ELSEVIER

Neurocomputing 38–40 (2001) 1025–1030

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Computer assisted neurophysiological analysis of cell assemblies activity

Alessandro E.P. Villa[a,*,1], Igor Tetko[a,b], Javier Iglesias[a]

[a]*Laboratory of Neuroheuristic, Institute of Physiology, University of Lausanne, Rue du Bugnon 7, CH-1005 Lausanne, Switzerland*

[b]*IBPC, Ukrainian Academy of Sciences, 253660, Kyiv, Ukraine*

Abstract

A critical feature of brain theories is whether neurons convey a noisy rate code or a precise temporal code. One of most valuable ways to test these theories consists in collecting the electrophysiological activity of cell assembles under several experimental conditions. The sequences of cell discharges—the spike trains—form time series whose dynamics is strongly related to the information processing carried out in the brain areas under study. Our purpose is to provide a user-friendly framework of a 'Virtual Laboratory' where computational neuroscience analyses and display of results can be distributed over a computer network, like Internet. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Multivariate time series; Multiple spike trains; Temporal coding; Virtual laboratory; Internet distributed software

1. Introduction

Internet activities have become in few years a major investment in information, business, communication and teaching technologies [8–11,4]. The World Wide Web (WWW) impact on society dramatically increases especially in the field of education and scientific research. Several large projects to develop object-oriented databases of neuronal data have been started in neuro-physiology by leading Universities [5]. One

* Corresponding author. Tel.: + 41-21-692-5534; fax: + 41-21-692-5505.
E-mail address: avilla@lnh.unil.ch, lnh@lnh.unil.ch (A.E.P. Villa).

[1] URL: www.neuroheuristic.org.

0925-2312/01/$ - see front matter © 2001 Elsevier Science B.V. All rights reserved
PII: S0925-2312(01)00379-4

of such project, the Human Brain Project funded by NIMH, will include somatosensory cortical neurons and characteristic neurophysiological data encapsulating these neurons' responses to specific stimuli. A second project will establish a database of molluscan identified neurons and the largely intracellular electrophysiological data for their description. However, in addition to becoming "a large database", Internet can also provide computational resources and can be a melting point of on-line algorithms and data analysis software (e.g., www.acdlabs.com/ilab/, www.spotfire.com/). In this article we describe a Virtual Laboratory for Data Analysis in Neurophysiology (DAN). The main idea of this project is to provide a user-friendly framework for analysis of neurophysiological data over Internet. Our focus is on methods of spike train data analysis recorded in multi-units experiments. A number of programs to analyse cell assemblies, including recently developed Pattern Grouping Algorithm, are available to the WWW user. This software can be easily extended by incorporation of new data analysis modules developed by other scientists.

2. Methods

The software is designed using programming languages C++ and Java. The Java language is used to develop the graphical interface of the program [3]. Since its beginning, the Java creators envisioned the same Java program running on many different types of computer chips and in many different operating environments—without modification. The Internet packages of Java make it possible to run the programs across Internet, to provide unprecedented possibilities for remote calculations, to search on large databases and to offer various dynamic client–server interactions. Another important feature of Java programs called applets is that they can be easily accessible over the Internet using the most popular WWW browsers, such as Netscape and Internet Explorer. This makes it possible to use low-cost computers as graphical terminals for interaction with the high-speed workstations and even to use specific Java programs, called 'applets', to be used through the interface of Internet browsers. The kernel of the analysis modules is written exclusively in C++ and the combination of both languages, Java and C++, is achieved by the Java Native Interface (JNI) [6]. The interface between Java and native code is programmed by declaring in Java program some native methods that are then implemented using C/C++ code. A call of such methods, i.e., C/C++ programs, allow a fast execution of time-critical code as well as re-use of the software C/C++ libraries previously developed by the users. The result of this call is directly parsed to Java program. At the same time the JNI framework lets the native method utilize Java objects in the same way that Java code uses these objects. A native method can access objects created by Java code, create its own Java objects and update them. The native method can call the existing Java method, pass it the required parameters, and get the results back when the method completes. The developed software consisted of three main programs:

- **Client Applet.** This is the only part of the software that is visible to the end-user. This program handles data files upload, specification of the parameters of analysis, task submission and display of the calculated results.
- **Super Server.** This program collects the requests from the client, dispatches the tasks to the registered Calculation Servers and sends the calculated results to the Client. The Super Server is very flexible because it does not need to know which kind of data or tasks are transferred. This makes it possible to extend the tasks handled by the Super Server and to add new applications without recompilation or even stopping the Super Server.
- **Calculation Servers.** One server can calculate several different tasks that are provided as native libraries. The number and specification of libraries are provided as external configuration files. Each specific task has its own library corresponding to a specific keyword. The list of keywords is uploaded from external configuration files from the Calculation Server and provided to the Super Server during its registration.

