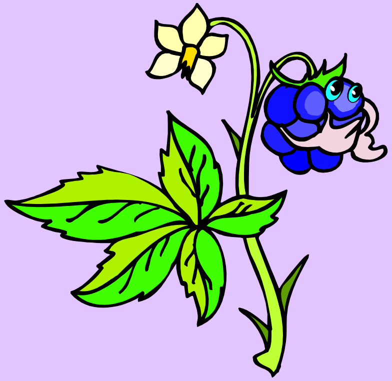
Polyploid data analysis
and…
How to gently transition from software user to software developer



Lindsay Clark, Genetics Graduate Group

February 28, 2011

UCDAVIS
PLANT SCIENCES

My study organism- invasive blackberry (*Rubus*)

- Many species of Rubus are invasive worldwide, particularly in the tetraploid, apomictic *R. fruticosus* agg. from Europe
- There is also native blackberry on the West Coast – *R. ursinus* – an allopolyploid mix of 6x, 8x, and 12x
- Diploid raspberry species, odd ploidy cultivars

Dissertation questions

- How much hybridization has gone on between native and non-native *Rubus* on the West Coast?
- Do the hybrids reproduce sexually or by apomixis? (Evolution of new invasive types?)
- What is the clonal diversity of the *R. fruticosus* invasion, and where did the clones come from?

