Sharing of Analytic Methods in Electrophysiology to Applications by Third Parties

Jan Stebetak [1,*], **Roman Moucek** [2]

[1] *University of West Bohemia, Department of Computer Science and Engineering, Pilsen, Czech Republic.*
[2] *NTIS*

Correspondence*:
Corresponding Author
Laboratory X, Institute X, Department X, Organization X, Street X, City X , State XX (only USA, Canada and Australia), Zip Code, X Country X, email@uni.edu

ABSTRACT

Neuroinformatics laboratories produce a lot of experimental data, which have to be stored and further processed. Storing and sharing data among laboratories is usual; however, sharing analytic methods is still not satisfactorily solved. The methods are usually accessible for users only via web browsers or downloadable and locally installable applications (e.g. EEGLAB). This paper presents a new approach of sharing analytic methods in the electrophysiology domain consisting in implementation of methods as Web Services. The proposed approach allows sharing of methods to third parties by integration the Web Services client into another application. The results demonstrate the implementation of this sharing technique to the EEG Data processor and the integration of methods into the EEG/ERP Portal.

Keywords: Electroencephalography; Event-related potential; Analytic methods; Signal processing; Web Services; Sharing

1 INTRODUCTION

Our research group specializes in the research of brain activity. The experiments in this area of research usually take long time and produce a lot of data. To analyse this data, we widely use the methods of electroencephalography (EEG) and event related potentials (ERP). The analytic methods that we use are presented.

Sharing experimental data or algorithms is very helpful for scientific community. They are not required to collect data or develop their own analytic methods. However, lack of standardized data format for shared data causes possible misinterpretation. To reduce the danger of misinterpretation, data must be very well documented by suitable metadata Teeters (2008). As members of the Czech National Node of International Neuroinformatics Coordinating Facility (INCF) INCF (2001) we defined and developed an EEG/ERP metadata description of electrophysiology experiments. Our efforts resulted in a custom solution the EEG/ERP Portal Jezek P. (2010). We also focused on design and implementation of analytic algorithms. We developed an analytic tools management system the EEG Data Processor Jezek P. (2013).

Currently, analytic methods are usually stored within a software infrastructure or a portal solution (as presented in the State of the Art). Scientists are able to use these methods via web user interface or download as an application and install it locally. The main disadvantage is a necessity of uploading data into a portal. There are also difficulties with the integration of methods into a different, third-party application. In this paper, we present a new approach of sharing methods in the electrophysiology domain with applications developed by third parties. The presented solution allows users integration of the analytic methods of the EEG Data Processor (where the proposed approach is implemented) into other applications and run them remotely.

This paper discusses the importance of sharing methods for the scientific community. It brings a brief introduction to existing neuroinformatics infrastructures allowing storing and sharing data and using analytic methods. It also discusses their approach of sharing data and methods. The next section describes the approach of sharing the analytic methods with third parties and its implementation. The last section of the paper shows the integration of the EEG Data Processor methods into the EEG/ERP Portal that serves as a data management application. This integration allows users of the EEG/ERP Portal executing the methods directly in the Portal where their experimental data are stored. They do not have to download their data from the Portal and upload it to the EEG Data Processor Jezek P. (2013) (see the Section 2.2).

2 MATERIAL & METHODS

This section discusses the importance of analytic tools and data sharing for scientific community. It also briefly presents existing neuroinformatics infrastructures and applications allowing sharing data and/or analytic methods. Then, the EEG Data Processor is presented. Finally, the proposed third-party sharing approach is presented.

2.1 State of the Art

There is a growing community of theorists in neuroscience who are trained in analytic methods but that had no direct access to experimental data. Therefore, they created mathematical models that were not guided by data and often only of limited relevance to neuroscience. Data sharing allowed theorists testing developed methods on real data Teeters (2008). There is also a community of experimentalists that obtain data but have limited access to analytic methods. Sharing of methods allows them analysing and validating their data. The following paragraphs bring an overview of existing contributions to providing and sharing experimental data and/or analytic methods.

The CARMEN Portal Watson P. and P. (2007) (Code Analysis, Repository & Modelling for e-Neuroscience) developed by the British National Node allows neuroscientists to save and share experimental data and services. CARMEN provides storage of services. Collaborators are able to upload and share experimental data, they also can upload their analytic tools as web services and share them. There are number of public services such as data filters, neural spike detection and spike sorting methods. All analytic tools are accessible for registered users via the CARMEN Portal. These tools can be run on the stored data that a user has access to. However, the services belong to owners that uploaded them to the CARMEN. When a scientist wants to use a CARMEN service anywhere else, he/she has to contact an owner of the service.

