

USING JWATCHER TO STUDY MARMOT BEHAVIOR

UTILISER JWATCHER POUR ÉTUDIER LE COMPORTEMENT DES MARMOTTES

ИСПОЛЬЗОВАНИЕ JWATCHER ПРИ ИЗУЧЕНИИ ПОВЕДЕНИЯ СУРКОВ

Janice C. Daniel & Daniel T. Blumstein

Department of Organismic Biology, Ecology and Evolution, 621 Young Drive South, University of California, Los Angeles, CA 90095-1606 USA. Email: janice@ucla.edu, marmots@ucla.edu

JWatcher est un logiciel d'enregistrement factuel, distribué gratuitement et écrit en Java™. Il fonctionne pratiquement avec tout type d'ordinateur. Nous donnons un aperçu des caractéristiques du programme et deux exemples de son utilisation lors de recherches sur les marmottes, pour en illustrer les caractéristiques clés. **Mots clés** : quantifier le comportement des marmottes.

JWatcher является бесплатной системной программой написанной на языке Java™ и производящей фактуальную запись. Она работает практически на всех видах компьютера. Мы даем технический обзор этой программы и два примера ее использования при проведении исследований на сурках для иллюстрации ее основных характеристик.

Ключевые слова: количественное выражение поведения сурков.

JWatcher is a freely distributed event recorder software program written in Java™ that runs on virtually any computer. We provide an overview of the program's features, and two examples of how this program has been used in marmot research to illustrate some of the program's key features. **Keywords**: quantifying marmot behavior.

JWatcher (Blumstein *et al.* 2000) is a software program for the quantitative analysis of behavior. It is written in the Java™ programming language and can run on virtually any modern micro-computer. JWatcher can be freely downloaded from two websites. It can be used to address any problem that requires a complex sequence of actions to be scored by a human observer.

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 animal sampling with continuous recording (i.e. a single subject is viewed continuously for a predetermined amount of time; Martin & Bateson 1986). The real power of JWatcher is, however, in its analysis routines. These calculate time budgets and provide statistics about the duration of behavioral states and the intervals between them.

JWatcher is based on a 'score once, analyze many times' philosophy. 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 the 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). Only the number of alphanumeric keys available limits the number of states that can be tracked simultaneously. 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.

JWatcher's interface (Fig. 1) allows the user to 'tab' between screens. Each screen has a specific function. For instance, screens walk the user through the process of defining 'master files' that are used to define behaviors of interest and to

score behavior. While reasonably straightforward to use, JWatcher is very flexible, and some of this flexibility can lead to difficulty when interpreting results. We strongly suggest that prospective users read the 33-page manual before use.

We describe how we used JWatcher to study marmot behavior. The purpose is to introduce you to some of the key features of the program, and to illustrate, by example, how it might be useful for your research. We have used JWatcher for both observational and experimental studies.

In one (on-going) observational study, we sought to determine whether marmots modified the amount of time allocated to foraging and vigilance as a function of perceived predation risk. To do this, we conducted two-minute focal samples on foraging marmots and recorded our observations into a small hand-held tape recorder. We noted when marmots were quadrupedally foraging with their head on the ground, rearing and foraging in order to grasp vegetation above ground level, looking while quadrupedally standing, looking while rearing, looking while rearing up on their hind legs, self-grooming, engaged in social interactions, walking or running. We also noted when individuals were out-of-sight. Each 2-minute focal was scored once by playing the tape in the lab and typing the behavioral code at the onset of each behavior. We then used JWatcher to calculate time budgets over the entire focal duration. Because individuals could occasionally wander behind a rock or become obscured by tall vegetation, and we therefore could not be absolutely sure about their behavior when out-of-sight, we used JWatcher to calculate the proportion of time in-sight for our time budgets. This feature of JWatcher's enabled us to specify one of our keycodes as an 'out-of-sight' key; the time spent out-of-sight was then automatically subtracted from the total focal duration, and therefore our time budgets automatically adjusted. If we had

wanted, we could have excluded any behavior, not just time out-of-sight, from our time-budget calculations. For example, if we wanted to know what fraction of time was spent in each of the three vigilant postures, given that the individual was vigilant, we could have excluded all behaviors except those three. For our final analysis, we were more interested in the total time spent vigilant rather than how much time was spent in each posture. We therefore used JWatcher's 'combine' feature to combine the three codes into one new category, total vigilance. We also were able to collapse walking and running into one new category, locomotion, using this same feature.

