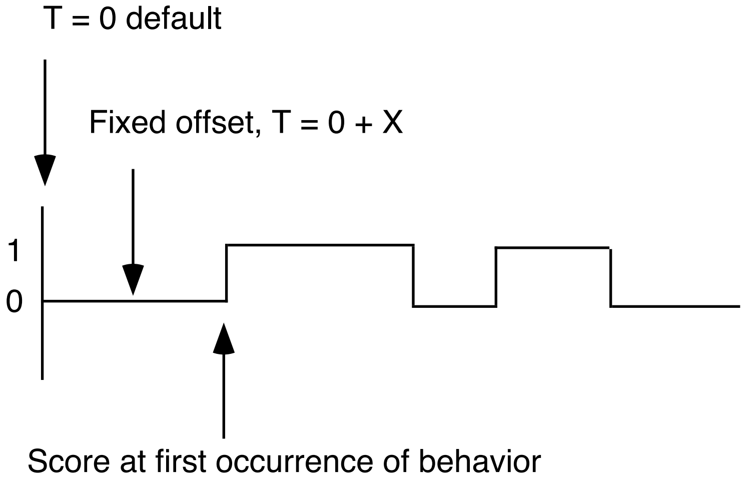
Ignore has precedence over subtract. The default is that no behaviors are either subtracted or ignored. Ignored behaviors are not reported in the results, while subtracted behaviors are. For subtracted behaviors only, the proportion of time and the proportion of time in sight statistics are calculated as the total time for the subtracted behavior divided by the total duration of the time bin. The total time for the subtracted behavior is not subtracted from the denominator in this case.

8.4 Focal Analysis Master > Time

Decisions made in this window influence the time over which behavior is analyzed. The default setting is for the entire data file (.dat file) to be analyzed. However, there are cases where users may wish to modify this in several ways.

Users may specify at what point to begin analyzing a file (Figure 7). Deselecting the "score from beginning of session" box allows modifications.

Figure 7. Score from beginning variations.



Fixed offset specifies the number of seconds from the beginning of the data file that should be ignored in analysis. You may wish to use a fixed offset to get around the problem of the time lag between when the start button is pressed and the first keystroke. Note, however, ALL key codes hit before this fixed offset will be ignored.

First occurrence defines which, if any, behavior will trigger the onset of analysis. When a specific code has been designated, all key codes hit before this code will be ignored. You may also analyze from the onset of any key code, regardless of its identity. Choose the <any> option from the pull down menu. This is another potential solution to the time lag problem discussed above.

Note: Using the <any> option will begin analysis at the first recorded keystroke regardless of whether it is defined in your .fmf, and regardless of whether it has been "ignored". In the latter case, a warning will be given during analysis, but analysis will proceed.

Time bin duration specifies how the data file (.dat file) is subdivided for analysis. For instance, you may wish to score a 5-minute focal sample but summarize behavior in 1-minute blocks of time. Specifying the time bin duration in 60 second blocks will take a 5-minute focal and generate 5 different 60-second blocks of results.

Note: "0" time bin duration is the default and specifies that the entire focal will be analyzed in one time bin.

Various combinations of time bin duration and offset times may lead to the case where the last time bin is truncated (i.e., it is shorter than previous time bins). Choose "End with last complete time bin" to only report those results for complete time bins or "Report all" to report those results for all time bins.

8.5 Focal Analysis Master > State Analysis

For each behavioral state, JWatcher calculates the number of bouts, the total time summed over all bouts, the average bout duration, the standard deviation of bout duration, and the proportion of time a focal subject engages in a given behavior calculated over the entire time bin.

If modifiers follow a behavior, the above statistics will be reported both for each behavior/modifier combination separately and also for that behavior globally.

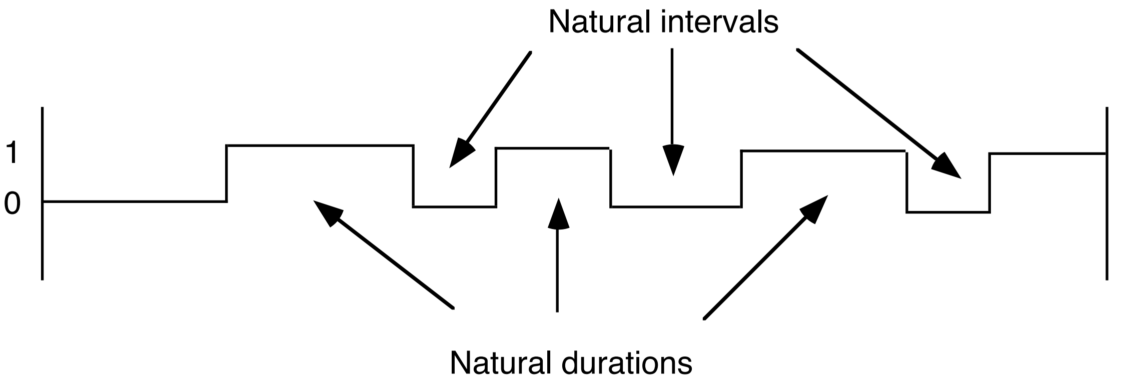
Note: The total observation period is derived from EOF ("end of focal") time in each .dat file and not the focal duration specified in the .fmf.

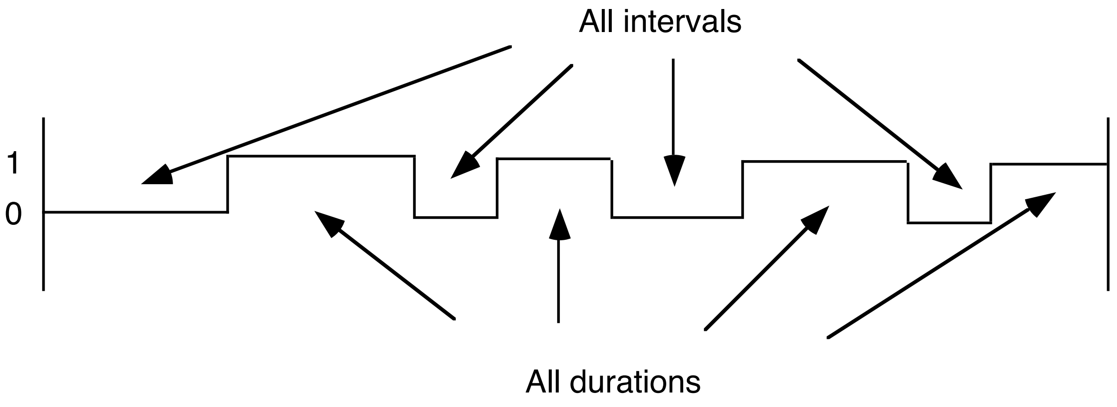
Subjects are not always in view. Users may wish to specify a key code used when the subject is out-of-sight (e.g., "o"). The proportion of time allocated to each behavior as a function of time in sight can also be calculated.

A behavioral state has a duration, and there is an interval of time between bouts of behavior. These durations and intervals may be either "natural" (i.e., they start and end within a time bin), or they may be truncated by either a time bin border or the end of file (Figure 8). Selecting "natural" interval and "natural" duration reports statistics only on those intervals and state durations not truncated. Selecting "all" interval and "all" duration reports statistics on all occurrences of interval and state durations regardless of whether they are truncated.

Note: The onset and termination of out-of-sight truncates intervals and durations. Scoring out-of-sight implies that you are unsure as to what behavior(s) the focal subject is engaged. If you are interested in natural durations or natural intervals, any bout of behavior adjacent to a bout of out-of-sight is, by definition, truncated.

Figure 8. Interval and duration definitions.

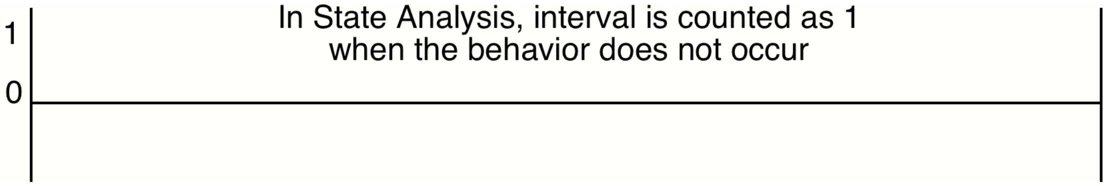




Note: For durations only, the beginning of the focal does not truncate a duration. However, the beginning of subsequent time bins when time windowing is employed would do so. Thus, if you are interested in natural durations, we suggest that you begin your focal at the true onset of a behavior.

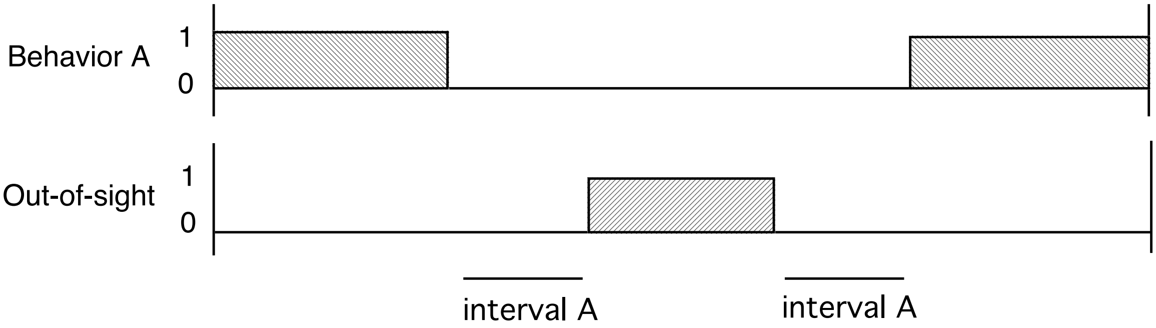
If a behavior does not occur during a time bin, the interval is considered to occur once (i.e., N = 1), with a total duration equal to the time bin (Figure 9).

Figure 9. Details about State Analysis.


1
0
In State Analysis, interval is counted as 1 when the behavior does not occur

Note: The above applies for "all" interval analysis. For "natural" interval analysis, the interval count would be 0 because the interval is truncated by both the beginning and end of the time bin.

Figure 10. Further details about State Analysis: how out-of-sight affects interval analysis



For "all" interval analysis, the count for interval A would be 2. Out-of-sight divides the interval into two because we don't know that behavior A did not occur during this time. For "natural" interval analysis, the count for interval A would be 0. In this case, out-of-sight truncates both intervals on either side of it, and so these intervals are excluded from analysis.

Conditional proportion of time generates statistics for up to four streams of simultaneously occurring behaviors. To generate these statistics, the specific behavioral combinations to be analyzed must also be specified in the Conditional States tab. For example, if B|A (B given A) is specified for analysis, then the conditional proportion of time will be reported as the following: given that a subject is engaged in behavior A, what proportion of time is it simultaneously engaged in behavior B, where A and B are non-mutually exclusive states? The number of bouts, total time, average bout duration, standard deviation of bouts, proportion of time, and proportion of time in sight will be reported for bouts of A and B occurring simultaneously, when those statistics are also selected above.

Conditional proportion of time may be calculated for "natural" or "all" durations, and also for "natural" or "all" intervals.

Figure 11. "All" duration conditional proportion of time for Stand given Look, where Stand and Look are non-mutually exclusive behaviors.

