R News

The Newsletter of the R Project

Volume 6/5, December 2006

Editorial

by Wolfgang Huber and Paul Murrell

Welcome to the fifth and final issue of R News for 2006, our third special issue of the year, with a focus on the use of R in Bioinformatics. Many thanks to guest editor Wolfgang Huber for doing a fantastic job in putting this issue together.

Paul Murrell
The University of Auckland, New Zealand
paul.murrell@R-project.org

Biology is going through a revolution. The genome sequencing projects that were initiated in the 1980's and undertaken in the 1990's have provided for the first time systematic inventories of the components of biological systems. Technological innovation is producing ever more detailed measurements on the functioning of these components and their interactions. The Internet has opened possibilities for the sharing of data and of computational resources that would have been unimaginable only 15 years ago.

The field of bioinformatics emerged in the 1990's to deal with the pressing questions that the massive amounts of new sequence data posed: sequence alignment, similarity and clustering, their phylogenetic interpretation, genome assembly, sequence annotation, protein structure. C and Perl were the languages of choice for the first generation of bioinformaticians. All of these question remain relevant, yet in addition we now have the big and colourful field of functional genomics, which employs all sorts of technologies to measure the abundances and activities of biomolecules under different conditions, map their interactions, monitor the effect of their perturbation on the phenotype of a cell or even a whole organism, and eventually to build predictive models of biological systems.

R is well suited to many of the scientific and computational challenges in functional genomics. Some of the efforts in this field have been pulled together since 2001 by the *Bioconductor* project, and many of the papers in this issue report on packages from the project. But Bioconductor is more than just a CRAN-style repository of biology-related packages. Motivated by the particular challenges of genomic data, the project has actively driven a number of technological innovations that have flown back into R, among these, package vignettes, an embracement of S4, the management of extensive package dependence hierarchies, and interfaces between R and

Contents of this issue:

other software systems. Biological metadata (for example, genome annotations) need to be tightly integrated with the analysis of primary data, and the Bioconductor project has invested a lot of effort in the provision of high quality metadata packages. The experimental data in functional genomics require more structured formats than the basic data types of R, and one of the main products of the Bioconductor core is the provision of common data structures that allow the efficient exchange of data and computational results between different packages. One example is the `ExpressionSet`, an S4 class for the storage of the essential data and information on a microarray experiment.

The articles in this issue span a wide range of topics. Common themes are *preprocessing* (data import, quality assessment, standardization, error modeling and summarization), *pattern discovery and detection*, and higher level statistical models with which we aim to gain insight into the underlying biological processes. Sometimes, the questions that we encounter in bioinformatics result in methods that have potentially a wider applicability; this is true in particular for the first three articles with which we start this issue.

Wolfgang Huber
European Bioinformatics Insitute (EBI)
European Molecular Biology Laboratory (EMBL) Cambridge, UK
huber@ebi.ac.uk

Graphs and Networks: Tools in Bioconductor

by Li Long and Vince Carey

Introduction

Network structures are fundamental components for scientific modeling in a number of substantive domains, including sociology, ecology, and computational biology. The mathematical theory of graphs comes immediately into play when the entities and processes being modeled are clearly organized into objects (modeled as graph nodes) and relationships (modeled as graph edges).

Graph theory addresses the taxonomy of graph structures, the measurement of general features of connectedness, complexity of traversals, and many other combinatorial and algebraic concepts. An important generalization of the basic concept of graph (traditionally defined as a set of nodes N and a binary relation E on N defining edges) is the hypergraph (in which edges are general subsets of the node set of cardinality at least 2).

The basic architecture of the Bioconductor toolkit for graphs and networks has the following structure:

- Representation infrastructure: packages **graph**, **hypergraph**;
- Algorithms for traversal and measurement: packages **RBGL**, **graphPart**
- Algorithms for layout and visualization: package **Rgraphviz**; **RBGL** also includes some layout algorithms;
- Packages for addressing substantive problems in bioinformatics: packages **pathRender**, **GraphAT**, **ScISI**, **GOstats**, **ontoTools**, and others.

In this article we survey aspects of working with network structures with some of these Bioconductor tools.

The basics: package graph

The **graph** package provides a variety of S4 classes representing graphs. A virtual class `graph` defines the basic structure:

```
> library(graph)
> getClass("graph")
Virtual Class

Slots:

Name:   edgemode  edgeData  nodeData
Class: character  attrData  attrData

Known Subclasses: "graphNEL", "graphAM",
   "graphH", "distGraph", "clusterGraph",
   "generalGraph"
```

A widely used concrete extension of this class is `graphNEL`, denoting the “node and edge list” representation:

```
> getClass("graphNEL")

Slots:

Name:      nodes     edgeL  edgemode  edgeData
Class:    vector      list character  attrData
```