We have also used JWatcher for several experimental studies in the field seeking to describe marmots' abilities to discriminate among the alarm calls of individuals, and other categories of callers. To do this, we constructed playback series of alarm calls that we then broadcast to marmots foraging on bait at a set location. For these experiments, we videotaped the responses of marmots to our playbacks. In general, for these experiments, we videotaped the subjects for one minute of baseline foraging prior to the playback, during the one-minute playback itself, and for one minute post-playback. The tapes were scored back in the lab where we logged the onset of each bout of foraging, looking while quadrupedally standing or while rearing or while rearing up, walking, running, grooming, as well as time spent inside the burrow.

For these experiments, we found JWatcher's time-windowing function to be particularly useful.

We scored each playback series once as a three-minute focal sample. We then were able to easily recalculate time budgets over a variety of time intervals (we examined 10s, 15s, 20s, 30s, and 1 min intervals) in order to find the most meaningful one for our analyses. Re-scoring the focal sample was not necessary.

In both of the above instances, we calculated time budgets only. However, we could also have calculated other statistics such the total time engaged in the behavior, or the average bout length (and standard deviation Fig. 2). Each behavior was treated as a 'state', with a duration determined by discrete onset and offset points. Instead, we could have specified a behavior to be an 'event,' with no associated duration, and then calculated its frequency of occurrence.

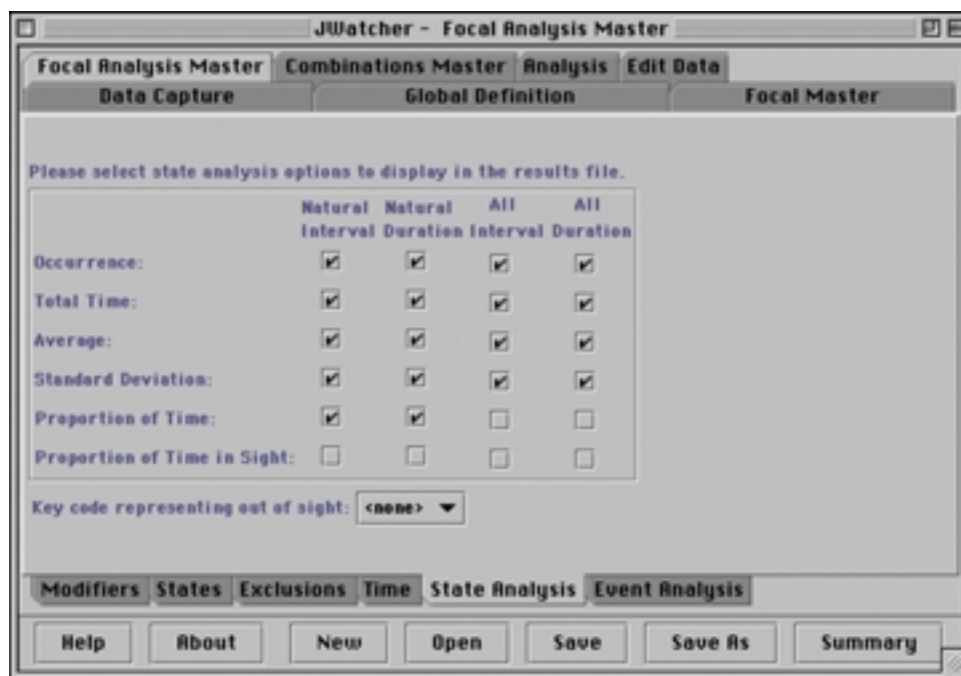
For anyone interested in bout length of a behavior (or in the intervals between behaviors), JWatcher has a powerful feature, which enables the user to exclude bouts of behavior from analysis that are 'truncated' by either the beginning or end of a time bin, or by the subject going out-of-sight. Thus, only naturally-occurring durations (or intervals) are included in the calculations (Fig. 2).

