Neural network based pattern matching and spike detection tools and services — in the CARMEN neuroinformatics project

Martyn Fletcher, Bojian Lianga, Leslie Smith, Alastair Knowles, Tom Jackson, Mark Jessop, Jim Austin

Neural Networks, Volume 21, Issue 8, October 2008, Pages 1076-1084

DOI link: http://dx.doi.org/10.1016/j.neunet.2008.06.009

Neural network based pattern matching and spike detection tools and services — in the CARMEN neuroinformatics project

Martyn Fletcher[a],[*] , Bojian Liang[a], Leslie Smith[b], Alastair Knowles[c], Tom Jackson[a], Mark Jessop[a], Jim Austin[a]

[a] Advanced Computer Architectures Group, Department of Computer Science, University of York, Heslington, York, YO10 5DD, UK [b] University of Stirling, Department of Computing Science and Mathematics, Stirling FK9 4LA, UK [c] School of Computing Science, Newcastle University, Newcastle upon Tyne, NE1 7RU, UK

Abstract

In the study of information flow in the nervous system, component processes can be investigated using a range of electrophysiological and imaging techniques. Although data is difficult and expensive to produce, it is rarely shared and collaboratively exploited. The Code Analysis, Repository and Modelling for eNeuroscience (CARMEN) project addresses this challenge through the provision of a virtual neuroscience laboratory: an infrastructure for sharing data, tools and services. Central to the CARMEN concept are federated CARMEN nodes, which provide: data and metadata storage, new, thirdparty and legacy services, and tools. In this paper, we describe the CARMEN project as well as the node infrastructure and an associated thick client tool for pattern visualisation and searching, the Signal Data Explorer (SDE). We also discuss new spike detection methods, which are central to the services provided by CARMEN. The SDE is a client application which can be used to explore data in the CARMEN repository, providing data visualization, signal processing and a pattern matching capability. It performs extremely fast pattern matching and can be used to search for complex conditions composed of many different patterns across the large datasets that are typical in neuroinformatics. Searches can also be constrained by specifying text based metadata filters. Spike detection services which use wavelet and morphology techniques are discussed, and have been shown to outperform traditional thresholding and template based systems. A number of different spike detection and sorting techniques will be deployed as services within the CARMEN infrastructure, to allow users to benchmark their performance against a wide range of reference datasets.

1. Introduction

An important challenge in the study of the nervous system is to understand the way in which neuronal signals are encoded, archived and decoded (Watson et al., 2007). This is not only a fundamental research question but has major application to industry, including: computer science, nanotechnology and neuromorphic systems, electronic engineering and central nervous system drug discovery. Coding is thought to occur in different ways at different levels of abstraction. These are characterised through application of various electrical, imaging and modelling techniques, addressing different levels of granularity.

Data, which are difficult and expensive to produce, yet sporadically shared, are typically voluminous and locally curated,

[*] Corresponding author. Tel.: +44 (0) 1904 567710.
Email address: martyn.fletcher@cs.york.ac.uk (M. Fletcher).

and exist in heterogeneous formats. They are therefore difficult to integrate and often not amenable to computation. Consequently, analysts are often deprived of data, and it is not common for methods to be verified on data from multiple sources. As a result, there is:

(a) A shortfall in generic analysis methods.
(b) An absence of readily accessible reference data.
(c) Limited organised curation or data optimisation.
(d) Limited cooperation between disparate research groups with complementary expertise.

This paper describes ongoing work into developing an infrastructure that will address these challenges. The CARMEN (Code Analysis, Repository and Modelling for eNeuroscience) Project is developing a virtual neuroscience laboratory: a platform for sharing data, tools and services. The paper discusses the evolution of the system and reports on the current state of development. An overview of the project is provided, followed by an outline of the integration of data, tools and services. The Signal Data Explorer (SDE) and underlying pattern match algorithms are described. Pattern matching will be used, for example, for detecting shapes which indicate a particular condition in an electrophysiology recording. Preliminary services for spike detection are then presented, followed by future work and conclusions.

2. Background

CARMEN is a $10M, 4 year UK eScience Pilot Project which began in October 2006 and is progressing through requirements, design and prototyping stages. Release in stable form is planned for October 2008. The aim is to provide a webbased computing infrastructure (Foster & Kesselman, 2004) to enable integration of data, software and knowledge from distributed neuroscientists. These innovations embody a virtual neuroscience laboratory, linking experimental and analytical neuroscientists in a translational pipeline which challenges contemporary neuroscience; offering potential for rapid and expedient advancement. Complementary to this, the publication and sharing of data is increasingly mandated by research funding organisations (Medical research council, n.d.; NIH data sharing policy, 2003).

The primary advantage of CARMEN is to reduce the requirement for expensive and often ethically contentious experimentation, by allowing maximum benefit to be derived from experimental data and analysis methods.

The initial requirements for the CARMEN system have been collected through extensive, iterative discussion with the ten neuroscience research groups on the project. These requirements can be summarised as follows:

- Allow ‘‘experimenter’’ users to describe, store and analyse data (time and image series) from various electrophysiology acquisition systems — in various proprietary and bespoke formats (the maximum data rate discovered so far is 1 Tbyte per week).
- Allow ‘‘analyst’’ users to describe, store and browse code (source and executable) for data analysis.
- Allow execution of code in MATLAB, R, Java, Python and C/C++, including highly parallel processes.
- Allow ‘‘simulation’’ users to describe, store and analyse data generated by simulation tools, including the NEURON (n.d.) and GENESIS (n.d.) simulators.
- Allow data derived from the results of analyses to be stored and bound to source data for further analysis.
- Allow users to specify access control rights to their data or analysis services.
- Allow both thin and thick client tools – including legacy tools – to access the

repository securely for data analysis and visualisation.

- Enable data translation between different proprietary and bespoke data formats. Retaining source data at all stages in the translation.
- Support a user community that is distributed and growing, with varying data preservation requirements.