The Galaxy project Goecks et al. (2010); Blankenberg et al. (2010); Giardine et al. (2005) is an open source workflows engine. A registered user is able to use methods and workflow tools provided by this system. Galaxy is focused on genome analysis; therefore, this system contains methods suitable for genome analysis. Registered users are not allowed to add their own methods, only administrators or developers can. The methods are well described for the users with description of parameters and examples. Theoretically, the implementation of methods as Web Services enables remote integration. Currently, there is no support for remote methods calling by external applications.

Modular Toolkit for Data Processing (MDP) Zito T. and P. (2008) is a data processing framework used for scientific data processing development. From the users perspective, MDP consists of a collection of supervised and unsupervised learning algorithms e.g. Principal Component Analysis, several Independent Component Analysis algorithms (CuBICA, FastICA, TDSEP, and JADE), Locally Linear Embedding. It also allows creating and executing processing flows. MDP is a modular framework written in Python that programmers can extend by additional modules MDP (2012). Common users can call implemented modules locally.

EEGLAB Delorme A. (2004) is a toolbox with graphic user interface for EEG signal analysis. Data are imported via GUI from local storage e.g. hard disc or USB devices. Available functions include EEG data, channel and event information importing, data visualization (scrolling, scalp map and dipole model plotting, plus multi-trial ERP-image plots), preprocessing (including artifact rejection, filtering, epoch selection, and averaging), Independent Component Analysis (ICA) and time-frequency decompositions. A 'plug-in' facility allows easy incorporation of new EEG modules into the main menu. EEGLAB is written in MatLab and is freely available (http://www.sccn.ucsd.edu/eeglab/) under the GNU public license for non-commercial use and open source development, together with sample data, user tutorial and extensive documentation.

A novel software and hardware infrastructure developed by the German Neuroinformatics Node (G-Node) Herz AVM (2008); GNode (2009) eases the acquisition, storage and analysis of experimental data. G-Node provides various services and tools to facilitate data access and data management (Neuroshare Neuroshare (2001) to HDF5 converter, etc.). They have been developing a Rest API to enable easy access to shared data from different applications Sobolev A. (2014). The G-Node focuses on cellular and systems neurophysiology. The G-Node portal is open source; the source code including analytic tools is available on GitHub: https://github.com/G-Node.

Taverna [] is an open source domain independent scientific workflow management application that allows designing, sharing and executing workflows. Taverna has been created by the myGrid project []. It allows users to integrate many different software component, including SOAP or Rest web services, or command tools. Taverna can access a large number of services in the fields of bioinformatics, astronomy, chemoinformatics, health informatics and others. A Suite of tools include: a Taverna Workbench (toolbox with GUI for local installation), Taverna Online web based tool for authoring workflows; Taverna Server and client that allows Taverna workflows to be run on other machines.

The overview in Table 1 shows ability to share methods to end users (using them via web portal or installing a stand-alone system including methods on a custom machine). However, it also shows lack of support for integration of methods in external applications developed by third parties. Therefore, we present such approach of sharing methods that supports the mentioned integration.

2.2 EEG Data Processor

EEG Data Processor Jezek P. (2013) is a custom system developed by ours research group that enables running of signal processing methods. We use the following methods for EEG/ERP signal analysis: Matching Pursuit Vareka (2012), Discrete and Continuous Wavelet transform Ciniburk J. (2010), FastICA

Table 1. Summary of infrastructures and method providers

	Type	Data Sharing	Sharing Methods	Integration to third parties
CARMEN	Web portal	YES	Via web portal	Contact owner
Galaxy	Web portal	YES	Via web portal	Not supported (enabled by used technology)
MDP	Framework	NO	Download and import	NO
EEGLab	Stand-alone system	NO	Download and install	NO
G-Node	Web portal	YES	Via web portal	NO
Taverna	Set of applications	NO	Download with applications	Not supported

Hyvarinen A. and E. (2001) and Fast Fourier transform DATAQ (2011). Since this system is a web based application, it does not need any installation, only web browser. It enables users upload and analyse data, then download results. The methods are stored locally in the file system of the server. Each method is a plug-in (a JAR file) and is called by an external method invoker. The source code of the EEG Data Processor and the analytic methods is open source and freely available to download from the SourceForge repository (https://sourceforge.net/projects/eegprocessor/). Note that the EEG Data Processor is not developed for storing data. It stores it only temporarily. The following paragraphs include technical information about the EEG Data Processor.

