

# Extending SEE for large-scale phenotyping of mouse open-field behavior

NERI KAFKAFI

University of Maryland, Baltimore, Maryland

SEE (Software for the Exploration of Exploration) is a visualization and analysis tool designed for the study of open-field behavior in rodents. In this paper, I present new extensions of SEE that were designed to facilitate its use for mouse behavioral phenotyping and, especially, for the problems of discrimination of genotypes and the replication of results across laboratories and experimental conditions. These extensions were specifically designed to promote a new approach in behavioral phenotyping, reminiscent of the approach that has been successfully employed in bioinformatics during recent years. The path coordinates of all animals from many experiments are stored in a database. SEE can be used to query, visualize, and analyze any desirable subsection of this database and to design new measures (endpoints) with increasingly better discriminative power and replicability. The use of the new extensions is demonstrated here in the analysis of results from several experiments and laboratories, with an emphasis on this approach.

SEE (Software for the Exploration of Exploration) is a visualization and analysis tool designed to study open-field behavior in rodents (Drai & Golani, 2001), using the time series of path coordinates exported from any automatic tracking system. SEE was specially developed on the basis of ethologically oriented studies (Benjamini, Drai, Elmer, Golani, & Kafkafi, 2001; Drai, Benjamini, & Golani, 2000; Drai, Kafkafi, Benjamini, Elmer, & Golani, 2001; Eilam & Golani, 1989; Golani, Benjamini, & Eilam, 1993; Kafkafi, Lipkind, et al., in press; Kafkafi, Mayo, Drai, Golani, & Elmer, 2001; Kafkafi, Pagis, et al., in press; Tchernichovski, Benjamini, & Golani, 1996, 1998; Tchernichovski & Golani, 1995) in rats and mice. The behavior patterns it is based on have also been proved to be useful in neurobehavioral studies (Cools, Ellenbroek, Gingras, Engbersen, & Heeren, 1997; Gingras & Cools, 1997; Whishaw, Cassel, Majchrzak, Cassel, & Will, 1994; Whishaw, Hines, & Wallace, 2001). In contrast to the conventional view of open-field behavior as a rather stochastic phenomenon, these studies showed that it is highly structured, consisting of typical behavior patterns. The most basic of these patterns, which can be regarded as the building blocks of open-field behavior, are lingering episodes (stops) and progression segments (Drai et al.,

2000; Kafkafi et al., 2001). Although SEE can readily be used to study a path as a continuous series, its main power lies in treating a path as a string of discrete units, each belonging to one of these two behavior patterns.

SEE is a sublanguage, or *package*, residing within the powerful, flexible programming environment of Mathematica (by Wolfram Research; see Wolfram, 1999), and can employ all the sophisticated tools available in Mathematica for its computations. Here are a few simple expressions in SEE that demonstrate the use of the language. As with any expression of Mathematica, pressing Shift+Enter immediately evaluates the expression and returns the result:

```
Length[movesegments]
```

The Mathematica function `Length[list]` returns the number of members in the list—in the above case, the number of progression segments in the currently loaded session:

```
movesegments[[1]]
```

The Mathematica expression of the form `list[[k]]` gives the *k*th member of the list—in the above case, the first progression segment in the loaded session—in the format {start time, end time}:

```
SegmentCoords[movesegments[[1]]]
```

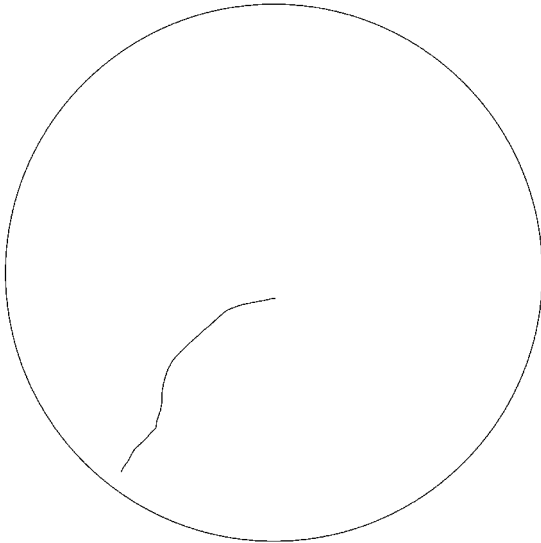
`SegmentCoords[segment]` is a function defined in SEE that returns the list of coordinates in a segment, either a progression segment or a lingering episode:

```
See[PathPlot[movesegments[[1]]]]
```

`PathPlot[segment]` creates a graphic object of the path in this segment. `See[graphics object]` displays this object in a visual representation of the arena. Evaluating the ex-

---

This research was supported by NIH Grant 1 R01 NS40234. The Experiment Explorer and Endpoint Manager packages, like the SEE software itself, are available upon request from the author or from Ilan Golani, Tel-Aviv University, <http://www.tau.ac.il/~ilan99/see/help>. They require, however, the Mathematica programming environment by Wolfram Research, Inc. Simpler stand-alone applications are also available. I thank Ilan Golani for his thoughtful comments and corrections of the manuscript. Correspondence concerning this article should be addressed to N. Kafkafi, Maryland Psychiatric Research Center, P. O. Box 21247, Baltimore, MD 21228 (e-mail: [nkafkafi@mprc.umaryland.edu](mailto:nkafkafi@mprc.umaryland.edu)).