3. Relation to existing infrastructures

The desire to share, preserve and make efficient use of scientific data is generic. Prior to the commoditisation of networked computing, peerreviewed articles provided the optimal media for distribution of results of research. Confronted simultaneously by escalating data rates, and rapidly increasing capacity to archive and transport data, the focus in many domains has diversified towards publishing experimental data, in some cases prior to peerreview. A range of initiatives in these domains, including the life sciences, therefore precedes CARMEN. CARMEN, where possible, utilises their technologies:

- The Storage Request Broker (SRB) (SRB, n.d.) was developed by the San Diego Supercomputer Center (SDSC). The Biomedical Imaging Research Network (BIRN) project (BIRN, n.d.) uses SRB to build datagrids, to integrate distributed image repositories. The DAME (Distributed Aircraft Maintenance Environment) (Austin et al., 2004) and BROADEN (Business Resource Optimisation for Aftermarket & Design on Engineering Networks) (Fletcher et al., 2006) projects used SRB to manage data from aeroengine sensors. CARMEN uses the SRB for storage of raw and derived data from neurophysiology experiments.
- The XACML security markup (XACML, n.d.) was used by the GOLD (n.d.) project at Newcastle University to allow intellectual property on new chemical entities to be shared securely by biotech companies. CARMEN utilises XACML tooling developed by the GOLD project for provision of federated security.
- The Taverna Workbench and Freefluo Enactment Engine (TAVERNA, n.d.) were developed by the myGrid project (myGrid, n.d.) to facilitate interchange of software services developed for microarray analysis. CARMEN will make use of Taverna and Freefluo for construction and orchestration of data analysis workflows.
- FuGE (n.d.) was developed by the functional genomics community to provide a generic data annotation model for experiments. FuGE will be extended by CARMEN to describe neurophysiology experiments.
- The SDE(n.d.)and distributed search technology was developed during the DAME and BROADEN projects. CARMEN will use this technology for data visualisation and searching.

To illustrate this point, we compare CARMEN to a preceding initiative in the High Energy Physics (HEP) domain. The ROOT (n.d.) system is an object oriented framework whose purpose is to manage and analyse large amounts of data from the Large Hadron Collider (LHC). The LHC generates data for both simulation and analysis which is (by orders of magnitude) larger than anything seen before; approximately 1 Tbyte per experimental run. Currently the core function of ROOT is restricted to data processing in C++ and Python, which differs from CARMEN where there is a requirement to support a much broader range of scripting languages, which will be met by embedding code within generic data messaging interfaces (Java, SOAP). It is intended that ROOT will be extended to cover: data acquisition, event reconstruction, detector simulation, and event generators. There is also is an extension of ROOT allowing transparent analysis of large sets of ROOT files in parallel, on clusters of computers or multicore machines, known as the Parallel ROOT Facility, PROOF (n.d.). It is intended that CARMEN will emulate and extend this functionality through use of Dynasoar (n.d.), which supports dynamic service deployment over loosely coupled compute grids. The CARMEN and ROOT projects are similar in that both intend to store data, and perform remote data analysis and simulation. However, their data formats, and analysis and simulation requirements are potentially very different. It is also not clear that ROOT intends to support sharing, social networking and client access, which are requisite features for CARMEN.

4. The CARMEN architecture

The architecture consists of federated CARMEN Active Information Repository Nodes (CAIRNs), which store, process and expose data (Fig. 1). A portal presents resources to clients in a conceptually centralised manner.

It is envisaged that CAIRNs will hold both raw time series data from electrophysiology recordings (e.g. Multi Electrode Array — MEA) and optical image data. Services will be provided to convert data to and from a translation format, providing a uniform data interface for analysis services. Presently, flat file data, including raw experimental recordings, are held in the Storage Request Broker. Metadata is held in a database, providing native search and indexing functions.

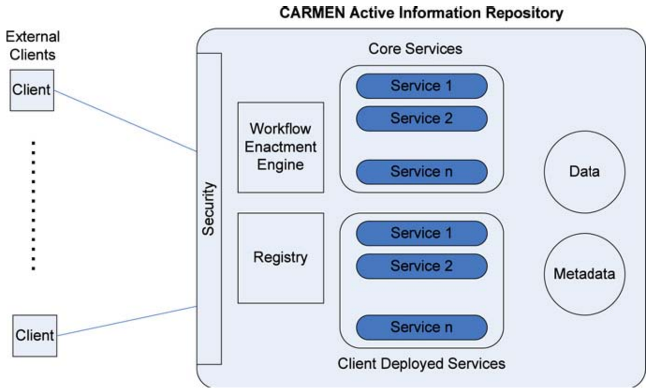


Fig. 1. CARMEN Active Information Repository Node (CAIRN).

The CAIRNs support the notion that both data and code may be managed. Furthermore, each CAIRN provides an execution environment comprising of: (a) Dynasoar, which allows services to be dynamically deployed over compute grids by way of virtual endpoints, and; (b) the Freefluo enactment engine, which allows workflows combining data and analysis services (specified in XML) to be deployed and reused. Together, these allow data transport to be optimised for fast, efficient processing.

Analysis functions are presented in the form of Web services (n.d.), which will be deployed by users supported by the CARMEN development team. It is anticipated that a core set of methods will be provided with the first stable release. Those currently in development range from spike detection and sorting, to information theory, correlation and causality measures, including Bayesian modelfitting.

Provenance functions are planned to allow processing threads to be transcribed and queried over time, binding raw data to derived data from analysis.

The security infrastructure allows users to grant other users with access to data and services. More sophisticated functions, including policy conflict resolution (e.g. where funder policies must be reconciled with policies specified by the recipient user) are earmarked for future research.

Section 5 presents further exploration on data, services workflows and tools. Sections 6 and 7 explore the Signal Data Explorer (SDE) tool and the pattern matching techniques used in more detail. Section 8 covers server based spike services, and Sections 9 and 10 describe future work and conclusions.

5. Data, services, workflows and tools

We have identified that typical users of the CARMEN infrastructure will perform one or more of the following:

- Collect raw data (e.g. time series or image series) and store this in the CARMEN infrastructure.
- Perform data translation and analysis using the tools and services (e.g. algorithms) available.
- Develop new algorithms and workflows.
- Produce synthetic data using models.
- Develop new models to generate synthetic data.
- Define model parameters by analysing real data.

Data, tools and services, and their integration, are considered in more detail in the following sub sections.

5.1. Data

Neurophysiology data are typically very large (>1 Tbyte) and growing due to advancements in data capture. Transport of intermediate data (e.g. data generated during analysis) is therefore undesirable. CARMEN addresses this in two ways. First, the federated data storage and processing architecture allows large datasets residing on a local CAIRN to be accessed and analysed remotely. Second, where network upload is unrealistic, it is planned that offline submission will be supported. Large data may be physically shipped to their eventual server location.

