20

A Cognitive Neurogenetics Screening System with a Data-Analysis Toolbox

C.R. Gallistel[1], *Fuat Balci*[2,1], *David Freestone*[3], *Aaron Kheifets*[1], *and Adam King*[4]

[1] *Department of Psychology, Rutgers University, New Jersey, USA*
[2] *College of Social Science and Humanities, Koç University, Istanbul, Turkey*
[3] *Center for Neural Science, New York University, New York, USA*
[4] *Department of Mathematics & Computer Science, Fairfield University, Connecticut, USA*

Cognitive neurogenetics is a genetics approach to the cellular and molecular neurobiology of cognitive mechanisms. It is predicated on the discovery (forward genetics)or production (reverse genetics) of mutant strains with a genetically based abnormality in a basic mechanism of cognition. The key to a successful prosecution of this approach is an effective behavioral screen for specific functional effects from genetic manipulations. An effective screen should have the following properties:

- It should focus on specified mechanisms of cognition (e.g., odometry, interval timing, circadian timing), not on categories of phenomena (e.g., spatial learning, temporal learning, fear learning) or on experimental procedures (e.g., fear conditioning).
- It should deliver functionally specific measurements, measurements that reflect the functioning of one particular mechanism, not the many different mechanisms that contribute to the measured strength or duration of a behavior.
- It should give a clear phenotype for each animal screened, not a group average.
- It should give physiologically meaningful measurements (e.g., the period of a circadian clock, the precision of interval timing), not quantities that have no physiological meaning (e.g., responses per minute).
- It should make it possible to screen large numbers of animals in reasonable periods of time with a reasonable investment of investigator/technician time and cost, and with minimal handling of the mice.
- It should archive the data in such a way that others can explore it in the light of future discoveries that pose new questions using old data. Put another way, it should contribute to a publically available and useable behavioral phenotype database.
- It should produce an intact data trail that enables others to trace the connection back from published summary statistics and graphs to the raw data.

We have developed and successfully used such a system. A major part of the development was the creation of an open source Matlab™ toolbox that facilitates the analysis of the rich time-stamped event records that modern computerized behavioral testing generates. The toolbox puts in the hands of investigators with limited programming experience the ability to extract from such records sophisticated statistics using only a few high level commands. The design of the toolbox ensures an intact data trail. Before presenting the system and the toolbox and some of the results obtained using them, we elaborate on the properties of an effective cognitive phenotype.

Handbook of Neurobehavioral Genetics and Phenotyping, First Edition. Edited by Valter Tucci.

Mechanisms, Not Procedures

In the experimental literature on the neurobiology of learning and memory, there is a tendency to assume something like a one-to-one mapping between various experimental procedures and underlying mechanisms. Classical (aka Pavlovian) conditioning and instrumental (aka operant) conditioning are different procedures. The behavioral changes produced by these two procedures are often assumed to depend on two different underlying mechanisms. Fear conditioning and appetitive conditioning are different procedures. These differences in procedures are sometimes taken to reflect different underlying mechanisms, so that one speaks of the mechanism of fear conditioning versus the mechanism of appetitive conditioning. Delay conditioning – teaching the subject to fear a brief light or tone – and context conditioning – teaching a subject to fear the experimental chamber – are also different procedures, and here again, investigators speak as if the learning that occurs in these different procedures were mediated by different mechanisms. And, for a final example, even such subtle differences in procedure as the difference between delay and trace conditioning procedures lead some investigators to write as if these procedures reflected the functioning of two different mechanisms.

AQ2 In our view, no learning procedure maps to any single underlying mechanism. They all reflect the operation of many different mechanisms, where by mechanism we understand some reasonably self-contained machinery whose principal components are specified by a modest number of genes. The Krebs cycle is one example of what we have in mind by "mechanism". We mention it because it is, on the one hand, a fairly complex chunk of molecular biological machinery, but, nonetheless, it is a self-contained functional unit, whose basic components are specified by a modest number of genes (on the order of 20). The circadian clock is a more behaviorally relevant example.

Mechanisms are clearly distinct from traits. Height is a trait, not a mechanism, and we mention it because we doubt that anyone imagines that there is a corresponding height mechanism, whereas we do imagine that there is a mechanism underlying aerobic metabolism. The behavioral measures commonly made in the above mentioned learning protocols are analogous to measurements of traits, not measurements of signature quantitative properties of underlying mechanisms.

Associative learning is a procedure insofar as it presents to the subject two distinct events (e.g., tone onsets and shock onsets) that are, at least in the mind of the investigator, temporally paired. However, associative learning is almost universally taken to be not just a procedure but also a mechanism. From an historical perspective, the belief in association formation as a basic mechanism of learning gave rise to the experimental procedures. They were devised in order to determine the laws that governed the operation of the presumed mechanism. One reason for questioning whether the procedures cited earlier (classical, instrumental, delay, trace, context, appetitive, and fear conditioning procedures) each depend on a distinct underlying mechanism is that the behavioral changes in all of these procedures are generally taken to be instances of associative learning. If that is true, and if association formation is in fact a mechanism, then those other procedures presumably measure various manifestations of one and the same underlying mechanism, not different mechanisms.

Functional Specificity

AQ3 Genetic alterations are notoriously pleiotropic. Therefore, one wants to be as confident as conditions permit that one's measure reflects primarily (or, in the best cases, exclusively) the properties of a single target mechanism. One does not want it to be the case that many different,

quite unrelated mechanisms could affect one's measurement in the same way. In short, there is a salient competence/performance distinction to be made. To make this distinction, it is important to be able to show that many other measures that also clearly involve basic mechanisms of cognition are not affected by the genetic manipulation. The reasoning here is that these different mechanisms will all invoke highly overlapping performance mechanisms when their operation is translated into observable behavior, so that if an altered measurement arises from changes in those performance factors rather than in the mechanism whose characteristics one intends to measure.

No Group Averages

It is not uncommon for investigators to average the trial-by-trial performance of the –/+ mice and of the +/+ mice in a learning protocol and compare the two group-average learning curves. The rationale for this is that the trial-by-trial performance of individual subjects within a group of a given genotype is highly variable. The averaging is intended to take out this "noise" and reveal the underlying "signal". The problem is that if the underlying signal (the time course of the change in performance) is different in each of the subjects, then the resulting average is meaningless. In particular, as was pointed out more than half a century ago(Estes 1956), if each individual subject makes a step transition in performance (one trial learning) but different subjects make steps of different sizes after differing numbers of trials, then averaging across subjects produces a smooth, continuous, negatively accelerated learning curve that bears no resemblance to the learning curve in the individual subjects. Examination of learning curves in both human and animal subjects in a wide variety of experimental paradigms shows that this is often the case (Estes 1956; Rock 1957; Estes 1960; Gallistel, Balsam *et al.* 2004; Roediger and Arnold 2012). Therefore, it is essential to compute summary statistics for each subject rather than after averaging across subjects. And, it is best practice to report not simply the mean and standard deviation of these summary statistics but rather the entire distribution. This is easily done by giving cumulative distributions instead of bar graphs. The cumulative distributions of the statistics from each subject within a group take up no more room, make visual comparisons at least as easy, and convey much more information than do the bar graphs commonly used to compare group means.

Physiologically Meaningful Measures

AQ4 A physiologically meaningful behavioral measure is a measure extracted from behavioral data that may be compared to neurobiological measures. The paradigmatic example is the scotopic spectral sensitivity curve of the human observer. This measure may be compared to the absorption spectrum of rhodopsin in the outer segments of rods (a photochemical measurement). The fact that the *in situ* absorption spectrum of rhodopsin superimposes on the behaviorally determined spectral sensitivity curve is an almost conclusive proof that the mechanism of light transduction in the dark-adapted observer is the isomerization of rhodopsin (because absorption spectra are molecular signatures; every compound has a different signature). Another example is the behavioral measurement of the period and phase of the free-running circadian clock. These measures may be compared to the measurement of, for example, the period and phase of the transcription of the period gene in the suprachiasmatic nucleus, as visualized by fluorescence. Other examples are behavioral measurement of the refractory periods and conduction velocities of the reward neurons in the medial forebrain bundle of the rat (Gallistel, Shizgal

et al. 1981; Shizgal and Murray 1989). Examples of behavioral measures that cannot be compared to neurobiological measurements are the mean time that a mouse takes to swim to the hidden platform in a water maze or the percentage of time that it spends in the quadrant where the hidden platform used to be.