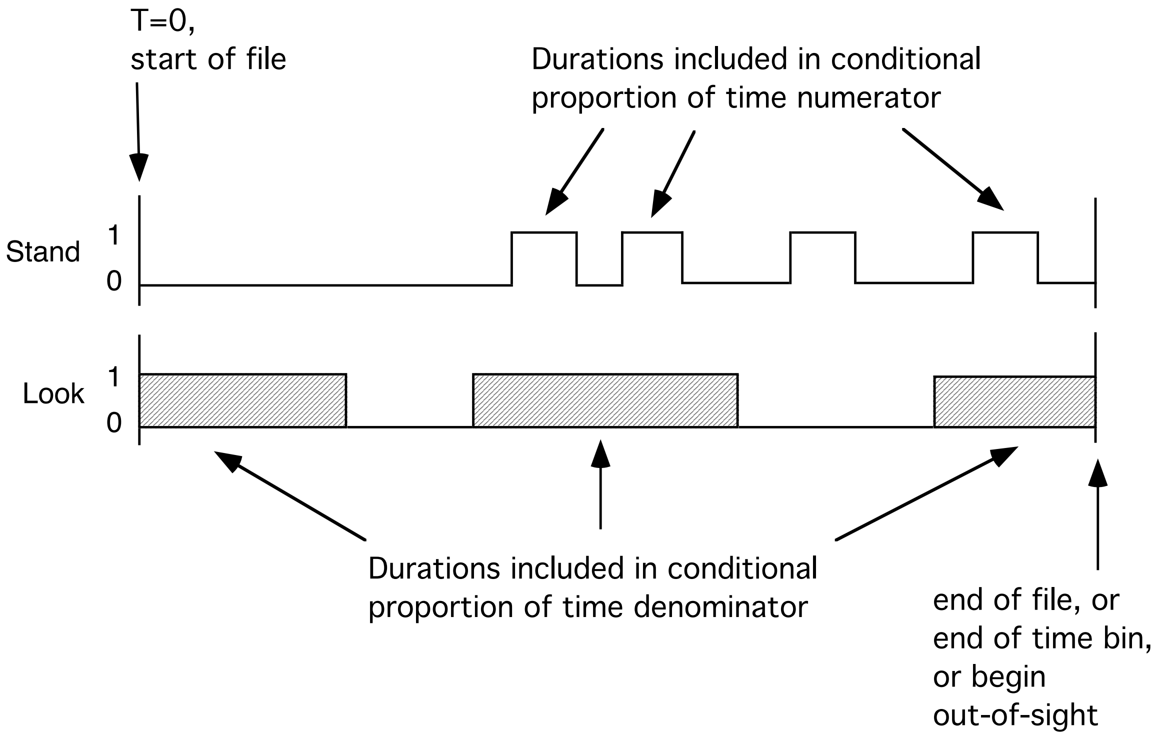
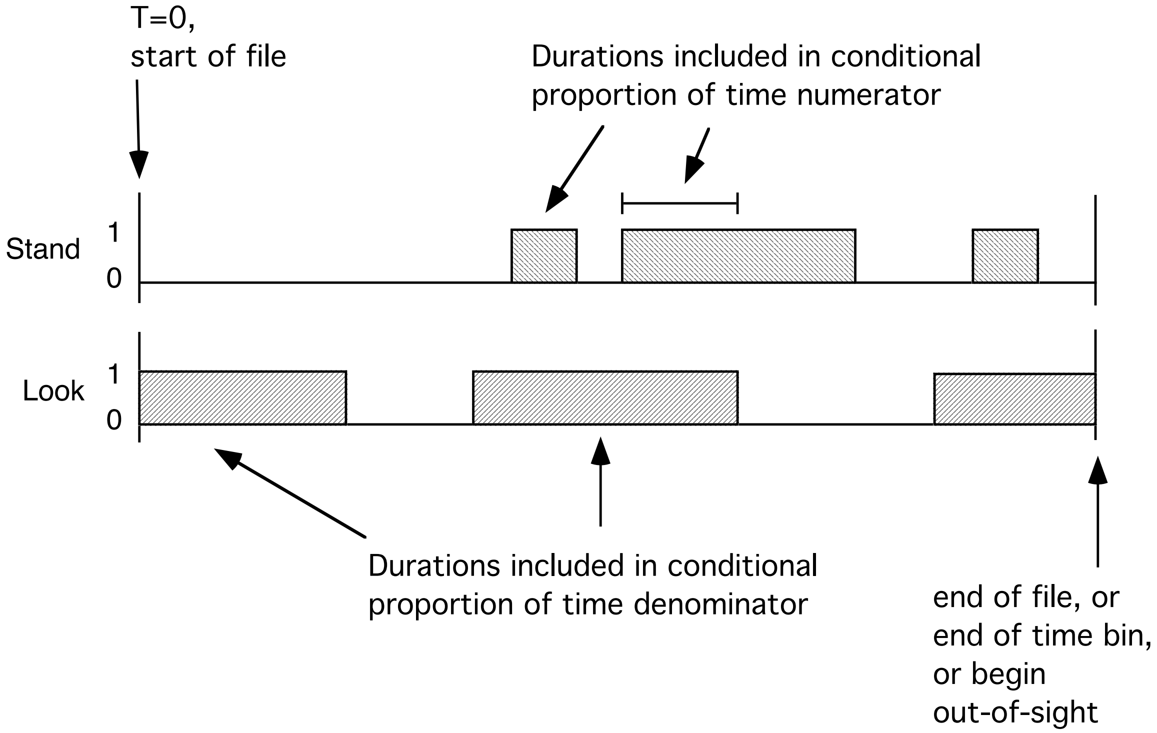

T=0,
start of file
Durations included in conditional proportion of time numerator
Stand
1
0
Look
1
0
Durations included in conditional proportion of time denominator
end of file, or end of time bin, or begin out-of-sight

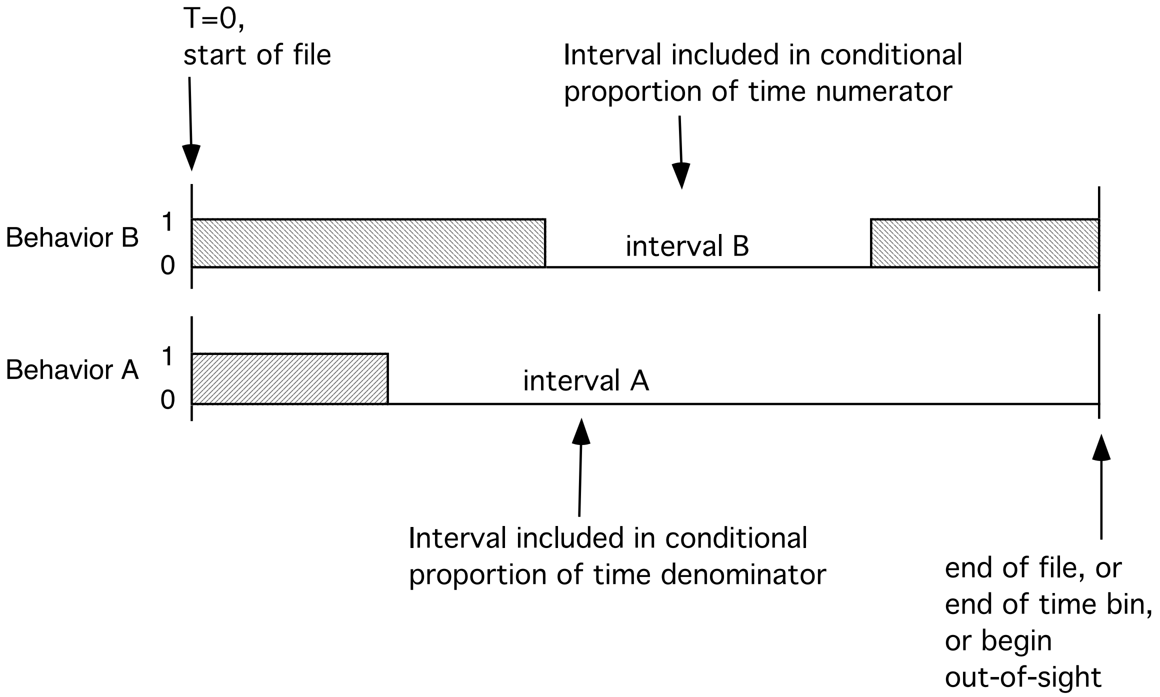
Figure 12. "Natural" duration conditional proportion of time for Stand given Look, where Stand and Look are non-mutually exclusive behaviors.


T=0,
start of file
Durations included in conditional
proportion of time numerator
Stand
1
0
Look
1
0
Durations included in conditional
proportion of time denominator
end of file, or
end of time bin,
or begin
out-of-sight

Note that the beginning of the focal (T=0) does not truncate state durations, but the end of the focal, the beginning and end of a time bin, and the beginning and end of out-of-sight do truncate state durations.

If B|A (B given A) is specified for interval analysis, the conditional proportion of time will report the proportion of time that a subject is not engaged in B, given that it is not engaged in A (i.e. the proportion of time that interval B occurs, given that interval A is occurring).

Figure 13. "All" interval conditional proportion of time for B|A, where B and A are non-mutually exclusive behaviors.


T=0,
start of file
Interval included in conditional proportion of time numerator
Behavior B
1
0
interval B
Behavior A
1
0
interval A
Interval included in conditional proportion of time denominator
end of file, or end of time bin, or begin out-of-sight

Note that for "natural" interval analysis, the conditional proportion of time = 0 in the above example, because interval A is truncated by the end of focal (or end of time bin or beginning of out-of-sight). Unlike durations, intervals are truncated by the beginning of a focal (T=0) as well.

See Conditional States tab (section 8.6) for more details.

8.6 Focal Analysis Master > Conditional States

Use this window to specify the combinations of behaviors to be analyzed as simultaneous streams. Up to four streams of simultaneously occurring behaviors may be analyzed.

Conditional states analysis can only be performed when at least two behaviors are non-mutually exclusive (or at least one behavior is modified - see special case below). The relationships between behaviors are defined in the States tab.

The statistics to be reported are specified in the State Analysis tab. The number of bouts, total time, average bout duration, standard deviation of bouts, proportion of time, and proportion of time in sight can be reported for bouts of up to four simultaneously occurring behaviors. For example, if A|B is specified, then the above statistics will be reported for all bouts of A and B simultaneously on. If A|B|C is specified, then statistics will be generated for all bouts of A, B and C simultaneously on. Likewise, if A|B|C|D is specified, then calculations will be based upon all bouts of A, B, C and D simultaneously on.

In addition, the conditional proportion of time can be reported for behaviors A|B (A given B), A|B|C (A given B given C), or A|B|C|D (A given B given C given D). For A|B, this statistic calculates the proportion of time that A is on, given that B is also on.

When A|B|C is specified for analysis, two separate conditional proportion of time statistics will be automatically generated: A|BC calculates the proportion of time that A is on, given that B and C are on simultaneously, and AB|C calculates the proportion of time that A and B are on simultaneously, given that C is on.

Similarly, for A|B|C|D, four such statistics are automatically generated: A|BCD calculates the proportion of time that A is on, given that B, C and D are simultaneously on. AB|CD calculates the proportion of time that A and B are simultaneously on, given that C and D are simultaneously on. AC|BD calculates the proportion of time that A and C are simultaneously on, given that B and D are simultaneously on. And, ABC|D calculates the proportion of time that A, B and C are simultaneously on, given that D is on.

The relevant conditional proportion of time statistic(s) will depend upon the question of interest.

1) Click [Add row] to begin.

2) Click [Conditional State] to select a pair of behaviors for conditional analysis (e.g A|B).

3) To add a third behavior (e.g to specify A|B|C), place cursor back within the cell containing the pair (e.g. A|B), and click on [Conditional State] button again. Select third behavior.

4) Repeat process to add up to four behaviors.

5) Click [Add row] to specify another combination of behaviors for analysis.