3. Results

The DAN software available from our Web site (www.neuroheuristic.org) includes Raster Viewer, Renewal Density Histograms and Dynamical System Analysis.

3.1. Input data format

The data format proposed by Abeles [1] was chosen as the standard for coding information related to multivariate discrete time series. Both neurons and stimulus are considered as events and coded using a triplet of numbers. The first number describes the type of the event (e.g., a neuronal discharge is coded by type equal to 1). The second number qualifies the event (i.e., 1,1 and 1,2 to indicate spikes of neurons 1 and 2, respectively). The third number is the time delay from the previous event. The triplet can be separated with any punctuation mark. There are also few reserved events (e.g. to indicate the end of data acquisition). This format is very simple and data in other formats can be easily converted to it.

3.2. Raster viewer (RV)

Discrete time series of point processes (e.g., sequences of neuronal discharges—the spike trains) are displayed by RV as dot rasters. Each line of the display corresponds to a time interval (e.g., a single trial or a repetition of a stimulus) that is repeated one line on top of the other. Each dot represents the occurrence of an event (e.g., a cell discharge). The rasters are aligned by a trigger event that is, for example, e.g. start of a trial or onset of the subject's response. The user can customize the layout of RV by selecting the number of rasters on screen, the number of rasters per line of display and their duration. The rasters are dynamically controlled with a slider.

3.3. Renewal density histograms

Histograms represent a standard and a convenient method to evaluate important parameters of neuronal spike train data. The Autocorrelation Histogram is used widely to describe the temporal pattern of an observed point process. Homogeneous Poisson processes are characterized by flat autocorrelograms. Instead, neurons

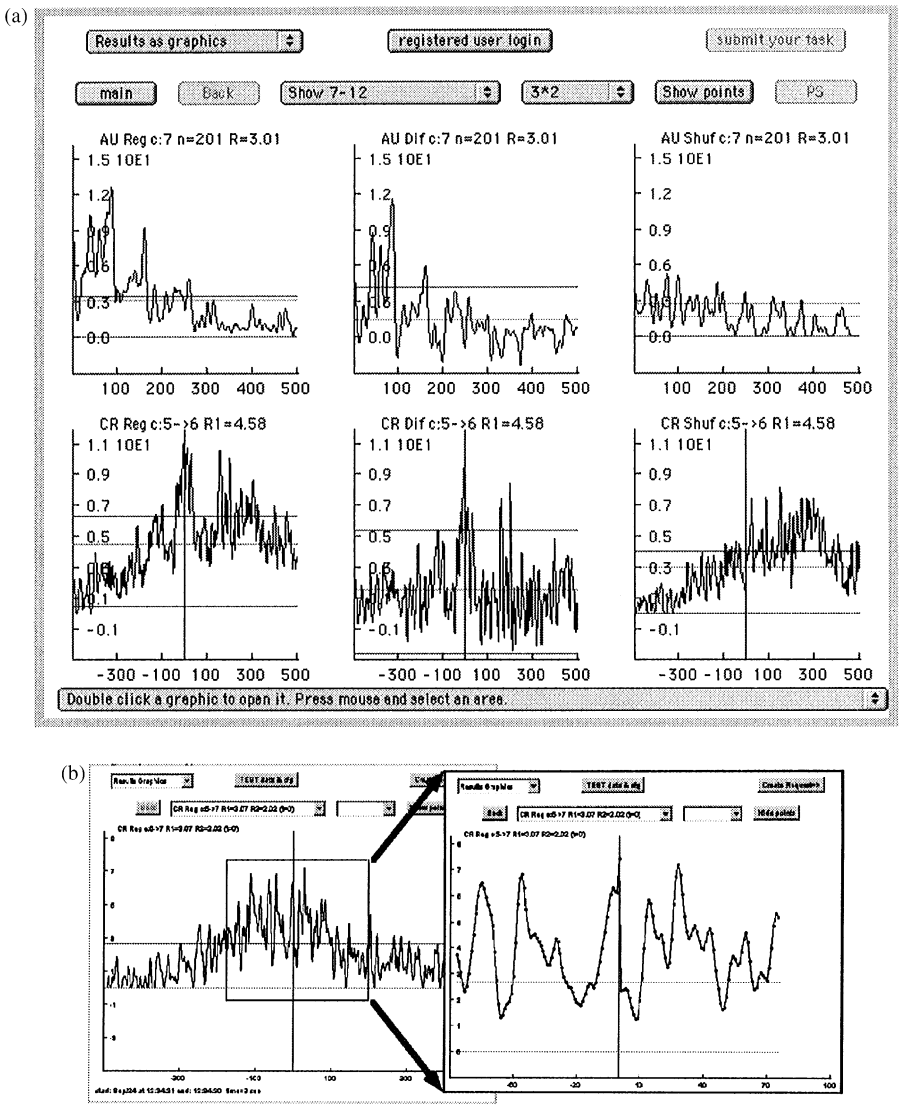

Fig. 1. (a) Several renewal density histograms of auto- and cross-correlations can be displayed on the same page or (b) can be displayed as a single graph and zoomed by the user.