A far greater challenge is the heterogeneity of the data; there is currently no standardized data format for neurophysiology. To counter this, an extensible, translation data format is being specified. This allows the format of binary array data, e.g. multichannel time series, to be arbitrarily specified by way of XML header documents, providing a uniform data interface for analysis services. Neuroshare (n.d.) was first investigated but was found to be suboptimal, as: (a) it does not write data out to a static file representation that can be passed by services in a standardised manner; (b) it inflates the data volume, presenting optimisation problems for operations such as streaming, or buffering for interactive visualisation. However, it is planned that the NeurosShare DLLs (Dynamic Linked Libraries) will be employed in the process of parsing data into the translation format.

Schemata and vocabularies are required to structure collaborative systems such as CARMEN.BrainML model repository (n.d.) was evaluated but was found to be unsuitable, due to the lack of available tooling and APIs to other models. Extensions to the FuGE data model are therefore being developed to describe neurophysiology experiments (Gibson et al., 2008).

Data entities are uniquely identified and associated with other data entities by Uniform Resource Identifiers (URIs). To support federation, it is anticipated that CARMEN will eventually utilise location independent identifiers, such as LSIDs (Life Science Identifiers) (LSID, n.d.).

5.2. Services and workflows

Existing and new algorithms will be implemented as web services to allow remote deployment. Due to the range of scripting languages employed by neuroscientists, some of which (MATLAB. R, Python) are interpreted and therefore require parsing engines to generate native instruction sets, there is no generic, scalable way of deploying analysis codes as services. The preferred approach is to embed scripts in Java to provide a uniform messaging interface (SOAP — Simple Object Access Protocol). Documentation is being produced to explain the embedding processes to users. It is hoped that in the later stages of the project it will be possible to codify these processes, in order to provide a drag and drop wizard for deployment of analysis services.

The use of MATLAB (n.d.) for coding analysis methods is prevalent in neuroscience. This raises challenges, both in terms of licensing, and distribution. As MATLAB is a widely used system for the analysis of data, CARMEN aims to support it to ensure rapid uptake. The approach taken has been to use the MATLAB compiler technology. However, this presents specific problems, in that: (a) the compilers are not free of charge, and cannot be shared by multiple licensees or deployed as third-party services; (b) some toolbox functions cannot be compiled. Further research is required to identify whether the latter represents a technical or commercial constraint.

Open source initiatives offer an alternative which may simultaneously encourage software vendors to consider more flexible licensing avenues. Discussions have taken place with the FIND Toolkit Project (FIND, n.d.) and may seed a collaborative effort along these lines. Given the current move towards service oriented computing, such an effort should not favour a particular scripting implementation, but should exploit web service standards (Web services, n.d.) to provide flexibility and crossplatform interoperability.

The CARMEN project will generate web services for neuroscience data analysis ranging from spike detection and sorting, to correlation and causality analysis including statistical modelfitting. The preliminary development of services for spike detection is described later. Users will be able to mobilise these services and associated datasets as modular workflow components in the Taverna system.

5.3. Interactive tools

Interactive tools will also be provided for visualisation and iterative searching. Due to complex user interaction with data it may not be possible to implement these tools within web browsers. However, other web services may be consumed to perform filtering and presentation; for example, a spike detection service may be mobilised to constrain the datasets over which an SDE pattern search operation may be made. CARMEN provides support for both native tools, such as the SDE, which are being developed within the project, and thirdparty tools, which (it is envisaged) will integrate with a programmatic interface (an API — currently in development).

5.4. Integration

The integration problem is characterised by:

- A variety of data formats produced by different acquisition systems. There is a need for data translation, and in the long term, standardisation.
- A variety of third party and legacy services and tools, some of which are bound to particular acquisition systems using particular data formats.
- New services which use translated or standardised data formats, or which can use any data format.
- New tools which use translated or standardised data formats, or which can use any data format.

As described in this paper, CARMEN can go some way towards mitigating these problems. However, the underlying challenges are socioeconomic. There is a requirement for the user community to work together to create incentives for manufacturers and other commercial beneficiaries to solve the problem. CARMEN, with other data sharing projects in the neuroscience domain, aims to precipitate this change by providing scalable platform technologies for widespread sharing of data and services.

5.5. The CARMEN portal

The CARMEN Portal is in development, and provides a conceptually centralised interface to federated resources in the CAIRNs. The portal (CARMEN portal, n.d.) currently supports user account creation, data upload, metadata ascription and control of security policies — in addition to basic searching. The SDE can also be used to download and search for patterns in specific datasets. These functions are undergoing refinement and enrichment in partnership with neuroscience users. The ability to upload services, run workflows, and scrutinise provenance is planned.

6. Signal Data Explorer (SDE) tool overview

The SDE tool supports viewing and search of the multichannel data in the CARMEN system. The tool, which was developed within a number of eScience projects, supports pattern matching over

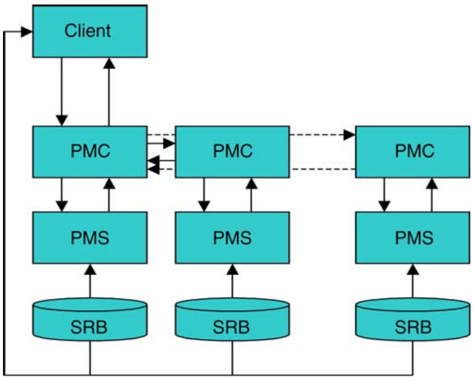


Fig. 2. SDE (Client) and Pattern Match Architecture, where PMC is Pattern Match Controller, PMS is Pattern Match Service, and SRB is the Storage Request Broker.

federated signal (time series) data repositories. The search techniques are based on research methods derived from Correlation Matrix Memories (Austin, 1995; Austin, Kennedy, & Lees, 1995).

The SDE is used as a client application that accesses data both on the client, or federated within the CARMEN system or other gridenabled data repositories. The SDE provides data visualisation, feature analysis and realtime search capabilities on complex experimental data.

6.1. The SDE as a client–remote data access and Pattern Match Services

The SDE is designed to interact with remote services and other software tools. Uniform Resource Locators (URLs, e.g. HTTP) or software commands can be used to access remote data repositories. Data can also be downloaded directly and loaded into the SDE on the client machine.