➢ Using microsats to address all of these

Polyploid HWE

- So if in a diploid

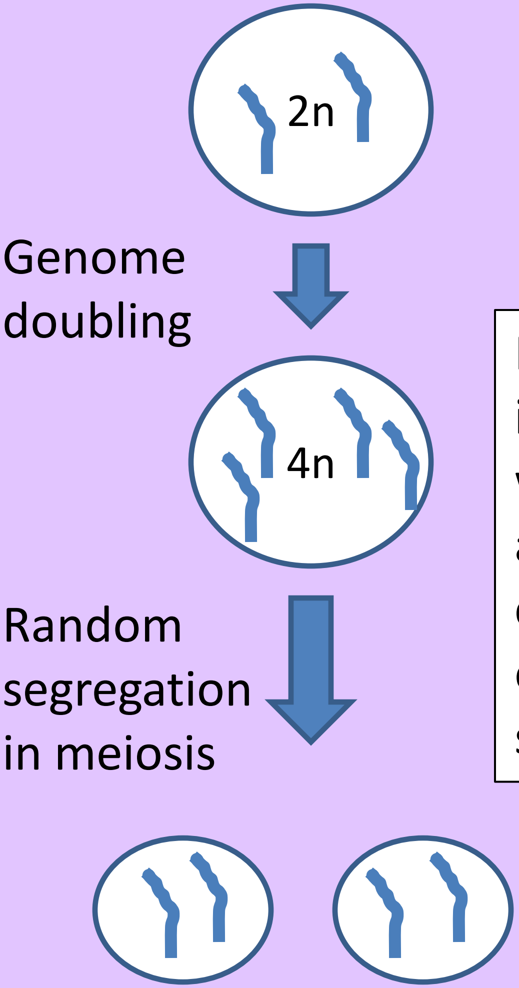
$$p^2 + 2pq + q^2 = 1$$

- Then in an autotetraploid

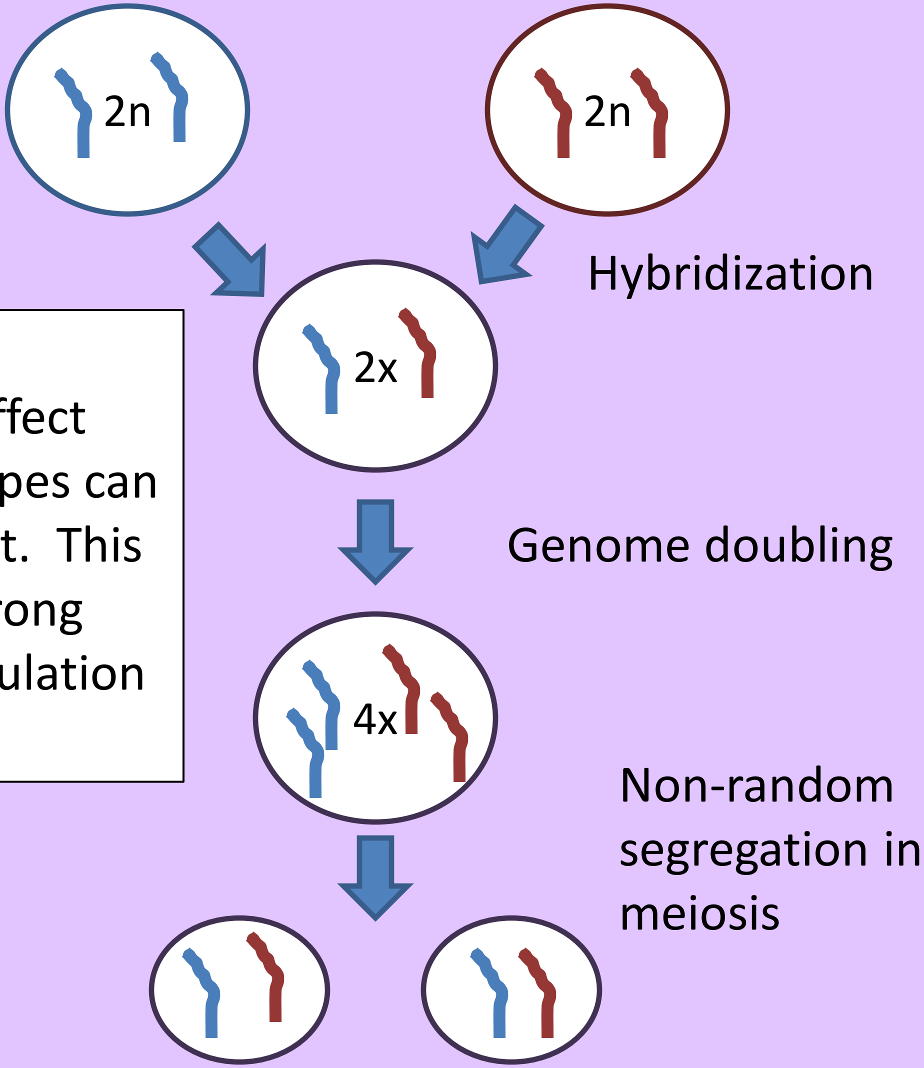
$$p^4 + 4p^3q + 6p^2q^2 + 4pq^3 + q^4 = 1$$

- But that’s the simplest situation…

Autopolyploidy vs. allopolyploidy



Patterns of inheritance affect which genotypes can and can’t exist. This can have a strong effect on population statistics!



Example: what genotypes are possible with self fertilization?

Autotetraploid (polysomic)

- Parent = ABCD
- Possible gametes = AB, AC, AD, BC, BD, CD
- Offspring = AABB, AACC, AADD, BBCC, BBDD, CCDD, AABC, AABD, ABBC, ABBD, ABCD, AACD, ABCC, ACCD, ABDD, ACDD, BBCD, BCCD, BCDD

Allotetraploid (disomic)

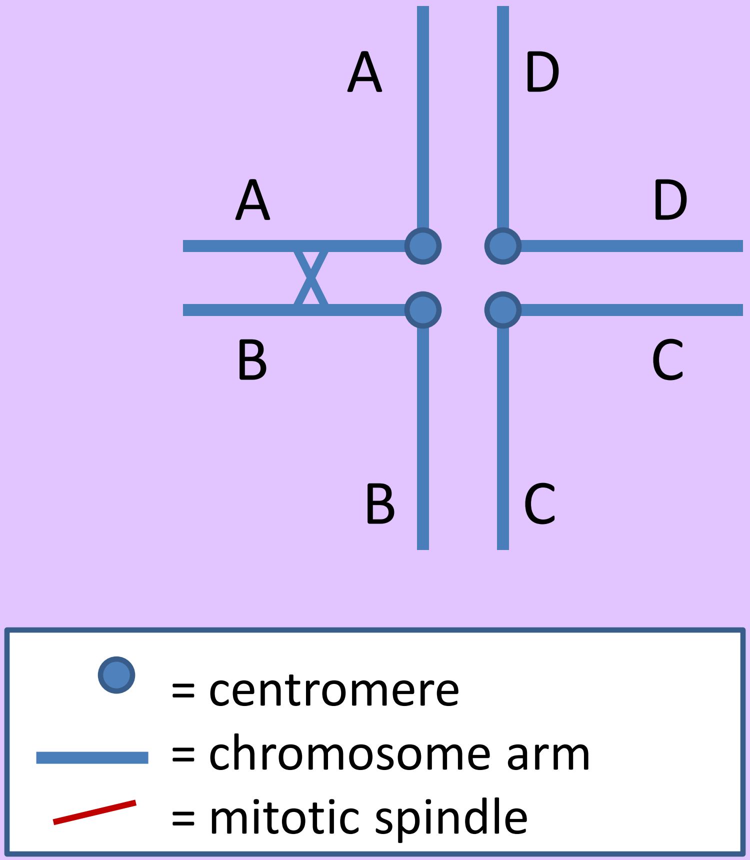
- Parent = AB CD
- Possible gametes = AC, AD, BC, BD
- Offspring = AACC, AADD, BBCC, BBDD, AACD, ABCC, ABCD, ABDD, BBCD

Ideally:
Know the inheritance pattern of your organism.
In an allopolyploid, know which alleles go with which isoloci.