**Figure 1.** The path plot of a single progression segment of a mouse. The actual diameter of the arena is 2.5 m.

pression above thus returns the path plot of the first segment in the arena (Figure 1).

Any function that is viable for each member of a list can be done over the whole list, using `Map[function, list]`. The expression

```
See[Map[PathPlot,movesegments]]
```

thus returns the path plots of all progression segments (Figure 2).

In addition to its flexible and elegant syntax, Mathematica contains hundreds of mathematical functions, many of them employing powerful numerical tools, which can readily be employed for analysis. For example,

```
Fourier[SegmentCoords[movesegments[[1]]]]
```

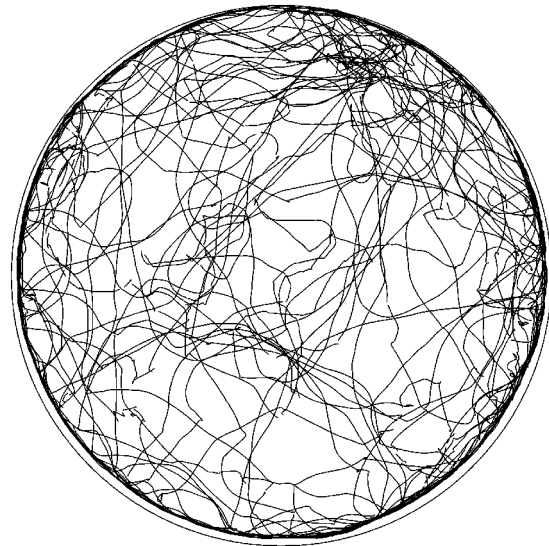
returns the Fourier spectrum analysis of the coordinates of the first progression segment.

The examples above demonstrate how the vocabulary of SEE can be used in conjunction with the vocabulary and syntax of Mathematica to build an unlimited number of expressions, ranging from simple to very complex, addressing many desirable aspects of the spatial behavior (for a more elaborate and complete description, see Draï & Golani, 2001). It was recently demonstrated that SEE can be employed for behavioral phenotyping of mice (Draï et al., 2001; Kafkafi, Lipkind, et al., in press).

Behavioral phenotyping is the field in which behavior patterns are characterized in order to associate them with particular gene loci in specific genotypes. In recent years, behavioral phenotyping has been increasingly employed in mouse studies, because of the unique technical possibilities for generating reproducible genotypes of this mammal (inbred strains) and manipulating them up to the level of single genes (single-site mutants, or *knock-outs*). The need for mouse behavioral phenotyping has

resulted in the design of batteries including many behavioral and physiological tests (e.g., Crawley, 2000; Crawley et al., 1997; Rogers et al., 1999; Tarantino, Gould, Druhan, & Bucan, 2000). A critical problem with many of the current behavioral tests, however, is that their results are difficult to replicate in other laboratories, despite careful standardization of conditions (Crabbe, Wahlsten, & Dudek, 1999). It was recently suggested that solving the replicability problem by even more rigorous standardization is not feasible and that it is preferable, instead, to choose behavioral tests that yield the most stable results (Wahlsten, 2001). Extending this suggestion further, we offered to design new behavioral methods and endpoints explicitly for improved discrimination and replicability and increased resistance to laboratory factors. This can be done efficiently if the raw path coordinates of open-field sessions from many experiments and laboratories are stored in a database, which can be explored with specialized algorithms. Instead of postulating a priori that a certain measure of open-field behavior is relevant, it is thus possible to explore many different optional measures and to select those that show high discrimination and replicability over most of the database. As the database increases with the accumulation of experiments, genotypes, and laboratories, these measures can be further modified, so as to make them increasingly discriminative, general, and resistant to laboratory factors. This approach was demonstrated in Kafkafi, Pagis, et al. (in press) by designing a new measure of a specific behavior pattern, the *darting* behavior, which discriminates two common inbred strains, C57BL/6 and DBA/2, reliably across three laboratories, different protocol conditions, and even under the effect of cocaine.