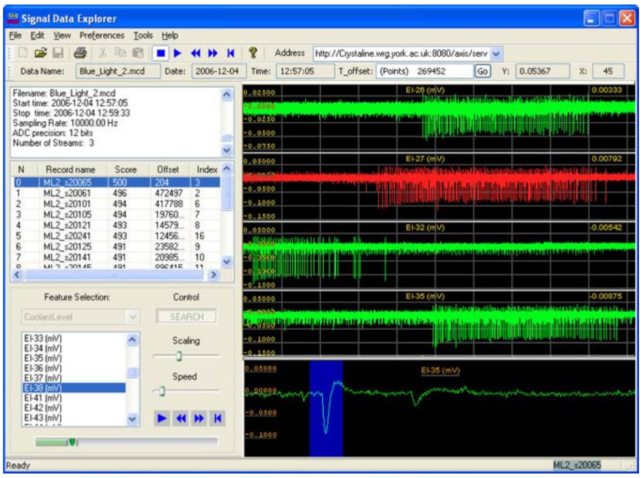
The SDE provides a plugin interface that integrates external search algorithms with the SDE environment. Users can establish their own search services by implementing new plugins. When using a remote search service, the plugin acts like a client to the remote service. The current default deployment of SDE provides a web service client plugin for accessing the Pattern Match Controller (PMC) based distributed Pattern Match Services (PMS).

The pattern match architecture (Fig. 2) consists of Pattern Match Controller (PMC) and Pattern Match Services (PMS). The Pattern Match Services access data in a distributed file system (implemented using the Storage Request Broker technology) and the Pattern Match Controller interfaces to the client for data communication relevant to the search task. A client can initiate a search task by contacting any of the PMC nodes over a distributed computer network. The PMC node contacted will automatically take control of the other relevant PMC notes for that particular search task and return the results to the client.

6.2. Visualisation and data processing

The SDE supports simultaneous and highly interactive viewing of multichannel time series data. This is a crucial feature for offline analysis of the complex experimental data in the CARMEN system, and also has potential applications in real-time computational steering.

An example of the SDE opening a MultiChannel Systems data file (mcd file from a MultiChannel Systems (MCS) acquisition system (Multi channel systems MCS GmbH, n.d.)) is shown inFig. 3. Using the SDE, a user can explore and view any portion of the data rapidly. The output of the data processing tools can be viewed immediately and compared to the raw data by displaying them in the same window.



Signal Data Explorer

Data Name: Blue_Light_2.mcd Date: 2006-12-04 Time: 12:57:05 T_offset: (Points) 269452 Y: 0.05367 X: 45

Address http://Cystaline.wrg.york.ac.uk:8080/axis/serv

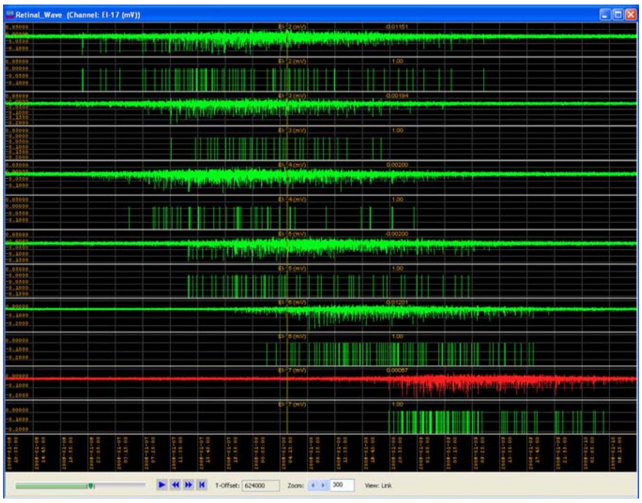
Filename: Blue_Light_2.mcd
Start time: 2006-12-04 12:57:05
Stop time: 2006-12-04 12:59:33
Sampling Rate: 10000.00 Hz
ADC precision: 12 bits
Number of Streams: 3

N	Record name	Score	Offset	Index
0	ML2_s20065	500	204	3
1	ML2_s20061	496	472497	2
2	ML2_s20101	494	417788	6
3	ML2_s20105	494	19760...	7
4	ML2_s20121	493	14579...	8
5	ML2_s20241	493	12456...	16
6	ML2_s20125	491	23582...	9
7	ML2_s20141	491	20985...	10

Feature Selection: CoolantLevel

Ready ML2_s20065

Fig. 3. Signal Data Explorer opening an mcd file.



Retinal_Wave (Channel: El-17 (mV))

T-Offset: 624000 Zoom: 300 View: Link

Fig. 4. Spike time data displayed in an auxiliary window.

A windowing capability permits auxiliary data views to be opened which permit the user to interactively zoom into or out of data, providing macro and micro views, as well as allowing the user to ''play and zoom'' very large datasets. All views of the same record are synchronized in time. If views have different zoom factors, they are synchronized using a time instance mark. Fig. 4 shows spike detector output together with the raw data in an auxiliary window. In the auxiliary window, a user can zoom in/out of the view and change the scaling factor of each subview separately.

The current version of the CARMEN SDE provides the capability to apply a range of data preprocessing tools and various viewing modes to complement the core search and visualisation functions. Currently, the SDE preprocessing toolbox includes a variety of filters, amplitude limiters, a firing rate converter, envelope detector and a template based spike detector. The SDE toolbox can be easily extended by adding new tools. In future, the toolbox will inherit the analysis services provided by the CARMEN system. The SDE may also be expanded to accommodate third party tools, including those developed using MATLAB.

6.3. The integrated environment for pattern matching

Pattern search and matching technology was developed by the University of York and Cybula Ltd. It provides the capability to search for patterns in temporal data signals across distributed repositories. The SDE can compare events in timeseries data based on existing events, those stored in a file, or new events sketched by the user. It has the ability to match across multiple timeseries as well as allowing the user to tag the data with Metadata to identify known events. A pattern template library allows a user to create, manage, edit and reuse interesting patterns.

The SDE provides an interactive and intuitive search capability, such that features of interest can be located in federated and local datasets. The search process is feature driven, in that the user can highlight a region of interest in a timeseries signal or select an instance from the pattern template library and request a pattern matching process to be carried out against the target datasets (see Fig. 3 the query pattern is highlighted in blue in the lower right window). Similarity measures are used to provide a ranking system that can score results for the search process. The search process has already been proven within the context of the DAME system and has been shown to be scalable to terabyte datasets. The SDE can search local and distributed data stores, and interfaces directly onto the datasets that are being held on the CAIRNs. It supports arbitrary variable length, and provides filter preprocessing and data segment selection, conversion and export.

7. Pattern matching

Pattern matching is one of the primary functions of the SDE. The native search engine provided within the SDE is based on a high performance binary neural network called a Correlation Matrix Memory (CMM). The pattern matching functions allow a user to search for particular patterns within or across variables in datasets. The SDE generates a search index based on binary vectors (explained below), similar to a conventional text search engine. The index is created on the fly and can include references onto remote data repositories, including relational databases. This allows remote functionality (e.g. text based querying in SQL) to be exploited to preconstrain pattern search requests.