An example of a physiologically meaningful behavioral measure in the study of the associative learning mechanism is the measurement of the width of the window of associability (cf Gluck and Thompson 1987): How close in time do two events have to be in order for an association to form? Schneiderman and Gormezano (1964) made such measurements in the rabbit eye-blink preparation. They measured variation in the trials to the acquisition of a conditioned blink as a function of variation in the interval between the conditioned stimulus (CS for short) and the unconditioned stimulus (US). Their results suggested that associability peaked when US onset followed CS onset at a latency of 0.5 s and fell to near 0 when the CS–US interval was increased to 2 s. In the early publications on the associative conditioning of the sensory synapses on *Aplysia* gill-withdrawal motor neurons, it was reported that the electrophysiologically measured process of alteration in synaptic conductance showed a quantitatively similar window of associability. This was taken as evidence that what was being measured at those *Aplysia*synapses was *the* mechanism of association(Hawkins, Carew *et al*. 1986). In other words, the measurements made by Schneiderman and Gormezano in the rabbit eye-blink preparation were taken to be those of an association-forming process at the synaptic level that was quantitatively the same in *Aplysia* and the rabbit. This strongly suggested an essentially universal, evolutionarily conserved synaptic mechanism of association formation.

Although the Schneiderman and Gormezano measures continue to be cited in recent electrophysiological work on the cerebellar substrate for eye-blink conditioning (Kalmbach, Voicu *et al*. 2011), their results are now known to be not generally valid. Schneiderman and Gormezano failed to take account of the critical role of the US–US interval in determining the effect of the CS–US interval on the rate at which conditioning progresses (Gallistel and Gibbon 2000). Increasing the US–US interval in rabbit eye-blink conditioning can reduce trials to acquisition by two orders of magnitude. It can render conditioning with a 2 s CS–US interval considerably faster than conditioning with a 0.5 s CS–US interval measured by Schneiderman and Gormezano(Brelsford and Theios 1965; Levinthal 1973; Levinthal, Tartell *et al*. 1985).

AQ5 More generally, the data on the window of associability for long-term potentiation (LTP) and long-term depression (LTD) and other quantitative properties of processes that modify synaptic conductances are in wide disagreement with the behaviorally measured properties of association formation (Gallistel and Matzel 2013; Hesslow, Jirenhed *et al*. 2013). This is a reason for skepticism about the claim that these synaptic processes are the mechanism of association.

A more promising example of a physiologically meaningful measure is the measurement of the accuracy and precision with which the conditioned response (CR) is timed. It has, for example, long been known that the conditioned eye-blink is like other conditioned responses in that the latency between CS onset and the conditioned response is proportional to the CS–US interval, with the result that the blink occurs at the time at which the US is expected, that is, it protects the eye at just the moment at which experience indicates that protection is most needed(Schneiderman and Gormezano 1964; Kehoe, Graham-Clarke *et al*. 1989; Kehoe and Napier 1991; White, Kehoe *et al*. 2000). As with the timing of other conditioned responses, the trial-to-trial variability in the latency of the CR increases in proportion to the mean latency (White, Kehoe *et al*. 2000). This is an example of the ubiquitous scalar variability in CR timing (Gibbon 1977). Scalar variability in response timing is an aspect of the more general phenomenon of time scale invariance in conditioning (Gallistel and Gibbon 2000).

The Purkinje cells in the second layer of the cerebellar cortex play a critical role in the conditioned eye-blink(Jirenhed, Bengtsson *et al*. 2007). Signals generated by the CS arrive at the

Purkinje cell by way of the parallel fibers. These parallel fibers are the axons of the tiny densely packed granule cells in the lowest of the three layers of the cerebellar cortex (the granular layer). The axons from the granule cells rise upward through the overlying two layers to branch in the top layer, forming the dense parallel fiber system that runs along the folds of the cerebellar cortex. The enormous, flattened dendritic tree of the Purkinje cells straddles these parallel fibers, so that each Purkinje cell gets inputs from tens of thousands of granule cells. By contrast, the signal from the US arrives at a Purkinje cell by way of a single climbing fiber from the olivary nucleus.

AQ6 The Purkinje cell has a high rate of spontaneous activity, even when the excitatory input from the parallel fibers is blocked (Cerminara and Rawson 2004). Stimulating pulses from an electrode on the mossy fibers driving the granule cell input to a given Purkinje cell elicit an additional spike at a 1–4 ms latency following each pulse, with a probability of about 0.5 (Hesslow, Jirenhed *et al*. 2013, Figure 3C).

When the onset of mossy fiber stimulation repeatedly predicts a stimulus to the climbing fiber innervating the same Purkinje cell, after a fixed CS interval in the range from 150 ms to at least 400 ms, there eventually appears a pronounced pause in the firing of the Purkinje cell. The conditioned pause lasts only as long as the CS–US interval, even on probe trials with no US, and even though the CS stimulation of the mossy fibers continues for several hundred ms after the end of the CS–US interval(Jirenhed, Bengtsson *et al*. 2007; Svensson, Jirenhed *et al*. 2010; Jirenhed and Hesslow 2011; Jirenhed and Hesslow 2011; Wetmore, Jirenhed *et al*. 2014). In other words, the output of the Purkinje cell shows a timed conditioned response. Therefore, it is possible to compare the accuracy and precision of the electrophysiologically measured pause in Purkinje cell firing to the accuracy and precision of the behaviorally measured timing of the conditioned response (although this has yet to be done).

Because the scalar timing of conditioned responses is a ubiquitous feature of conditioned behavior, extending over orders of magnitude variation in the CS–US interval, measuring the mean and coefficient of variation in CR latenciesis a promising means of forging strong links between the behavioral phenomenon and the underlying neurobiological mechanisms. So, too, are other timing-based measures, such as the function relating trials to acquisition to the ratio between the CS–US interval and the average US–US interval (Gallistel and Gibbon 2000; Balsam and Gallistel 2009; Ward, Gallistel *et al*. 2012).

Importance of Large-scale Screening and Minimal Handling

A genetic approach, particularly a forward genetics approach, requires screening mutant mice on a large scale. Practical, large-scale behavioral screening, like large-scale gene sequencing, requires highly automated methods that make minimal demands on human labor. Because handling of the mice is highly stressful to them, and because stress has ubiquitous behavioral and physiological effects, the screening method should minimize the handling of the subjects while maximizing the amount and number of physiologically meaningful quantitative measures obtained.

AQ7

Utilizable Archived Data with Intact Data Trails

Phenotyping data are like genotyping data in that when properly gathered and archived they are likely to be useful well into the future, in ways not foreseen when they were first gathered and analyzed. Large-scale highly automated phenotyping generates massive electronic data

files. Extracting meaningful results requires extensive computerized analysis of the raw data. Investigators who direct a laboratory that depends on computerized analysis of masses of raw data are likely to be familiar with the obstacles to the maintenance of clear data trails under these conditions. The computer code to do the analysis is often produced by several different programmers, with varying degrees of skill at code development and error checking. The code is often poorly documented and hence almost unusable by anyone but the author, who may have left the lab years ago. There is an accumulation of results files, produced by code at varying stages of development, applied to varying subsets of the data. It often becomes unclear which code applied to which data generated which results, which is to say that the trail from published summaries and graphs back to the raw data becomes messy and obscured. It is therefore highly desirable that the data-analysis environment be so designed that all the results, both intermediate, and published, remain together in a single file with the raw data from which they were extracted, organized in such a way as to make clear the trail from the published results back to the raw data and in such a way as to give subsequent investigators in other labs effective access to the raw data and various intermediate statistics extracted from it.