To facilitate the use of SEE analysis within this framework, I programmed two additional Mathematica *pack-*



**Figure 2.** The path plot of all progression segments of 1 mouse, during a 30-min session.

ages that can be used in conjunction with the regular SEE package. The Experiment Explorer package is used to define a database of path coordinates, including several experiments, and to enable any SEE or Mathematica expression to be calculated in any desired subsection of this database. The Endpoint Manager package is used to define measures (endpoints), calculate them in any selected subsection of this database, and use them as primitives in the design of more complex endpoints. The operation of these two packages will be demonstrated in the next two sections, with an emphasis on this approach. The data used for this demonstration were taken from our experiments comparing the mouse inbred strains C57BL/6J (C57) and DBA/2J (DBA) across three laboratories (Kafkafi, Lipkind, et al., in press; Kafkafi, Pagis, et al., in press).

Within the context of this study, a *session* means an unbroken track of a locomotor path of a single animal—in the case of our setup, exploring a large (~2.5-m radius) arena for a predetermined duration (up to several hours). A *group* is a list of sessions within an experiment, usually with a single controlled variable being changed—that is, several different sessions of the same animal or several animals (one session each) of the same genotype or with the same treatment. An *experiment* is a list of groups that were tracked in similar conditions (location, time, handling protocol, tracking parameters, etc.). An *endpoint* is some measure (i.e., a defined variable) used to evaluate the properties of the behavior. We use this term because it is part of the jargon used in behavioral phenotyping, although it is not exactly appropriate within our approach, where any endpoint may be modified or used as the basis for more elaborate endpoints.

### The Experiment Explorer

In the original SEE package (Drai & Golani, 2001), a single session had to be loaded in order to be queried. Expressions such as those demonstrated in the introduction returned the result for this currently loaded session. Querying many sessions and many experiments can still be done this way, but it is cumbersome and does not encourage the user to explore. The Experiment Explorer package was thus developed to facilitate exploring, by using SEE, any part of a database including several experiments, each containing several groups and several sessions in each group. For example,

```
Length[movesegments]~InSession~{c57,1}
~InExperiment~"C57 vs DBA in MPRC"
```

returns the number of progression segments in the first session of the group named C57 in the experiment named "C57 vs DBA in MPRC." The expression Length[movesegments] is only an example here and can be substituted for any SEE or Mathematica expression, however complex, that is viable for a single session. For example,

```
See[Map[PathPlot,movesegments]]~InSession~{c57,
1}~InExperiment~"C57 vs DBA in MPRC"
```

will return the path plot of this session, as in Figure 2. It is also possible to execute, in each session, a list of many commands. A similar syntax is used to access any other desirable subsection of the database. For example, the expression

```
Length[movesegments]~MapOverGroups
~experimentGroups~MapOverExperiments~{"C57
vs DBA in MPRC","C57 vs DBA in TAU"}
```

returns a list of the number of progression segments in all sessions in all groups in both experiments, "C57 vs DBA in MPRC" and "C57 vs DBA in TAU."

It is also possible to address specific parts of sessions, using the functions "InTimeSlice" and "ByTimeBin" interchangeably with all the other functions. For example,

```
Length[movesegments]~ByTimeBin~{5
minutes}~InSession~{c57,1}~InExperiment~"C57
vs DBA in MPRC"
```

returns a list of the numbers of progression segments in time bins of 5 min in this session.

In order to be added to the database, each experiment should be defined once by the user. This definition includes the names of experimental groups and the directories in the computer where the data files are located. The list experimentGroups that we used in several of the examples above is defined as the list of basic groups in the experiment, but the user is free to add any additional ad hoc grouping over the sessions in an experiment and to use it with the same syntax as that applied to experimentGroups.

The user may also formulate any desirable variable that is specific to the experiment and include it in the experiment definition. These variables will then get the appropriate values whenever this experiment is addressed by functions such as InExperiment and MapOverExperiments and can thus be used in queries. Several such examples of experiment-specific variables that are currently employed are arenaCenter, arenaRadius, and sessionDuration.

Almost any desirable query can thus be formulated over any part of a database, using the flexible language of Mathematica interchangeably with the additional vocabulary of the SEE and Experiment Explorer packages. This vocabulary was designed to be user friendly and as natural as possible, so the meaning of each expression can usually be discerned, at least approximately, from a first reading. The use of a specialized language to define queries might seem to involve a lot of typing by the user, but this problem is mostly solved in Mathematica by using the automatic completion of defined words and by palettes of the required functions and variables, so they can be inserted in any desired place or wrapped around any desired expression by a click of a mouse (the electronic one, of course).

### The Endpoint Manager

In order to facilitate our approach, it is important that any expression, once found useful for quantifying a cer-

tain desirable property of the behavior, could be immediately defined as a new endpoint and become available for use, like any previously defined endpoint, in the previously demonstrated commands. The Endpoint Manager package was programmed to achieve this in conjunction with the Experiment Explorer and SEE packages. In addition, this package supplies graphical and statistical tools for studying endpoint results from many sessions, groups, and experiments.