The system is a layered architecture. This architectural style is supported by used programming languages and technologies (Java, Maven, Spring, Hibernate, Apache CXF Web services etc.) Jezek P. (2013).

2.3 Third-Party Sharing Approach

The possibility to share analytic methods is important as well as possibility to share experimental data. Sharing of analytic methods improves the efficiency of scientific work. Scientists can use published tools e.g. EEGLab or use already implemented methods.

In this paper, we present an extension of the EEG Data Processor that allows sharing analytic methods by third-party applications. The presented concept of sharing methods consists in registration of applications developed and run by third parties into the EEG Data Processor. When an application is registered and accepted by the administrator of the EEG Data Processor, the application is able to use methods provided by the Processor. Figure 1 shows the communication between the EEG Data Processor and a client application.

The main idea of sharing the methods provided by the EEG Data Processor is to allow using them directly in other applications. This approach removes the necessity of manually data transfer from the data storage to the EEG Data Processor. Since the methods are available and runnable from client applications where data are stored, the data transfer is performed automatically. The proposed approach also solves the difficulties with downloading and updating methods. Since the methods are accessed remotely, there is no need to download or update them manually.

This solution also provides an opportunity to call shared methods remotely from different applications independently of programming language. The type of a client application (web or desktop application) is

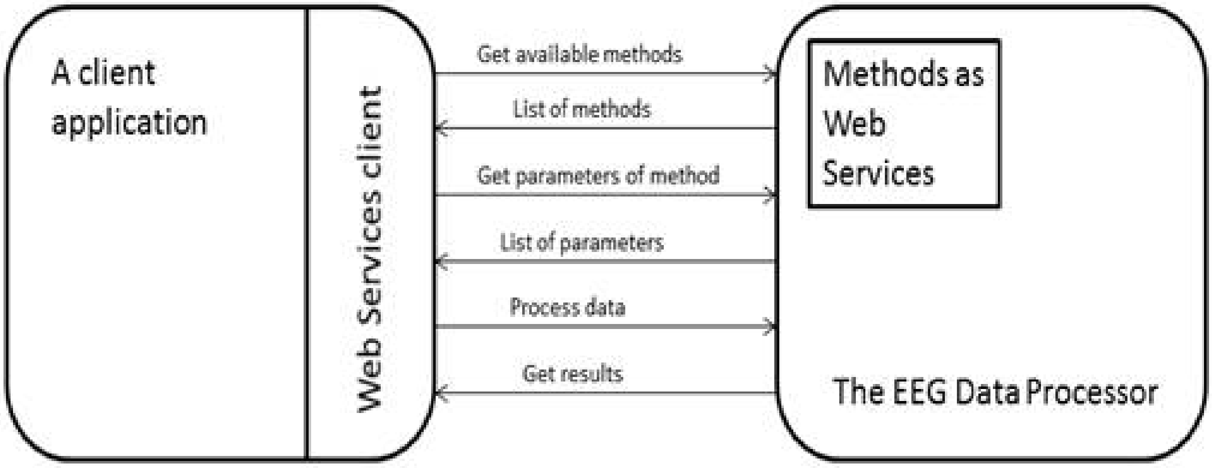

Figure 1. Communication between applications

also not relevant. For successful integration, the internet connection and the implemented client for remote methods call are required.

3 RESULTS

3.1 Third-Party Sharing Implementation

This section presents the third-party sharing technique and its implementation in the EEG Data Processor. Since we use the Web Services technology for the implementation, this section also provides its brief description. We propose the third-party sharing mechanism implemented in the EEG Data Processor that extends the common sharing of methods and enables integration of the provided methods to other applications and using them there.

The proposed approach of sharing methods is based on Remote Procedure Call technique (RPC). RPC is widely used for constructing distributed, client-server based applications. A client application calls a remote procedure (method), transfers data to a server application, and waits for a result. For web applications Web Services technology Jie Liu and yong Lv (2006) is used. For implementation of this approach, the SOAP Web Services technology is used. SOAP Web Services Jie Liu and yong Lv (2006) use XML messages, HTTP protocol, and XML Namespaces for objects identification.

