SESSION 4 ANALYTICAL AND VISUALIZATION TOOLS

*Chaired by Darrell L. Butler, Ball State University
and John H. Flowers, University of Nebraska, Lincoln*

Analysis and visualization of complex psychophysiological data: Coordinated use of multiple applications on a microcomputer

STEVEN P. REIDBORD and DANA J. REDINGTON
University of California, San Francisco, California

Personal computers have become ubiquitous for data analysis and visualization. Complex analyses and visualizations of large data sets prove challenging for microcomputer environments and highlight a tradeoff between customized programming to enhance specificity and processing speed and the use of commercially available software packages. We frame this choice in terms of a continuum that exists from low-level programming languages to higher-level languages, and beyond this to object-oriented programming environments and other advanced software products; we argue that there is no sharp conceptual distinction between programming and not programming. In this light, we discuss the pros and cons of using several commercial software packages in concert on a single platform to facilitate complex research tasks and discuss the implications of this choice for innovative research.

The analysis and visualization of complex data present a challenge for microcomputer environments. Large data sets, complex data transformations and statistical analyses, and the need for innovative data visualizations tax the capabilities of these relatively modest systems. Yet the ubiquity and low cost of personal computers in research settings argue for (and often necessitate) their use over more powerful minicomputers or mainframes. In this paper, we focus on the use of personal computers for complex scientific investigations and discuss the tradeoff between using commercial software and relying on special-purpose programming for these complex tasks.

It is always an option to write customized programs to satisfy each step of a complex analysis. Programming languages run the gamut from low level (e.g., C and assembly language) to higher level (e.g., BASIC and Smalltalk).

One typically weighs the faster processing speed of lower-level languages against the ease and rapidity of programming in higher-level languages.

This tradeoff extends readily to the choice to forgo formal programming and to use commercially available general-purpose software instead. In doing so, one trades specificity, customization, and usually processing speed for more economical use of human resources (see Figure 1). Our position is that for even relatively esoteric data analyses, commercial software employed with forethought and attention to data compatibility presents the more productive and economical alternative.

While these issues are quite generalizable, we will illustrate the requirements of a relatively complex research paradigm by referring to our own studies of psychophysiological data. Since we use the Apple Macintosh, our examples refer to Macintosh software, but the issues discussed here are applicable to other platforms as well.

The analysis and visualization of our large data sets involve several steps, which we first describe in general terms and later illustrate with actual data. We will not address data collection, which is beyond the scope of this paper. Our analytic approach calls for convenient tabular

This research was supported in part by the John D. and Catherine T. MacArthur Foundation Program on Conscious and Unconscious Mental Processes. Correspondence should be addressed to D. J. Redington, Department of Psychiatry, University of California, San Francisco, 401 Parnassus Ave., San Francisco, CA 94143 (e-mail: reidbord@macpsy.ucsf.edu or redwolf@macpsy.ucsf.edu).

Figure 1. Programming ease versus processing speed.

and graphical inspection of the raw data; preliminary statistical analyses of the time series, that is, autocorrelation and power spectrum plots; mathematical transformations of the time series to generate derived data sets; exploratory data visualization and spatial typologizing using three-dimensional scatterplots; other statistical analyses and plotting; and report and document generation. No single software application meets all these needs.

Tabular and graphical inspection of data. A great many simple data-handling tasks, including initial inspection and plotting of raw data, transformation, and tabular summaries, can be accomplished with spreadsheets. In our view, advanced spreadsheets are underutilized in research settings; they are remarkably flexible tools for data storage, manipulation, and graphic representation.¹ For example, the most recent version of Microsoft Excel adequately handles 3,000 data points in a single chart. Simple spreadsheet rules can be written to flag erroneous data, summarize intermediate results, and so forth. Various other packages, for example, DeltaGraph (Delta Point) and KaleidaGraph (Synergy Software), can be used in concert with spreadsheets to provide enhanced graphics, such as error bars on column plots.

Statistical analyses. While simple descriptive statistics are available in spreadsheets, more complex multivariate analyses are often only practical with intermediate- to high-level statistical packages. Both SYSTAT 5 (Systat) and StatView II (Abacus Concepts), for example, prove more than adequate for most parametric and nonparametric tests, summaries, and plotting. Odesta's DataDesk seems preferable for pilot efforts in data inspection and exploratory method prototyping. National Instrument's LabVIEW 2, discussed below, is a highly flexible product that, among its many functions, can perform customized statistical analyses and plots.

Mathematical transformations to generate derived data sets. Transformation of data can be accomplished with lesser or greater ease using spreadsheets, using statistical packages, or more elaborately with symbolic and mathematical processors such as Mathematica (Wolfram Research) or MathCad (MathSoft). The latter products directly interpret equations to transform data. Again, the need to include another software product depends largely on the researcher's desire to trade the cost of the software for savings in human programming time and effort.

