

An Introduction to R and Bioconductor

Robert Gentleman

© Copyright 2011, all rights reserved

The R Language

- **R is a fully functional programming language and analysis environment for scientific computing**
- **it contains an essentially complete set of routines for numerical computations, statistical analysis and has extensive graphics capabilities**
- **computations/algorithms are organized by packages (there are over 3000) and these can easily be downloaded and installed on your computer**
- **users can create and share their own packages**
 - **two main repositories are CRAN and Bioconductor**
 - **packages will contain source code, documentation etc**

R Language

- **R is updated twice a year, on a very regular schedule**
 - typically in late April and in late Sept
 - you should keep your local versions of R and Bioconductor up to date
- **you should always use `biocLite` for Bioconductor packages and `install.packages`, or `update.packages` for R**
- **packages contain source code, documentation**
 - man pages with examples
 - vignettes: self-contained runnable documents that describe how the code in the package can be used on an analysis problem

Bioconductor

- Bioconductor is an **open source** and **open development** software project for the analysis of biomedical and genomic data.
- The project was started in the Fall of 2001 and includes developers in many countries
- **R** and the **R package system** are used to design and distribute software.
- A goal of the project is to develop integrated and interoperable software modules to provide comprehensive software solutions to relevant problems.

Why are we Open Source

- **so that you can find out what algorithm is being used, and how it is being used**
- **so that you can modify these algorithms to try out new ideas or to accommodate local conditions or needs**
- **so that they can be used as components (potentially modified) in other peoples software**

Overview

- **biology is a computational science**
- **problems of data analysis, data generation, reproducibility require computational support and computational solutions**
- **we value code reuse**
 - many of the tasks have already been solved
 - if we use those solutions we can put effort into new research
- **well designed, self-describing data structures help us deal with complex data**

Goals

- Provide access to powerful statistical and graphical methods for the analysis of genomic data.
- Facilitate the integration of biological metadata (**GenBank, GO, Entrez Gene, PubMed**) in the analysis of experimental data.
- Allow the rapid development of extensible, interoperable, and scalable software.
- Promote high-quality documentation and reproducible research.
- Provide training in computational and statistical methods.

Bioconductor packages

Release 2.8, 466 Software Packages!

- **General infrastructure**
Biobase, Biostrings, biocViews
- **Annotation:**
annotate, annaffy, biomaRt, AnnotationDbi → data packages.
- **Graphics/GUIs:**
geneplotter, hexbin, limmaGUI, exploRase
- **Pre-processing:**
affy, affycomp, oligo, makecdfenv, vsn, gcrn, limma
- **Differential gene expression:**
genefilter, limma, ROC, siggenes, EBArrays, factDesign
- **GSEA/Hypergeometric Testing**
GSEABase, Category, GOstats, topGO
- **Graphs and networks:**
graph, RBGL, Rgraphviz
- **Flow Cytometry:**
flowCore, flowViz, flowUtils
- **Protein Interactions:**
ppiData, ppiStats, ScISI, Rintact
- **Sequence Data:**
Biostrings, ShortRead, chipseq, rtracklayer, IRanges
- **Other data:**
xcms, DNACopy, PROcess, aCGH, rsbml, SBMLR, Rdisop

Component software

- **most interesting problems will require the coordinated application of many different techniques**
- **thus we need integrated interoperable software**
- **of primary importance is well designed and shared data structures**
- **you should design your contributions to be a cog in a big machine**

Data complexity

- Dimensionality.
- Dynamic/evolving data: e.g., gene annotation, sequence, literature.
- Multiple data sources and locations: in-house, WWW.
- Multiple data types: numeric, textual, graphical.

No longer $X_{n \times p}$!

We distinguish between biological metadata and experimental metadata.