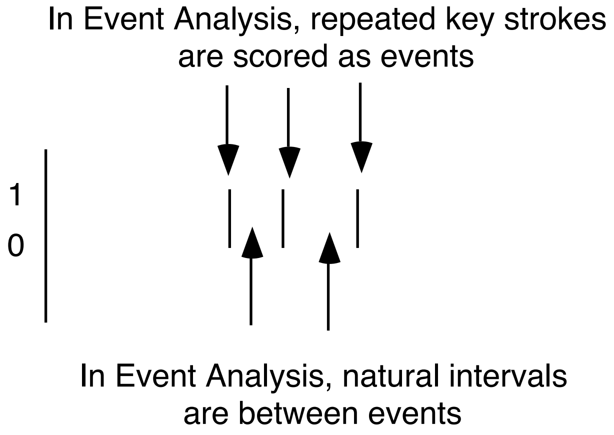
Conditional state combinations may also be typed directly into the table.

Modified behaviors may also be included within the combination of behaviors to be analyzed, and will be treated as described above for regular, unmodified behaviors. For example, if behaviors A and B are followed by modifiers 1 or 2, then combinations such as A1|B2 or A|B1|C|D may be analyzed. Special case: it is also possible to analyze A1|A or A2|A with conditional states analysis. The conditional proportion of time for A1|A would be the proportion of time that A1 is on, given that any A is on, where "any A" includes A1, A2, or A alone not followed by a modifier.

8.7 Focal Analysis Master > Event Analysis

Events are behaviors that have no duration (Figure 14). Thus, they can be counted and intervals between them quantified. For event analysis, JWatcher only quantifies "natural" intervals (see section 8.5 for more about "natural" versus "all" intervals).

Figure 14. Details about event analysis.


In Event Analysis, repeated key strokes are scored as events
1
0
In Event Analysis, natural intervals are between events

Note: JWatcher permits behaviors to be analyzed as both events and states.

Select those behaviors for which event analysis is requested and select the output required. The first row ("Events") reports results for standard event analysis. Event count is a tally of the number of times an event occurred, while the natural interval count is the number of natural intervals between events. "Average interval between" and "standard deviation" are based on natural intervals. Remember that if you specified an out-of-sight key in the State Analysis window, out-of-sight will truncate all natural intervals including those in event analysis. If modifiers follow a behavior analyzed as an event, then the above statistics will be reported both for each behavior/modifier combination separately and also for that behavior globally.

The second and third rows ("Conditional Events Natural" and "Conditional Events All") report statistics for conditional event analysis. Conditional event analysis tallies the number of events that occur while a specified state is on (or while up to three simultaneously occurring states are on). Statistics for natural intervals occurring between those events are also reported. The rate is the number of events that occur while the state is on, divided by the total time (in milliseconds) that the state is on. "Natural" conditional event analysis does not tally events occurring during truncated states. Once again, remember that if you specified an out-of-sight key in the State Analysis window, out-of-sight will truncate state durations. "All" conditional event analysis tallies events occurring during states regardless of whether states are truncated.

Figure 15. Details of "conditional events all" analysis.

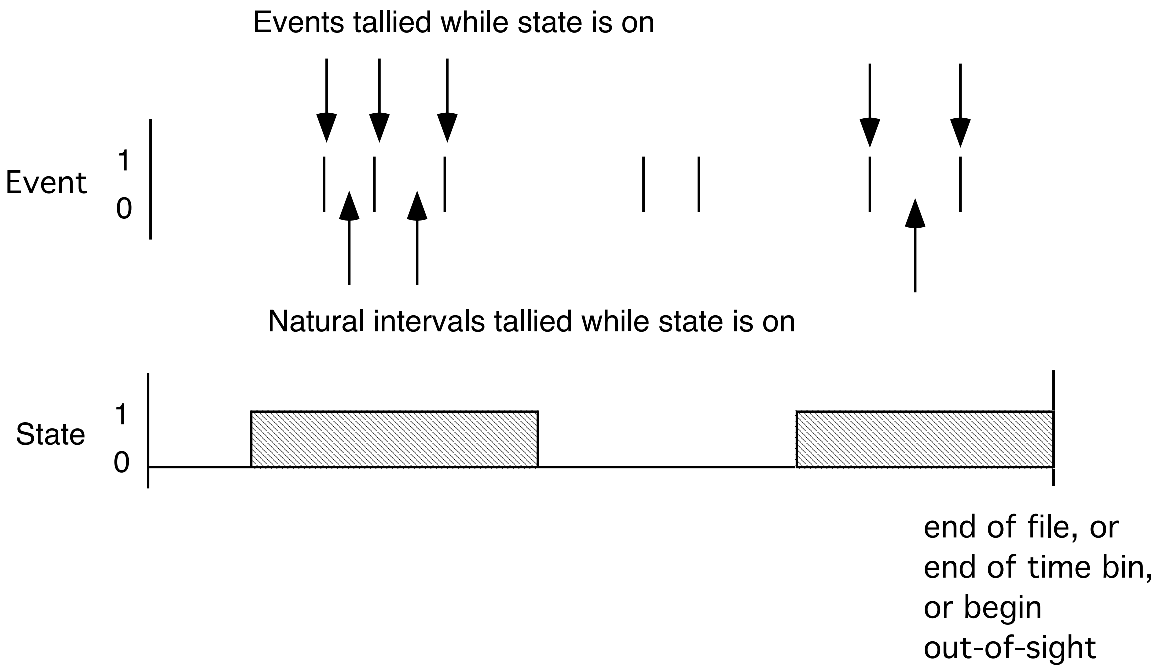
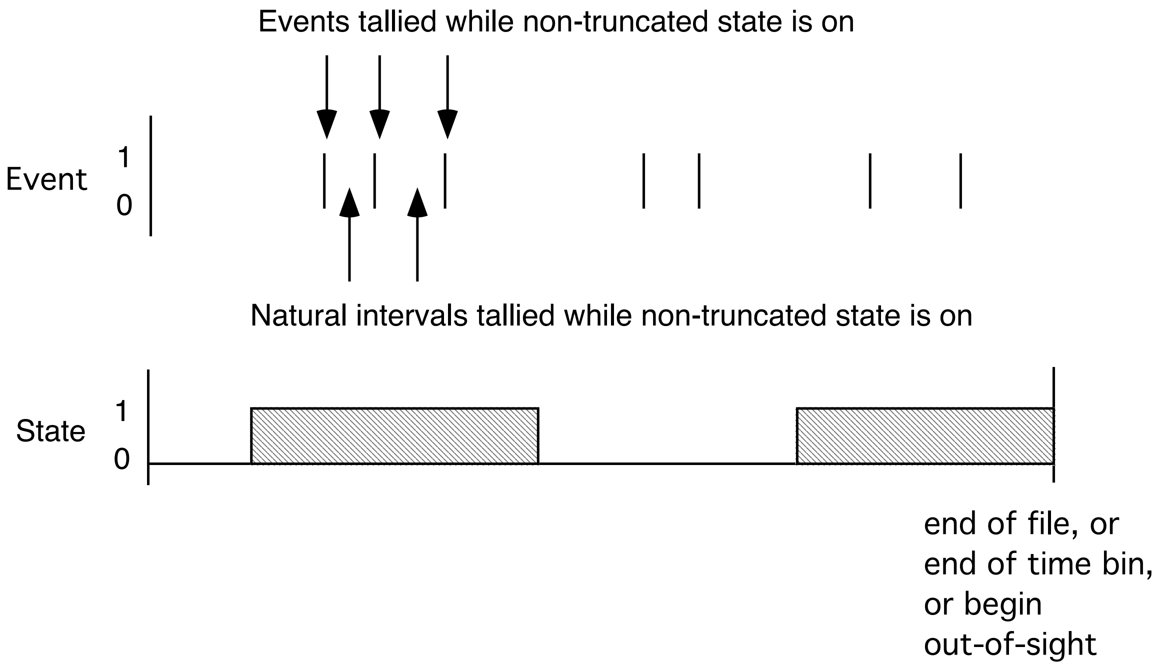

Events tallied while state is on
Event
1
0
Natural intervals tallied while state is on
State
1
0
end of file, or
end of time bin,
or begin
out-of-sight

Figure 16. Details of "conditional events natural" analysis.


Events tallied while non-truncated state is on
Event
1
0
Natural intervals tallied while non-truncated state is on
State
1
0
end of file, or
end of time bin,
or begin
out-of-sight

The combination of events and states to be analyzed must be specified in the Conditional Events tab. See Conditional Events tab (section 8.8) for more information.

WARNING: For each combination of event and state(s) to be analyzed, the event must be made non-mutually exclusive with the state(s) in the States tab. Otherwise, each occurrence of the event will turn off the state(s) of interest, and thus produce potentially unintended results. The program does not currently provide any error checking or warnings with respect to this issue.

8.8 Focal Analysis Master > Conditional Events

Use this window to specify the combinations of events and states to be analyzed as a conditional event. Conditional event analysis calculates statistics for events that occur while a specified state is on. Events occurring during up to three streams of simultaneously occurring states may be analyzed.

To begin, you must specify the behaviors to be analyzed as events in the Event Analysis tab. For conditional event analysis, behaviors are considered to be either events or states, but not both. All behaviors not specified to be an event in the Event Analysis tab will be considered to be states for this analysis.

WARNING: Events must be made non-mutually exclusive with state(s) in the States tab for this analysis to proceed correctly. Otherwise, each occurrence of the event will turn off the state(s) of interest, and thus produce potentially unintended results. The program does not currently provide any error checking or warnings with respect to this issue.

The statistics to be reported are specified in the Event Analysis tab. For A|B, statistics will be reported for event A given that state B is on. For A|B|C, statistics will be reported for event A given that states B and C are simultaneously on. For A|B|C|D, statistics will be reported for event A given that states B, C and D are simultaneously on. See Event Analysis tab (section 8.7) for more details about particular statistics.

Remember that if you specified an out-of-sight key in the States Analysis tab, then out-of-sight truncates natural durations of states (for "Conditional Events Natural" analysis only). Out-of-sight may not be included as part of a conditional event.

Modified behaviors may also be included within the combination of behaviors to be analyzed, and will be treated as previously described for regular, unmodified behaviors. For example, if behaviors A and B are followed by modifiers 1 or 2, then combinations such as A1|B2 or A|B1|C|D may be analyzed.

1) Click [Add row] to begin.

2) Click [Conditional Event] to select a pair of behaviors for conditional analysis (e.g A|B).

3) To add a third behavior (e.g to specify A|B|C), place cursor back within the cell containing the pair (e.g. A|B), and click on [Conditional Event] button again. Select third behavior.

4) Repeat process to add up to four behaviors.

5) Click [Add row] to specify another combination of behaviors for analysis.

Conditional event combinations may also be typed directly into the table.

9 Combinations

Combinations Master allows you to combine key codes in a data file (.dat files) and to ultimately create a new "combined" data file. By "combine key codes" we mean replace one or more existing key codes with another different code that you specify. To do so, you must first create a combinations master file (.cmf) in the Define Combinations window. Next, you will use the Combine Key Codes routine to generate a new set of "combined" data files containing the substituted codes.

1) Click the [New] button and, using the navigator box, select a pre-existing focal master file (.fmf). This inherits a set of key codes that can now be combined.

Note: You will not see these codes at this point. Thus, you may want to have a print-out of the .fmf file you are modifying next to you when creating a .cmf file. To print it, open the file in a word processor or text editor. Alternatively, you could view your .fmf by opening it in the Focal Master window and tabbing between windows.

2) Use the [Add row] button to add a row.

3) Type a key code in the Combined Key Code column that will replace one or more existing key codes. For example, f--forage may be used to replace a series of scored foraging behaviors (e.g., s--stand and forage, r--rear and forage, c—-crouch and forage)

4) Define the newly created key code in the Behavior/Modifer text box. For example, for "f" above, you might type in "forage all postures".