Exploratory data visualization. Data visualization comes in varying degrees of sophistication. Spreadsheets now offer numerous two- and three-dimensional charts, although these tend to be limited in terms of viewing options and interactive visual manipulations. Some statistical packages that incorporate exploratory data-analysis techniques (e.g., DataDesk) are more versatile and interactive. Maximum versatility is found in dedicated three-dimensional plotting packages. For example, Abacus Software's MacSpin permits rotation of three-dimensional scatterplots and identification of data subsets that can be labeled by color and independently selected and visualized.

Report and document generation and presentation software. Report and document generation may be handled through virtually any word processing package. Advanced word processors and publishing programs can incorporate formatted tables and graphics into scientific documents. When used with a high-quality printer, such packages can produce camera-ready copy and in some instances can obviate the need for professional figure artwork. Aldus Persuasion and MacroMind Director are multimedia products that can enhance conference presentations, lectures, and the like by creating color slides and handouts of words and images imported from other applications.

Coordinating Among Software Packages

The use of multiple software applications would be a nightmare if not for a standard data format that allows easy transfer of information among applications. In almost all cases, it is possible to import and export data as ASCII files. In addition, proprietary formats can sometimes be read by other applications (e.g., SYSTAT 5 can read Excel files directly), but we have not found this feature partic-

	A	B	C
1	Heart Rate		
2	103.55		
3	99.45		
4	101.70		
5	102.86		
6	102.08		
7	100.56		
8	107.41		
9	103.00		
10	100.61		
11	100.00		
12	97.34		
13	95.96		
14	98.59		
15	100.56		
16	100.79		
17	100.00		
18	100.00		
19	101.27		
20	98.37		
21	95.47		
22	92.58		
	90.68		

Figure 2. Data in spreadsheet format.

ularly useful since the format is not shared among all applications.

Many commercially available programs run on several computer platforms. It is clearly advantageous to standardize one's "analytic workbench" by using a single, widely available platform that has a consistent and fully integrated user interface. The Macintosh operating system is designed for consistent user interface across different software products and has advanced graphic manipulation capabilities. This enables switching among a wide variety of software packages with minimal effort (since basic functions are similar across applications) and without the need to master multiple operating systems.

Additional Strengths and Weaknesses of Using Commercial Software

Before illustrating these points with actual data, there are additional strengths and weaknesses of commercial software to consider. Among the strengths are the following. Studies can be replicated more readily by other workers, who do not need to acquire (and understand) noncommercial, special-purpose software. Similarly, data formats are more likely to be standardized, allowing the sharing of data with other sites or investigators. Commercial software is more likely to be error free, and most vendors offer helpful software support. The user interface is often more intuitive and standardized in commercial software designed for specific platforms. Finally, the packages can be used multiple times for different kinds of studies and for other uses.

There are, of course, weaknesses as well. Commercial software may demand frustrating compromises in data

collection or analysis. For example, as noted above, Excel can chart 3,000 data points; our time series, however, usually have 3,300. MacSpin does not provide for copying the names or contents of defined data subsets out of its proprietary format and into something more standard (e.g., ASCII), so this desirable step cannot be automated. Some packages, for example, SYSTAT 5 and LabVIEW 2, offer so many options that one must wade through many settings to adjust them for a specific purpose. Finally, custom-made software can be a marketable product in its own right, and the use of commercial packages precludes special-purpose software sales.

An Example Using Actual Data

Novel approaches to psychophysiological analysis require all the functionality described above. In our investigations (Redington & Reidbord, 1991, in press), we study complex collections of physiological and clinical data taken from patients undergoing psychotherapy. Our focus is on nervous-system activity in both patient and therapist, with a specific interest in the nonlinear or "chaotic" dynamics of heart function and how this relates to psychological events during therapy. Each physiological data stream is collected over a 50-min therapy session plus a brief baseline period and constitutes a time series of approximately 3,300 data points (one datum per second per channel). We employ a multiple-step strategy to process and analyze these data.

The selection of hardware forms the foundation of this approach. All of our analyses are performed on a Macintosh IIfx. This platform balances the speed and power needed for these demanding tasks while at the same time

Figure 3. Example of time and frequency analyses.

Figure 4. Phase space construction in DataDesk.

being less expensive and easier to use than true workstations, minicomputers, or mainframes.

In this example, the data consist of 1 subject's heart rate for a single therapy hour. Approximately 3,300 values, constituting a time series of one heart rate value per second, are imported into Microsoft Excel in ASCII format. Such data inevitably contain artifacts; these erroneous values are identified by visual inspection of the spreadsheet and plots and by using spreadsheet rules to flag outliers. A small portion of the data at this stage is depicted in Figure 2.

A first step in nonlinear dynamic analysis is to conduct time and frequency domain analyses, which speak to the suitability of the data for further study under this paradigm. The data are imported into LabVIEW 2 to produce the requisite autocorrelation and power-spectrum plots; these are depicted in Figure 3.