Experimental metadata

- **Gene expression measures**
 - scanned images, i.e., raw data;
 - image quantitation data, i.e., output from image analysis;
 - normalized expression measures,
 - Reliability/quality information for the expression measures.
- **Information on the probe sequences printed on the arrays (array layout).**
- **Information on the target samples hybridized to the arrays.**
- **See Minimum Information About a Microarray Experiment (MIAME) standards and the `MAGEML` package.**
- **standards and requirements for sequence data are evolving**

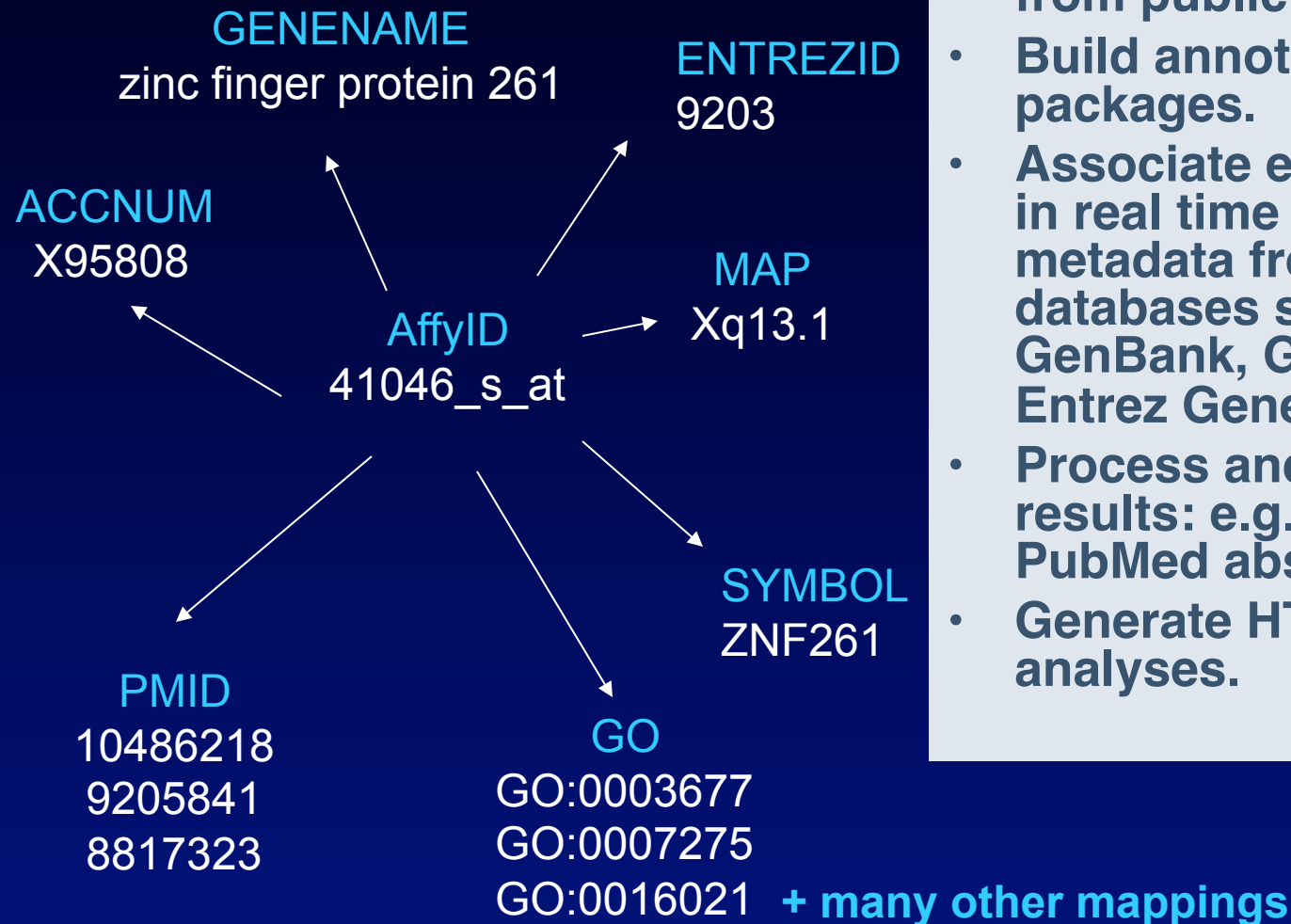
Biological metadata

- **Biological attributes that can be applied to the experimental data.**
- **E.g. for genes**
 - chromosomal location;
 - gene annotation (Entrez Gene, GO);
 - gene models
 - relevant literature (PubMed)
- **Biological metadata sets are large, evolving rapidly, and typically distributed via the WWW.**
- **Tools:** `annotate`, `biomaRt`, and `AnnotationDbi` **packages, and annotation data packages.**