5) Click on the "Replaced Key Codes" text box and either type the key codes to be replaced, separated by a comma (e.g. s, r, c), or click on the now highlighted [Behaviors…] button to see a list of the key codes. Use the control key to select multiple key codes to be replaced. Clicking OK places those key codes into the "Replaced Key Codes" text field. Alternatively, click [Modifiers] to replace key codes defined in the .fmf as modifiers.

Note: You can replace or re-define behaviors or modifiers, but not a mix of behaviors and modifers.

6) Save this file by clicking on the [Save] or [Save As] box.

7) Click on the [Next] box to shift to the Combine Key Codes window. You must save all changes to enable the [Next] button.

8) Select the focal data file(s) or folder that you wish to combine. The default prefix for combined data files is "c"; change this is you wish.

Modify the default location where the combined files will be placed if you wish.

9) Click on [Combine] to generate new "combined" data file(s).

Note: When your newly created .cmf file is saved, a new .fmf file with the replaced key codes is automatically created and placed in the same folder as your .cmf file. By default, it is named c*.fmf, where * = the name you gave the .cmf. You may view the newly created "combined" data file(s) in Edit Data. The .fmf displayed (along with associated behaviors and modifiers) will be the c*.fmf. Before analyzing combined files, you will have to create a new .faf which is based on the c*.fmf. To create a .faf, see section 8.

10 Analysis

In the Analysis tab, you can either run your .faf to analyze .dat files (in either single or batch processing mode), or check for the reliability of your scoring of two files.

10.1 Analysis > Data

You must have a focal analysis master file (.faf file) and at least one data file (.dat file) to run Analyze Data.

Click on the file icon to specify the .faf file used to analyze the data.

Click on either the file icon or the folder icon to specify a .dat file or a folder with one or more .dat files to be analyzed. All .dat files in a folder will be analyzed if the folder option is selected.

Use the folder icon to specify a folder in which the results file(s) will later be placed. JWatcher will create a new folder if one does not already exist.

Note: JWatcher names the results files by changing the suffix of the original .dat file name. For example, the results file generated by analyzing a data file named "test.dat" will be "test.cd.res" and "test.tr.res" (see below for the distinction between *.cd.res and *.tr.res). If you modify the .faf file and re-run an analysis, be sure to specify a different folder for the results; the default is to write over existing results files. JWatcher will ask if it is OK to overwrite files before doing so.

There are two types of results files output; both are comma delimited text files.

*.cd.res is a "codes down" results file suitable for importing into a spreadsheet or word processing program. After a specified header, each row of results summarizes a behavior or a behavior and modifier. Each column in the results file contains a summary statistic specified when creating the focal analysis master file (.faf file).

*.tr.res provides an easy way to graph "traces" of behavior over time using a spreadsheet or graphics program. All 0-1 and 1-0 "transitions" along with the transition times are specified for each behavior. If behaviors have been subtracted, 0-"-1" transitions illustrate where subtracted behaviors have occurred in a time window.

Checking the box "Include summary of the focal analysis master in the results file" will do just that.

Decide whether to report the results for all behaviors and modifiers in the focal analysis master file, or only for those which occurred in a data file by selecting the appropriate radio button.

Note: If you wish to later summarize all of your results files into a single summary sheet using the Summarize Results routine, then you MUST print results for all behaviors.

Select the [Analyze] button to run the program. A progress box is updated each time a data file is analyzed. A text box provides an option to examine a file which contains a list of any errors or warnings. If an "error" occurs, no results files will be produced. However, in the case of a "warning", the .dat file will be processed. This log file (.log file) is saved into the results file folder with a unique date/time-based name.

Select the [Reset] button to clear the text fields, but note that the pathways and file names are editable.

Select the [View Log] button to view the log file created when analyzing data.

To view results files, open them in a word processing or spreadsheet program. Results files are comma-delimited text.

10.2 Analysis > Reliability

The Reliability algorithms calculate the percent agreement between the two files and a kappa coefficient. It also provides a "confusion" matrix and a line-by-line comparison of the two files. Martin & Bateson (1993) provides a good description of these statistics.

To calculate the reliability:

1) Select the two data files you wish to compare using the file selection tools to the right of the First Data File and Second Data File text fields. Note: both data files must begin with the same code.

2) Type in a name for your new reliability output file in the Reliability File Name text field. Do not specify the directory in this field. The suffix of this comma delimited text file will automatically be *.rel.res

3) Specify the directory where your new reliability output file should be saved using the folder selection icon to the right of the Reliability File Folder text field.

4) Click on the [Calculate Reliability] box to calculate reliability.

5) Use a spreadsheet program to open the comma delimited text *.rel.res file.

Note: The reliability routine compares the sequence of key codes in two data files, without respect to the times that they occur. Results are reported for the set of codes found within the two data files, without respect to any focal master file. No distinction is made between codes defined to be modifiers or behaviors for this analysis.

Reliability results file:

The confusion matrix tabulates the number of agreements between two data files, broken down by key code.

The statistics section tabulates results from the confusion matrix. The "Key Codes" column lists the codes that occur within the two data files, including the category "no entry" where relevant. The "filename1.dat" column reports the number of times that each code occurred in the first file. The "filename2.dat" column reports the number of times that each code occurred in the second file. The "both agree" column reports the number of positions in the sequence for which both files contained the same code. "YN" reports the number of times that a code occurred in the first file but did not occur in the second file. "NY" reports the number of times that a code did not occur in the first file but did occur in the second file. The "disagree" column reports the total number of disagreements for each code, which is the sum of "YN" and "NY". The "% agreement" is calculated as the total agreements for a code divided by the sum of the total agreements and the total disagreements, times 100, for each code.

The total percent agreement is the number of positions in the sequence where the codes agree (e.g. are the same) divided by the total number of available positions, multiplied by 100. Cohen's kappa is a modification of the percent agreement statistic that adjusts for agreements due to chance alone, and is reported as a proportion.

The list-by-line section juxtaposes the sequence of codes in the two data files to facilitate direct comparison of the two files. The sequence number is simply the position in the sequence. The time associated with each key press is also listed for each data file, but is not used for any calculations described above.

11 Summarize Results

Summarize Results allows you to conduct simple analyses on multiple data files and it allows you to combine results files into one file that can be imported into a spreadsheet or statistical package for additional analyses.

Note: This routine applies to .cd.res results files from basic analysis only. Reliability and sequential analysis results files cannot be summarized here.

1) Select the folder containing your results files (.cd.res files) using the folder selection tool to the right of the Result Files Folder text field. Note: all results files must have been analyzed using the same .faf. In addition, you must have selected the “Print results for all behaviors” button in the Analysis > Data tab. The summarize routine will compare all results files to the first file in the result files folder to ensure matching settings.

2) Type in the name for your new summary file in the Summary File Name text field. Do not enter the directory in this field. The output file will be a comma delimited text file with the suffix *.rsm.

3) Modify, if necessary, the location where you wish to save your new summary file, using the folder selection icon to the right of the Summary File Folder text field.

4) Specify which summary statistics you would like to calculate from the set of results files. All summary statistics are calculated for each behavior and every statistic in the results file.

5) Select whether you would like your summary statistics in columns, or in rows. Note: some spreadsheet programs limit the number of columns able to be imported. If you have many behaviors and/or statistics in your results files, and you encounter this problem, try “Analysis results in rows”.

6) Select whether you would like the behaviors/modifiers reported as key codes (saves space) or as full names. Note: conditional states and conditional events will always be reported as key codes.

7) Click on [Summarize Results Files] to summarize your results.

Your comma delimited *.rsm file is located in the summary file folder that you specified. If you did not specify a summary file folder, then the *.rsm file will be placed by default in the result files folder.

This *.rsm file contains a list of results from each .cd.res file in the result file folder, followed by the requested summary statistics calculated across all those .cd.res files.

Each analysis heading is comprised of the statistic name (as reported in the .cd.res file), followed by the time bin (as reported in the .cd.res file), followed by the key code or behavior name. For example, StateAllDur N-0-f refers to State Analysis for "All" Durations, Count, for time bin zero, for key code "f".

Under some circumstances, analysis headings will be generated without any associated data. That is, some columns (or rows) may be blank. This occurs whenever the cd.res files have blank cells within the result matrix. For example, whenever conditional proportion of time is selected, blank cells will be produced for the list of regular behaviors. Or, whenever regular event analysis statistics have been selected, blank cells will be generated for those behaviors not specified to be events. These blank columns (or rows) in the summary file are essentially irrelevant. You will need to delete them yourself in your word-processing or spreadsheet program. Note: if you are uncertain as to why specific columns (or rows) are blank, we suggest examining the original cd.res files for clarification.

Additional note about error checking: JWatcher derives the analysis headings from the first cd.res file in your result files folder. JWatcher will then compare each subsequent cd.res file in the folder with this first file to ensure that the same .faf file was used. In addition, JWatcher tests whether the number of rows and columns of the result matrix for each cd.res file is the same as the first file. This is why you must "Print results for all behaviors" when generating cd.res files. Note that if you time window and the session duration (EOF time) varies drastically among data files, then you could potentially generate cd.res files with a different number of time bins (and hence a different number of rows and columns). In this case, you will not be able to summarize this collection of files.

Known issue: For the Result Files Folder text field, if you type in the name of a non-existent folder name, then nothing will happen when you click on [Summarize Result Files] button. No warning message or output will be generated. The log will not be updated, and will continue to display the previous analysis. You may be asked if it is OK to overwrite existing file, when relevant, but file will not be overwritten, as no analysis will occur.

12 Sequential Analysis

JWatcher's sequential analysis routines enable you to describe various patterns in a sequence of behaviors. To run these routines, you will need a focal master file (.fmf) and at least one data file (.dat).

WARNING: For sequential analysis, you must specify a focal master file when capturing data. Data files created without an associated focal master file cannot be analyzed with sequential analysis routines.

All calculations are based upon the sequence only of key codes in a data file, without respect to the specific times that the key codes were pressed. Thus, these analyses assume a single stream of events. There can be no simultaneously occurring behaviors for these types of analyses, so data files must be scored (or edited) accordingly.

1) To begin, specify the focal master file and other parameters in the General tab. See section 12.1 for more details. General tab specifications apply to all sequential analysis tabs except for the runs test. Omit this step for the runs test.

2) Next, open the window for the type of analysis desired by clicking on it. There are five options: Runs, Lag Sequential, Sequence Analysis, Complex Sequences, and Markovian Analysis.

3) You must specify the data file(s) to be analyzed and the folder in which result files will be placed, for all tabs.

   For the focal data file(s) field, click on either the file icon or the folder icon to specify a data file (.dat) or a folder with one or more .dat files to be analyzed. All .dat files in a folder will be analyzed if the folder option is selected.

   For the results folder field, use the folder icon to specify a folder in which the results file(s) will later be placed. JWatcher will create a new folder if one does not already exist.

Note: JWatcher names the results files by changing the suffix of the original data file (.dat) name. For example, the results file generated by the runs test for a data file named "test.dat" will be "test.run.res". The suffixes for sequence analysis, Markovian analysis, lag sequential analysis, and complex sequences are seq.res, mar.res, lag.res, and com.res, respectively. If you re-run an analysis with different parameters, be sure to specify a different folder for the results; the default is to write over existing results files. JWatcher will ask if it is OK to overwrite files before doing so.

4) Specify statistics to be calculated where relevant.

5) Click the [Analyze] button to run the program. A progress box is updated each time a data file is analyzed. A text box provides an option to

examine a file containing a list of any errors or warnings. If an "error" occurs, no results files will be produced. However, in the case of a "warning", the .dat file will be processed. This log file (.log) is saved into the results file folder with a unique date/time-based name.

6) Select the [Reset] button to clear the text fields and restore statistics selections to default values. Note that the file and folder names within text fields are directly editable.

7) Select the [View Log] button to view the log file of the last analysis that you ran.

8) To view results files, open them in a word-processing or spreadsheet program. Results files are comma-delimited text.