characterized by bursting activity are characterized by a hump near zero lag. For two time series recorded simultaneously the Cross-correlation Histogram is often used to describe the interactions between the point processes. In case of an external event, it is possible to study whether the time series are locked to that event by computing the Peri-event-time Histogram. All three types of histograms can be calculated by the user from the Virtual Laboratory page. The configuration parameters of histograms are specified with the help of keywords. The software allows to display 1, 2, 6 or 12 histograms simultaneously, zoom over the region of interest as well as to create postscript files 1 (Fig. 1).

3.4. Dynamical system approach

The software available on-line allows to calculate important parameters of discrete time series, such as the embedding and correlation dimensions using the Grassberger and Procaccia algorithm [7] as well as maximum Lyapunov exponents [12]. Experimental neuronal data are often composed of a limited number of points (in the order of few hundreds to several thousands). On the A method available on the Virtual Laboratory site [2] can be applied to test low-dimensional determinism of a short time series (400–1000 points) and it provides a reliable estimate of the maximum Lyapunov exponents.

4. Discussion

We have developed a framework of a 'Virtual Laboratory' where computational neuroscience analyses and display of results can be distributed over a computer network. Our approach can be used to integrate specialized spike train analyses already developed in programming languages like C++ or Fortran and make them accessible to a broad audience over Internet because new plug-in modules can be easily incorporated in the existing software. It can be run on different computer platforms of the end-users and it does not require multiple installation steps. Such software is also very secure and cannot propagate viruses or Trojan horses. While mainly developed for neurophysiologists, this software can also be applied to analyze any type of time series data, from financial forecasting to celestial mechanics and heart disorders. The code will be disseminated free of charge to partner laboratories and non-profit educational and research centers.

Acknowledgements

This study was partially supported with Swiss NSF 2053-055753.98 and INTAS-OPEN 97-0168 and 97-0173 grants. We thank Dmytro Filipov, Tetyana Aksenova, Alessandra Celletti, Vladimir Volkovich, Luc Jeandenans, Michel Gautero and Vsevolod Tanchuk for their contribution to the software development and the courtesy of Alexa Riehle for contributing some demonstration data files.

References

[1] M. Abeles, Corticonics: Neural Circuits of the Cerebral Cortex, Cambridge University Press, New York, 1991.
[2] A. Celletti, C. Froeschlé, I.V. Tetko, A.E.P. Villa, Deterministic behaviour of short time series, Meccanica 252 (1999) 1–8.
[3] P. Chan, Java Almanac, Addison Wesley, Reading, MA, 1999.
[4] T.J. Filler, J. Jerosch, E.T. Peuker, Live interdisciplinary teaching via the internet, Comp. Meth. Prog. Biomed. 61 (2000) 157–162.
[5] D. Gardner, S.M. Erde, Database schemas for electrophysiological datasets may complement genomic and protein sequences, Biophys. J. 72 (1997) A118.
[6] R. Gordon, Essential JNI: Java Native Interface, Prentice Hall, Colorado, 1998.
[7] P. Grassberger, I. Procaccia, Characterization of strange attractors, Phys. Rev. Lett. 50 (1983) 346–349.
[8] S.R. Heller, Chemistry on the internet—the road to everywhere and nowhere, J. Chem. Inf. Comput. Sci. 38 (1998) 956–965.
[9] M. Ryan, C.W. Mulholland, W.S. Gilmore, Applications of computer-aided learning in biomedical sciences: considerations in design and evaluation, Brit. J. Biomed. Sci. 57 (2000) 28–34.
[10] M.A. Stoodley, M. Foroohar, R.L. Macdonald, B.K.A. Weir, Multimedia computer database for neurosurgery, Neurosurgery 47 (2000) 178–186.
[11] A.P. Tonge, H.S. Rzepa, H. Yoshida, Authentication of internet-based distributed computing resources in chemistry, J. Chem. Inf. Comput. Sci. 39 (1999) 483–490.
[12] A. Wolf, J.B. Swift, H.L. Swinney, J.A. Vastano, Determining Lyapunov exponents from a time series, Physica D 16 (1985) 285–317.