The Simple Object Access Protocol (SOAP) is a lightweight, XML-based protocol for exchanging information in a decentralized, distributed environment. SOAP-based requests and responses are combined with a transport protocol, such as HTTP Oracle (2001). The main advantage of SOAP is protocol, language, and platform independence Oracle (2001). The Remote Procedure Call technique is used in the EEG Data Processor for third-party sharing of the analytic methods. The EEG Data Processor provides a following set of features.

- List of available methods This method returns the list of available signal processing method provided by the EEG Data Processor.

Figure 2. The form for registration of application

- List of parameters of chosen method It returns required parameters of selected method with data type and restrictions (possible values etc.).
- Running the selected method Result of selected analytic method is provided in XML format.
- Number of available threads It returns information about available processing threads. If there is no available thread, a request is put into a queue.

3.2 Registration

When users want to use the EEG Data Processor, they also have to register first by filling the simple registration form (including full name of user, his/her email and password). The successful registration grants access to the EEG Data Processor and its analytic methods to users.

In case of any uncommon or violent behaviour of a registered user, administrators of the Processor are able to block the user.

As we mentioned before, we implemented the third-party sharing mechanism allowing using presented methods in other applications. A registered user is able to register his/her application via registration form (Figure 2). The fields name of application and password will be used as credentials for the Web Services client. The site URL field is used for checking a newly registered application by the EEG Data Processor administrators. There are defined several statuses for applications.

- Pending It is a status for newly registered application before review. This application is not able to use the analytic methods.
- Rejected The registered application is rejected by a reviewer.

Figure 3. Administration of applications

- Confirmed The registered application is accepted by a reviewer and is able to use the analytic methods.
- Blocked administrators of the EEG Data Processor are able to block an application when any uncommon or violent behaviour is discovered.

Administrators are able to change a status of an application via administration of applications interface (Figure 3). Confirmed applications are able to integrate the analytic methods by implementation of a Web Services client. The EEG Data Processor serves as a Web Services server.

3.3 Integration with the EEG/ERP Portal

This section describes integration of the analytic methods provided by the EEG Data Processor into the EEG/ERP Portal. It mainly provides the technical information about used technology and the implementation of Web Services in the EEG Data Processor and the WS client in the EEG/ERP Portal. The EEG/ERP Portal is already registered in the Processor and confirmed.

The EEG/ERP Portal has been developed by our research group Jezek P. (2010). This portal enables research groups to store, manage and download their experimental data and metadata. The portal is developed as a standalone product running on servers in our department. The usage of the Portal does not require any special software installation, only a web browser.

Apache CXF CXF (2005) is an open source services framework that makes web service development easy, simplified, and standard based. This technology is used for client-server based integration of web applications. The client application will prepare an order and send it to the EEG Data Processor through a business method call. For implementation we use code-first approach. It includes three following steps:

- Create a Service Endpoint Interface (SEI) and define a business method to be used with the web service.
- Create the implementation class and annotate it as a web service.
- Create an xml configuration of the service class and an instance of the class using JAX-WS frontend.

We have created the Service Endpoint Interface with defined methods allowing accessing and running analytic methods, which the EEG Data Processor provides. This interface is given below.

```
@WebService
@Secured("ROLE_APP")
public interface ProcessService{
/**
* Getter of available algorithm names.
* @return algorithm names
*/
public String[] getAvailableMethods();
/**
* Returns number of currently available processing units.
* @return available processing units
*/
public int availableProcessingUnits();
/**
* Getter of parameters necessary for method to run.
* @param fileFormat supported file format
* @param methodName name of desired process method
* @return array of parameters of chosen method
*/
public MethodParameters[] getMethodParameters(SupportedFormat fileFormat,
String methodName);
/**
* Returns byte array of processed data
* (will be replaced by output format in time)
* @param data files to be processed
* @param fileFormat one of supported file formats
* @param algorithmName name of processing algorithm
* @param params other parameters
* @return bytes of processed data
*/
public byte[] processData(DataFile[] data, SupportedFormat fileFormat,
String methodName, String[] params);
}
```

For integration of the service into a client application (e.g. the EEG/ERP Portal), it is necessary to configure the Web Services client. The client configuration in the EEG/ERP Portal using the Apache CXF is given below. Id specifies a unique identifier for the client instance. Address specifies the URL address where the endpoint of the service is published. ServiceClass specifies the interface of the service in the