12.1 Sequential Analysis > General

Use this window to specify the focal master file and other settings for all sequential analysis tabs except the runs test.

1) To specify the focal master file (.fmf), click on the file icon to the right of the text field, and use the navigation window to select the file. The focal master file specified here should be the same as the focal master file associated with your data files.

2) Select any behaviors that have modifiers following them. For instance, aggression ("A") may be directed to different individuals (e.g. "1", "2", or "3"). The default is that no modifiers follow any behaviors.

Note: Only modifiers that immediately follow a behavior specified as having modifiers will be reported in the results files. Any modifiers that do not follow a modified behavior will be recorded in the .dat file, but will be ignored during analysis.

For a description of how modified behaviors are treated during analysis, see sequential analysis sections 12.3-12.6.

3) Select any behaviors to be ignored. "Ignore" literally ignores selected behaviors within the sequence. Thus, the sequence will be tabulated as if the ignored code never occurred. The default is that no behaviors are ignored.

4) Specify a break code from the pull-down menu. A break code can be any code that indicates a break in the sequence, when you do not know exactly what behavior followed or preceded other codes in the sequence. For example, a subject may go out-of-sight, or a sequence in a data file may be comprised of several shorter sequences scored on different occasions. The default is that no break code is specified. See section 12.3 for specifics on how break codes affect analyses.

5) Specify whether consecutive, repeated codes are to be tallied or ignored. Depending upon your question, it may or may not make logical sense for a code to follow (or precede) itself. If codes cannot logically be repeated, then check the "ignored" button. If codes are allowed to repeat, then check the "tallied" button.

Note: When "ignored" is checked, any repeated codes in your data file will be treated as if they do not exist in the sequence. We strongly suggest that you check the "ignore" button whenever you have logically defined your codes to be non-repeating, even if you are sure that such repeated codes do not exist in your data set. Specifically, for lag sequential analyses, statistics are calculated differently depending upon which of these buttons is checked (see section 12.5).

All results files will contain a list of general tab settings in place at the time of analysis (except for the runs test). The following table gives an example:

Focal master file from general tab:	/my folder/data files/test.fmf
Number of behaviors (or behavior/modifiers):	7
Behaviors that are modified:	A
Number of entries in data file:	30
Break code:	X
Number of break code entries:	1
Number of continuous records:	2
Behaviors that are ignored:	none
Number of ignored codes:	0
Consecutive repeated codes tallied:	yes

Notes: The number of behaviors found in the data file sequence is reported, regardless of the total number of behaviors listed in the .fmf. When behaviors are modified, then each behavior/modifier combination (e.g. A1, A2, A3, where A is a behavior followed by modifiers 1, 2, or 3) is reported as a distinct event. Thus, a sequence comprised of A1, A2, A3, and B would be reported as containing 4 behaviors (or behavior/modifiers). The break code, if specified, and any ignored codes are not included.

The number of entries in the data file is the total number of times that either a behavior key code or a behavior/modifier combination has been pressed. For example, the number of entries for the sequence {A1 B A2 A1 C} is 5, where A, B, and C are behaviors, and A is modified by 1, 2 or 3. Break codes are not included, if specified and present, nor are any ignored codes.

The number of continuous records will always be 1, unless a break code is specified and present. For example, a single break code occurring in mid-sequence will divide a sequence into two separate, continuous records. The number of continuous records will always be one greater than the number of break code entries, except when those break codes occur at the very beginning or very end of the overall sequence.

All results files will also contain a list of general statistics (except for the runs test). The following table gives an example, where A, B, C, and D are behaviors, and A is modified by 1, 2, or 3:

#GENERAL STATISTICS

Behavior/Modifier	Description	Count	Probability
A	beh A	9	0.3000
A*	beh A/ *none	1	0.0333
A1	beh A/ *mod 1	4	0.1333
A2	beh A/ *mod 2	2	0.0667
A3	beh A/ *mod 3	2	0.0667
B	beh B	9	0.3000
C	beh C	7	0.2333
D	beh D	5	0.1667

Notes: The count is the number of times that the behavior occurred in the sequence. The probability is the count for each behavior divided by the total number of entries in the data file. For modified behaviors, counts and probabilities are reported for each behavior/modifier combination separately (e.g., A1, A2, A3) and also for the behavior in its entirety (A). A modified behavior occurring without a modifier following it is treated as a distinct event, and is reported as the behavior followed by an asterisk (e.g., A*).

12.2 Sequential Analysis > Runs Test

The Runs Test examines whether a sequence of dichotomous behaviors is random. Data files must be comprised of no more than two behaviors. Analysis proceeds independently of any selections made in the General tab. Thus, behaviors may not be modified, ignored, or specified to be a break code.

Note: Key codes defined to be modifiers in the focal master file associated with the data file(s) will be ignored.

The following statistics are reported in the results file for a sequence comprised of two behaviors A and B:

Cases(A)
Cases(B)
Total cases
Number of runs
z-score
p-value (2-tailed)

The null hypothesis for this test is that the sequence is random.

See Sokal & Rohlf (1995) for further description and interpretation of the runs test, including z-score formula.

12.3 Sequential Analysis > Sequence Analysis

Sequence analysis algorithms quantify the frequency with which one behavioral event follows another behavior event. For example, you may be interested to know the frequency that behavior B follows behavior A in a sequence. Statistics are reported in matrix form for all possible event pairs found within a data file. Additionally, Odds Ratio and Yule's Q may be calculated for specific event pairs.

To use this tab, you must first specify a focal master file (.fmf) and other settings in the general tab. However, the results reported are based upon the key codes occurring within each data file itself, and not the complete list found in the fmf.

The goal is to tabulate a "given" behavior (always called "Lag 0"), and identify what immediately follows it at different lags (e.g., "Lag 1", "Lag 2", etc.). The behavior that follows the given event is referred to as the target event. Lag refers to the position of the target event relative to the given event. Lag 1 means that the target event is one position ahead of the given event in the sequence. Lag 2 means that the target event is two positions ahead of the given event in the sequence. JWatcher calculates sequence analysis statistics up through Lag 5. For example, assume the following sequence: {A B C D E}. If A is the given event, then B occurs at lag 1, C occurs at lag 2, D occurs at lag 3, and E occurs at lag 4.

A behavior event pair refers to a specific given and target event combination, regardless of lag (e.g., B|A refers to behavior B given behavior A, such that A is the given event and B is the target event).

All sequence analysis tabulations are based upon overlapped sampling of the sequence. Specifically, overlapped sampling means that you slide your sampling "window" from the first code to the second code, to the third code, etc. You are moving from one position in the sequence to the next. For example, for the sequence {A B C D E}, overlapped sampling for lag 1 would produce the following series: AB, BC, CD, DE.

When break codes are used, sampling stops at each break code, and resumes at the first code immediately thereafter. For example, for the sequence {A B C X D E F} where X is the break code, overlapped sampling for lag 1 would produce the following series: AB, BC, DE, EF. For lag 2, the series would be: AC, DF.

When codes are ignored, sampling occurs as if these codes never occurred in the sequence. For example, for the sequence {A B C D E F}, if B is ignored, the following series would be produced for lag 1: AC, CD, DE, EF.

When repeated codes are ignored, sampling occurs as if the repeated code never occurred in the sequence. For example, for the sequence {A A B B C C}, the following series would be produced for lag 1: AB, BC.

When behaviors are modified, each behavior/modifier combination is treated as a distinct event. For example, for the sequence {A1 A2 B A1 C}, where A, B, C are behaviors and A is modified by 1 or 2, the following series would be produced for lag 1: A1A2, A2B, BA1, A1C. If a behavior specified to have modifiers following it occurs without a modifier in the sequence, then this unmodified behavior will be treated as a distinct event, and will be designated by an asterisk in the results file. For example, for the sequence {A1 A A2 B}, the following series would be produced for lag 1: A1A*, A*A2, A2B.

JWatcher's Sequence Analysis routines calculate the following statistics:

Observed matrix
Simple probability matrix
Transitional probability matrix
Expected values, z-scores, and p-values
Odds Ratio
Yule's Q

See Bakeman & Gottman (1997) for description and interpretation of these statistics.

Warning: JWatcher's algorithms to calculate expected values and z-scores (and hence p-values) apply only when behaviors are allowed to repeat, that is, each behavior is allowed to follow itself. If behaviors are not allowed to repeat, we advise exporting your tabulated data with JWatcher's Complex Sequences routine, and then analyzing with a general-purpose log-linear program such as SPSS. See Bakeman & Gottman (1997) for further discussion on this topic.

Odds ratio and Yule's Q require that you specify specific event pairs for analysis using the key codes pair table. For each specified event pair, a collapsed 2x2 matrix will be generated, comprised of A, not A, B, not B. Calculations for Odds ratio and Yule's Q are based upon this collapsed matrix.

1) To specify key code pairs for Odds ratio and/or Yule's Q, click on [Add row] button in the key codes pairs table.

   Note: To enable [Add row] button, you must first specify an .fmf file in the general tab.

2) Click on [Key Code Pair] button and select key code pair in the pop-up window. Alternatively, key code pairs may be typed directly into the table such that A|B represents A given B.

3) Click [Add row] to add another pair. Use [Delete row] to delete one or more rows.

Additional notes:

WARNING: It is essential that you specify the same focal master file in the general tab as was used to create your data files. A mismatch in the list of behaviors and modifiers between these focal master files may produce erroneous results. You will be given a warning in the log when the name and pathway of the focal master files do not match. Depending upon the nature of the mismatch, your results files may contain a combination of the two lists of behaviors and modifiers and/or you may see −1's in some of your observed matrices. −1's in observed matrices will lead to erroneous calculations for all other matrices (simple probabilities, transitional probabilities, etc) because these are derived from the observed matrix. This applies to Odds Ratio and Yule's Q as well.

WARNING: When specifying key code pairs for Odds ratio and Yule's Q, the available list of behaviors and modifiers will be based upon the focal master file specified in the general tab. However, it is possible to specify pairs based upon one focal master file in the general tab, and then later change that focal master file, while retaining your original selections in the key code pairs table. If you do not delete these selections, the program will report results for them; however, if there is a mismatch between the first focal master file and the focal master file used to create the data files, then erroneous results may be produced, as described above.

When behaviors are modified, then the matrices will include tabulations for each behavior/modifier combination separately and also tabulations for that behavior globally. For example, consider the sequence {A1 A2 B A1 B A2 B B A A1 A1 B} where A and B are behaviors and A is modified by 1 or 2. The following observed matrix would be produced for lag 1:

		Lag 1					
		A	A*	A1	A2	B	total
Lag 0	A	3	0	2	1	4	7
	A*	1	0	1	0	0	1
	A1	2	0	1	1	2	4
	A2	0	0	0	0	2	2
	B	3	1	1	1	1	4
	total	6	1	3	2	5	11

Note that A* represents an A followed by no modifier in the sequence, and that A* is treated as a distinct event for analysis. Also, note that A refers to the global behavior A in this case. Thus, A|A refers to "any A" followed by "any A." A1|A refers to A1 following "any A" (i.e. A1 following A*, A1, or A2). A|A1 refers to "any A" following A1 (i.e. A*, A1, or A2 following A1).

12.4 Sequential Analysis > Markovian Analysis

Markovian analysis establishes whether a series of behaviors is independent (i.e. whether behaviors are constrained or predicted by behaviors that precede them). Shannon-Wiener information statistics are used to analyze these models ($-\Sigma\ p_i \log_2 p_i$). The logic of this is to calculate increasingly higher order "Markovian models" and analyze how the "uncertainty" is reduced with each successive step. Following Hailman & Hailman (1993), JWatcher calculates statistics up to second order Markovian models because ethological studies are usually limited in their duration and prevent meaningful higher-level analyses.