Annotation packages

annotate, annafy, biomaRt, and AnnotationDbi

Metadata package `hgu95av2` mappings between different gene IDs for this chip.



- Assemble and process genomic annotation data from public repositories.
- Build annotation data packages.
- Associate experimental data in real time to biological metadata from web databases such as GenBank, GO, KEGG, Entrez Gene, and PubMed.
- Process and store query results: e.g., search PubMed abstracts.
- Generate HTML reports of analyses.

Vignettes

- Bioconductor developed a new documentation paradigm, the vignette.
- A **vignette** is an **executable document** consisting of a collection of documentation text and code chunks.
- Vignettes form **dynamic, integrated, and reproducible statistical documents** that can be automatically updated if either data or analyses are changed.
- Vignettes can be generated using the `Sweave` function from the R `tools` package.

Short Courses/Conferences

- we have given many short courses
 - see bioconductor.org for more details on upcoming courses
- **BioC2011 - Seattle, July 28-29**

Bioconductor Software

- concentrate development resources on a few important aspects
- **Biobase**: core classes and definitions that allow for succinct description and handling of the data
- **annotate**: generic functions for annotation that can be specialized
- **genefilter**: fast filtering via virtually every mechanism
- **graph/Rgraphviz/RBGL**: code for handling graphs and networks
- **Biostrings/ShortRead/IRanges**: string manipulations, sequence analysis