The System

The system is schematized in Fig. 20.1. The salient features are:

- Each mouse lives 24/7 in a test environment consisting of a standard polypropylene tub connected to a Med Associates™ mouse test chamber by an acrylic tube. It moves back and

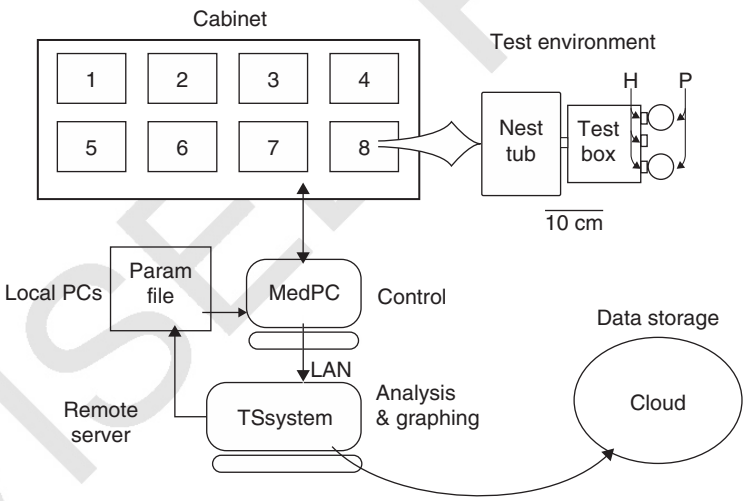


Figure 20.1 Schematic of the automated cognitive neurogenetics screening system. Each mouse lives in a test environment consisting of a polypropylene nest tub connected to a Med Associates™ mouse test chamber by an acrylic tube. Eight environments are housed in a steel cabinet 1.22 m wide, 0.61 m deep and 1.98 m high. A Windows™ PC running MedPC™ controls two cabinets (16 test environments). A local area network (LAN) connects the experiment-control computers to a remote server running a data-analysis and graphing program written in TSsystem©, an open source Matlab™ toolbox. The control computers update the data files for each mouse every 10 minutes. The server reads those files every 15 minutes, copying the data into files that live "in the cloud" as well as on the server's hard disk. Two or more times per day, the TSsystem© software analyzes and graphs the data, giving investigators quasi real time access to the progress of each mouse wherever and whenever there is internet access. The server software applies investigator-specified decision code to the incoming data to decide when a mouse has completed a given protocol. If so, it automatically advances the mouse to the next protocol by writing to a parameter file (Param File) that is read by the MedPC™ code every few minutes. *Source:* Gallistel, Balci *et al*. 2014.

forth between the tub and test chamber at will, obtaining its daily ration of food pellets by performing the programmed tasks.

- There is no handling of the mouse in the course of testing it with several different protocols
- Each mouse is individually advanced from protocol to protocol by decision software that applies user-specified decision criteria to the incoming data from the current protocol.
- The TS system software is so designed that the raw data, the MedPC code that controlled the protocols, and all of the results derived from the raw data are automatically stored in a single Matlab™ structure, insuring an intact data trail and greatly facilitating the archiving of the experimental results in a form accessible by and intelligible to others.
- Commercial cloud-based automatic file synchronization provides state-of-the-art back-up, off-site storage, and ready access to the latest results for investigators in the lab and off site.

The Toolbox

The principles that guided the development of the toolbox are:

- *Keep it all together*. The raw data and the results extracted from them should be processed in such a way that they are inseparable one from the other and from the code that governed the operation of the testing equipment and the logging of the data when the data were gathered. This latter process-control code is in essence the protocol for the experiment that generated the data.
- *Make a clear trail*. It must be possible to regenerate the published analyses from the raw data, that is, the trail from the raw data to the figures and tables and numerical values that appear in the published reports should be retraceable without difficulty.
- *Make complex analyses easy.* One can pose many different questions when one has a rich time-stamped database, discovering important results that were not foreseen when the experiment was designed. However, this is only likely to happen when it is relatively easy and quick for an appropriately trained researcher to pose a question to the data and quickly get an answer. If one has to write more than a few lines of code to get an answer to a new question, there will be many fewer questions posed to the data.
- *Better buy than build*. The analytic software should be embedded in a powerful general purpose, widely used, securely and extensively supported programming, statistics, and graphing system, such as Matlab™, Python, Mathematica™, or R.

In conformity to the last principle, we decided to have our data-analysis software take the form of a Matlab™ open source toolbox.

AQ8 In conformity to our first principle – *keep it all together* – the software system puts the raw data and all the results that come from analyses of that raw data into a single "structure". A structure in Matlab™ is a data type that gives flexibility and intelligibility in organizing vast and diverse data, while making all the data, both numerical and textual, accessible to computation. A Matlab™ structure is a hierarchically structured set of data fields with user-chosen names. The flexibility comes from the hierarchical structuring and from the fact that there are no restrictions on what can be put into a field; one can put in anything from a single number to a long text to another complex structure (that is, structures may be embedded within other structures). The intelligibility comes from the hierarchical arrangement of numerically indexable fields and from the fact that the user creates the field names, just as they would create headings in a spreadsheet.

The accessibility comes from the fact that the hierarchically structured field names are used to access the data to be processed. For example, the command:

```
mean(Experiment.Subject(3).Session(5).VisitDurations)
```

computes the mean of the column (that is, field) of visit durations during the fifth session for the third subject. "Experiment" is the name of the entire structure. "Subject" is an indexed field, with a different index number for each subject. Under each "Subject" field, there are indexed "Session" fields. Each of these session fields often has arrayed under it a large structure with many additional layers of fields. In this simple example, one of those fields gives the durations of successive visits to some location of interest, such as a feeding hopper. As this command illustrates, the data are accessed by way of the field hierarchy.

A Matlab™ structure is analogous to the hierarchically organized data arrays that users create in spreadsheets. The headings above the columns of numbers in a spreadsheet tell you what the numbers in a column represent and the hierarchical arrangement of the headings tells you how the columns of numbers are related to one another. Spreadsheet treatment of voluminous time-stamped data is not feasible. There are too many columns; they are too long; the hierarchical structure is too complex; the computations provided by the spreadsheet are not diverse and powerful enough; those provided take much too long when they operate on really large masses of data; and the graphic presentation resources are not sufficient.

The numbers specifying visit durations in the above example are not in the raw data. The durations of visits, like essentially all numbers of any interest, must be computed from the raw data. These computations generally begin by finding sequences of (generally non-contiguous) events. Whenever a sought-for sequence is found, a statistic, such as an event-duration, is computed from the time stamps associated with the events that compose the sequence. The statistic is stored within the same structure that contains the data from which it was computed – in a field created by the user. Suppose, for example, that one wants to know the durations of successive pokes into a feeding hopper. The onset of each poke is indicated by a time-stamped beam interruption. The offset of the poke is indicated by the next occurring beam-completion event (when the mouse withdraws its head from the hopper). This latter event will often not be the next event in the sequence of recorded events. Other events, such as pellet deliveries or light offsets or onsets, may intervene. To compute the duration of single poke, the data-analysis program must find the onset event, find the next later offset event, and then subtract the time stamp of the former from the time stamp of the latter. It must do this for each of the thousands of pokes that may well occur in the course of 24 hours.

Behavioral events have hierarchical structure. A sequence of pokes into the same hopper uninterrupted by any events that happened elsewhere (for example, a poke into another hopper or an exit from the test box) constitutes a visit event. The duration of the visit event encompasses the durations of all the pokes and interpoke intervals that comprise that visit. The data analysis code must make it easy to abstract multiple levels of structure from the sequence of time-stamped microevents, and compute statistics for each level of behavioral structure. Our system accomplishes this by allowing users to define different kinds of "trials" in extremely versatile ways and then to look for substructure within a kind of trial.