Defining a new endpoint is done by using the function `EndpointDefinition`. Any expression in SEE or Mathematica can be defined as an endpoint. For example, the expression used in some of the examples in the previous sections, `Length[movesegments]` (the number of members in the list `movesegments`—i.e., the number of progression segments), can be defined as the endpoint `NumberOfProgressionSegments`:

```
EndpointDefinition[NumberOfProgression
Segments]:= Length[movesegments]
```

A slightly more complex example is

```
EndpointDefinition[DistanceTraveled]:= Apply[Plus,
Map[Activity,segments]]
```

In this example, we employed built-in functions of Mathematica (`Apply`, `Plus`, and `Map`) and a list and a function defined in SEE (`segments` and `Activity`). It is possible to define endpoints of arbitrary complexity by using any number of previously defined functions and variables.

An expression of the form `EndpointName[ ]` is used to compute the value of the endpoint in any subsection of the data, using the same syntax as that in the previous section. Such an expression can also be combined interchangeably with all other expressions of SEE and Math-

ematica, including other endpoints. In what follows, I will demonstrate the use of Endpoint Manager and Experiment Explorer in deriving a new endpoint capturing a behavior pattern that was found to discriminate C57 and DBA mice across laboratories (Kafkafi, Pagis, et al., in press). I will start with a standard endpoint of SEE, the maximal speed to duration ratio (MSDR), which computes the ratio between the maximal speed attained in a progression segment and its duration (Kafkafi, Lipkind, et al., in press). It is thus a measure of the acceleration, or *abruptness*, of movement in this segment. The function returns the median of this value over all segments in the session. The results in this endpoint across three laboratories can be displayed using a graphics function of Endpoint Manager, `MeanSEPlot`, and the usual syntax of Experiment Explorer:

```
MeanSEPlot[
MaxSpeedToDurationRatio[]~MapOverGroups
~experimentGroups~MapOverExperiments~{"C57
vs DBA in NIDA","C57 vs DBA in MPRC","C57 vs
DBA in TAU"}]
```

Evaluating this expression will return the graph in Figure 3. Using the function `TwoWayAnova` instead of `MeanSEPlot` in the above expression will return an analysis of variance table of strains by laboratories for the same data. As can be seen in Figure 3, MSDR highly discriminates the two strains [ $F(1,45) = 109.1, p < .0001$ ], but its replicability across laboratories is low, as is shown by a significant laboratory effect [ $F(2,45) = 30.5, p < .0001$ ].

Another standard endpoint in SEE is lingering mean speed (LMS). This endpoint divides the distance traveled in the lingering mode (i.e., within stops) by the total du-

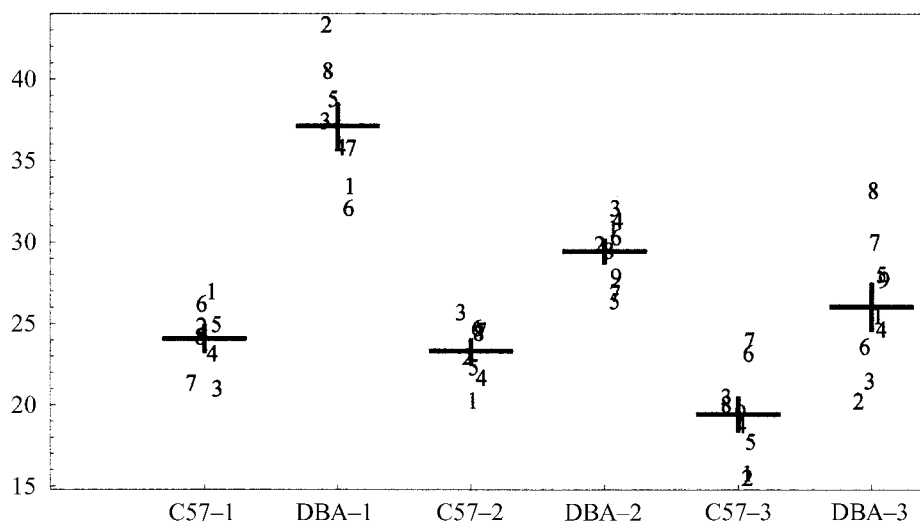


Figure 3. Maximal speed-to-duration ratio: group means and standard errors for C57 and DBA over three experiments in different laboratories (1, 2, and 3). Numbers label single sessions according to their order in the group.