In both of the above examples, we either videotaped or tape-recorded a verbal transcript of behavior in the field, and subsequently scored these with JWatcher back in the lab. It is also possible to directly score 'live' behavior in the field with a laptop computer. A hand-held recording device is planned for future development.

Figure 1. JWatcher's interface allows the user to 'tab' between screens. The 'Data Capture' window is shown in which a user presses key codes and JWatcher logs the time when they are pressed. *L'interface de JWatcher permet à l'utilisateur de naviguer entre les écrans. La fenêtre affichée « Data capture » permet d'appuyer sur les touches codées et à JWatcher d'enregistrer le moment.*



Figure 2. The 'Focal Analysis Master' window is where a user specifies exactly which statistics are to be calculated. There are several sub-windows: the 'State Analysis' window is shown here. Statistics for 'natural' durations (or intervals) exclude those durations (or intervals) that are 'truncated' by the beginning or end of time bins, or by a subject moving out-of-sight. *La fenêtre « Focal Analysis Master » permet de spécifier les statistiques à calculer. La fenêtre option « State Analysis » est présentée. Les durées ou les intervalles de séquences tronquées soit au début soit à la fin soit par un épisode hors de vue du sujet, sont exclus.*



JWatcher (Blumstein et al. 2000), logiciel d'analyse quantitative du comportement est écrit en JavaTM. Il fonctionne pratiquement sur tous les ordinateurs. Il est téléchargeable gratuitement sur 2 sites. Il peut être utilisé pour tout problème qui demande l'enregistrement par un observateur humain d'une suite complexe d'actions.

La fonction la plus simple de JWatcher est d'enregistrer des faits en notant le moment auquel les touches sont enfoncées. Il peut réaliser un échantillonnage focalisé continu (le même sujet est observé pendant une période de temps prédéfinie ; Martin & Bateson 1986). L'intérêt de JWatcher réside dans ses programmes qui calculent bilans et statistiques sur la durée des états comportementaux.

La philosophie de JWatcher est de noter une fois, et d'analyser plusieurs fois. Ainsi, le même fichier de données peut être utilisé pour calculer le budget-temps à la fois pour des périodes courtes et longues, ou pour repérer les variations dans divers sous-groupes de comportements observés. Ceux-ci peuvent être analysés, comme un événement instantané (fréquence) ou comme un état (durée totale). Leurs relations logiques peuvent être réexaminées (un état comportemental donné peut être redéfini comme indépendant ou comme exclusif par rapport aux autres). Le nombre d'états n'est limité que par le nombre de touches alpha-numériques disponibles. L'utilisateur peut choisir d'exclure ou d'ignorer certains comportements dans les analyses, ou de combiner des catégories pour en générer de nouvelles. Des analyses groupées permettent des traitements

efficaces.

L'interface de JWatcher (Fig. 1) permet de passer d'un écran à l'autre, chacun ayant une fonction spécifique. Ainsi, les écrans guident l'utilisateur lors du processus de définition des "dossiers maîtres" des comportements intéressants, et lors de leur enregistrement. D'utilisation relativement aisée, JWatcher est souple, mais cette souplesse peut entraîner des difficultés d'interprétation des résultats. Nous recommandons aux utilisateurs potentiels de lire le manuel avant usage.

L'objectif est de faire découvrir quelques éléments clés de JWatcher, et d'illustrer son utilité, aussi bien pour des études expérimentales que d'observations du comportement des marmottes.