A first order Markov process means that the next event is predicted entirely by the immediately preceding event. A second order Markov process means that the next event is predicted entirely by two immediately preceding events. Uncertainty values of zero at some order indicate that the data follow a Markov process at that order. Large drops in uncertainty between two orders suggest that the data approach a Markov process at that order. Data that strongly approach a Markov process are often called "semi-Markovian." For example, a large drop in uncertainty between zero- and first-order uncertainty values suggests that the immediately preceding event predicts the next event with a high likelihood, but not entirely.

To use this tab, you must specify a focal master file (.fmf) and other settings in the general tab. However, statistics are calculated based upon the behaviors (and modifiers) occurring in each data file only. In other words, a behavior (or modifier) that is listed in the focal master file but that does not occur in the data file will not be included in the results.

See sequence analysis (section 12.3) for a description of how general tab settings affect tabulation.

Markovian analysis assumes standard overlapped sampling of the sequence (see section 12.3).

JWatcher's Markovian analysis will produce the following statistics for every analysis, in addition to the specific statistics selected in the menu:

U_m (maximum uncertainty)
U_0 (zero-order uncertainty)
U_1 (first-order uncertainty)
U_2 (second-order uncertainty)

JWatcher will also produce the following statistics (when selected) for first and second order models:

Observed matrix
Simple probability matrix
Transitional probability matrix

See Bakeman & Gottman (1997) and Hailman & Hailman (1993) for description and interpretation of the above statistics.

Note regarding modifiers: Markovian analysis treats each behavior/modifier combination as a distinct event. For example, if behavior A is specified to be modified by 1 or 2, then the behavior/modifier combinations A1 and A2 (or A*, if A were not followed by any modifier) are considered to be distinct events. No calculations are made with respect to behavior A globally.

12.5 Sequential Analysis > Lag Sequential

Lag sequential analysis algorithms quantify the frequency with which one behavior event follows or precedes another behavior event for a range of lags. For each event pair that you specify, JWatcher will report observed frequencies, expected mean frequencies, and 95% confidence intervals around these means, for a series of successive lags prior to or following the focal behavior. See Douglas & Tweed (1979) and Van Hooff (1982) for more details.

To use this tab, you must first specify a focal master file (.fmf) and other settings in the general tab.

WARNING: This is the only tab where calculations differ based upon whether codes are allowed to repeat or not. Ensure that you properly specify whether repeated codes are to be tallied or ignored in the general tab. If you have logically defined your codes to be non-repeating, then you MUST check the "ignored" box even if you are certain that no repeating codes exist in your sequence. All statistics are affected by this choice, except for "Observed frequency B|A" and "Observed frequency A". More details about statistics are below.

You must then specify at least one event pair for analysis in the key code pairs table. A behavior event pair refers to a specific given and target event combination, regardless of lag (e.g., B|A refers to behavior B given behavior A, such that A is the given event and B is the target event). See sequence analysis (section 12.3) for definitions of given event, target event, and lag. Modified behaviors may be included within an event pair (e.g., X5|Y may be specified, where X and Y are behaviors, and X is modified by 5).

1) To specify key code pairs for analysis, click on [Add row] button in the key codes pairs table.

Note: To enable [Add row] button, you must first specify an .fmf file in the general tab.

2) Click on [Key Code Pair] button and select key code pair in the pop-up window. Alternatively, key code pairs may be typed directly into the table such that A|B represents A given B.

3) Click [Add row] to add another pair. Use [Delete row] to delete one or more rows.

JWatcher's Lag Sequential routines calculate the following statistics:

Observed frequency B|A
Observed frequency A
Probability B
Expected frequency B|A
Standard deviation
Confidence interval 95%
z-score
p-value

Each of these statistics is calculated for a specified event pair, such that B|A refers to B given A. We use B|A generically in the statistics selection menu to refer to any combination of behaviors where A is the "given" or "lag 0" behavior and B is the "target" behavior. Thus, "Observed frequency A" tabulates the observed frequency of your given behavior, and "Probability B" calculates the probability of your target behavior.

The target behavior (B) may either precede or follow the given behavior (A), depending upon the range of lags specified. Statistics will be calculated for every lag within the specified range. A negative lag means that the target behavior will precede the given behavior, whereas a positive lag means that the target behavior will follow the given behavior. For example, lag –2 means that the target behavior occurs two positions before the given behavior, whereas lag +2 means that the target is two positions ahead of the given behavior in the sequence. For the sequence {A B C D E}, if the given behavior is C, then the target will be A for lag –2, and E for lag +2. The default range of lags is –5 to +5. Modify these values as desired.

All lag sequential tabulations are based upon overlapped sampling of the sequence. See section 12.3 for definition of overlapped sampling.

See section 12.3 for a description of how break codes and ignored codes affect tabulation.

Notes regarding modifiers:

Lag sequential analysis treats each behavior/modifier combination as a distinct event. For example, if A is a behavior specified to be modified by 1 or 2, then the behavior/modifier combinations A1 and A2 (or A*, if A were not followed by any modifier) are considered to be distinct events.

Event pairs may include global behaviors when codes are allowed to repeat. For example, if A and B are behaviors specified to be modified, then A|B may be requested, where A|B refers to "any A" following "any B." However, some combinations such as A1|A and A|A1 are not permitted.

Event pairs may not include global behaviors when codes are not allowed to repeat.

12.6 Sequential Analysis > Complex Sequences

The purpose of this tab is two-fold: 1) it enables you to create your own sampling regimens which may differ from those provided in the other tabs, and 2) it enables you to export tabulated data for analysis in other statistical packages. Results files are comma-delimited text (with the suffix com.res). Word-processing or spreadsheet programs may be used to easily reformat the results files for use in a variety of statistics programs.

To use this tab, you must first specify a focal master file (.fmf) and other settings in the general tab. However, the calculations for the observed table and export data are based upon the key codes occurring within each data file itself, and not the fmf. In other words, a behavior may be listed in the fmf, but if it never occurs then it will not be included in the table or export data. The same applies for modifiers.

NOTE: To generate data for export using sampling regimes identical to those found in the sequence analysis, Markovian, or lag sequential tabs, see the end of this section.

For each analysis, you may generate an observed frequency table and/or a list of tabulated data for export. If the "Report all results" box is selected then the observed frequency table will list results for all possible behavioral combinations, including those with counts of zero. The "Report all results" box does not affect the list of export data.

You may use either the Basic Sampling or Custom Sampling routines of this tab.

BASIC SAMPLING:

Basic sampling requires that you specify the number of consecutive codes to sample at one time (the length of the sample unit), and whether the sampling will be overlapped or non-overlapped.

Overlapped sampling means that you slide your sampling "window" from the first code to the second code, to the third code, etc. You are moving from one position in the sequence to the next.

Non-overlapped sampling means that your sampling "window" skips one or more positions in the sequence, such that no position is sampled more than once. Sampling occurs at an interval equal to the length of the sample unit.

For example, for the following sequence {A B C D E F G}, you may wish to sample units of length 3. In this case, you would generate the following series with overlapped sampling: ABC, BCD, CDE, DEF, EFG. For non-overlapped sampling with length 3, the following series would be produced: ABC, DEF. In other words, sampling occurs at position 1 (A) and then again at position 4 (D) in our example.

CUSTOM SAMPLING:

Custom sampling enables you to vary the length of both the given and the target units, and to vary the relative position of one with respect to the other (i.e. the lag). You may also vary the sampling interval to create overlapped sampling, non-overlapped sampling, or some other sampling interval. In addition, you may sample portions of the sequence by varying the start and end positions.

Given Length:

We use "given" to mean the criterion event, or "lag 0" event. It is the event for which you want to know what precedes or follows. The length refers to how many codes comprise this event. For example, if the sequence is X Y Z A B C, and the given length is 2, then the first given event would be XY, if we assume we are starting from the beginning of the sequence. If the given length is 4, then the first given event would be XYZA.

Target Length:

We use "target" to mean the event that precedes or follows the "given" event. The length refers to how many codes comprise this event. The "target" event could occur at any specified lag, either positive or negative (see below).

Lag:

Lag refers to the position of the "target" event relative to the "given" event. Lag 1 means that the target is one position ahead of the given event in the sequence. Lag –1 means that the target is one position behind the given event. If the given event is comprised of more than one code, then positive lags are counted from the last code of the given event. However, negative lags, in this case, are counted from the first code of the given event, not the last code.

For example, consider the sequence: A B C D E F G H I J K. If the given length is 3, the target length is 2, and the lag is +3, then the first sequence would be ABCFG. The given is ABC. The lag is counted as 3 positions ahead of the last code in the given sequence, so three positions ahead of the C brings us to F. The target is FG.

Consider the same sequence. If the given length is 3, the target length is 2, and the lag is -3, then the first sequence would be DEFAB. The first code in the sequence from which we can count back 3 positions (lag –3) is D. Given length = 3, so given is DEF. Counting back three positions brings us to A. Target length = 2, so the target is AB.

Sub-sample interval:

The sub-sample interval refers to how you move along the sequence when sampling. Standard overlapped sampling would have a sub-sample interval of 1. This means that you slide your "window" from the first code to the second code, to the third code, etc. You are moving from one position in the sequence to the next (defined to be an interval of 1). A sub-sample interval of 2 would mean that you start at the first code, and then move your sampling "window" to the third code, then to the fifth code, etc.

For example, for the sequence {A A B C A}, if we use a given length =1, target length =1, and lag=1, then a sub-sample interval =1 would produce the following series: AA, AB, BC, CA. The sampling window slides along the sequence one position at a time. A sub-sample interval = 2 would produce the series: AA, BC. The sampling window slides along by moving from first to third to fifth position (interval is 2). For this sequence there is nothing to tally beyond the third position.

Note that if the given event is comprised of more than one code, then the interval is counted from the first code of the given event. For example, consider the sequence: A B C D E F G H I J K. If the given length is 3, the target length is 2, and the lag is 3, then the first sequence would be ABCFG, as described above. If our sub-sample interval is 3, then next sequence would be DEFIJ. We move 3 positions ahead from the first position in the given event (A), which brings us to D. We then consider our given event to be DEF (length = 3). Because lag = 3, we count 3 positions ahead from the last position in the given (F), which brings us to I. The target length = 2, so the target is IJ. Continuing, the subsequent sequence should begin with GHI, but there are no codes available at lag 3, so no other sequence is produced.

Sampling portions of the sequence:

Begin position:

This is the position in the overall sequence where tallying begins. For example, for the sequence {A B C X D E F}, begin position 1 means start at A. Begin position 3 means start at C. Note that break codes are included when specifying positions in the sequence. For example, if X is specified to be the break code, X is considered to be at position 4 and D is considered to be at position 5.

End position:

This is the position in the overall sequence where tallying ends. For example, for the sequence {A B C D E F G H I J K}, end position 0 means end at K. End position 6 means end at F.

Begin code:

Tallying begins at the first occurrence of the specified code. For example, for the sequence {A B C D A B C D A B C D}, if the begin code is D, then tallying starts at the first D, located at position 4 in the sequence.

End code:

Tallying ends at the first occurrence of the specified code, after tallying has begun. For example, for the sequence {A B C D A B C D A B C D}, if the begin code is D, and the end code is B, then tallying will start at D (position 4) and end at B (position 6).

Known issue when sampling a portion of the sequence: the observed frequency table and the data for export will correctly report results based upon the portion of the sequence requested. However, calculations for the general settings section and general statistics section are based upon the sequence in its entirety.

HOW TO REPLICATE SAMPLING FROM OTHER TABS:

To replicate sampling for sequence analysis, select custom sampling and specify given length = 1, target length = 1, sub-sample interval = 1. Vary the lag from 1 to 5, depending upon the lag desired. Ensure "Generate data for export" box is selected.