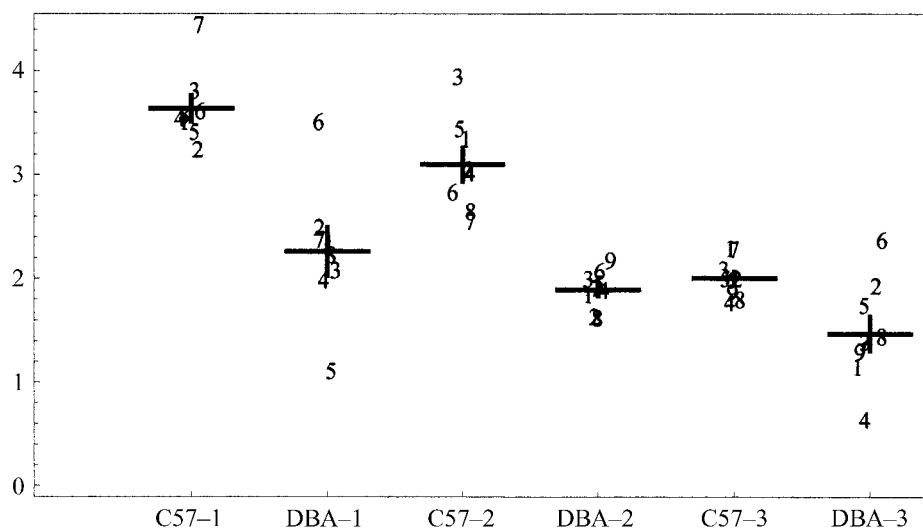


Figure 4. Lingering mean speed: group means and standard errors for C57 and DBA over three experiments in different laboratories (1, 2, and 3). Numbers label single sessions according to their order in the group.

ration of lingering. This endpoint thus gives indication of the mobility inside stops, which includes local movements, such as scanning, rearing, and sideways steps (Kafkafi, Lipkind, et al., in press). Using MeanSEPlot with the LMS across the three laboratories returns Figure 4, and using TwoWayAnova confirms that LMS also discriminates the strains [ $F(1,45) = 36.25, p < .0001$ ] but

is not well replicated across laboratories [ $F(2,45) = 4.8, p < .013$ ].

It is also possible, however, to look at the relation between these two endpoints. The expression

```
CorrelationPlot[
  {LingeringMeanSpeed[],MaxSpeedToDuration
  Ratio[]}
  ~MapOverGroups~experimentGroups~MapOver
  Experiments~
  {"C57 vs DBA in NIDA","C57 vs DBA in MPRC",
  "C57 vs DBA in TAU"}]
```

returns Figure 5.

Note that the combination of both endpoints discriminates the two genotypes across all three laboratories better than does each endpoint by itself. Moreover, within each genotype, the laboratories are similarly ordered and the group medians are approximately aligned along an imagined ray projecting from the coordinate (0, 10). The results all lie within a range of approximately 4.5 cm/sec for the LMS and 35 cm/sec<sup>2</sup> for the MSDR. As was elaborated in Kafkafi, Pagis, et al. (in press), such a relationship may be captured by defining a new endpoint termed darting (DART):

$$DART = \text{ArcTan}[(MSDR - 10)/35 / (LMS / 4.5)].$$

Such a definition can be readily implemented in the Endpoint Manager:

```
EndpointDefinition[DART]:=
  ArcTan[(MaxSpeedToDurationRatio[] - 10)/35 /
  (LingeringMeanSpeed[]/4.5)]
```

After this definition, DART can be used in any command that is viable for predefined endpoints. Applying MeanSEPlot to DART returns Figure 6, showing both

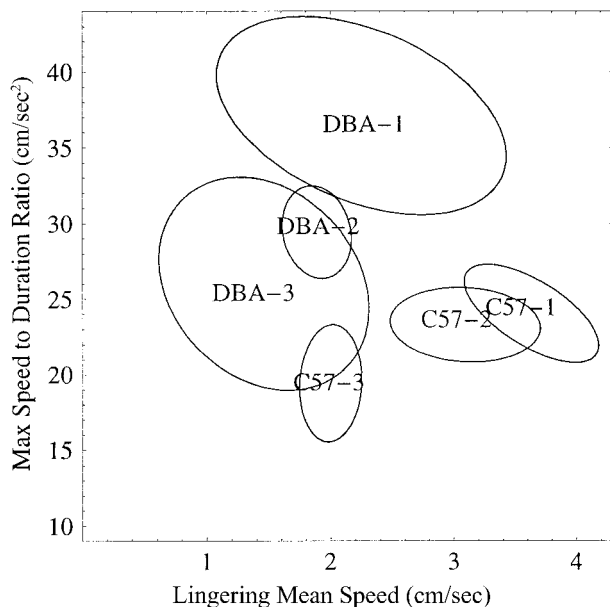


Figure 5. A correlation plot of the endpoints LingeringMeanSpeed and MaxSpeedToDurationRatio in two genotypes (C57 and DBA) over three experiments in different laboratories (1, 2, and 3). Ellipsoids include 75% of group results (i.e., individual sessions, one session per animal). Group titles are centered on the group medians.