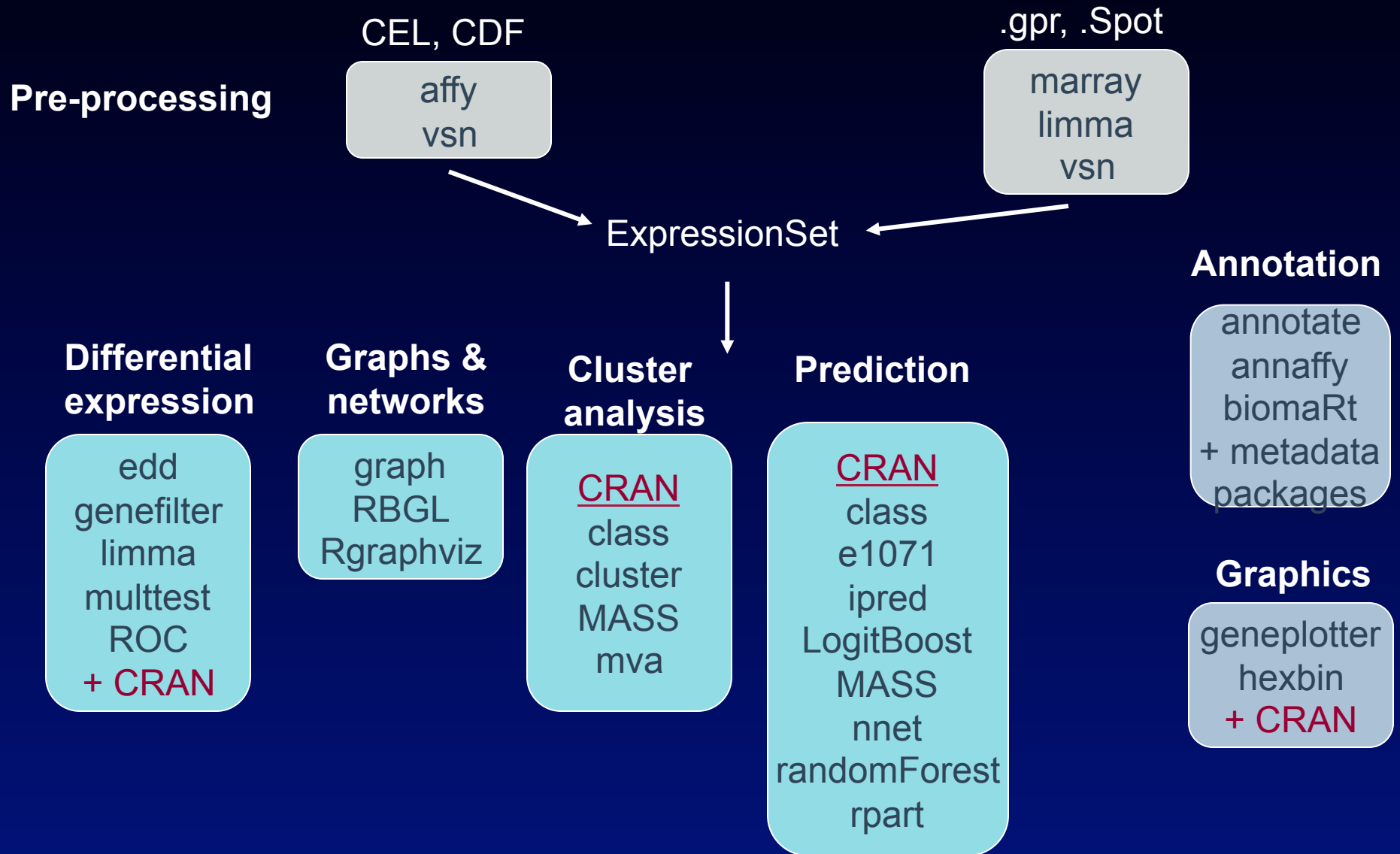
Quality Assessment

- ensuring that the data are of sufficient quality is an essential first step
- **arrayQuality Metrics**: comprehensive QA assessment of microarrays (one color or two color)
 - now modular
 - output is easy to browse HTML
- **ShortRead**: tools for QA of short reads, primarily Illumina

Biobase:ExpressionSet

- **software should help organize and manipulate your data**
- **this was the intention of the original exprSet class**
- **the data need to be assembled correctly once, and then they can be processed, subset etc without worrying about them**
- **exprSet was too limited (and too oriented to single channel arrays)**
- **we developed the new ExpressionSet class**