An example application of the SDE is template based spike detection. By applying the SDE search engine, spikes within raw time series data can be detected quickly and (if the noise level is low) accurately. The search engine is currently being extended to allow fast, reliable spike train pattern matching at higher noise thresholds.

In addition to the local search engine, a gridenabled, distributed search service cluster is marshalled by the PMC in the CARMEN system. This provides the capability for the CARMEN pattern matching system to efficiently manage and search large volumes of distributed data federated over a Grid (Foster & Kesselman, 2004). In this case, the SDE acts as a (thick) client to the CARMEN Grid services.

7.1. AURA and CMM

AURA (Advanced Uncertain Reasoning Architecture) is a set of generalpurpose methods for searching large unstructured datasets (Austin, 1995; Austin et al., 1995) and is used in SDE for pattern searching. AURA, which is based on CMMs, can perform extremely fast parallel pattern matching on distributed data.

The CMM is a type of binary associative neural network. A CMM with input width *n* and output width *m* can be represented as a *n x m* binary matrix **M.** For a given input binary vector $\mathbf{I}_k$ and associated binary output vector $\mathbf{O}_k$, the kth training update of a CMM is defined as:

$$\mathbf{M}_k = \mathbf{M}_{k-1}\ \mathbf{U}\ \mathbf{I}_k^{T}\ \mathbf{O}_k$$

where $\mathbf{M}_k$ and $\mathbf{M}_{k-1}$ are the CMM after and before the training ($\mathbf{M}_0\mathbf{0}$)., and **U** denotes a logical OR operation. The recall vector $\mathbf{S}_i$ associated to the input $\mathbf{I}_i$ is defined as:

$$\mathbf{S}_i = \mathbf{I}_i\mathbf{M}$$

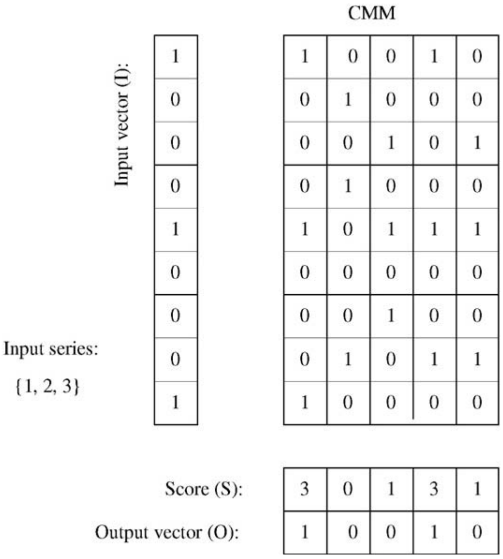


Fig. 5. The CMM recall technique.

Recall vector $\mathbf{S}_i$ is, in general, an integer vector and the integer value of each element of the recall vector is called the ‘‘score’’ of the CMM matching on the relevant column vector. The recall vector can then be thresholded to a binary output vector, as shown in Fig. 5.

A detailed discussion of AURA and CMM is out of the scope of this paper, more detail can be found in other papers (Austin, 1995; Austin et al., 1995; Furber et al., 2007).

7.2. Preprocessing and postprocessing

Many of the fast pattern matching algorithms for large datasets use a similar approach in which a small set of data instances are obtained using a fast, approximate searching algorithm, then a conventional approach is applied to the candidate instances to obtain the final results speedily. In this context, the final results are constructed from the data candidates obtained from the approximate match via ‘‘postprocessing’’ or ‘‘fine tuning’’. Postprocessing is any procedure that uses the approximate search results as input, and outputs the final results.

Usually, preprocessing is applied to the raw data before the approximate search stage. The main purpose of preprocessing is noise attenuation, feature enhancement, data mapping and data encoding.

The SDE system provides a set of configurable tools for preprocessing and postprocessing. These tools can be used to convert the raw data to a required format, to extract a particular component from the raw data or to enhance the required features from the data to improve the performance of the pattern searching. For example, local field potential (LFP) can be obtained by applying a lowpass filter to the raw data. A particular shape selected from the LFP data series can then be used as a search query in a search of the dataset using the SDE search engine. Filters and envelope detection can be applied to the data sequentially to obtain a suitable signal for the subsequent processing.

7.3. The k-Nearest Neighbour (k-NN) method

The current system uses the k-Nearest Neighbour (k-NN) pattern matching method, implemented using AURA and CMMs for pattern matching. K-NN is a simple algorithm that is widely used in data clustering, classification and prediction. Based on a specific distance metric or similarity measure, k-NN searches for the k instances of data (from all available data examples) that are ‘‘nearest’’ to the point representing the query.

Zhou, Austin and Kennedy proposed an approach using the AURA technology for fast k-NN searching (Zhou, Austin, & Kennedy, 1999). This approach applies a CMM to quickly produce a small number of instances of candidate data and then applies the conventional k-NN approach to the candidate data to obtain the required results. Liang and Austin further improved the approach in several ways (Liang & Austin, 2003). First, a particular kernel is applied to the query vector in order to obtain the specific distance measure for the CMM output that is consistent with the distance metric applied on the following postprocessing (e.g. Euclidean or Cityblock distance measure). Secondly, split kernels are used to separate the searching space into a number of subspaces and k-NN searching is applied to each subspace until the least of k instances are obtained. Third, an ‘‘asymmetry kernel’’ is suggested to cope with quantisation errors introduced by the procedure of encoding the real data value into binary vectors. The improved AURA based k-NN approach provides consistent distance measure both on the CMM matching stage and the following finetuning stage and ensures that the correct k-NN candidates are included in the search results. The split kernel makes it possible to control the searching space rather than either searching the whole dataset or missing some of the subspaces that may contain the k-NN candidates.

7.4. Multi criteria searching

In many applications, events or conditions are defined by more than one variable over time and also across data channels. Multicriteria searching makes it possible to characterize a condition or event by using multiple template patterns, parameters and measures to search for patterns. Searching for multiple patterns from timeseries data is a more complicated problem than searching for a single pattern. This is because it is not just a simple combination of separate single pattern searching procedures but also requires efficient collaboration of all searching procedures, intelligent management of the searching constraints applied and a clear, meaningful output that reasonably interprets the search results.

To search efficiently for multiple variables across channels, data from each channel of the same record in databases must be associated by use of an identifier such as a groupname or index. All data in the same record are synchronized by time. In this way, the search service can efficiently locate the data channels and compute the final matches by interpreting the parameters from the query.