AQ9 Our concept of a trial was introduced in scare quotes, because, although it was inspired by the practice common in learning experiments of organizing events into trials, we have generalized the notion to any sequence of events or even any of several disjointed such sequences. Thus, "trial" simply means "interesting stretch of data". To a good approximation, any sequence of non-contiguous events and non-events that a human record scorer could be instructed to look for can define a trial and become the basis for the automated parsing of the raw data into "trials". For example, a visit to Feeding Hopper 1 might define a trial. It might be defined by the OR of the following two sequences:

[PokeOn1 PokeOff 1 –-PokeOn1 PokeOn2] *or* [Poke On1 PokeOff1 –-PokeOn1 Tube1]

The first sequence detects the mouse coming to Hopper 1, making a sequence of pokes, and then going to Hopper 2, while the second detects the mouse coming to Hopper 1, making a sequence of pokes, and then leaving the test box. The detection of either sequence detects a visit to Hopper 1. Figure 20.2 illustrates the progress of the computation that detects the first of these two trial-defining defining sequences. Detection of either sequence serves to define (this particular kind of) trial.

The sequence of trials of any given kind forms an indexed array of fields under the field for that kind of trial. Suppose, for example, that we have defined two kinds of trials, Hopper1_Visits and Hopper2_Visits. Suppose further that there were 48 visits to Hopper 1 and 27 visits to Hopper 2 in a given session (say, Session(3)) by a given subject (say, Subject(8)). Then, subordinate to "Experiment.Subject(8).Session(3).TrialHopper1_Visits", one would find:

Trial (1)
Trial (2)
•
•
•
Trial (48)

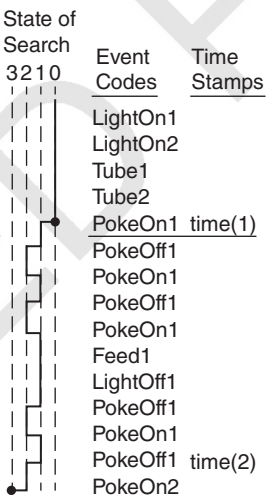


Figure 20.2 The progress of the computation that detects one of the two different sequences that define a visit to Hopper 1 (see text for the defining sequences) is illustrated by the steps in the solid line to the left of the hypothetical sequence of events. When the first event in the definition is encountered, the search steps from the 0 state to the 1 state; when the second is encountered, it steps to the 2 state. When a negating event (in this case, PokeOn1) is encountered, it steps back to the previous state. When the third (positive) event is detected while the search is in State 2, it steps to State 3, which, in this example, is the terminal state. Attainment of a terminal state signifies the discovery of a stretch of data constituting a user-specified "trial" of a particular kind. The user may specify any number of different kinds of trials. Any one kind may be defined by the ORing of any number of different trial-defining sequences. A trial-defining sequence may have several negating events in between any two positive events anywhere in the sequence. Encountering any one of the negating events steps the search back to its preceding state. Once that backward step has been made, further encounters with negating events in that same sequence of negating events will not step the search back to a still earlier state. All of the events have time stamps, but we indicate here only the two that may be used to compute the duration of the visit.

and subordinate to "Experiment.Subject(8).Session(3).TrialHopper2_Visits", one would find:

Trial (1)
Trial (2)
•
•
•
Trial (27)

Each of these indexed subfields (one for each instance of a given kind of trial in a given session with a given subject) would itself contain a structure of user-specified fields. These fields would contain user-specified statistics computed from the sequence of the events falling between the initial event and the final event in one of the sequences that defined that kind of trial – all of the events in that stretch of data, not just the events used to define the trial, that is, to pick out that stretch of data.

Core Commands

AQ10 The TS (for time-stamped) toolbox has over 50 commands. However, the core commands, which do most of the work, and which the user must master in order to use the toolbox, number only eight. Several of these functions routinely call helper functions. A helper function

AQ11

- *TSmatch.* This is a largely hidden workhorse. The user rarely calls it directly, but other core commands often call it. It does the above-described searching for sequences of not-necessarily-contiguous events. TSmatch can search simultaneously for more than one sequence. When it finds any one of the sequences that it is searching for, it returns a number indicating which sequence it found and it returns the row numbers for all of the positive events in that sequence.
- *TSparse.* This command computes first-order statistics from the sequences found by TSmatch. For example, the computation of the durations of the intervals between two events within a found sequence (e.g., Visit Duration = [time of EndVisitEven] – [time of StartVisitEvent]). The user specifies the computation to be performed. This command is usually the user-specified helper function for TSsessionstat and TStrialstat. Those commands specify the fields into which the results of its computations are put.
- *TSsessionstat.* This command computes user-specified session-level statistics, such as the times of occurrence of every feeding or every poke or peck or lever press in a session (and considerably more complex statistics). It puts the results in a field whose name is specified by the user. This command often calls TSparse as a helper function, but it can call any user-specified function instead, including user-created functions that do arbitrarily complex computations.
- *TStrialstat.* This command does the same thing as TSsessionstat but it operates on the sequences demarcated by user-defined "trials" instead of operating on the data from entire sessions. It extracts statistics from sequences found within data stretches demarcated by trial-defining sequences (for example, from sequences with stretches of data that constitute "visits"). This allows the user to find and extract statistics from hierarchical structure in behavior (for example, how far into each visit a given event occurs).
- *TSapplystat.* This command computes one or more higher-order statistics from one or more lower-order statistics. For example, the poking rate during a trial is computed from the number of pokes and the duration of the trial ([number pokes during trial]/[duration of trial]). The user specifies the fields containing the input statistic(s), the names of the fields to contain

the output statistic(s), and the computation that extracts the output statistics from the input statistics.

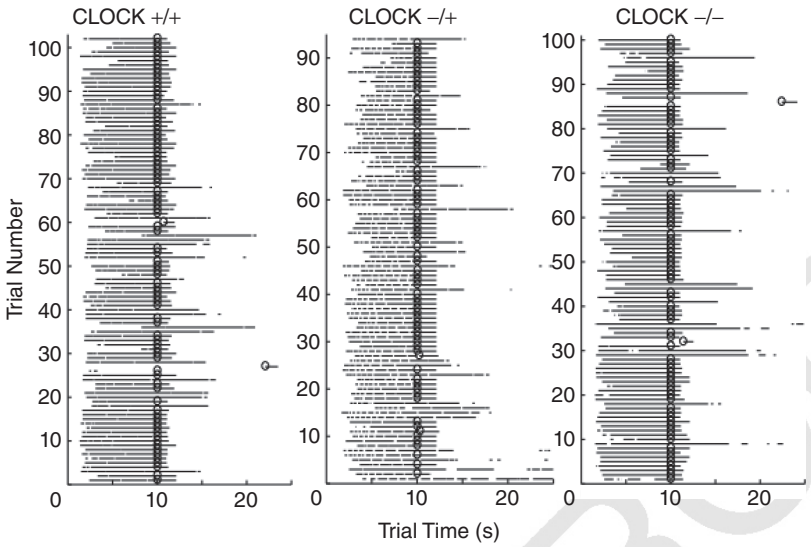
- *TScombineover*. This command compiles statistics from lower levels (e.g., the trial level or session level) into fields at higher levels (e.g., the session, subject, or experiment level), which contain, for example, the poking rates from every trial of a given kind in a given session, or across all sessions for a given subject.
- *TSlimit*. This command limits the scope of the other commands. Many analyses only make sense when conducted on a subset of the data or the available statistics. One can limit the scope to a range of trial index numbers, to a range of session index numbers, to a range of subject index numbers, or on the basis of the phase (experimental condition/protocol) during a given session.

The core commands permit the user to specify, in powerful and flexible ways, the statistics that are to be computed, using the immense resources of the Matlab™ platform – in conformity with our *better buy than build* principle. They operate on data contained in previously created fields within the overarching "Experiment" structure, and they store the results in a newly created field within the same structure – in conformity to our *keep it all together* principle. Each trial-oriented command operates on every trial (of a specified kind) in every session for every subject – unless the user restricts the range of application. The ability to restrict the range of application is also powerful, flexible, and general. The commands themselves may be elaborated in small sets, indeed, even as single commands, using the "cell" feature of the Matlab™ script editor. This feature allows the user to issue subsets of commands ("cells") without leaving the script editor. Thus, there is only one code file for the entire experiment. The code in that file is a sequence of (mostly) core commands. These sequences of core commands are grouped into more or less self-contained small cells, containing as few as one command or, rarely, as many as perhaps 10 commands. Each cell answers a question that the experimenter has posed to the data. The commands are so high level that they are almost self-documenting; that is, it is generally obvious what the command is computing from which data and where it is putting the result. The script grows naturally one cell at a time as the investigator poses one question after another.

