

ISMB 2001 – Tutorial Proposal

The Ajax Command Description language: a framework for application integration and user interface building

Tutors

Alan Bleasby, Ph.D Biochemistry. Node manager of the UK EMBnet (European Molecular Biology Network) and Head of Proteomics Applications at the UK MRC Human Genome Mapping Project Resource Centre.

Alan Bleasby is the current co-ordinator of the EMBOSS project. He managed the UK SEQNET service for over 10 years before the service merged with the HGMP-RC. He has given many bioinformatics training courses in the above roles

Bruno A. Gaëta, Ph.D Biochemistry. Director of Education and Senior Scientist, Entigen
Bruno Gaëta manages educational resources, products and applications at Entigen. Previously he was in charge of education and training at the Australian Genomic Information Centre. He is also adjunct lecturer at the University of Sydney. He has taught numerous bioinformatics courses and workshops and is principal author of the ANGIS Bioinformatics Handbook, a 4-volumes practical sequence analysis textbook.

Timothy G. Littlejohn, Ph.D. Molecular Genetics, Chief Scientific Officer, Entigen
Tim Littlejohn was director of bioinformatics for the Canadian Organelle Genome Program before becoming head of the Australian Genomic Information Centre. He is also CSO and founder of Entigen and associate professor in bioinformatics at the University of Sydney and the University of NSW. He has taught bioinformatics in ANGIS courses and at the University of Sydney since 1996.

Michael Poidinger, PhD Virology, PGDipSci Comp Sci. Project Manager, Entigen
Michael Poidinger is responsible for the design and implementation of the ACD integration framework in BioNavigator. He has 8 years of postdoctoral experience, which included numerous presentations at large conferences, undergraduate lecturing and under/post graduate supervision.

Tutorial length

Half-day (4 hours)

Motivation

Many bioinformatics software developers focus on algorithm development and implementation rather than on user interface construction. This often results in software that the intended end users (biologists) find difficult to use because they are familiar only with desktop-based graphical user interfaces. When an effort is made to develop user-friendly interfaces, the results are often non-standard, and make the software difficult to integrate with other programs.

One solution to this issue has been the development of integrated bioinformatics front-ends that can provide a common, familiar user interface to a collection of disparate applications. These front-ends are typically web-based (PISE, W2H, BioNavigator) or fully interactive GUIs (SPIN/Staden package), and most of them include an application programming interface (API) to facilitate the addition of new programs. However in most cases, the addition of a new program still requires a substantial programming effort to develop a suitable command line wrapper for the application.

The Ajax Command Description (ACD) is a language originally developed for the EMBOSS sequence analysis package that can considerably simplify the integration of new applications into a common front-end. ACD provides a syntax for describing the command line parameters of an application in a configuration file. A number of user interface and integrated package providers, including SPIN (Staden package), BioNavigator, W2H-EMBOSS, SRS and PISE have adopted the ACD standard or a slight variation as part of their API. As a result, most command line-driven applications can be incorporated into these packages and the EMBOSS package simply by creating ACD configuration files describing their command line parameters.

An ACD file can describe not only the various parameters of an application but also its input file types. It can indicate whether any of the parameters are mandatory (like an input sequence file) or that certain parameters are within certain limits (a gap penalty for an alignment must be higher than 0 for instance). It can also indicate whether one parameter's value is dependent on the value or the presence of another. (An example: If the input sequence for an alignment program is DNA, it should not accept a protein comparison matrix).

ACD is a very 'forgiving' language in that it does not restrict the available syntax any more than is strictly necessary. ACD files can therefore be created for many command line-driven sequence analysis applications without any modification.

A formal definition of the ACD as used in the EMBOSS package is available at <http://www.hgmp.mrc.ac.uk/Software/EMBOSS/Acd/>.

Goals

- Introduce the ACD language and the various packages that make use of it
- Identify the ACD requirements: what makes an application "ACD compliant"?
- Demonstrate how to create an ACD file for a command line application
- Demonstrate how ACD files can be used to facilitate the integration of diverse applications into one common user interface

Intended audience

This tutorial is intended for bioinformatics application developers who are interested in creating ACD-compatible applications and include ACD files with their applications so that they can be incorporated easily into existing integrated packages that make use of ACD. The tutorial is also directed at bioinformatics system engineers, bioinformatics core facility developers and service providers who want to make use of the ACD

framework to integrate a collection of disparate applications into a standardized user interface.

The audience should have basic programming/scripting experience and UNIX knowledge. No biology expertise is required.

Contents and presentation outline

1. What is ACD? (15 min)
An introduction to the Ajax Command Description, its history and use in the EMBOSS package
2. Examples of ACD applications and interfaces (30 min)
Demonstrations of the various packages and interfaces that make use of ACD, including EMBOSS, EMBASSY, BioNavigator, Staden/SPIN, PISE, W2H-EMBOSS
3. The EMBOSS ACD syntax (45 min)
An overview of the structure of ACD files as used in the EMBOSS package, including application definition, data types, parameter description syntax, variables, arithmetic and conditional operations.
4. Extensions to the ACD syntax (30 min)
A discussion of variations to the ACD syntax used in systems other than EMBOSS, and of projected extensions to the standard
5. What makes an application “ACD-compliant”? (10 min)
A discussion of the command line format that can be described using ACD (applications that do not conform to this format may require a ‘wrapper’ in order to be used with ACD)
6. Writing an ACD file - a step by step guide (45 min)
A practical demonstration of writing an ACD file for a standard command line application
7. Putting it all together: a worked example of integrating a command line application into EMBOSS and BioNavigator (30 min)