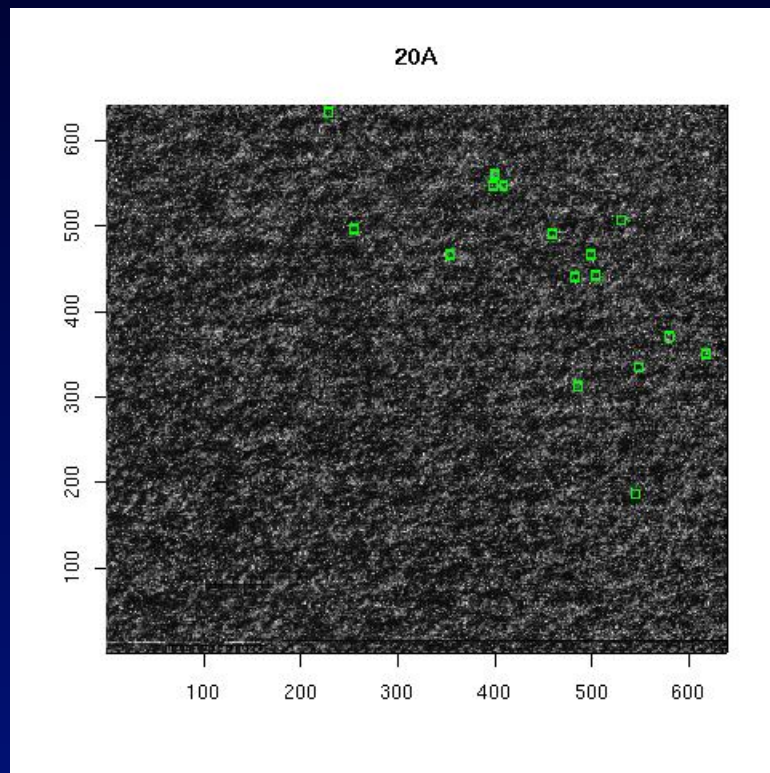
Microarray data analysis



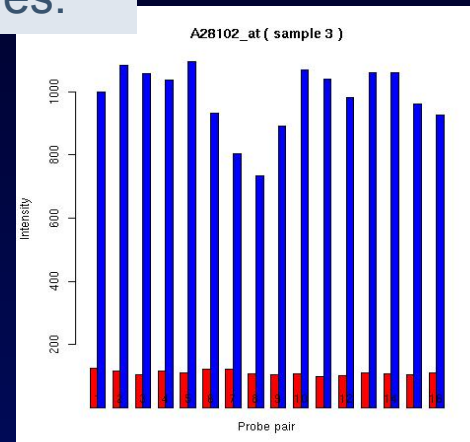
affy/oligo package

Pre-processing oligonucleotide chip data:

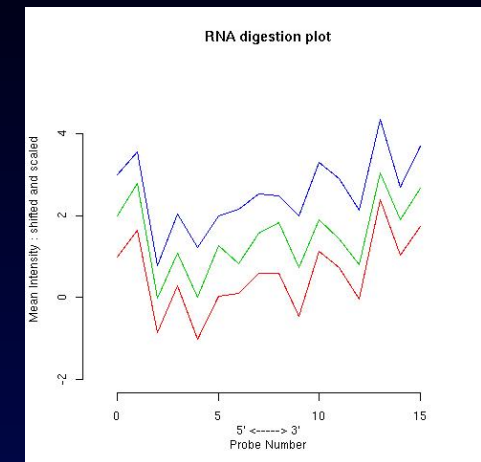
- diagnostic plots,
- background correction,
- probe-level normalization,
- computation of expression measures.