Although the data-analysis script grows one small cell at a time, calling the script from the Matlab™ command window executes the entire analysis, creating *de novo* the Experiment structure, filling it with the raw data (assuming that those files themselves are still accessible), creating and filling all the fields within the Experiment structure that contain statistics derived from the raw data. This conforms to our *keep it all together* and our *keep a clear trail* principles. If the archived raw data files have themselves been lost – it has been known to happen! – there is no problem, because the raw data are obligatorily(!) copied into the Experiment structure prior to any analysis of them. The script file enables the user or anyone else to reconstruct the analysis that led to the published results.

Powerful Graphics Commands

AQ12 The toolbox contains three powerful graphics commands that enable users to create graphics such as the raster plots in Figs 20.3 and 20.4, which we have found particularly useful for visualizing behavior while sticking very close to the raw data. In addition to the TSraster command, the TSplotcumrecs command plots cumulative records, and TSplotcdf plots cumulative distributions and. Cumulative records are a powerful and versatile way of visualizing changes in a measure over time or over trials, such as occur, for example, in individual-subject learning curves (see Figs 20.5 and 20.6). Cumulative distributions provide a much more powerful and information-rich way of comparing measures from different conditions or different strains



AQ13 **Figure 20.3** The command TSraster in our Matlab™ toolbox generates raster plots like these. Each plot shows the performance of a mouse over approximately 100 trials in a "peak procedure" protocol. In this protocol, a pellet is delivered into the hopper in response to any beam interruption at or after 10 s has elapsed since the illumination of the hopper, which illumination serves as a signal that food is to become available at the end of the fixed 10-s interval. On some trials, called probe trials, the food is not delivered. These plots are computer generated but they are best understood by imagining that there is a pen that traverses the paper horizontally from left to right on each trial, moving at an unvarying speed. When the mouse's head is in the hopper, interrupting the infrared beam across the hopper opening, the pen is "down", writing on the paper; whenever the head is not in the hopper (not interrupting the beam), the pen is "up", not writing on the paper. Thus, the black lines show the intervals when the head was in the hopper; the white interruptions, the intervals when it was not. When food is delivered, a small circle is superposed on the black line. On food trials, the head is withdrawn soon after the pellet is released. On probe trials, there is no pellet and the mouse keeps its head in the hopper well beyond the expected time of pellet release. The variations in the well marked onset of poking may be automatically extracted to constitute the so-called start times statistic. The variations in the times of last head withdrawal on probe trials may be automatically extracted to constitute the so-called stop-times statistic. Together these statistics define the peak interval, the interval that brackets the time when the mouse expects food delivery. The +/+ mouse was a wild-type C57/B6; the +/– and –/– mice were, respectively, heterozygous and homozygous CLOCK null mutants. These are ~~in~~ plots of the raw data, yet they enable one to see at a glance that the differences between the strains are at best very small. They are all timing the pellet release with the same degree of accuracy and precision (Cordes and Gallistel 2008). Plots like these may be automatically generated and emailed to the researchers while the testing is in progress.

(see Figs 20.7 and 20.8). The high-level graphics commands in the Toolbox supplement the immense graphical resources provided by the Matlab™ platform.

Results

The mechanisms that enable animals to locate themselves in time and space are among the most basic mechanisms of cognition. Konopka and Benzer (1971)showed the way to a forward genetics approach to temporal localization mechanisms when they successfully screened for genetic abnormalities in the circadian clock of *Drosophilamelanogaster*. The circadian clock is a mechanism of cognition because one way in which animals locate themselves in time is by reading the phase of their circadian clock and because one way in which they localize events in time is by remembering the phase of their circadian clock at the time the event occurred (Gallistel 1990). Computing the difference between the current phase of their clock and the

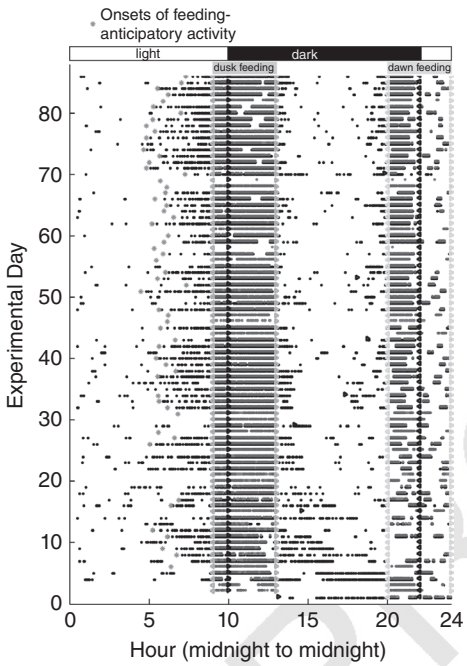


Figure 20.4 Raster plot used to visualize the development of circadian food-anticipatory activity in a mouse with two periods of access to food (dawn feeding and dusk feeding) in each 24-hour period. The automatically scored daily onsets of activity anticipating the dusk feeding are marked by purple asterisks. Each line records every poke made into either of two feeding hoppers (black dots) throughout a 24-hour day/night cycle. The red and green dots mark every pellet delivery. The base-left black triangles mark the daily offsets of the house light in the test box; the base-right black triangles mark the daily onsets of the house light. Similarly, the cyan triangles mark the (silent) onsets and offsets of food availability in a 4-hour period beginning one hour before dusk and the magenta triangles mark the onsets and offsets of food availability in the 4-hour period beginning 2 hours before dawn. During these intervals, the mouse earned pellets delivered into the two hoppers by performing the tasks required by our experimental protocols (the matching, 2-hopper autoshaping, and switch protocols). Thus, the gathering of data on the learning of the time of day at which important changes occur proceeds concurrently with the gathering of other measures. This complex data-rich plot was produced by a single command in the Toolbox (the TSraster command). *(See color plate section for the color representation of this figure.)*

remembered phase at which food becomes available makes possible the circadian food-anticipatory activity shown in Fig. 20.4 (Gallistel 1990).

Computing the difference between the times at which two events occurred enables the timing of intervals, which underlies most forms of conditioned behavior (Gallistel and Gibbon 2000; Balsam and Gallistel 2009; Gallistel and Balsam 2014).Just as one can measure the phase and period of a clock both behaviorally and neurobiologically, so, too, one can measure the accuracy and imprecision of interval timing both behaviorally and at the sites in the nervous system where the intervals between events are remembered and used to time conditioned responses. As reviewed in the introduction, one such site has already been discovered: the Purkinje cells in the cerebellum show a pause in their firing whose duration is approximately that of the CS–US interval used to artificially condition the cerebellar eye-blink circuit, using electrical stimulation of mossy fibers and climbing fibers in place of the CS and the US, respectively. It has also been shown that the firing of place cells in the hippocampus signals both the rat's location within a familiar environment and its location within a familiar temporal interval (MacDonald, Lepage *et al*. 2011; Eichenbaum 2013).