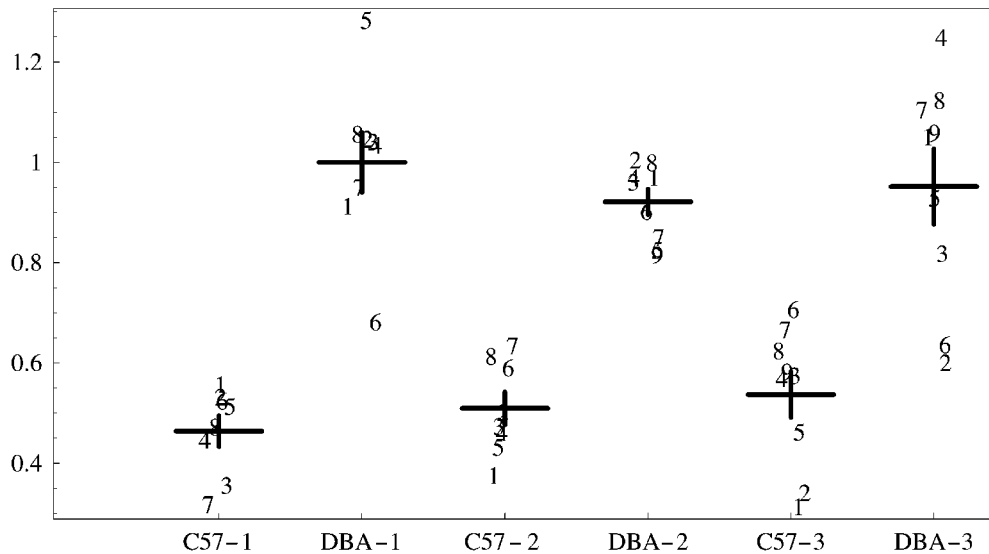


Figure 6. DART: group means and standard errors for C57 and DBA over three experiments in different laboratories (1, 2, and 3). Numbers label single sessions according to their order in the group.

high strain discrimination and high replicability across laboratories. This is tested by applying TwoWayAnova to DART [ $F(1,45) = 128.9, p < .0001$ , for strains;  $F(2,45) = 0.18, p = .83$ , for laboratories; and  $F(2,45) = 1.0, p = .37$ , for the strain  $\times$  laboratory interaction]. The experimenters' subjective observations agree that DBA mice, which have a high DART index, tend to freeze during stops and then shoot very quickly and abruptly to the next stop, where they freeze again. Such behavior is typical of wild rodents in their natural habitat and may also constitute a model for genetic inclination to anxiety or stress. Note that it is still required that any new endpoint be tested with data that were not used for its design. DART was indeed found to have high discrimination and replicability in three additional experiments (Kafkafi, Pagis, et al., in press).

### Discussion

In order to minimize manual labor and subjective judgment, researchers in behavior genetics and other fields of the behavioral neurosciences strive to automate behavioral tests. As ethologists, researchers' main intuition about behavior is that it consists of patterns with specific structures that usually can be identified immediately by the experienced observer. When examined closely, however, these structures are almost always found to be complex and difficult to capture algorithmically. To take a typical example, the algorithmic definition of the seemingly trivial pattern *stopping* (Drai et al., 2000) took almost 2 years to be developed and is still being fine-tuned on the basis of ongoing results. In the several years' experience with the programming of algorithms for automatic categorization and recognition of behavior patterns, we found that it is very easy to fail because the inherent structure of the pattern has not been

properly understood or even because of simple reasons, such as not using the right transformation. In such frequent cases of failure, the algorithm is most likely to measure noise and artifacts, and it should not be surprising if many genotypes cannot be differentiated and genotype differences cannot be reproduced in other laboratories, despite rigorous standardization (Crabbe et al., 1999). In order to handle this difficult task of algorithmically recognizing complex structures of behavior, it is much preferable to use flexible measurement systems that let the user visualize the behavior and to reprogram them accordingly. SEE was programmed with such flexibility in mind. In comparison, the analysis modules of most photobeam cages and tracking systems were designed by professional programmers, not by students of behavior, and they give the user little or no possibility of changing their programming.

The Experiment Explorer and Endpoint Manager presented here demonstrate how readily the functionality of SEE can be extended. Experiments conducted in several laboratories provide a unique opportunity for designing better measures, since they enable one to extract the robust aspects of behavior—those that depend on factors of interest, such as genotype or treatment—and to disregard aspects that depend on local, idiosyncratic factors, such as measurement artifacts or small differences in housing. Hence, designing endpoints from multilaboratory studies has a double benefit. For behavior geneticists and psychopharmacologists, it supplies measures that can be expected to better discriminate different genotypes in other laboratories. For ethologists, it supplies measures that properly and robustly capture innate behavior patterns and thus are more appropriate for the study of behavior in many other situations. The Experiment Explorer and Endpoint Manager were thus designed to

apply SEE to exploring a database of path coordinates from many experiments over many laboratories. For this, they have the following properties, some of which were demonstrated in the previous sections: (1) Any expression of SEE can be applied to any part of the database, using the same simple and intuitive syntax; (2) the user can easily add experiments to the database; (3) the user can easily define any alternative grouping in an experiment and then use it the same way the predefined grouping is used; (4) the user can easily define and set any experiment-specific variables, which will be then available for use in SEE expressions; (5) many established behavioral endpoints are already available for queries, using the same query format as that in the introduction; (6) endpoints can be combined with other endpoints or any other expression to generate complex queries; (7) any expression, including a combination of current endpoints, can be easily defined as a new endpoint, which can then be used the same way as predefined endpoints; (8) endpoint results can be visualized and statistically analyzed using specialized functions or the many tools available in Mathematica; and (9) the syntax is as intuitive as possible, so even a novice user can usually discern the meaning of an expression, at least approximately, from a first reading.