To replicate sampling for Markovian analysis, select basic sampling and specify overlapped. For first order models, use length = 2. For second order models, use length = 3. Ensure "Generate data for export" box is selected.

To replicate sampling for lag sequential analysis, select custom sampling and specify given length = 1, target length = 1, sub-sample interval = 1. Vary lag depending upon lag desired. Ensure "Generate data for export" box is selected.

HOW GENERAL TAB SETTINGS AFFECT TABULATION:

Modified behaviors:

When behaviors are modified, each behavior/modifier combination is treated as a distinct event. For example, for the sequence {A1 A2 B A1 C}, where A, B, C are behaviors and A is modified by 1 or 2, the following series would be produced for basic, overlapped sampling with length = 2: A1A2, A2B, BA1, A1C. When a behavior is modified, observed frequencies will not be provided for the behavior globally (e.g. for A in the above example, where A refers to "any A"). If such statistics are desired, then you should re-run the analysis with the behavior unmodified. That is, deselect it in the modifier table of the general tab.

Ignored behaviors:

When codes are ignored, sampling occurs as if these codes never occurred in the sequence. For example, for the sequence {A B C D E F}, if B is ignored, the following series would be produced for basic, overlapped sampling with length = 2: AC, CD, DE, EF.

Repeated key codes:

When repeated codes are ignored, sampling occurs as if the repeated code never occurred in the sequence. For example, for the sequence {A A B B C C}, the following series would be produced for basic, overlapped sampling with length = 2: AB, BC.

Break codes:

When break codes are used, sampling stops at each break code, and resumes at the first code immediately thereafter. For example, for the sequence {A B C D E X F G H I} where X is the break code, the following series would be produced for basic, overlapped sampling with length = 3: ABC, BCD, CDE, FGH, GHI. For custom sampling, given = 1, target = 1, lag = 3, sub-sample interval =1, then the following series would be produced: AD, BE, FI. For custom sampling, given = 1, target = 1, lag = -3, sub-sample interval = 1, the following series would be produced: DA, EB, IF.

Note: For custom sampling with negative lags and sub-sample intervals greater than 1, the following example illustrates what happens for this special case. Consider the sequence {A A B A C B X B C A B C} where A, B, and C are behaviors and X is the break code. The following series will be generated for given = 1, target = 1, lag = -1, and sub-sample interval = 2: BA, CA, AC, CB. Sampling will begin at the first position in the sequence or after a break code, and if that position cannot be used to sample, then it will skip to the next position based upon the sub-sample interval specified, which is 2 positions ahead in this example.

RESULT FILE FORMAT:

The following example illustrates the format of exported data. Assume the sequence: {A A B A C A}. Basic, overlapped sampling with length = 3 will produce the series: AAB, ABA, BAC, ACA. This series will be reported in the result file as follows:

#EXPORT DATA

Sample number	Sequence	code1	code2	code3
1	AAB	A	A	B
2	ABA	A	B	A
3	BAC	B	A	C
4	ACA	A	C	A

Note that the series of sampled sequences are listed in the order that they are sampled, accompanied by the corresponding sample number. Each sampled sequence is additionally broken down into separate columns containing no more than one code per column, in order to facilitate importation into some statistical packages. The output associated with custom sampling is identical as above, but will also contain two extra columns, listing the given and target sequences, respectively, for each line.

13 Strategies for using JWatcher

To use JWatcher, you must first create an overall ethogram for your species in the .gdf. Then create a .fmf from a subset of those behaviors. Once you have a .fmf, you're ready to score behavior. However, observing behavior ultimately requires you to have an explicit idea of what you're aiming to quantify.

For basic analysis, we suggest that you create an .faf before scoring behavior because the process will force you to decide whether or not behaviors are mutually-exclusive, whether or not you will ultimately want to subtract or ignore behaviors, which behaviors may or may not have modifiers, whether or not you're interested in understanding something about events or states, and whether or not you will be using conditional analyses. The types of analyses that you wish to undertake will ultimately influence the best way to score behavior. Considering these issues before scoring behavior does not prevent you from changing the .faf at a later time. In fact, we find the ability to "time window" data one of the most useful functions of JWatcher and we routinely analyze data with several .fafs.

For state analyses, it behoves you to think very clearly about the difference in meaning between "natural" and "all" analyses. This decision applies to both duration and interval statistics. Remember that "natural" analyses will report results only for those durations and intervals that begin and end completely within a time bin (that is, they are not truncated). Both durations and intervals will be truncated by the end of the focal, end of a time bin, beginning of a time bin other than the very beginning of the focal (T=0), and the onset and termination of a key code specified to be out-of-sight. Additionally, intervals will be truncated by the beginning of the focal (T=0). Because durations are not truncated by the beginning of the focal, if you are interested in natural durations, you should begin scoring at the true onset of the first behavior. Also note that out-of-sight will have an additional consequence for "all" interval statistics (see section 8.5), in that the onset of out-of-sight will terminate (but not truncate) an interval, and will continue to do so for the entire time that out-of-sight remains on.

There are two ways one could conceivably use JWatcher to calculate the proportion of time in sight. One way is to "subtract" the behavior that represents out-of-sight by using the subtract function in Focal Analysis Master > Exclusions and then examine the proportion of time statistic. The other way is to specify an out-of-sight key in Focal Analysis Master > State Analysis and examine the "proportion of time in sight" statistic. However, designating a key as out-of-sight in the State Analysis window will have additional consequences beyond simple subtraction if you are interested in natural durations or natural intervals. Specifically, it truncates natural durations and natural intervals, as described above. If a focal subject is truly out of sight, this is a desirable characteristic.

If you are interested in calculating time budgets, then you should be aware that proportion of time (and proportion of time in sight) statistics are calculated using the duration of the entire time bin as the denominator (or the entire time bin minus the time out-of-sight). This applies to both "all" duration and "natural" duration calculations. For "natural" duration calculations, the time associated with any truncated states is not subtracted from the denominator in this case.

There are several file management issues relevant to both basic and sequential analyses. When .dat, .faf, and .cmf files are created, the location (i.e., pathway) to the associated .fmf at the time of creation is recorded. Editing .dat files, or using .faf or .cmf files for later analysis, requires the .fmf either to be in the original location where it was when the above files were created, or in the same folder as the .dat, .faf, or .cmf files. We suggest that users create a directory for each project and not move around the .fmf, .faf, or .cmf files. Data and results files can usually be reorganized as long as the .fmf stays in its original location.

14 Warnings and Errors

Log File Warnings

Reliability

1. At least one of the key code entries in [data file] contains a comma. Commas will alter the format of the reliability results file. Analyzing anyway.

2. Specified .dat files were not created with the same fmf. Analyzing anyway.

All Tabs

3. Code [some code] at offset [some time] is not in focal master [fmf file]. Ignoring code.

4. Modifier [some modifier] at offset [some time] does not follow a behavior. Ignoring modifier.

5. Modifier [some modifier] at offset [some time] follows an already modified behavior [some behavior-modifier pair]. Ignoring modifier.

6. There are no code entries in the focal data file. Analyzing anyway.

7. Scoring from first code [some code] but this behavior is ignored.

8. Length of time bin [time bin duration] is greater than length of focal data capture session [data capture duration].

Sequential Analysis – Complex Sequences

9. The range of positions specified is invalid since either BEGIN code or END code does not appear in the data file. Analysis will proceed for all positions available within the data file.

10. The range of positions specified is invalid since BEGIN position is greater than END position. Analysis will proceed for all positions available within the data file.

11. The range of positions specified is invalid since either BEGIN position or END position or both exceed those found in the data file. Analysis will proceed for all positions available within the data file.

Sequential Analysis – Lag Sequential

12. The range of lags specified exceeds that found in the data file. Analysis will proceed for those lags available within the data file.

13. The range of lags specified is invalid since FROM lag is greater than TO lag. Analysis will proceed for those lags available within the data file.

14. For data file [data file], at least one z-score, p-value and/or confidence interval may not be valid. The z-score, p-value and confidence interval are calculated with a normal approximation that is valid only when N*P > 5 and N*(1-P) > 5, where N is the observed frequency of the given event (lag 0) and P is the probability for the target event (precedes or follows the given event). The following pairs/lags are affected: [pairs].

Sequential Analysis – Runs

15. The sample size of at least one of the dichotomous variables is less than 20 in the data file [data file]. This test is calculated using a normal approximation that may not be valid when the sample size is small. Analyzing anyway.

16. General tab settings do not apply to the runs test. The runs test requires data files to contain a sequence of dichotomous variables only, such that each variable consists of a single key code. Any selections made in the general tab with respect to focal master file, modifiers, ignored behaviors, break code, or consecutively repeating codes will be disregarded for this test. Analyzing anyway.

17. Code [some code] at offset [some time] is either not defined in the focal master file associated with this data file, or is defined to be a modifier. Undefined codes and modifiers will be ignored for the runs test. Ignoring code.

Summary Results

18. The summary result file has more than 254 columns. This file may be incompatible for use with some spreadsheet programs such as Microsoft Excel.

19. The summary result file has more than 254 columns. This file may be incompatible for use with some spreadsheet programs such as Microsoft Excel.

Data Analysis

20. The focal master for focal data is [fmf name]. This does not match the focal master [fmf name] in focal analysis master. Analyzing anyway.

Combinations – Combine Key Codes

21. There are no code entries in the focal data file. Combining anyway.

22. The focal master for focal data is [fmf file]. This does not match the focal master [fmf file] in combinations master. Combining anyway.

23. Focal master file associated with specified combinations master file [fmf name] does not exist or could not be loaded. Proceeding with analysis anyway.

Sequential Analysis – All but Runs

24. The focal master file is not specified in the General tab. Analyzing anyway using the focal master file in the data file(s). Analysis will proceed assuming that all behaviors are unmodified.

25. The name and directory of the focal master file [fmf name] associated with the data file [dat file] does not match the name and directory of the focal master file [fmf name] specified in the General tab. A mismatch in contents between focal master files could potentially produce erroneous results.

26. Code [some code] at offset [some time] is specified to have a modifier following it but is unmodified. Analyzing anyway.

27. Modifier [some code] at offset [some time] follows an already modified behavior. Ignoring modifier and analyzing anyway.

28. Code [some code] at offset [some time] is not defined in the focal master file associated with this data file. Ignoring code.

Log File Errors

Analysis

1. Offset [some time] matches or exceeds the duration of the data capture session [" + dataDuration + "].

2. No complete time bins to report on.

3. No complete or truncated time bins to report on.

4. The number of time bins calculated is [number of bins]. Greater than MAX_BINS is too many to attempt to score!

5. Code [some code] at offset [some time] is later than focal data EOF time. Skipping analysis.

6. Code [some code] at offset [some time] is earlier than previous entry. Skipping analysis.

7. Behavior Code [some code] in FMF but missing from focal analysis master [faf file]. Ignoring behavior.

8. Offset [offset to score from] matches or exceeds the duration of the data capture session [data capture duration]. Skipping analysis.

9. Scoring from first occurrence of code [behavior to score from] but this behavior is ignored. Skipping analysis.

10. Behavior to score from [behavior to score from] not found in focal data. Skipping analysis.

11. Length of focal data session [data capture duration] is less than or equal to scoring offset [offset to score from]. Skipping analysis.

Sequential Analysis – Comples Sequences

12. There are no codes to analyze in the data file [data file]. Skipping this file.

Sequential Analysis – Lag Sequential

13. There are no codes to analyze in the data file [data file]. Skipping this file.

Sequential Analysis - Markovian

14. There are no codes to analyze in the data file [data file]. Skipping this file.

Sequential Analysis - Runs

15. Data file [data file] has zero or more than two variables, where each variable consists of a single key code. Runs test cannot be performed. Skipping this file.

16. Unable to find the focal master file [fmf file] associated with this data file [data file]. Skipping analysis of this data file. Please place the focal master in the same directory with the data file.