A ''hit'' (or match) of a search is defined as the local maximum of the matches within a given tolerance threshold. The output of a similarity search may contain all the possible hits or may only be the best match on a dataset depending on the application. An event window Tw defines a maximum time interval that contains a set of valid hits across channels. Tw is an important searching constraint to define the valid hit set for multiple searching. Proper use of the event window constraint can also reduce the overall searching space. Each set of valid hits from a multiple searching procedure with n variables defines an ndimensional output vector. Conventional processes such as classification or indexing can then be applied to the output vector to obtain further information.

SDE provides an addon called 'The task planner' that allows a user to define and manage multicriteria searching tasks. Using the task planner, a user can define the query patterns from a pattern library or selected from the current dataset. Each query pattern defines a subtask for pattern matching. All the search parameters such as measures, event windows together with a set of filters/preprocessing and post-processing methods can be selected and attached to each subtask by using the task planner interface. A user can then dispatch the multi criteria search task and manage the search results using the task planner and the SDE.

7.5. Pattern matching on neuronal signals — application examples

To illustrate the use of SDE on the CARMEN system we consider the problem of searching for spikes and local field potentials in electrophysiology data.

The user can view the raw data and highlight a desired pattern
(i.e. a user selected template for a spike) and then apply the search to the whole dataset of one or more specified channels and produce a list of spikes based on the shape of the selected pattern. Details of the spike time data and the raw data can be viewed and compared in the SDE auxiliary view window. The SDE template based spike detector allows a user to define the template shape from the action potential waveforms, the threshold of the similarity measure and the amplitude range (the upper and lower bounds of the signal amplitude). Fig. 4 shows raw data and detected spikes from each channel, alternately.

Another example use of the SDE for neuronal signal pattern matching is to pick up the highly correlated channels by matching a template pattern to the Local Field Potential (LFP) signals. Through the use of the task planner, a user can define the width of the event window to ensure that a pattern belongs to the same experimental event when the pattern template is taken from the current dataset or from the template library. If the LFP signals are not directly available, a lowpass filter can be applied to the raw data before the search. This is carried out by defining it in the preprocessing procedures using the task planner. Other processing can also be defined at this stage in the same way. Searching can be carried out within the current data record or a set of records from the data repository. After the search task is defined, a user can invoke the search task by clicking on the search button; the SDE will automatically carry out the task and return the results. A user can then explore in detail the dataset of the result records by clicking on the relevant results from the displayed list.

8. Preliminary service development — spike detection and sorting

As discussed, one of the major tasks applied to spike train data is the detection and analysis of spikes (e.g. characteristic events). Typical analysis protocols range from spike detection and sorting to techniques for detecting mutual information between spike trains, to techniques for investigating the semantic content of higher level spike sequences. Higher level analyses rely on the accuracy and reproducibility of lower level analyses: hence in the initial phases of the CARMEN project, we have been concentrating on ensuring accuracy and flexibility of spike detection and sorting.

In general it is difficult to evaluate the effectiveness of spike detection and sorting systems because of the lack of ‘‘ground truth’’ information. Currently the SDE tool uses a templatebased pattern detector, where the template is selected manually by the operator. This is one example of the many different types of spike detection systems in usage. Recent work has investigated the effectiveness of a range of spike detection systems using a biophysically realistic signal generation system (Smith & Mtetwa, 2007). This approach allows comparisons between different spike detection and spike sorting algorithms, although one can always criticise the particular biophysical model used for the generation of the data. In Smith and Mtetwa (2007) a simple threshold based spike detector was used, and KlustaKwik (KlustaKwik home page, n.d.) and Waveclus (Quian Quiroga, Nadasdy, & BenShaul, 2004) (spike sorting) techniques were compared. These spike sorting techniques extract a segment from the original signal around where a spike has been detected, and attempt to assign this spike to one of a (small) number of classes, each representing a spike from a different neuron. The segment should be rather longer than the feature to be classified: in this case, of the order of 3ms. At usual sampling rates (25 kHz is often used), 3ms of signal has a large number of dimensions; too many to permit effective clustering. The segment is therefore projected down to a lower dimensional space in which clustering is possible. The projection must summarise the critical aspects of the signal for spike detection purposes. Both Principal Component Analysis (PCA) and wavelet based techniques have been used for this (Lewicki, 1998). In this test, it was found that a wavelet based dimensionality reduction technique outperformed the more commonly used PCA

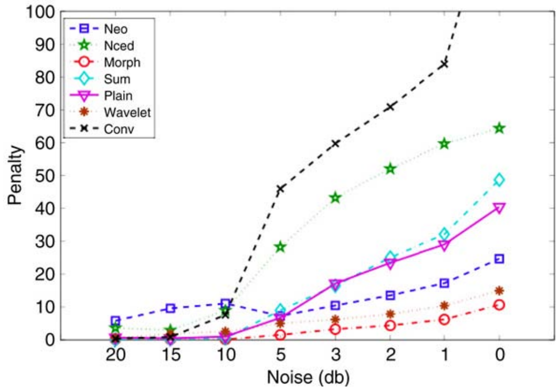


Fig. 6. Best performance achieved by a number of different spike detection techniques on synthetically generated data (over 50 runs each). The xaxis shows the peak:peak signal:noise ratio, and the yaxis shows the penalty calculated by adding the number of inserted spikes and the number of missed spikes. Techniques: Neo, Nced: energy based techniques, Morph: morphology based technique, Sum: averaging based technique, Plain: simple thresholding, Wavelet: wavelet based technique, Conv: template based. The techniques are described in detail in Mtetwa and Smith (2006) and Smith, Shahid, Vernier, and Mtetwa (2007).

based techniques.

Recently the same approach has been used to compare spike detection systems. This work (reported in Mtetwa and Smith (2006) and Shahid, Walker, and Smith (2008)) suggests that traditional thresholding and template based systems can be considerably outperformed by wavelet and morphology based systems. In this work, several different spike detection systems were applied to some synthetic datasets – generated with a variety of different levels of realistic noise, and tested over 50 different runs – see Fig. 6. Each technique has parameters which need to be tuned (for example, threshold level in simple thresholding) so that the technique works appropriately. Every technique was tested throughout the valid range of these parameters, and the best results chosen. In this way, each technique is allowed to perform optimally. Although one might expect that every technique would perform at about the same level (since each is allowed to perform optimally) this is clearly not the case. The wavelet and morphology techniques (described in detail in Mtetwa and Smith (2006)) outperform all the others. This shows that even discounting the parameter search problem, choosing the best spike detection technique is critically important, particularly when the SNR is below 10dB.