The approach that is advocated here is reminiscent of the approach that has been successfully employed in bioinformatics during recent years (e.g., Attwood, 2000; Pitt et al., 2001). The two components of this approach are (1) a database that includes the raw data (in this case the path coordinates in an open-field arena) from many experiments, conducted in several different laboratories, where increasing the size and scope of this database should directly increase the potential power of the produced measures, and (2) software for addressing any desirable subsection of this database and performing desirable computations over it, so as to recognize recurring patterns.

The path of the animal in the arena, in analogy to the base sequence of a DNA strand, is considered here as a structured and information-rich sequence that can be stored and reanalyzed. As in molecular genetics, the segmentation of this series into discrete and meaningful units (in our case, lingering episodes and progression segments) highly facilitates the analysis. Each of these units has simple properties, such as the distance traversed, duration, maximal speed, and so on, that can be used for analysis, and the animal usually performs hundreds of them during the session, providing a large sample size for statistical analysis. Another possible advantage of this approach is that many experimenters can contribute data to the database, whereas many data analysis specialists can study it without having to conduct their own experiments. Any progress in the design of new endpoints can be immediately employed by the experimenters in order to design new experiments and to reevaluate old experiments. This approach is, thus, more suitable for tackling the inherent complexity of behavioral phenotypes. Our database currently contains about 20 experiments, conducted in three laboratories with differ-

ent arenas, and includes 13 genotypes and hundreds of individual animals. The approach was practically demonstrated in Kafkafi, Lipkind, et al. (in press) and Kafkafi, Pagis, et al. (in press).

Behavior geneticists and psychopharmacologists are naturally apprehensive about their ability to learn the use of Mathematica and SEE in a short time. In our experience, students with little or no previous programming experience (including the author) were able to start working with Mathematica and SEE almost immediately, by using given examples and substituting in them the desired groups, experiments, endpoints, and so on. These students learned the use of the Mathematica language while working with it and, after several months, were able to program complex and powerful algorithms of their own. The interactive nature of Mathematica encourages novice users to scale up their programming by combining and transforming simple expressions into increasingly more elaborate ones. The elegance and generality of the syntax (which I aspired to follow in the packages presented here) educate the user in mathematical thinking without being overly didactic.

The single disadvantage of SEE stems from the same property of Mathematica that endows it with its unique advantages—that is, being a noncompilable language. This means that SEE is not a stand-alone program but requires Mathematica in order to run. A simple stand-alone program for calculating a set of endpoints is thus under way. The translation of endpoint algorithms developed in SEE into a compilable language, such as C++, is straightforward but still requires programming knowledge of this language. This stand-alone program, however, is designed to be a simple tool for conducting automated behavioral tests and does not include the visualization, analysis, database handling, and development capabilities presented here. For such capabilities, the Mathematica environment is necessary and practically irreducible. For the task of continuous development of measures by students of behavior, rather than by professional programmers, SEE and its extensions thus constitute the best alternative.

## REFERENCES

- ATTWOOD, T. K. (2000). The role of pattern databases in sequence analysis. *Brief Bioinformatics*, **1**, 45-59.
- BENJAMINI, Y., DRAI, D., ELMER, G. I., GOLANI, I., & KAFKAFI, N. (2001). Controlling the false discovery rate in behavior genetics research. *Behavioural Brain Research*, **125**, 279-284.
- COOLS, A. R., ELLENBROEK, B. A., GINGRAS, M. A., ENGBERSEN, A., & HEEREN, D. (1997). Differences in vulnerability and susceptibility to dexamphetamine in Nijmegen high and low responders to novelty: A dose-effect analysis of spatio-temporal programming of behavior. *Psychopharmacology*, **132**, 181-187.
- CRABBE, J. C., WAHLSTEN, D., & DUDEK, B. C. (1999). Genetics of mouse behavior: Interactions with lab environment. *Science*, **284**, 1670-1672.
- CRAWLEY, J. N. (2000). *What's wrong with my mouse? Behavioral phenotyping of transgenic and knockout mice*. New York: Wiley.
- CRAWLEY, J. N., BELKNAP, J. K., COLLINS, A., CRABBE, J. C., FRANKEL, W., HENDERSON, N., HITZEMANN, R. J., MAXSON, S. C., MINER, L. L., SILVA, A. J., WEHNER, J. M., WYNshaw-BORIS, A., & PAVLOV, R. (1997). Behavioral phenotypes of inbred mouse strains: Implications