Sequential Analysis - Sequence

17. There are no codes to analyze in the data file [data file]. Skipping this file.

Data Analysis

18. No results have been selected for reporting in this focal analysis master

19. Behaviors in the related focal master [fmf file] have changed.

20. Unable to save results to [results file]. [Exception message]

Combinations

21. Combinations master [cmf name] does not specify any codes to replace. No files were combined.

22. Unable to save combined data to [cmf file]. [Exception message]

Reliability

23. Encountered problems while loading one of the data files. Processing aborted: [exception message].

24. Specified .dat files do not begin with the same key code since at least one of them has no key code entries. Skipping analysis.

25. Specified .dat files do not begin with the same key code. Skipping analysis.

26. Unable to save results to [reliability file].

Sequential Analysis – All

27. Encountered problems while loading data file [file]. Skipping this file: [exception message]

28. Unable to save results to [sequential result file]. [Exception message]

Summary

29. Encountered problems while loading result file [result file]. File skipped.

30. Result file [result file] will not be summarized with other result files in the directory [user directory], since it has not been analyzed in the same way as the first result file in the directory. File skipped.

31. Nothing to summarize, since there are no result files in the specified directory.

32. Unable to save results to [summary file]. [Exception message]

Sequential Analysis – All but Runs

33. Unable to find the focal master file [fmf file] associated with this data file [data file]. Skipping analysis of this data file. Please place the focal master in the same directory with the data file.

15 Frequently asked questions about JWatcher

Last updated 11 Jan 06

Q: What is JWatcher?
A: JWatcher is an event-recording program written in Java, a computer language developed by Sun Microsystems that runs in both the Windows and Macintosh (and UNIX and LINUX…) operating environments.

Q: What is an event recorder?
A: Event recorders are tools to help quantify behavior. By logging key presses that signify the onset or occurrence of behaviors, event recorders allow you to estimate the time animals or people allocate to different activities. To learn more about quantifying behavior, read the outstanding book *Observing Behavior* (now in it's second edition) by Paul Martin and Pat Bateson (Cambridge University Press).

Q: How much does JWatcher cost?
A: JWatcher is free to use. Please tell others about it and use it in your laboratories and classes. To ensure you're using the correct version, download it directly from the JWatcher page (http://www.jwatcher.ucla.edu).

Q: Who developed JWatcher 1.0?
A: JWatcher was developed by Dan Blumstein, Janice Daniel, and Chris Evans with the generous support of the US National Institute of Health. While we are interested in bug reports (email to: marmots@ucla.edu), and we will update this FAQ periodically, we are currently unable to correspond with users about the routine workings of JWatcher. JWatcher comes with no warranties expressed or implied.

Q: What operating system does JWatcher require?
A: JWatcher 0.9 was developed in a Windows (98/NT) environment and tested extensively using both Windows (95/98/NT) and the Mac OS (System 8.5, 8.6 and 9.0). It should run on any computer that can run the Java Virtual Machine (also called the Apple Applet Runner in the Mac OS). It was developed to work on the Windows Java Runtime Environment 1.1.8 and the Mac OS Runtime For Java version 2.2.3. Needless to say, it runs faster and more smoothly on newer machines with a lot of RAM.

JWatcher 1.0 was developed with the J2SE and Java 1.4.2. It requires more recent operating systems (Windows XP or Mac OS-X).

Q: I have an older computer. Which version of JWatcher should I use? If you are running Windows 95/98/NT or Mac OS-9, you will have to use JWatcher 0.9. Newer computers can take advantage of the new features in JWatcher 1.0.

Q: I can score behavior but JWatcher freezes when I try to analyze it. What's happening?
A: You should use JWatcher's analysis routines on a computer with more RAM. Data Capture works well with limited RAM but analyzing or combining key codes work better with more RAM.

Q: How do I open/print a results file?
A: Results files (.res) are comma delimited text files. They can be opened within any word processing or spreadsheet programs.

Q: JWatcher does not redraw the screen properly and suddenly begins behaving badly. Why?
A: You may have too many applications open. Quitting JWatcher and closing un-used applications may solve this problem.

Q: How can I print my .faf to verify what it does?
A: In Focal Analysis Master, click on the [Summary] button, select all of the contents of the summary file using a mouse, and then copy (control + c) and paste the contents into a word processor.

Q: Why do I have to specify the path where I want files to be stored every time I restart JWatcher?
A: JWatcher does not create a preference file. Thus, the default location where files are created and stored is the folder where the application resides. When we use JWatcher, we store the application in the default installation location (in the applications folder), and create a folder elsewhere where we store JWatcher files. Each time we restart Jwatcher, we have to navigate over to that folder. Within a session, however, JWatcher remembers the last location used.

Q: How can I print my data files?
A: Data files (.dat) are comma delimited text files. They can be opened within any word processing or spreadsheet programs.

Q: How do I edit a pre-existing data file?
A: Use the Edit data function built into JWatcher, or edit using a word processor or spreadsheet program.

Q: If I analyze the same data file using first one and then another focal analysis master file, my results file is overwritten. How can I avoid this?
A: Select another destination folder into which the second batch of results can be put. By design, results are named by adding either cd.res or tr.res to an existing .dat file.

Q: When I re-size the JWatcher screen, some of the buttons disappear. Why?

A: JWatcher is designed to run on a 640 x 480 screen. Re-sizing it below that may cause some problems that should be easily reversed by making it bigger.

Q: My on-line help doesn't work when I'm running JWatcher.
A: We"ve had some documented problems with Netscape Navigator^{TM} running on a Macintosh recognizing our help files. Try downloading Microsoft's Internet Explorer and setting it as your default web browser. Additionally, depending on your memory allocations, JWatcher may not be able to run properly with a web browser. Print the JWatcher guide directly from the .pdf file provided and use it, rather than the on-line help.

Q: How much memory is needed to run JWatcher?
A: JWatcher 0.9 only requires about 2 mb of RAM. However, the Java Virtual Machine requires more and takes it. For best performance, we suggest having about 64 mb of RAM free when running JWatcher. Allocating different amounts of RAM to JWatcher and the Java Virtual Machine may result in strange performance. JWatcher 1.0 requires more RAM, and once again, the more RAM you have the more efficiently JWatcher operates.

Note: Strange performance in no way influences the results, but you may have to quit the program and restart it periodically.

Q: What is the .tr.res file?
A: The .tr.res file is a results file set up in a way that you can easily graph a "trace" of behavior. Behavioral traces show the onset and termination of bouts of behavior.

Q: When I want to quantify a behavior and treat it as both an event and a state, I hit the same behavior code more than once in a row. However, in state analysis, each key entry is counted as a bout with a 0 second interval between them. Is this correct?
A: Yes. Logically, each behavior turns itself off. Therefore, you must be careful to decide whether you want the same behavior code to be used to signify events within states or to solve this problem by assigning different codes to signify the state and event.

Q: I don't want to have to remove my hands from the keyboard to start recording behavior in Data Capture. Is there a way I don't have to use the mouse to click start a focal observation?
A: Yes. Using the tab key, move the focus to the start button. When the focus is on the start button it will be highlighted. Once highlighted, hitting the space bar will start the clock. Or, use alt-s.

Q: I'm working with a lot of slightly different .faf files. How can I keep track of them?

A: Data management is always difficult when conducting complex analyses. Copy the contents of the .faf summary into a word processor, add relevant comments, and print and save it.

Q: How can I view my .fmf file?
A: Open it and print it from a word processor.

Q: Why do I get unrecognized key codes in the analysis log file?
A: Either you have mistakenly typed an unrecognized key code or the .fmf used to score data is different from that used to create the .faf.

Q: How come I can't save my focal analysis master file (.faf file)?
A: Ahh, that nasty RAM again. Close other programs and/or try restarting JWatcher (but also see next question).

Q: It sometimes takes a long time to save my focal analysis master file (.faf file). Why?
A: This is a result of the .faf validation process in Version 1.0 that examines your file for logical inconsistencies before saving. It is most likely to occur on computers with limited RAM. Simply wait and your file will eventually be saved.

Q: It takes long time to open About. Why?
A: Same issue about RAM as above. Simply wait and the About file will eventually be opened.

Q: I opened my .dat file to edit it and received a message saying that the associated behaviors and modifiers are missing. What's up?
A: In order to edit data files, JWatcher must locate and load the .fmf used when scoring data. The .fmf must either be in the location it was when scoring data, or it can be in the same folder as the .dat file.

Q: I batch processed a number of files but in some cases no results files are produced. Why?
A: Check the .log files for errors. Files with serious errors will not be processed.

Q: I was given the error message: "Behaviors in the related focal master <name> have changed. Edit the focal analysis master to update." What do I do?
A: Reopen the .faf in the Focal Analysis Master tab. A dialog box will ask if it is OK to update your .faf file. Click OK to update.

Q: Are files scored in 0.9 compatible with 1.0 and later releases?
A: Data files scored with Version 0.9 are compatible for use with all applicable Version 1.0 routines; therefore, no rescoring of existing data files is necessary (but see exception below). Global definition and focal master files created by Version 0.9 should be opened and resaved in Version 1.0,

after verifying that no unintended alterations have occurred. Note that a number of key codes previously allowed in Version 0.9 global definition and focal master files are no longer able to be defined in Version 1.0 (see section 4 - Global Definition for more details - the six key codes are \ | : = , .). If your previous global definition and focal master files contain any of these codes, we suggest that you create entirely new files in the new version. Version 0.9 data files containing these codes will be able to be analyzed, but these codes will be ignored, and thus information may be lost. If your Version 0.9 data files intentionally contained any of these key codes then you will need either to edit them or rescore.
Focal analysis master files created by Version 0.9 must be updated within the new version before use to prevent unexpected results. Open, fix any inconsistencies, and re-save. All of your original selections should be preserved except for one case within the Event Analysis window (see below). Files not updated may still be analyzed, but we have not extensively tested these results. Event Analysis window: In Version 0.9, two analyses (event count and interval occurrence) were combined into one check box. In the new version, these two analyses are now separated into two independent check boxes. If you had selected the "Event Count and Interval Occurrence" box in Version 0.9, you won't get the "Event Count" box checked in the new version. Therefore, you must update this portion manually. Combinations master files created by Version 0.9 should not be used with Version 1.0. We suggest creating entirely new combination master files with Version 1.0, and rerunning the combine key codes routine. See the JWatcher 1.0 manual, section 9 - Combinations.

Q: Are version 1.0 files compatible with version 0.9?
A: Version 1.0 files are not compatible for use with Version 0.9.

Q: Are some key codes prohibited?
A: Yes, the six prohibited key codes are \ | : = , . This means that you should not use these to quantify behavior?

16 References

Bakeman, Roger & Gottman, John M. Observing interaction: an introduction to sequential analysis. Cambridge, U.K.; New York: Cambridge University Press, 1997.

Douglas, J. M. & Tweed, R. L. Analysing the patterning of a sequence of discrete behavioural events. Animal Behaviour, 1979, 27, 1236-1252.

Hailman, E.D & Hailman, J.P. UNCERT User's Guide. University of Wisconsin Zoology Department, Madison. 1993.
http://www.animalbehavior.org/Resources/CSASAB/

Martin, Paul & Bateson, Patrick. Measuring behaviour: an introductory guide. Cambridge, U.K.; New York: Cambridge University Press, 1993.

Sokal, Robert R. & Rohlf, F. James. Biometry: the principles and practice of statistics in biological research. 3rd edition. New York: W.H. Freeman, c1995 (1998 printing).

Van Hooff, J. A. R. A. M. Categories and sequences of behavior: methods of description and analysis (Chapter 7). In: Handbook of methods in nonverbal behaviour research. Cambridge University Press. 1982. Eds: K. R. Scherer & P. Ekman.