Lors d'une étude d'observation, nous avons cherché à déterminer si les marmottes modifiaient le temps alloué à l'alimentation et à la vigilance en fonction du risque de prédation perçu. Des échantillonnages focaux de 2 minutes de marmottes s'alimentant étaient enregistrés sur un magnétophone portable. Les comportements suivants ont été notés : alimentation à quatre pattes et tête au sol, ou tête relevée pour saisir la végétation au-dessus du sol ; observation à quatre pattes, en relevant la tête, ou en se dressant sur les pattes arrières ; auto-toilette ; interactions sociales ; marche ou course et quand les individus étaient hors de vue. Chaque échantillon a été décodé en écoutant l'enregistrement au laboratoire et le code entré au début de chaque nouveau comportement. Puis, JWatcher a calculé les budgets-temps de la période échantillonnée. Les animaux

pouvant être, parfois, hors de vue (masqués par un rocher ou par la végétation), et leur comportement indéterminable, JWatcher peut calculer la proportion de temps pendant lequel les animaux sont visibles. Cette caractéristique permet de définir l'option "hors de vue"; le temps passé hors de vue est alors automatiquement soustrait de la durée totale de l'échantillon et les budgets-temps recalculés. Nous pouvons, à volonté, exclure de nos calculs, n'importe quel comportement. Ainsi, pour connaître la fraction de temps consacrée à chacune des trois postures de vigilance, à condition que les sujets aient été vigilants, il est possible d'exclure tous les autres comportements. Pour l'analyse finale, nous avons réuni ces trois postures en une seule catégorie : la vigilance totale grâce à l'option "combine" de JWatcher. Comme, nous pouvons aussi réunir la marche et la course en une seule catégorie : le déplacement.

Pour tester, expérimentalement sur le terrain, les capacités des marmottes à discriminer les cris d'alarme individuels et des autres catégories d'émetteurs, nous avons enregistré des cris d'alarme qui sont ensuite réémis auprès de marmottes s'alimentant à des appâts dans un site donné. Leurs réponses sont filmées en vidéo durant une minute avant, une minute pendant et une minute après la retransmission. Les bandes vidéo sont analysées au laboratoire et le début de chaque comportement de l'animal est noté. Pour ces expériences, la fonction de temporisation de JWatcher est très utile. Après avoir analysé chacune des séries de trois minutes, nous pouvons alors aisément recalculer

les budget-temps de n'importe quel intervalle de temps (les intervalles 10 s, 15 s, 20 s, 30 s, et 1 minute ont été examinés) de façon à déterminer l'intervalle le plus significatif pour nos analyses, sans qu'un réexamen des échantillons focaux ne soit nécessaire.

Dans les exemples précédents, nous n'avons calculé que les budget-temps. Mais, nous aurions pu calculer la durée totale, ou la durée moyenne des bouts de comportement (et l'écart-type, Fig. 2). Chaque comportement est traité comme un "état", sa durée est déterminée à partir de son début et de sa fin. Au lieu de cela, nous pouvons définir le comportement comme un "événement", sans durée associée, et calculer sa fréquence.

Pour ceux qui sont intéressés par la durée des séquences comportementales (ou par les intervalles entre les comportements), JWatcher possède une fonction qui permet à l'utilisateur d'exclure de l'analyse les bouts tronqués au départ ou à la fin, ou par sortie du champ d'observation du sujet. Ainsi, seules les durées entières (ou les intervalles) sont pris en compte lors des calculs (Fig. 2)

Dans les exemples précédents, nous avons enregistré ce comportement sur le terrain soit sur un magnétoscope, soit sur un magnétophone, et ensuite analysé ces documents au laboratoire. Il est aussi possible de décrypter directement le comportement sur le terrain avec un ordinateur portable. Le développement d'un appareil d'enregistrement portable est envisagé dans le futur.

BIBLIOGRAPHIE / REFERENCES

BLUMSTEIN D.T, EVANS C.S. & DANIEL J.C. 2000. JWatcher 0.9: An introductory user's guide. <http://www.jwatcher.ucla.edu> & <http://galliform.psy.mq.edu.au/jwatcher>.

MARTIN P. & BATESON P. 1986. Measuring Behaviour: an introductory guide, 2nd edition. Cambridge University Press. Cambridge, UK.