- and recommendations for molecular studies. *Psychopharmacology*, **132**, 107-124.
- DRAI, D., BENJAMINI, Y., & GOLANI, I. (2000). Statistical discrimination of natural modes of motion in rat exploratory behavior. *Journal of Neuroscience Methods*, **96**, 119-131.
- DRAI, D., & GOLANI, I. (2001). SEE: A tool for the visualization and analysis of rodent exploratory behavior. *Neuroscience & Biobehavioral Reviews*, **25**, 409-426.
- DRAI, D., KAFKAFI, N., BENJAMINI, Y., ELMER, G., & GOLANI, I. (2001). Rats and mice share common ethologically relevant parameters of exploratory behavior. *Behavioural Brain Research*, **125**, 133-140.
- EILAM, D., & GOLANI, I. (1989). Home base behavior of rats (*Rattus norvegicus*) exploring a novel environment. *Behavioral Brain Research*, **34**, 199-211.
- GINGRAS, M. A., & COOLS, A. R. (1997). Different behavioral effects of daily or intermittent dexamphetamine administration in Nijmegen high and low responders. *Psychopharmacology*, **132**, 188-194.
- GOLANI, I., BENJAMINI, Y., & EILAM, D. (1993). Stopping behavior: Constraints on exploration in rats (*Rattus norvegicus*). *Behavioural Brain Research*, **53**, 21-33.
- KAFKAFI, N., LIPKIND, D., BENJAMINI, Y., MAYO, C. L., ELMER, I. G., & GOLANI, I. (in press). SEE locomotor behavior test discriminates C57BL/6J and DBA/2J mouse inbred strains across laboratories and protocol conditions. *Behavioral Neuroscience*.
- KAFKAFI, N., MAYO, C. L., DRAI, D., GOLANI, I., & ELMER, G. I. (2001). Natural segmentation of the locomotor behavior of drug-induced rats in a photobeam cage. *Journal of Neuroscience Methods*, **109**, 111-121.
- KAFKAFI, N., PAGIS, M., LIPKIND, D., MAYO, C. L., BENJAMINI, Y., GOLANI, I., & ELMER, I. G. (in press). Darting behavior: A quantitative movement pattern designed for discrimination and replicability in mouse locomotor behavior. *Behavioural Brain Research*.
- PITT, W. R., WILLIAMS, M. A., STEVEN, M., SWEENEY, B., BLEASBY, A. J., & MOSS, D. S. (2001). The Bioinformatics Template Library: Generic components for biocomputing. *Bioinformatics*, **17**, 729-737.
- ROGERS, D. C., JONES, D. N., NELSON, P. R., JONES, C. M., QUILTER, C. A., ROBINSON, T. L., & HAGAN, J. J. (1999). Use of SHIRPA and discriminant analysis to characterise marked differences in the behavioural phenotype of six inbred mouse strains. *Behavioural Brain Research*, **105**, 207-217.
- TARANTINO, L. M., GOULD, T. J., DRUHAN, J. P., & BUCAN, M. (2000). Behavior and mutagenesis screens: The importance of baseline analysis of inbred strains. *Mammalian Genome*, **11**, 555-564.
- TCHERNICHOVSKI, O., BENJAMINI, Y., & GOLANI, I. (1996). Constraints and the emergence of freedom in the ontogeny of rat exploratory behavior. *Behaviour*, **133**, 519-539.
- TCHERNICHOVSKI, O., BENJAMINI, Y., & GOLANI, I. (1998). The dynamics of long term exploratory behavior in the rat: Pt. I. *Biological Cybernetics*, **78**, 423-432.
- TCHERNICHOVSKI, O., & GOLANI, I. (1995). A phase plane representation of rat exploratory behavior. *Journal of Neuroscience Methods*, **62**, 21-27.
- WAHLSTEN, D. (2001). Standardizing tests of mouse behavior: Reasons, recommendations, and reality. *Physiology & Behavior*, **73**, 695-704.
- WHISHAW, I. Q., CASSEL, J. C., MAJCHRZAK, M., CASSEL, S., & WILL, B. (1994). "Short-stops" in rats with fimbria-fornix lesions: Evidence for change in the mobility gradient. *Hippocampus*, **4**, 577-582.
- WHISHAW, I. Q., HINES, D. J., & WALLACE, D. G. (2001). Dead reckoning (path integration) requires the hippocampal formation: Evidence from spontaneous exploration and spatial learning tasks in light (allothetic) and dark (idiothetic) tests. *Behavioural Brain Research*, **127**, 49-69.
- WOLFRAM, S. (1999). *The Mathematica book* (4th ed.). Champaign, IL: Wolfram Media.

(Manuscript received January 10, 2002;  
revision accepted for publication August 7, 2002.)