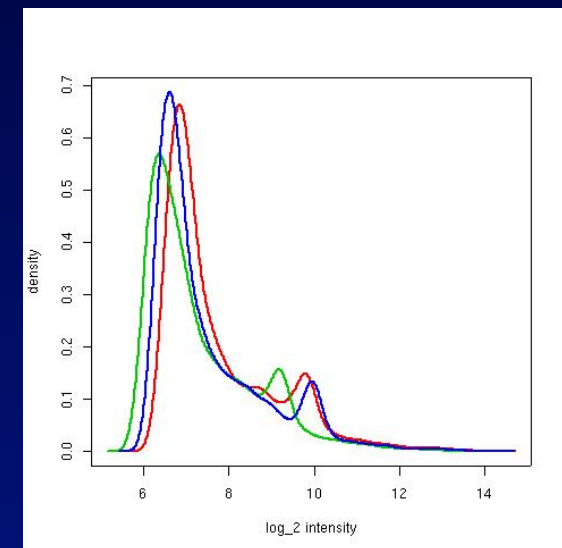
image



barplot.ProbeSet



plotAffyRNAdeg



plotDensity

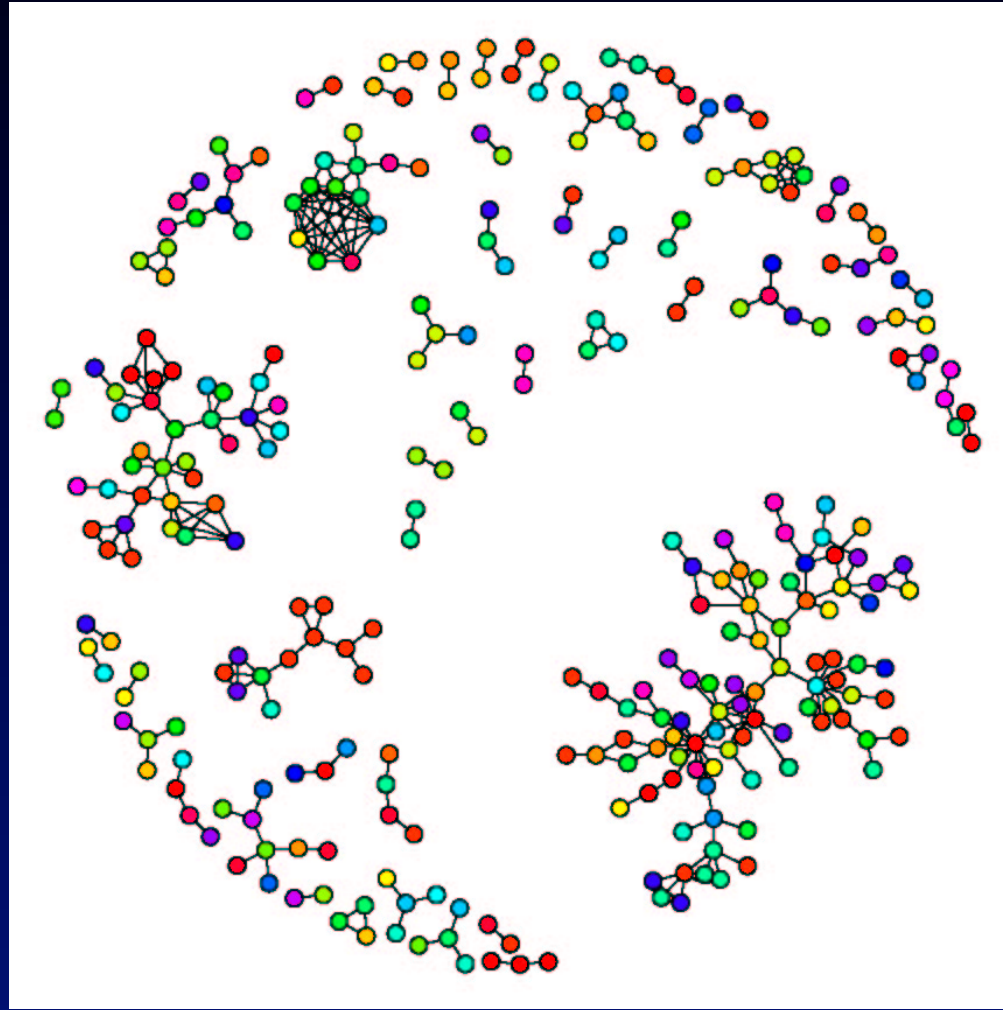
Differential Expression

- **limma**: provides a linear models interface for DE
 - uses a moderated variance
 - a variety of p-value correction methods are provided
- **DESeq and edgeR**: for sequence data
 - similar approach to limma
 - make use of count data (Neg Binomial)

Machine Learning

- **MLInterfaces**
- provides uniform calling sequences and return values for all machine learning algorithms
- **MLearn** is the main wrapper function
 - methods, eg knni, are passed to the wrapper
- return values are of class **MLOutput**
- see the **MLInterfaces** vignette for more details

