BIOINFORMATICS APPLICATIONS NOTE
Vol. 24 no. 8 2008, pages 1102–1103
doi:10.1093/bioinformatics/btn085

Genome analysis

A suite of Perl modules for handling microarray data

James A. Morris*, Simon A. Gayther, Ian J. Jacobs and Christopher Jones
Translational Research Laboratories, UCL EGA Institute for Women's Health, University College London, London, United Kingdom

Received on December 14, 2007; revised on February 14, 2008; accepted on March 1, 2008
Advance Access publication March 18, 2008
Associate Editor: John Quackenbush

ABSTRACT

Summary: We describe PerlMAT, a Perl microarray toolkit providing easy to use object-oriented methods for the simplified manipulation, management and analysis of microarray data. The toolkit provides objects for the encapsulation of microarray spots and reporters, several common microarray data file formats and GAL files. In addition, an analysis object provides methods for data processing, and an image object enables the visualisation of microarray data. This important addition to the Perl developer's library will facilitate more widespread use of Perl for microarray application development within the bioinformatics community. The coherent interface and well-documented code enables rapid analysis by even inexperienced Perl developers.

Availability: Software is available at http://sourceforge.net/projects/perlmat

Contact: james.morris@ucl.ac.uk

1 INTRODUCTION

DNA microarrays have become a ubiquitously used technology in scientific research. Their popularity is due to the way they have enabled researchers to perform molecular and genetic experimental techniques on a genome wide scale in a high-throughput manner. Microarray experiments require computational processing steps for the management and analysis of the resulting data. The steps required include the extraction of raw data, storage of the data, quality assessment of the experiment, normalisation of the raw values to remove systemic biases, feature annotation and finally analysis of the microarray data to yield biological results. Although packages are available which perform all of these tasks, it is often necessary to develop tailor-made solutions in order to fulfil individual laboratories' requirements. Bespoke software applications provide a level of flexibility in the treatment of the data which is not possible through the use of existing commercial and non-commercial packages. The use of bespoke applications also enables the rapid incorporation of new or improved algorithms for any of the processing steps. The programming language Perl (http://www.perl.org/) has been described as the 'Swiss Army chainsaw of programming languages' and is an excellent option for developing microarray data handling solutions. Perl has established itself as a standard in the bioinformatics community due to a number of features; built in support for text processing; advanced yet simple to use database support; comprehensive support for internet services, such as common gateway interface (CGI) programming; and a very active and helpful developer community including the Comprehensive Perl Archive Network (CPAN; http://www.cpan.org/). However, there is limited support for dealing with microarray data; all available Perl code related to microarray data has been developed for use with the commercial Affymetrix platform, with no development tools available for other platforms. As a result, we have developed the Perl MicroArray Toolkit (PerlMAT), providing object-oriented methods for the manipulation, management and analysis of microarray data.

2 SOFTWARE OVERVIEW

PerlMAT has been developed as a suite of object-oriented Perl modules. The toolkit is segmented into logical groups such as file, analysis and image, organised in an inheritance hierarchy (Fig. 1) to provide reusable code that behaves in a consistent manner.

2.1 A Microarray experiment

The PerlMAT Microarray module works as a container for different PerlMAT objects. Upon the initialisation of a new Microarray object a Data object is created and stored, from which Spot objects are created for each arrayed spot. Replicate spots are represented by Reporter objects for each distinct genetic feature, and the Analysis object performs post-processing of either Spot or Reporter output. At any stage a variety of data plots can be exported using methods from the Microarray::Image class, either from a Microarray object or directly from a Data object.

2.2 Generating data plots

The Microarray::Image module is built using the Perl GD module and GD image library (http://www.libgd.org/). It contains methods for the creation of a number of different plots for visualisation of microarray experiment data, including; raw intensity scatter plots; MA/RI plots; Log2 ratio and individual channel intensity heatmaps; and chromosome comparative genomic hybridisation (CGH) plots.

*To whom correspondence should be addressed.