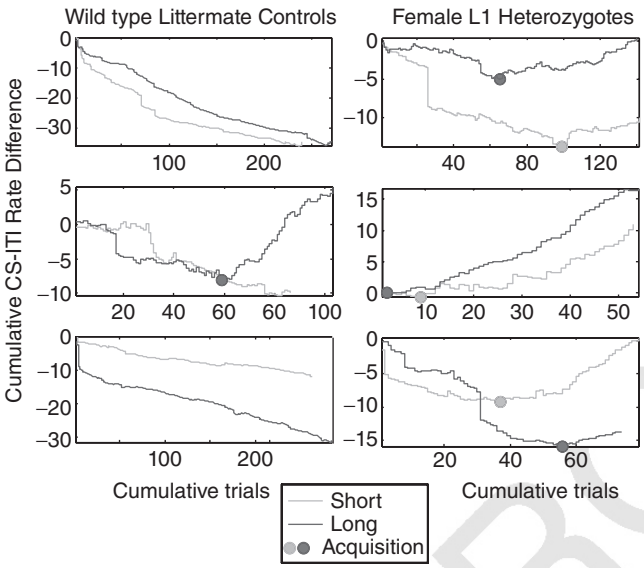


Figure 20.5 Cumulative records visualizing the acquisition of autoshaped hopper entry in a 2-hopper autoshaping protocol with intermixed trials during which there is either a short (4 s) pellet-delivery latency on one hopper (Short) or a long (12 s) delivery latency on the other (Long). In hopper-entry autoshaping, the pending delivery of a pellet is signaled by the illumination of the hopper into which the pellet will be delivered. The measure plotted in these cumulative records is the difference between the poking rate during the intertrial interval when the hopper in question is not illuminated and the poking rate during a trial when it is illuminated. The average value of this measure is 0 or negative before a mouse learns to poke into the illuminated hopper in anticipation of the pellet delivery and positive after it learns to do so. The automatically detected inflection point in each plot, where the slope changes from 0 or negative to positive, is marked by a filled circle. (Note that some mice never developed anticipatory poking in one or both hoppers.) Multipanel plots of cumulative records are generated ~~with a single command~~ in the Toolbox ~~(the TSplotcumrecs command.)~~ The data are from Gallistel, Tucci *et al*. 2014. *Source:* Gallistel, Balci *et al*. 2014, with slight modifications.

We use the switch protocol (Balci, Papachristos *et al*. 2008; Balci, Freestone *et al*. 2009) to measure the accuracy and imprecision of interval timing. In this protocol, the center hopper (see Fig. 20.1) lights up to signal the mouse that it can start a trial. When the mouse pokes into the illuminated center hopper, the light there extinguishes and the two flanking hoppers light up. The computer randomly selects one of the flanking hoppers to be the hot hopper. If the "short" hopper is selected, then a pellet is delivered there in response to the first poke at or after 4 s; if the "long" hopper is selected, then a pellet is delivered there in response to the first poke at or after 12 s. The mouse soon learns to go first to the short hopper, and then to switch to the long hopper on the long trials, when a pellet is not forthcoming at the short hopper at the end of 4 s. If the mouse switches from the short to the long hopper on a short trial so that its first poke at or after 4 s is into the long hopper, the trial ends without a pellet delivery. If it switches too late on a long trial, so that its first poke at or after the long latency is into the short hopper, the trial again ends without a pellet release. However, the mouse soon learns to switch midway between the short and long delivery latencies, so that it gets a pellet on the great majority of trials. When it has learned to do this, we shorten the long release-latency to 8 s and then to 7 s and then to 6 s, thereby increasing the incentive to make tightly timed switches.

Our measure of its ability to time is the latency to depart from the short hopper on long trials. We call this the switch latency. The extent to which the mean switch latency is accurately positioned halfway between the short and long pellet-delivery latencies measures the accuracy

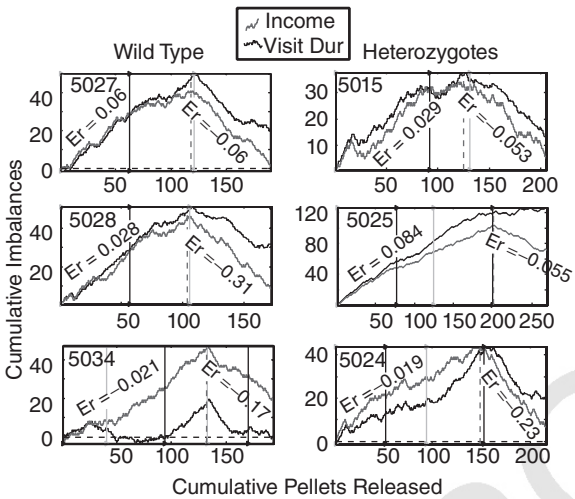


Figure 20.6 Concurrent cumulative records of feeding-by-feeding income imbalance (gray) and feeding-by-feeding visit duration imbalance (black) reveal matching behavior at a feeding-by-feeding level of resolution. The feeding imbalance measure is $(F_2–F_1)/(F_2+F_1)$, where F_i is the number of pellets obtained from Hopper i. The Visit Duration imbalance measure is $(T_2–T_1)/(T_2+T_1)$, where T_i is the time spent visiting Hopper i. These measures are linear transformations of the Herrnstein fractions commonly used to characterize matching. The Herrnstein fractions are the percent total income obtained from a given hopper and the percent of total visit time spent visiting that hopper. When the slopes of the cumulative imbalance records are the same, the mouse is matching the ratio of its average visit durations to the two hoppers to the ratio of the average incomes obtained from them. Put another way, the percent time spent visiting a hopper matches the percent income obtained from it. "Er" is the difference in the average slopes, which is twice the difference in the Herrnstein fractions. The mice are those in Fig. 20.5. The programmed income imbalance was reversed at the dashed gray vertical lines, producing the downward inflection in the gray plots. Note the rapid and abrupt answering change in time-allocation (the downward inflection in the black plots). The data are from Gallistel, Tucci *et al*. 2014. *Source:* Gallistel, Balci *et al*. 2014, with modifications.

of timing (Balci, Freestone *et al*. 2009). The coefficient of variation, that is, the ratio between the standard deviation of the Gaussian component of the switch-latency distribution and its mean (see Fig. 20.7) measures the imprecision of its timing.

Using the fully automated system and three standard protocols: matching (Fig. 20.6), 2-hopper autoshaping (Fig. 20.5), and the switch protocol (Fig. 20.7), Gallistel, Tucci and others (2014) screened two strains of heterozygous mutant mice, *Bfc* (batface) and L1, and their wild-type littermate controls. The Batface (Bfc, MGI:2656734) mouse mutants carry a single point mutation within the β-catenin gene (*Ctnnb1*), which was identified in a large-scale *N*-ethyl-*N*-nitrosourea (ENU) mutagenized screen (Nolan, Peters *et al*. 2000). Female L1 heterozygotes (*x/L1I*) are heterozygous at the level of the L1-expressing cells, in that each cell expresses either the normal or the mutated gene. The L1 glycoprotein is expressed in neurons but not in astrocytes or oligodendrocytes in the central nervous system. It plays a role in neuronal migration, neurite outgrowth and guidance, fasciculation of axons, and myelination during development(Moos, Tacke *et al*. 1988; Hortsch 1996; Maretzky, Schulte *et al*. 2005; Maness and Schachner 2007).

The Batface heterozygotes showed reduced interval-timing imprecision (increased imprecision) relative to their littermate controls (Fig. 20.8, bottom right panel, compare dashed versus solid plots); that is, the Batface heterozygotes were less precise than their wild-type controls. Surprisingly, the L1 heterozygotes showed increased precision (reduced imprecision) relative both to their wild-type controls and to the Batface wild-type controls (Fig. 20.8, top right panel)

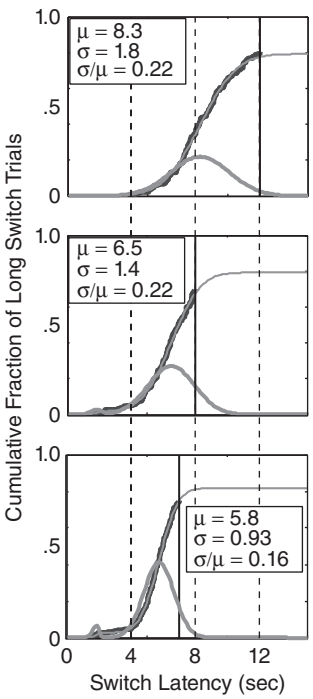


Figure 20.7 Measuring interval timing accuracy and precision using the switch paradigm. The slightly wiggly thick blue lines are the cumulative distribution of switch latencies. The thin red smooth curves superimposed on them are the best-fitting cumulative Weibull–Gauss mixture distributions. The thick red bell-shaped curves are the corresponding probability density distributions (the derivatives of the best fitting cumulative distributions). The thick dashed vertical running across all three plots marks the "short" (4 s) delivery latency, which was not varied. The thick black vertical within each plot marks the "long" delivery latency, which varied from 12 s to 8 s to 7 s in successive blocks of trials and from top to bottom panels in this figure. Notice that the Gaussian component of the probability density function is positioned approximately midway between the short and long latencies and that the precision (σ/μ) improves when the interval between the latencies is narrowed. *Source:* Gallistel, Balci *et al*. 2014. *(See color plate section for the color representation of this figure.)*

and, *a fortiori*, to the Batface heterozygotes (compare top and bottom panels on the right of Fig. 20.8); that is, the L1 heterozygotes were more precise than their wild-type controls.