graph and Rgraphviz



apComplex

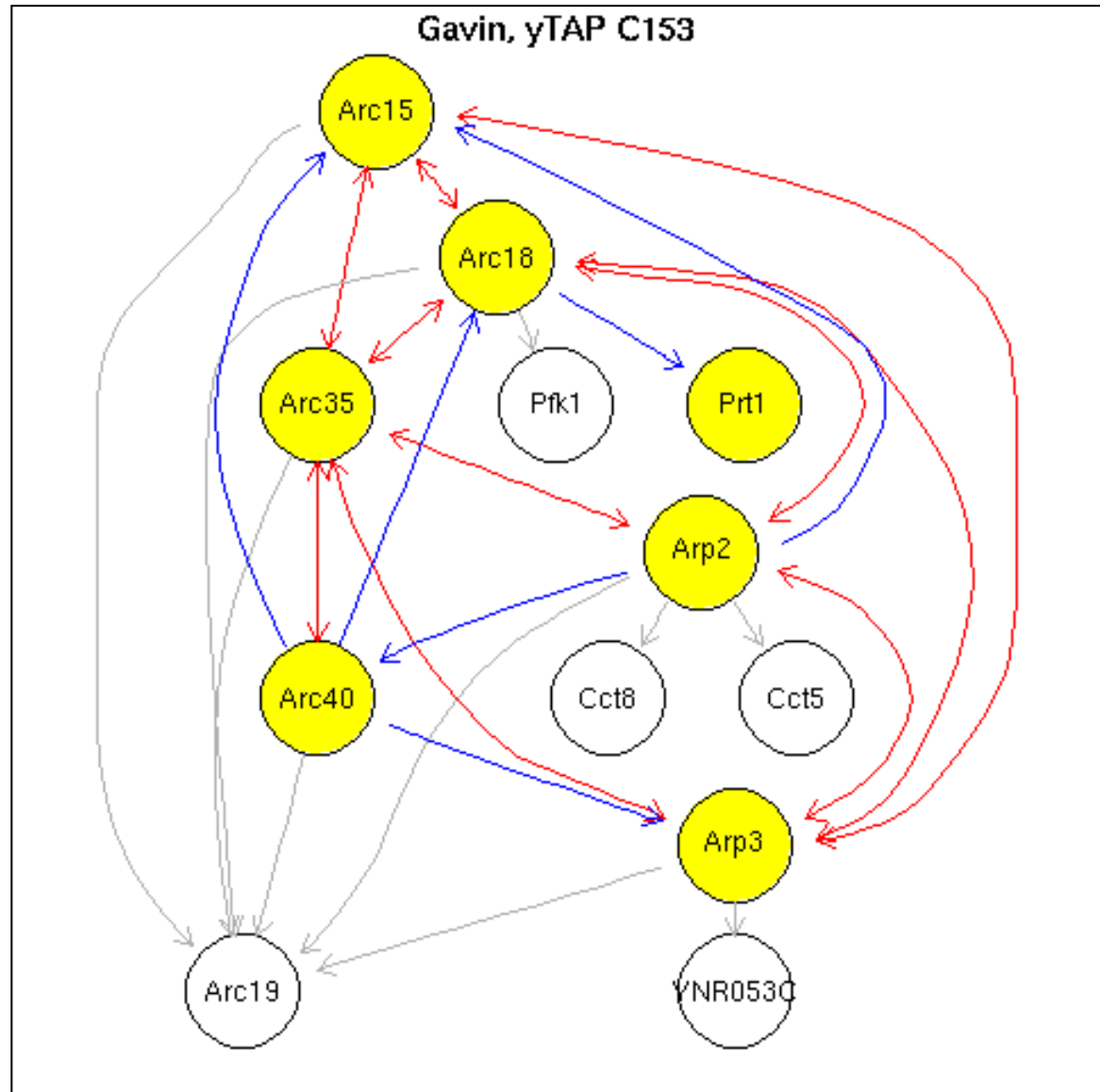
Arp2/3

Arp2/3 complex:

- Arp2
- Arp3
- Arc15
- Arc18
- Arc19
- Arc35
- Arc40

'The Arp2/3 complex is a stable multiprotein assembly required for the nucleation of actin filaments in all eukaryotic cells and consists of seven proteins in human and yeast.'

Winter, et al (1997). *Curr Biol.*
Higgs and Pollard (2001). *Annu Rev Biochem.*



Publications

- **Bioconductor: Open software development for computational biology and bioinformatics, *Genome Biology* 2004, 5:R80, <http://genomebiology.com/2004/5/10/R80>**
- **Bioinformatics and Computational Biology Solutions using R and Bioconductor, Springer, 2005, R. Gentleman, V. Carey, W. Huber, R. Irizarry, S. Dudoit eds.**
- **Bioconductor Case Studies, Springer**
- **R Programming for Bioinformatics, Chapman Hall**

References

- **R** www.r-project.org, cran.r-project.org
 - software (CRAN);
 - documentation;
 - newsletter: R News;
 - mailing list.
- **Bioconductor** www.bioconductor.org
 - software, data, and documentation (vignettes);
 - training materials from short courses;
 - mailing list (please read the posting guide)