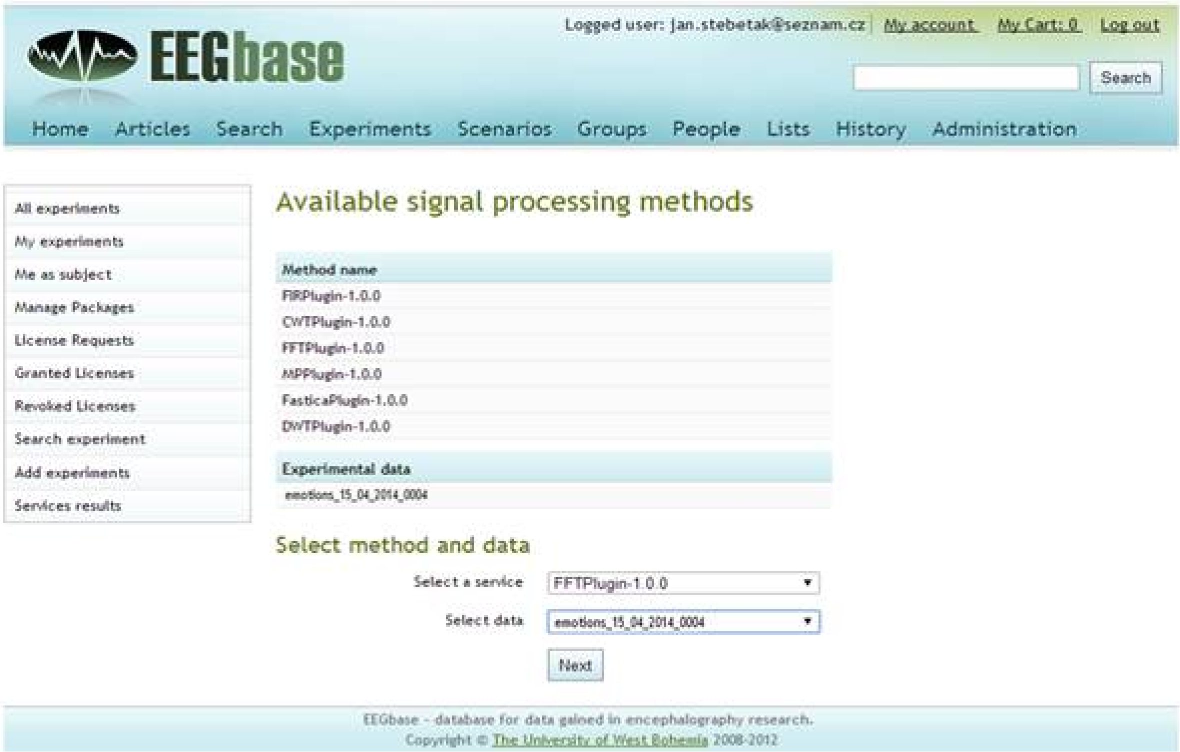

Figure 4. Integration of methods into the EEG/ERP Portal

client application. In the EEG/ERP Portal, we used the Apache CXF for generating necessary files for the client implementation from the description file .wsdl.

```
<jaxws:client id="eegService" address="http://147.228.63.134:8080/
eegdataprocessor/webservice/ webservice/processService"
serviceClass="cz.zcu.kiv.eegdatabase.webservices.EDPClient.ProcessService"
username="EEGBase"
password="***********" />
```

When the client is configured, the interface shown above is available to use in the client application. In the EEG/ERP Portal, we use it as follows: The getAvailableMethods gets all available methods from the EEG Data Processor. Then a user selects method and data. The required parameters of the selected method are obtained by getMethodParameters method. When all parameters are filled, the method processData starts data processing.

For users of the EEG/ERP Portal, we have prepared a web interface, where he/she select an experiment including data and chooses services option. The EEG/ERP Portal displays available methods and data (Figure 4). The next web page includes form where a user fills parameters of selected method. The Run button starts the processing. The EEG/ERP Portal waits for processing result on background, so a user can continue to work. When finished, the processing result is stored as XML file into the database for further using. A user can obtain results using Services Results button in Experiment section (Figure 5).

