Journal of Neuroscience Methods 211 (2012) 245–264

Contents lists available at SciVerse ScienceDirect

Journal of Neuroscience Methods

journal homepage: www.elsevier.com/locate/jneumeth

Computational neuroscience

# nSTAT: Open-source neural spike train analysis toolbox for Matlab

I. Cajigas[a,b,c,*], W.Q. Malik[a,c], E.N. Brown[a,b,c]

[a] Department of Anesthesia and Critical Care, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114, United States
[b] Harvard-MIT Division of Health Sciences and Technology, Massachusetts Institute of Technology, Cambridge, MA 02139, United States
[c] Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139, United States

## HIGHLIGHTS

- We have developed the neural spike train analysis toolbox (nSTAT) for Matlab®.
- nSTAT makes existing point process/GLM methods for spike train analysis more accessible to the neuroscience community.
- nSTAT adopts object-oriented programming to allow manipulation of data objects rather than raw numerical representations.
- nSTAT allows systematic building/testing of neural encoding models and allows these models to be used for neural decoding.

## ARTICLE INFO

*Article history:*
Received 15 January 2012
Received in revised form 6 August 2012
Accepted 7 August 2012

*Keywords:*
Open-source
Point process
Generalized linear models
Neuroscience
Statistics
Data analysis
Signal processing
Spike trains

## ABSTRACT

Over the last decade there has been a tremendous advance in the analytical tools available to neuroscientists to understand and model neural function. In particular, the point process – generalized linear model (PP-GLM) framework has been applied successfully to problems ranging from neuro-endocrine physiology to neural decoding. However, the lack of freely distributed software implementations of published PP-GLM algorithms together with problem-specific modifications required for their use, limit wide application of these techniques. In an effort to make existing PP-GLM methods more accessible to the neuroscience community, we have developed nSTAT – an open source neural spike train analysis toolbox for Matlab®. By adopting an object-oriented programming (OOP) approach, nSTAT allows users to easily manipulate data by performing operations on objects that have an intuitive connection to the experiment (spike trains, covariates, etc.), rather than by dealing with data in vector/matrix form. The algorithms implemented within nSTAT address a number of common problems including computation of peri-stimulus time histograms, quantification of the temporal response properties of neurons, and characterization of neural plasticity within and across trials. nSTAT provides a starting point for exploratory data analysis, allows for simple and systematic building and testing of point process models, and for decoding of stimulus variables based on point process models of neural function. By providing an open-source toolbox, we hope to establish a platform that can be easily used, modified, and extended by the scientific community to address limitations of current techniques and to extend available techniques to more complex problems.



## 1. Introduction

Understanding and quantifying how neurons represent and transmit information is a central problem in neuroscience. Whether it involves understanding how the concentration of a particular chemical present within the bath solution of an isolated neuron affects its spontaneous spiking activity (Phillips et al., 2010) or how a collection of neurons encode arm movement information (Georgopoulos et al., 1986), the neurophysiologist aims to decipher how the individual or collective action potentials of neurons are correlated with the stimulus, condition, or task at hand.

Due to the stereotypic all-or-none nature of action potentials, neural spiking activity can be represented as a point process, a time series that takes on the value 1 at the times of an action potential and is 0 otherwise (Daley and Vere-Jones, 1988). Many other common phenomena can be described as point processes ranging from geyser eruptions (Azzalini and Bowman, 1990) to data traffic within a computer network (Barbarossa et al., 1997). Generalized linear models (GLMs) (McCullagh and Nelder, 1989), a flexible generalization of linear regression, can be used in concert with point process models to yield a robust and efficient framework for analyzing neural spike train data. This point process – generalized linear model (PP-GLM) framework has been applied successfully to a broad range

* Corresponding author. Tel.: +1 617 407 3367.
*E-mail address:* iahn@mit.edu (I. Cajigas).

0165-0270/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jneumeth.2012.08.009

of problems including the study of cardiovascular physiology (Chen et al., 2011, 2010a,b, 2008b, 2009a, 2008a; Barbieri and Brown, 2004, 2006a,b; Barbieri et al., 2005a), neuro-endocrine physiology (Brown et al., 2001), neurophysiology (Schwartz et al., 2006; Frank et al., 2002, 2004, 2006; Eden et al., 2004b; Vidne et al., 2012), and neural decoding (Barbieri et al., 2004, 2008; Eden et al., 2004a; Srinivasan et al., 2006, 2007; Wu et al., 2009; Ahmadian et al., 2011). Truccolo et al. (2005) and Paninski et al. (2007) provide a broad overview of the PP-GLM framework.

While much progress has been made on the development and application of PP-GLM methods within neuroscience, the use of these methods typically requires in-depth knowledge of point process theory. Additionally, while there are widely available implementations for the estimation of GLMs within software packages such as Matlab® (The Mathworks, Natick, MA), S, or R programming languages, their use to analyze neural data requires problem specific modifications. These adaptations require much work on the part of the experimentalist and detract from the goal of neural data analysis. These barriers are exacerbated by the fact that even when software implementations are made publicly available they vary greatly in the amount of documentation provided, the programming style used, and in the problem-specific details.