AQ14
AQ15

Our screen showed a high degree of functional specificity for these opposing effects of genotype in the two strains in that no other measure was affected by the within-strain differences in genotype. As just noted, the effect of genotype was in both cases specific to the imprecision of interval timing, not to its accuracy. In this same task, we also extract from the distribution of switch latencies, a measure of impulsivity and a measure of trials on task. The measure of impulsivity is the percent of the Weibull–Gauss mixture distribution ascribed to the Weibull component. This component captures the tendency of some mice not to time their switch on some trials; that is, they go first very briefly to the short hopper, but they leave it well before the short delivery latency has expired. The little bump at the left of the probability distribution in the bottom panel of Fig. 20.7 is the Weibull component of the mixture. The strength of this tendency (the size of the bump at short latencies) varies strikingly from mouse to mouse and also with conditions: when the long trials are made highly probable, this bump usually gets larger. We call this impulsivity because these short departures are not optimal (unlike the departures that are timed to fall after the short delivery latency and before the long one).The measure of trials on task is the height of the asymptote in the fit to the mixture distribution. If the mouse were always on task, this asymptote would be at 1, but the mouse observer, like the human observer is psychophysical tasks, is rarely on task on every trial. Therefore, it is necessary to include a parameter that allows the cumulative distribution to asymptote below 1. Both impulsivity and trials on task would seem to be very likely to reflect genetic variation in performance factors, but there was the within-strain genotypic differences had no effect on these measures.

The within-strain genotypic differences also had no effect on the accuracy with which the mice matched the ratio of their visit durations to the ratio of the incomes in the matching protocol. And it had no effect on the rapidity with which the experimentally naïve mice began to cycle between the two hoppers in that protocol (the first protocol the mice were exposed to).

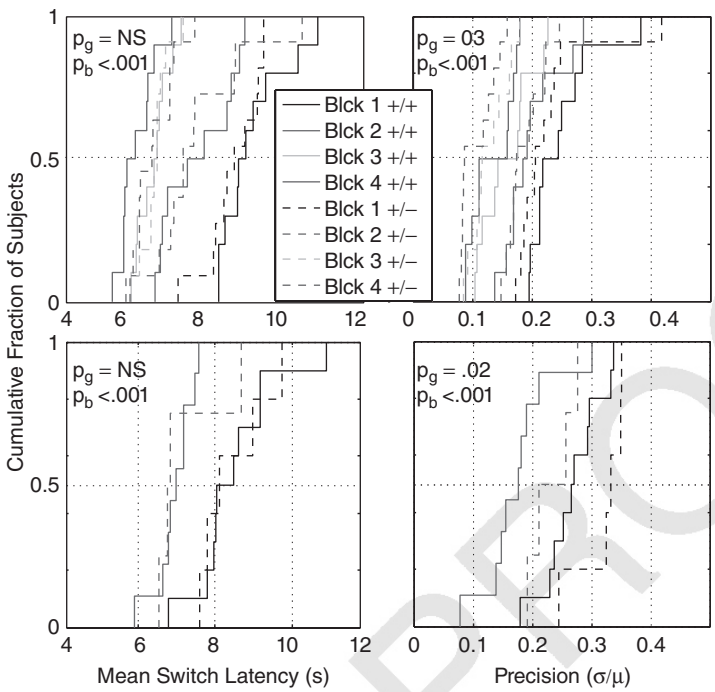


Figure 20.8 Cumulative distribution of Gaussian means and Gaussian coefficients of variation (~~s/m~~ σ/μ) for heterozygotes (dashed plots) and wild-type littermate controls (solid plots) for L1 strain (top panels) and Batface strain (bottom panels) across successive blocks of trials in which the long delivery latency was reduced from 12 s (Blck 1) to 8 s (Blck 2) to 7 s (Blck 3) to 6 s (Blck 4). The p values come from 2-way ANOVAs: p_g is for main effect of genotype; p_b is for main effect of block. Note that the block (that is the variation in long delivery latency) had a highly significant effect on the mean of the Gaussian component of the Weibull–Gauss mixture distribution: the shorter the long delivery latency, the shorter the mean switch latency, regardless of genotype, as shown by the fact that the plots for successive blocks move substantially leftward (toward shorter mean switch latencies) in the panels on the left. The genotype had no significant effect on this mean, as shown by the fact that the solid and dashed plots more or less superimpose in the panels on the left. Thus, the accuracy of interval timing is unaffected by genotype. The main effect of block on precision is also highly significant (panels on the right): the narrower the difference between the short and long release latencies, the more precisely the mice time their switches. On this measure, however, the main effect of genotype is also significant, but in opposite directions: in the L1 strain, the heterozygotic females time their switches more precisely than do their wild-type controls; whereas in the Batface strain, the opposite is true (compare solid versus dashed plots in the panels on the right).

There was a similar lack of genotypic effects on the speed with which mice learned to initiate trials in the 2-hopper autoshaping protocol (a measure of the rate of instrumental, aka operant, learning), on the distribution of trial-initiation latencies in that protocol, and on the trials to acquisition of anticipatory hopper entries in that protocol (a measure of the rate of classical conditioning, aka Pavlovian conditioning).

Summary

AQ16 A fully automated screening system, using a test environment in which the mice live 24/7, makes physiologically meaningful quantitative measures of several different basic mechanisms of cognition on many mice simultaneously (high screening volume) with very little demand on the time of investigators and technicians (low investment of human time). The mice are not handled during the screening (minimizing the effects of handling stress). Many of the measures

are physiologically meaningful in that comparable measures can be made at the neurobiological level of analysis. An open source Matlab™ toolbox greatly facilitates the extraction of meaningful measures from length records of time-stamped events. The toolbox is designed in such a way as to guarantee an intact trail from raw data to published measures. The eight core commands in the toolbox operate at such a high level that answers to most questions posed of the data are extracted with a small number of commands (1–6), in code that is much more transparent than code written in lower-level languages.

The mechanism mediating the measuring of interval durations is a particularly promising focus for a genetic approach to basic mechanisms of cognition. Screens made with the here-described fully automated screening system have revealed functionally specific effects of within-strain genotypic differences on the precision with which mice measure intervals measured in a few seconds. These results may make possible the kind of genetic approach to the neurobiology of interval timing that has been conspicuously successful in the case of the circadian clock.

References

Balci, F., D. Freestone, *et al.* (2009). "Risk assessment in man and mouse." *Proceedings of the National Academy of Science U S* A 106(7): 2459–2463.

Balci, F., E. B. Papachristos, *et al.* (2008). "Interval timing in the genetically modified mouse: A simple paradigm." *Genes, Brains & Behavior* 7: 373–384.

Balsam, P. and C. R. Gallistel (2009). "Temporal maps and informativeness in associative learning." *Trends in Neurosciences* 32(2): 73–78.

Brelsford, J. and J. Theios (1965). "Single session conditioning of the nictitating membrane in the rabbit: Effect of intertrial interval." *Psychonomic Science* 2: 81–82.

Cerminara, N. L. and J. A. Rawson (2004). "Evidence that climbing fibers control an intrinsic spike generator in cerebellar Purkinje cells." *Journal of Neuroscience* 24: 4510–4517.

Cordes, S. and C. R. Gallistel (2008). "Intact interval timing in circadian CLOCK mutants." *Brain Research* 1227: 120–127.