Figure 5. Services Results section

4 DISCUSSION

This paper summarizes basic approaches of sharing analytic methods. The common approach is to publish methods via a portal or a web interface. Users upload data on the server and are able to use a method. We extended this approach by allowing third-party sharing of our methods.

We implemented our methods as web services via Web Services technology. Then we published the web services description file (.wsdl). Once a user registers his/her application in the EEG Data Processor, the methods can be integrated into this application. The methods are still accessible via the web interface for registered users.

Since the wsdl description file and also communication between client and server are based on XML language, the Web services technology brings the language independency. Methods and also the EEG Data Processor are implemented in the Java language; however, client applications can be implemented in different programming language.

The main disadvantage of this approach is the necessity to modify the client application. Developers have to create the WS client and put it into the application. However, there are possibilities to generate the WS client automatically from the wsdl file. The widely used tools are wsdl2Java CXF (2001), Web Services Description Language Tool for Java or C#, or SOAPy for Python. When a client is implemented, it automatically reacts on changing (adding, editing, or removing) methods in the EEG Data Processor.

We successfully integrated the methods from the EEG Data Processor into the EEG/ERP Portal. This integration allows us using the methods directly in the Portal. This integration is described in Section 3.3.

Currently, the EEG Data Processor allows addition of methods developed in Java language. Therefore, our future work includes the development of advanced plug-in engine for automatic addition of analytic methods in different programming languages (Java, Python, Matlab).

ACKNOWLEDGMENTS

Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text. Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text.

Funding: Text Text Text Text Text Text Text Text.

REFERENCES

Blankenberg, D., Kuster, G. V., Coraor, N., Ananda, G., Lazarus, R., Mangan, M., et al. (2010). Galaxy: A web-based genome analysis tool for experimentalists. *Current protocols in molecular biology* , 19–10

Ciniburk J., M. P.-R. T., Moucek R. (2010). Erp components detection using wavelet transform and matching pursuit algorithm. In *DCII* (Prague)

CXF (2001). Wsdl2java

CXF (2005). Apache cxf

DATAQ (2011). Waveform analysis using the fourier transform

Delorme A., M. S. (2004). Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis. In *PubMed*

Giardine, B., Riemer, C., Hardison, R. C., Burhans, R., Elnitski, L., Shah, P., et al. (2005). Galaxy: a platform for interactive large-scale genome analysis. *Genome research* 15, 1451–1455

GNode (2009). German neuroinformatics node

Goecks, J., Nekrutenko, A., Taylor, J., and Team, T. G. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol* 11, R86

Herz AVM, N. M.-S. W. Z. T., Meier R (2008). G-node: An integrated tool-sharing platform to support cellular and systems neurophysiology in the age of global neuroinformatics. In *Neural Networks*. 1070–1075

Hyvarinen A., K. J. and E., O. (2001). Independent component analysis. In *Adaptive and Learning Systems for Signal Processing* (Prague)

INCF (2001). International neuroinformatics coordinating facility

Jezek P., M. R. (2010). System for storage and management of eeg/erp experiments generation on ontology. In *Databases and Information System Integration* (Madeira), vol. 1

Jezek P., M. R. (2013). Electroencephalography data processor. In *HEALTHINF 2013 - International Conference on Health Informatics* (Barcelona), 357–361

Jie Liu, J.-f. X., Er-peng Zhang and yong Lv, Z. (2006). Deployment of web services for enterprise application integration. In *(EAI) System* (Berlin)

MDP (2012). Mdp documentation

Neuroshare (2001). Neuroshare

Oracle (2001). Soap web services

Sobolev A., L. A. P. R. P. L. K. C. J. G. C. W. T., Stoewer A. (2014). Integrated platform and api for electrophysiological data. *Frontiers in neuroinformatics* 8

Surname, B. (2002). The title of the work. In *The title of the book*, ed. E. Name (The city: The name of the publisher). 201–213

Teeters, e. a., Jeffrey L. (2008). Data sharing for computational neuroscience. *Neuroinformatics* 6, 47–55

Vareka, L. (2012). Matching pursuit for p300-based brain computer interfaces (Prague)

Watson P., P. G. G. F. A. J. F. M. L. B., Jackson T. and P., L. (2007). The carmen neuroscience server. In *Proceedings of the UK e-Science All hands Meeting*. 135–141

Zito T., W. L., Wilbert N. and P., B. (2008). Modular toolkit for data processing (mdp): A python data processing framework. *Frontiers in neuroinformatics* 2