Within CARMEN, we intend to supply a number of spike detection and sorting techniques, allowing the user to compare the performance of their complete workflow with different services. These range from wellknown techniques (e.g. simple thresholding, wavelet based (Kim & Kim, 2003; Lewicki, 1998; Nenadic & Burdick, 2005)) to new techniques based on higher order statistics (Shahid et al., 2008): it is imperative that these services be described and evaluated – both declaratively and as a product of their use – as higher level services consume their outputs.

9. Future work

CARMEN aims to provide a secure, endtoend virtual laboratory, linking data capture with analysis, modelling and publication, allowing digital assets to be exploited by large, distributed collaborator groups prior to publication and under longer term curation. This methodological shift is imperative to our understanding of the complexity of the nervous system. Our work to date has been concerned with mapping the problem space; specifically highlevel user requirements and software R&D priorities. A demonstration system has been released and usability testing is underway to identify lowlevel requirements. Further, system data and metadata format specifications have been defined, and published for community review.

CARMEN is a rarity in the neuroinformatics domain, in that it encapsulates research into computer science methods, translation of this research into proof of concept software demonstrations, and transfer of the knowledge gained during the first two stages into formalised software development. Traversing this process endto-end, within a distributed academic consortium, is highly challenging.

Immediate next steps are to formalise our internal development process, allowing fully benchmarked system releases to be distributed for evaluation. We expect early releases to support secure data exchange, annotation and data format translation. Provision of a set of core analysis services, potentially drawing from the spike detection and sorting techniques described, is scheduled soon thereafter. Our preliminary release is set for October 2008.

Our eventual aim is to supply a range of services, allowing the user to compare the performance of their complete workflow using different components to process different sets of data. Services and workflows may be provided by users in a similar way to data and metadata. A number of higher order analysis techniques will be provided by user groups funded by the CARMEN project grant. While it will be important to provide support for current luminary measures such as Rotter and Diesmann's distance (Rotter & Diesmann, 1999), Victor's distance (Victor, 2005) and Rossum's distance (Van Rossum, 2001), the key consideration is that the study of the nervous system is a longliving problem, requiring prolonged optimisation of data and analysis methods.

Using the current technology available in the SDE pattern matching system, it will be possible to implement a fast spike train comparison algorithm to compute distance measures. This will provide a highly flexible and useful tool for spike train comparison across multiple channels and also for higher level pattern processing. Currently, the SDE is used in a supervised manner. However, it will be useful to provide automatic usage for certain scenarios; for example, the use of predefined templates and other parameters – possibly drawn from a living database of data annotations – for automated detection of patterns. Work to this end is in progress and will continue to integrate the ability to index raw data against patterns derived from local field potentials, firing rates and spike trains.

Tools developed for studies in neurophysiology utilise both proprietary and open source scripting technologies. Each of these has advantages and disadvantages, and a mechanism to integrate different scripts is required. Work will continue in CARMEN to provide generic web service interfaces and tooling to address this problem, allowing the SDE to plug and play with services developed in a range of scripting formats, including MATLAB (n.d.).

In the long term, CARMEN aims to enable a truly global user community, building from the support of our early adopters participating in the CARMEN project. The mechanisms, and specifically business models, for doing this are highly complex and require deep consideration as our production infrastructure hardens into being.

10. Conclusions

Through the detailed discussions with the CARMEN research groups it is apparent that the system needs to support a wide range of technologies for services and address the difficulties involved in integrating multiple components developed by different research groups. A strategy for the transfer (wrapping and testing) of services to the system is being developed. Deployed services will need to be well tested and documented so that they can be easily shared. A strategy for interchange of data between services and support for the various data formats provided by the many acquisition systems is also being developed. System development times were underestimated at the outset, due in part to unanticipated technical barriers but more significantly to the deep complexity of the incentive structure uniting the many contributors to the programme. The latter should be a primary consideration for future platform initiatives in the neuroinformatics domain.

However, the first release of CARMEN supporting the methods described in this paper will be available towards the end of 2008. Third party and legacy tools and services are being integrated into the CARMEN infrastructure, together with the capability to develop and run workflows. CARMEN will provide an environment where data, tools, services and workflows can be shared. It will also provide facilities for development and testing of new services and tools.

The Signal Data Explorer (SDE) is an example of a CARMEN tool that provides visualisation and pattern matching on timeseries data (raw and derived neuroscience data). The SDE uses AURA, a Correlation Matrix Memory based neural network to perform extremely fast pattern matching. The technique includes preprocessing, AURA approximate matching (using k-NN and similarity measures) followed by post processing. The SDE and the AURA pattern match services can be used to detect patterns in neuronal data and can be used to search for complex conditions comprising of many different patterns across the large datasets. The SDE will provide a valuable means of querying and visualising patterns in the data held in the CARMEN repository, complementing conventional text based metadata query.

During preliminary CARMEN service development, spike detection using wavelet and morphology techniques was shown to outperform traditional thresholding and template based systems. The workflow tools that will be provided in subsequent releases of CARMEN will allow users to compare the impact of different spike detection and sorting techniques across their analysis protocols.

Acknowledgements

The CARMEN project is funded by the UK Engineering and Physical Sciences Research Council (EPSRC) as an eScience Pilot project (reference: EP/E002331/1). The CARMEN consortium is made up of the Universities of Cambridge, Leicester, Manchester, Newcastle, Plymouth, St Andrews, Stirling, Sheffield, Warwick, York; Imperial College, Cybula Ltd. and Lectus Therapeutics Ltd. We wish to acknowledge the roles each of our consortium members and collaborative partners have played in providing the impetus for this manuscript.

References

Austin, J. (1995). Distributed associative memories for high speed symbolic reasoning. In Working notes of IJCAI95 workshop on connectionistsymbolic integration: From unified to hybrid approaches (pp. 87–93).

Austin, J., Jackson, T., Fletcher, M., Jessop, M., Cowley, P., & Lobner, P. (2004). Distributed aircraft engine diagnostics. In I. Foster, & C. Kesselman (Eds.), The Grid: Blueprint for a new computing. Infrastructure (2nd ed.) (pp. 69–79). Morgan Kaufmann, ISBN: 1558609334. Austin, J., Kennedy, J., & Lees, K. (1995). The advanced uncertain reasoning architecture. In Proceedings of the Weightless Neural Network Workshop '95 (WNNW'95), University of Kent. BIRN — Biomedical Informatics Research Network. (n.d.). http://www.nbirn.net/ index.shtm. (Retrieved June 4, 2008).