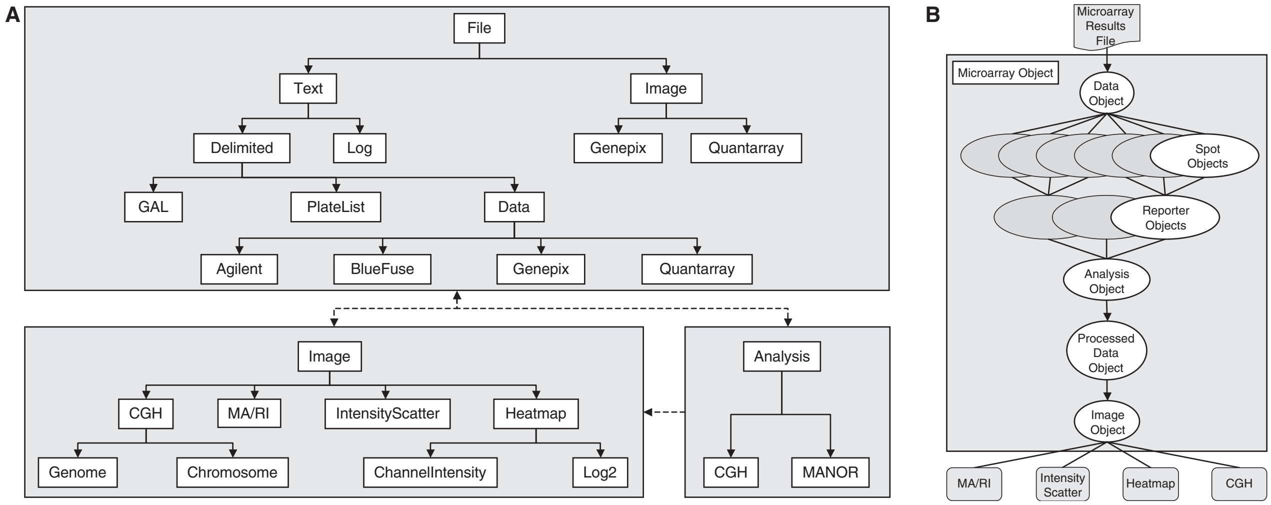

Fig. 1. The PerlMAT object organisation. (**A**) The PerlMAT toolbox is organised into three functional units: file, image and analysis. The heirarchy of the object sub-classes is shown, and their inheritance can be followed from the top to bottom using the arrow connectors. The dotted lines represent the ways data may be imported into the modules. (**B**) A schematic illustration of how the PerlMAT Microarray object acts as a container for other PerlMAT objects, simplifying data import and analysis. Upon initialisation using one of the supported data file modules, the Microarray object creates Spot objects and resolves Reporter replicates, passing processed data to the Analysis and Image modules.

2.3 Analysing microarray data

The Microarray::Analysis module enables the processing and filtering of raw microarray data. Microarray::Analysis::MANOR interfaces with R *via* RSPerl (http://www.omegahat.org/RSPerl/) to implement the MANOR (Neuvial *et al*., 2006) normalisation algorithm. Microarray::Analysis::CGH provides methods for the analysis of CGH microarray data, such as data smoothing.

2.4 Microarray reporters and spots

The Microarray::Spot module encapsulates a spotted feature on a microarray slide. Data from individual spots are imported as Spot objects from a Data object, which provide methods for retrieving all spot data. Microarrays often contain replicates of the same spotted reporter, and to model this PerlMAT uses a Reporter object which serves as a container into which replicate Spot objects are placed. The Reporter object then returns collated information about the replicate spots, such as the number of spots in the reporter and average signal intensity. Reporter annotation information can also be stored and retrieved using sub-classes of the Reporter object inheriting from any class describing a genetic feature.

2.5 Handling microarray files

PerlMAT provides modules that encapsulate a number of common microarray file formats, providing various methods for the retrieval and manipulation of file data. The Microarray::File module, from which all file objects inherit, contains basic general purpose methods for file manipulation and management including filehandle management methods for creation, return, closure and resetting.

Scanning images are handled by Microarray::File::Image, which provides methods for retrieving information contained within image file headers, including the image barcode, the scanning conditions (such as laser and PMT power) and other image details. Microarray::File::GAL provides methods for managing GAL files, including methods for returning details of the GAL file layout, the array blocks and the number of spots on the array. Microarray::File::Data provides methods for retrieving data from microarray data files. The Agilent, BlueFuse, Genepix and Quantarray file formats are all supported by individual modules.

3 DISCUSSION

PerlMAT provides tools for the manipulation, management and analysis of microarray data using the Perl programming language, providing easy to use objects that encapsulate all aspects of the microarray experimentation process. The use of inheritance and the modularity of the code will help to promote the development of PerlMAT by the extension and addition of objects. The object-oriented style also presents an intuitive interface, which together with extensive documentation facilitates ease of use by even inexperienced Perl developers.

The project is hosted on the SourceForge web site (http://sourceforge.net/projects/perlmat) with code maintained in a Concurrent Versions System (CVS) and available for download. This important addition to the Perl developer's library will facilitate more widespread use of Perl for microarray application development within the bioinformatics community.

ACKNOWLEDGEMENT

Authors thank The Mermaid charity (Copenhagen, Denmark).

Conflict of Interest: none declared.

REFERENCE

Neuvial,P. *et al*. (2006) Spatial normalization of array-CGH data. *BMC Bioinformatics*, **7**, 264.