Once shown to be suitable, the data are then "lowpass filtered" to remove the short-term component of heart rate variation. This is done readily in Excel by creating a moving average of the data in an adjoining spreadsheet column. These filtered, but still one-dimensional, data are next transformed into three related time series through a standardized lagging procedure used in nonlinear analyses (Takens, 1981). We have used either Excel or DataDesk for this step. Figure 4 shows DataDesk's interface.

The data now consist of three time series that are imported into MacSpin and visualized as an intricate three-dimensional plot. This depiction, known as a phase portrait, is then visually typed to discern characteristic three-dimensional data patterns. As such patterns are identified, they are color coded and labeled as defined subsets in

MacSpin. Figure 5 is a black-and-white depiction of one such subset in the phase portrait.

The type, location, and duration of the defined data subsets, called "trajectories," are entered (by hand, unfortunately) back into Excel and from there imported into SYSTAT 5 or StatView II to produce descriptive statistics and to test hypotheses concerning the relation of specific trajectory types to other time-coded clinical phenomena.

Finally, Microsoft Word is used to create reports and documents; statistical tables and graphics are imported directly from the applications that produced them. Presentation software is similarly used to create slides and other educational and presentation aids.

A flow diagram depicts the path taken by the data from one application to the next (see Figure 6).

Discussion

Quick, efficient means of prototyping ideas and evaluating data are often among the first steps of innovative scientific research. After an idea or method has been tested, it may then be refined for faster repeated performance, perhaps on larger data sets. This can mean optimizing the computational procedure by translating it into a lower-level programming language, although speed can also be purchased in the form of faster hardware or accelerator cards.

The tradeoff between processing speed and ease of prototyping extends from low-level programming languages, to higher-level languages, to object-oriented programming environments like LabVIEW 2, and finally to flexible general-purpose commercial applications. There is no sharp conceptual distinction between programming and not pro-

Figure 5. Phase space portrait.

gramming. In theory, a wide variety of applications could be programmed to produce a desired type of analysis. For example, advanced spreadsheets have macro capabilities that can be used to build analytic equations. Thus, complex data transformation is possible with spreadsheets, but this requires much programming time and expertise to create the needed macros. Analytic equations can be specified far more easily in Mathematica and MathCad with a real saving in human effort.

Advanced packages increasingly combine development environments and specific tools. MathCad and Mathematica allow the user to collect and save previously entered

mathematical formulae for later use; similarly, a programming language resembling Basic is included in SYSTAT 5 to extend its functions. A particularly compelling example of this trend is LabVIEW 2, an iconic, object-oriented programming environment. Customized "virtual instruments" can be "diagrammed" (i.e., programmed) and used repeatedly. In addition to its panoply of standard building-block functions, LabVIEW 2 also features add-on modules that can be purchased separately or that can be programmed on site. LabVIEW 2 straddles the fence between programming languages and software applications, typifying an accelerating trend in software design.

Figure 6. Data-flow diagram.

Such products are destined to become more popular as they become more affordable.

Flexibility in the use of different application packages provides a greater degree of freedom in accomplishing research tasks. There is never just one answer. By trying a variety of available products, one minimizes the tendency to opt for the familiar, or to be suspicious of products "not invented here." In particular, familiarity with the most advanced examples of each major class of software helps the researcher evaluate intelligently the capabilities and limitations of these products so he/she can purchase only those that offer real savings in time, effort, and money. While it may seem expensive to buy these packages, the labor cost of programming in a lower-level environment often proves far more expensive overall.

In this paper, we have discussed the benefits of constructing an "analytical workbench" using several commercially available software products in a sequential, coordinated fashion and have illustrated this approach with actual data. A conceptual continuum exists from low-level programming languages to special-purpose commercial software products, with an underlying dimension of processing speed versus ease of prototyping. Despite the lack of specificity and often slower processing speed, commer-

cial software is usually preferable to low-level programming when the researcher wishes to prototype an innovative method or analysis. These computational tasks may be streamlined and optimized later, after the method or analysis has proved its worth.

REFERENCES

- GALLISTEL, C. R. (1992). Classical conditioning as a nonstationary, multivariate time series analysis: A spreadsheet model. *Behavior Research Methods, Instruments, & Computers*, **24**, 340-351.
- REDINGTON, D. J., & REIDBORD, S. P. (1991). Toward non-linear psychodynamics: Exploring the chaotic nature of mind and body processes. In D. Duke & W. Pritchard (Eds.), *Measuring chaos in the human brain* (pp. 216-231). Singapore: World Scientific Publishing.
- REDINGTON, D. J., & REIDBORD, S. P. (in press). Non-linear dynamics in mind-body processes. In W. Ditto, L. Pecora, & T. Carrol (Eds.), *Proceedings of the First Experimental Chaos Conference*. Singapore: World Scientific Publishing.
- TAKENS, F. (1981). Detecting strange attractors in turbulence. In D. A. Rand & L. S. Young (Eds.), *Dynamical Systems and Turbulence, Warwick, 1980* (pp. 366-381). Berlin: Springer-Verlag.

NOTE

1. See, for example, the paper by Gallistel (1992) in this issue—*Editor*