Numerous investigators have successfully addressed common problems within neuroscience (such as spike sorting, data filtering, and spectral analysis) through the creation of freely available software toolboxes. Chronux (Bokil et al., 2010), FIND (previously MEA-Tools) (Meier et al., 2008; Egert et al., 2002), STAToolkit (Goldberg et al., 2009), and SPKtool (Liu et al., 2011) are a few examples of such tools. Chronux offers several routines for computing spectra and coherences for both point and continuous processes along with several general purpose routines for extracting specified data segments or binning spike time data. STAToolkit offers robust and well-documented implementations of a range of information-theoretic and entropy-based spike train analysis techniques. The FIND toolbox provides analysis tools to address a range of neural activity data, including discrete series of spike events, continuous time series and imaging data, along with solutions for the simulation of parallel stochastic point processes to model multi-channel spiking activity. SPKtool provides a broad range of tools for spike detection, feature extraction, spike sorting, and spike train analysis. However, a simple software interface to PP-GLM specific techniques is still lacking.

The method for data analysis within the PP-GLM framework is consistent and amenable to implementation as a software toolbox. There are two main types of analysis that can be performed: (1) encoding analysis and (2) decoding analysis. In an encoding analysis, the experimenter seeks to build a model that describes the relationship between spiking activity and a putative stimulus and covariates (Paninski et al., 2007). This type of analysis requires model selection and assessing goodness-of-fit. A decoding analysis estimates the stimulus given spiking activity from one or more neurons (Donoghue, 2002; Rieke, 1999). An example of this type of analysis would aim to estimate arm movement velocity given the population spiking activity of a collection of M1 neurons and a model of their spiking properties such as that developed by Moran and Schwartz (1999).

We use the consistency of the data analysis process within the PP-GLM framework in the design of our neural spike train analysis toolbox (nSTAT). Our object-oriented software implementation incorporates knowledge of the standard encoding and decoding analysis approaches together with knowledge of the common elements present in most neuroscience experiments (e.g. neural spike trains, covariates, events, and trials) to develop platform that can be used across a broad range of problems with few changes. Object-oriented programming (OOP) represents an attempt to make programs more closely model the way people think about and interact with the world. By adopting an OOP approach for software development we hope to allow the user to easily manipulate data by performing operations on objects that have an intuitive connection to the experiment and hypothesis at hand, rather than by dealing with raw data in matrix/vector form. Building the toolbox for Matlab®, we make sure that users can easily transfer their data and results from nSTAT to other public or commercial software packages, and develop their own extensions for nSTAT with relative ease.

nSTAT address a number of problems of interest to the neuroscience community including computation of peri-stimulus time histograms, quantification of the temporal response properties of neurons (e.g. refractory period, bursting activity, etc.), characterization of neural plasticity within and across trials, and decoding of stimuli based on models of neural function (which can be prespecified or estimated using the encoding methods in the toolbox). Importantly, the point process analysis methods within nSTAT are not limited to sorted single-unit spiking activity but can be applied to any binary discrete spiking process such as multi-unit threshold crossing events (see for example Chestek et al., 2011). It should be noted that while all of the examples presented in the paper focus on the PP-GLM framework, nSTAT contains methods for analyzing spike trains when they are represented by their firing rate and treated as a Gaussian time-series instead of a point process. These include time-series methods such as Kalman Filtering (Kalman, 1960), frequency domain methods like multi-taper spectral estimation (Thomson, 1982), and mixed time-frequency domain methods such as the spectrogram (Cohen and Lee, 1990; Boashash, 1992). For brevity, and because these methods are also available in other toolboxes, we do not discuss these elements of the toolbox herein.

This paper is organized as follows: Section 2.1 summarizes the general theory of point processes and generalized linear models as it applies to our implementation. We include brief descriptions of some of the algorithms present in nSTAT to establish consistent notation across algorithms developed by distinct authors. Section 2.2 describes the software classes that make up nSTAT, the relationships among classes, and relevant class methods and properties. Section 2.3 describes five examples that highlight common problems addressed using the PP-GLM framework and how they can be analyzed using nSTAT. Lastly, results for each of the different examples are summarized in Section 3. nSTAT is available for download at http://www.neurostat.mit.edu/nstat/. All examples described herein (including data, figures, and code) are contained within the toolbox help files. The software documentation also includes descriptions and examples of the time-series methods not discussed herein for brevity.

## 2. Materials and methods

### 2.1. Summary of the PP-GLM framework

In this section, we describe the PP-GLM framework and the model selection and goodness of fit techniques that can be applied within the framework to select among models of varying complexity. The peri-stimulus time histogram (PSTH) and its PP-GLM analogue, the GLM-PSTH, are then presented together with extensions that allow for estimation of both within-trial and across-trial neural dynamics (SSGLM). We then discuss how point process models can be used in decoding applications when neural spiking information is used to estimate a driving stimulus.

#### 2.1.1. Point process theory

Due to the stereotyped all-or-none nature of action potentials, neural spiking activity can be represented as a point process, i.e. as a time series that takes on the value 1 at the time of an action