Eichenbaum, H. (2013). "Memory on time." *Trends in Cognitive Science* 17(2): 81–88.

Estes, W. K. (1956). "The problem of inference from curves based on group data." *Psychological Bulletin* 53: 134–140.

Estes, W. K. (1960). "Learning theory and the new "mental chemistry"." *Psychological Review* 67: 207–223.

Gallistel, C. R. (1990). *The organization of learning*. Cambridge, MA, Bradford Books/MIT Press.

Gallistel, C. R. and P. D. Balsam (2014). "Time to rethink the neural mechanisms of learning and memory." *Neurobiology of Learning and Memory* 108: 136–144.

Gallistel, C. R. and J. Gibbon (2000). "Time, rate, and conditioning." *Psychological Review* 107(2): 289–344.

Gallistel, C. R. and L. D. Matzel (2013). "The neuroscience of learning: Beyond the Hebbian Synapse." *Annual Review of Psychology* 64: 169–200.

Gallistel, C. R., F. Balci, *et al.* (2014). "Automated, quantitative cognitive/behavioral screening of mice: for genetics, pharmacology, animal cognition and undergraduate instruction." *Journal of Visualized Experiments (JoVE)* 84: e51047.

Gallistel, C. R., P. D. Balsam, *et al.* (2004). "The learning curve: Implications of a quantitative analysis." *Proceedings of the National Academy of Sciences* 101(36): 13124–13131.

Gallistel, C. R., P. Shizgal, *et al.* (1981). "A portrait of the substrate for self-stimulation." *Psychological Review* 88(3): 228–273.

Gallistel, C. R., V. Tucci, *et al.* (2014). " Cognitive assessment of mice strains heterozygous for cell-adhesion genes reveals strain-specific alterations in timing." *Philosophical Transactions of the Royal Society. B.* 369(1637): 20120464.

Gibbon, J. (1977). "Scalar expectancy theory and Weber's Law in animal timing." *Psychological Review* 84: 279–335.

Gluck, M. A. and R. F. Thompson (1987). "Modeling the neural substrates of associative learning and memory: a computational approach." *Psychological Review* 94(2): 176–191.

Hawkins, R. D., T. J. Carew, *et al.* (1986). "Effects of interstimulus interval and contingency on classical conditioning of the Aplysia siphon withdrawal reflex." *Journal of Neuroscience* 6: 1695–1701.

Hesslow, G., D.-A. Jirenhed, *et al.* (2013). "Classical conditioning of motor responses: What is the learning mechanism?" *Neural Networks* 47: 81–87.

Hortsch, M. (1996). "The L1 family of neural cell adhesion molecules: old proteins performing new tricks." *Neuron* 17: 587–593.

Jirenhed, D. A. and G. Hesslow (2011). "Learning stimulus intervals—adaptive timing of conditioned Purkinje cell responses." *The Cerebellum* 10: 523–535.

Jirenhed, D. A. and G. Hesslow (2011). "Time course of classically conditioned Purkinje cell response is determined by initial part of conditioned stimulus." *Journal of Neuroscience* 31: 9070–9074.

Jirenhed, D. A., F. Bengtsson, *et al.* (2007). "Acquisition, extinction, and reacquisition of a cerebellar cortical memory trace." *Journal of Neuroscience* 27(10): 2493–2502.

Kalmbach, B. E., H. Voicu, *et al.* (2011). "A subtraction mechanism of temporal coding in cerebellar cortex." *The Journal of Neuroscience* 31(6): 2025–2034.

Kehoe, E. J. and R. M. Napier (1991). "Temporal specificity in cross-modal transfer of the rabbit nictitating membrane response." *Journal of Experimental Psychology: Animal Behavior Processes* 17: 26–35.

Kehoe, E. J., P. Graham-Clarke, *et al.* (1989). "Temporal patterns of the rabbit's nictitating membrane response to compound and component stimuli under mixed CS-US intervals." *Behavioral Neuroscience* 103: 283–295.

Konopka, R. J. and S. Benzer (1971). "Clock mutants of *Drosophila melanogaster*." *Proceedings of the National Academy of Sciences (USA)* 108(25): 2112–2116.

Levinthal, C. F. (1973). "The CS-US interval function in rabbit nictitating membrane response conditioning: Single vs multiple trials per conditioning session." *Learning and Motivation* 4: 259–267.

Levinthal, C. F., R. H. Tartell, *et al.* (1985). "The CS-US interval (ISI) function in rabbit nictitating membrane response conditioning with very long intertrial intervals." *Animal Learning and Behavior* 13(3): 228–232.

MacDonald, C. J., K. Q. Lepage, *et al.* (2011). "Hippocampal "time cells" bridge the gap in memory for discontiguous events." *Neuron* 71(4): 737–749.

Maness, P. F. and M. Schachner (2007). "Neural recognition molecules of the immunoglobulin superfamily: signaling transducers of axon guidance and neuronal migration." *Nature Neuroscience* 10: 19–26.

Maretzky, T., M. Schulte, *et al.* (2005). "L1 is sequentially processed by two differently activated metalloproteases and presenilin/gamma-secretase and regulates neural cell adhesion, cell migration, and neurite outgrowth." *Molecular and Cellular Biology* 25: 9040–9053.

Moos, M., R. Tacke, *et al.* (1988). "Neural adhesion molecule L1 as a member of the 986 immunoglobulin superfamily with binding domains similar to 987 fibronectin." *Nature* 334: 701–703.

Nolan, P. M., J. Peters, *et al.* (2000). "Implementation of a large-scale ENU mutagenesis program: towards increasing the mouse mutant resource." *Mammalian Genome* 11: 500–506.

Rock, I. (1957). "The role of repetition in associative learning." *The American Journal of Psychology* 70: 186–183.

Roediger, H. and K. Arnold (2012). "The one-trial learning controversy and its aftermath: remembering Rock (1957)." *American Journal of Psychology* 125: 127–143.

Schneiderman, N. and I. Gormezano (1964). "Conditioning of the nictitating membrane of the rabbit as a function of CS-US interval." *Journal of Comparative and Physiological Psychology* 57: 188–195.

Shizgal, P. and B. Murray (1989). Neuronal basis of intracranial self-stimulation. *The Neuropharmacological Basis of Reward*. J. M. Liebman and S. J. Cooper. Oxford, Clarendon Press:106–163.

Svensson, P., D. A. Jirenhed, *et al*. (2010). "Effect of conditioned stimulus parameters on timing of conditioned Purkinje cell responses." *Journal of Neurophysiology* 103: 1329–1336.

Ward, R. D., C. R. Gallistel, *et al*. (2012). "Conditional stimulus informativeness governs conditioned stimulus—unconditioned stimulus associability." *Journal of Experimental Psychology: Animal Behavior Processes* 38(1): 217–232.

Wetmore, D. Z., D.-A. Jirenhed, *et al*. (2014). "Bidirectional plasticity of Purkinje cells matches temporal features of learning." *The Journal of Neuroscience* 34: 1731–1737.

White, N. E., E. J. Kehoe, *et al*. (2000). "Coefficients of variation in timing of the classically conditioned eyeblink in rabbits." *Psychobiology* 28(4): 520–524.

Author Queries

AQ1 items in this list are applicable to all models?
AQ2 "All such procedures"?
AQ3 please check this sentence – is the end missing? If not, what does the "if" refer to?
AQ4 italics?
AQ5 please check that abbreviations have been spelled out correctly.
AQ6 does this refer to a figure in the paper by Hesslow et al please?
AQ7 please check that levels of heading (coded as [H1], [H2] etc.) are as intended from this point onwards.
AQ9 "numerous"?
AQ11 please check – "square"?
AQ12 only 7 items are listed below?
AQ13 please check – something missing here?
AQ14 something missing here or delete final "and"?
AQ15 delete "in"?
AQ17 please check – I wasn't sure if something was missing from the midpart of this sentence or the second comma should be deleted?
AQ18 please check – delete "there was the"?
AQ19 "lengthy"?