BrainML model repository. (n.d.). http://brainml.org/. (Retrieved June 4, 2008).

CARMEN portal. (n.d.). https://hildr.wrg.york.ac.uk/carmenportal/ CARMENPortal. html. (Retrieved June 4, 2008).

Dynasoar — Dynamic deployment of web services on a grid or the internet. (n.d.). http://www.neresc.ac.uk/projects/ dynasoar/. (Retrieved June 4, 2008).

FIND — Finding information in neural data. (n.d.). http://find.bccn.unifreiburg.de. (Retrieved June 4, 2008).

Fletcher, M., Jackson, T., Jessop, M., Klinger, S., Liang, B., & Austin, J. (2006). The BROADEN Distributed Tool, Service and Data Architecture. UK eScience All Hands Meeting 2006.

Foster, I., & Kesselman, C. (2004). The Grid: Blueprint for a new computing infrastructure (2nd ed.). Morgan Kaufmann Publishers, ISBN: 1558604758.

Furber, S. B., Brown, G., Bose, J., Cumpstey, J. M., Marshall, P., & Shapiro, J. L. (2007). Sparse distributed memory using rankorder neural codes. IEEE Transactions on Neural Networks, 18(3).

FuGE — Functional genomics experiment. (n.d.). http://f uge.sourceforge.net/.

(Retrieved June 4, 2008).

GENESIS — General neural simulation system. (n.d.). http://www.genesissim.org/ GENESIS/. (Retrieved June 4, 2008).

Gibson, F. et al. (2008). Minimum information about a neuroscience investigation. In Nature Precedings. http://precedings.nature.com/documents/1720/version/ 1. (Retrieved June 4, 2008).

GOLD — Gridbased information models to support the rapid innovation of new high value added chemicals. (n.d.). http://www.neresc.ac.uk/projects/ GOLD/. (Retrieved June 4, 2008).

KlustaKwik home page. (n.d.). http://klustakwik.sourceforge.net. (Retrieved June 5, 2008).

Kim, K. H., & Kim, S. J. (2003). A waveletbased method for action potential detection from extracellular neural signal recording with low signaltonoise ratio. IEEE Transactions on Biomedical Engineering, 50(8), 999–1011.

Lewicki, M. (1998). A review of methods for spike sorting: The detection and classification of neural action potentials. Network: Computation in Neural Systems, R53–R78.

Liang, B., & Austin, J. (2003). Improved high performance k-nn classifier using a binary neural network. In Eighth International Conference on Engineering Applications of Neural Networks (EANN03), (pp. 148–153).

LSID — Life science identifiers. (n.d.). http://l sids.sourceforge.net/. (Retrieved June 4, 2008).

MATLAB. (n.d.). http://www.mathworks.com/products/ matlab/. (Retrieved June 4, 2008).

Multi channel systems MCS GmbH. (n.d.). http://www.multichannelsystems.com/. (Retrieved June 4, 2008).

Medical research council — Policy on data sharing and preservation. (n.d.). http://www.mrc.ac.uk/PolicyGuidance/EthicsAndGovernance/ DataSharing/ PolicyonDataSharingandPreservation/i ndex.htm#P16_ 1333. (Retrieved June 4, 2008).

Mtetwa, N., & Smith, L. S. (2006). Smoothing and thresholding in neuronal spike detection. Neurocomputing, 69(10–12), 1366–1370.

myGrid. (n.d.). http://www.mygrid.org.uk/. (Retrieved June 4, 2008).

Nenadic, Z., & Burdick, J. W. (2005). Spike detection using the continuous wavelet transform. IEEE Transactions on Biomedical Engineering, 52(1), 74–87.

NEURON. (n.d.). http://www.neuron.yale.edu/neuron/. (Retrieved June 4, 2008).

Neuroshare. (n.d.). http://neuroshare.sourceforge.net/ i ndex.shtml. (Retrieved June 4, 2008).

NIH data sharing policy. (2003). http://grants.nih.gov/grants/guide/noticefiles/ NOT-OD03032.html. (Retrieved June 4, 2008).

PROOF — the parallel ROOT facility. (n.d.). http://root.cern.ch/twiki/bin/view/ ROOT/ PROOF. (Retrieved June 4, 2008).

Quian Quiroga, R., Nadasdy, Z., & BenShaul, Y. (2004). Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. Neural Computation, 16, 1661–1687.

ROOT — An object oriented data analysis framework. (n.d.). http://root.cern.ch/. (Retrieved June 4, 2008).

Rotter, S., & Diesmann, M. (1999). Exact digital simulation of timeinvariant linear systems with applications to neuronal modelling. Biological Cybernetics, 81, 381–402.

Shahid, S., Walker, J., & Smith, L.S. (2008). A new spike detection algorithm for extracellular neural recordings. Under revision for Journal of Neuroscience

Methods.

SDE — Signal data explorer. (n.d.). http://www.cybula.com/flyers/ SignalData.pdf. (Retrieved June 4, 2008).

Smith, L. S., & Mtetwa, N. (2007). A tool for synthesizing spike trains with realistic interference. Journal of Neuroscience Methods, 159(1), 170–180.

Smith, L. S., Shahid, S., Vernier, A., & Mtetwa, N. (2007). Finding events in noisy signals. In Proceedings of the IET Irish signals and systems conference 2007 (pp. 31–35). The Institution of Engineering and Technology.

SRB — Storage resource broker. (n.d.). http://www.sdsc.edu/srb/ index.php/ Main_ Page. (Retrieved June 4, 2008).

TAVERNA. (n.d.). http://taverna.sourceforge.net/. (Retrieved June 4, 2008).

Van Rossum, M. C. W. (2001). A novel spike distance. Neural Computation, 13, 751–763.

Victor, J. D. (2005). Spike train metrics. Current Opinion in Neurobiology, 15, 585–592.

Watson, P. et al. (2007). The CARMEN neuroinformatics server. UK eScience All Hands Meeting 2007.

Web services. (n.d.). http://www.w3.org/2002/ ws/. (Retrieved June 4, 2008).

XACML — eXtensible Access Control Markup Language. (n.d.). http://www. oasis-open.org/committees/tc_home.php?wg_ abbrev=xacml. (Retrieved June 4, 2008).

Zhou, P., Austin, J., & Kennedy, J. (1999). A high performance kNN classifier using a binary correlation matrix memory. In Advances in neural information processing systems: Vol. 11. CA: MIT press.