Double reduction

- In tetrasomic ABCD, expected gametes are AB, AC, AD, BC, BD, CD
- More rarely, you can get AA, BB, CC, or DD from double reduction
- Most analysis has “no double reduction” assumption

In prophase of meiosis I:



● = centromere

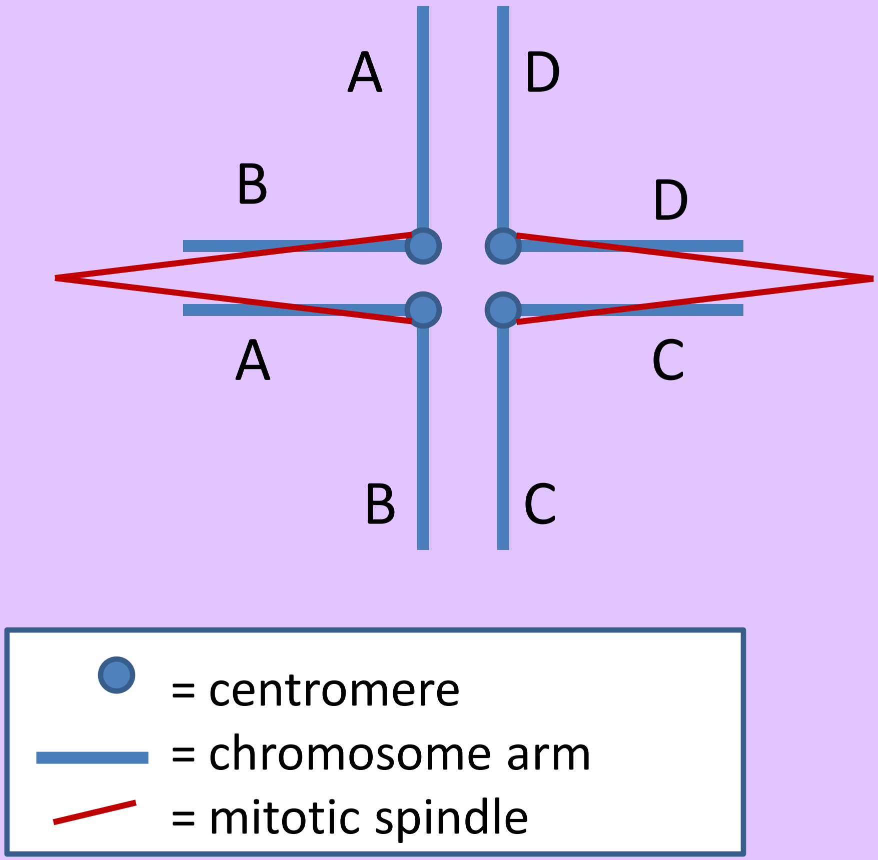
— = chromosome arm

╱ = mitotic spindle

Double reduction

- In tetrasomic ABCD, expected gametes are AB, AC, AD, BC, BD, CD
- More rarely, you can get AA, BB, CC, or DD from double reduction
- Most analysis has “no double reduction” assumption

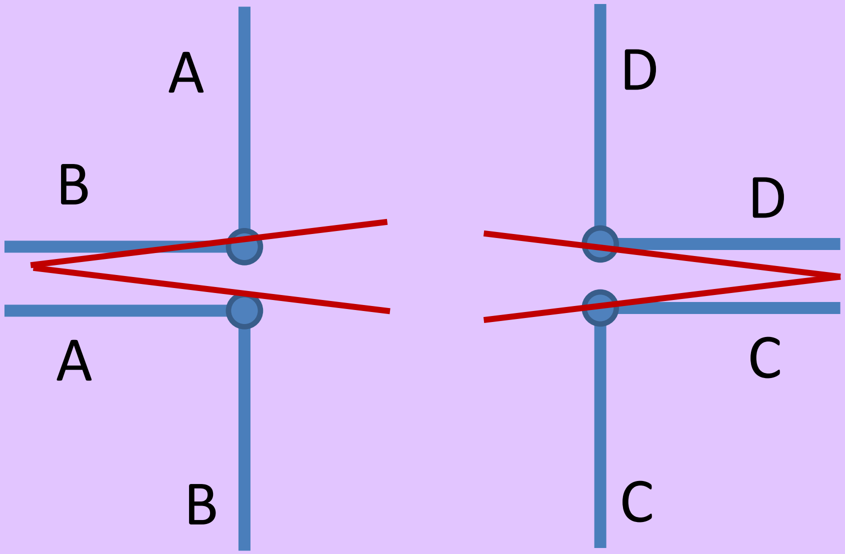
In metaphase of meiosis I:



Double reduction

- In tetrasomic ABCD, expected gametes are AB, AC, AD, BC, BD, CD
- More rarely, you can get AA, BB, CC, or DD from double reduction
- Most analysis has “no double reduction” assumption

In anaphase of meiosis I:

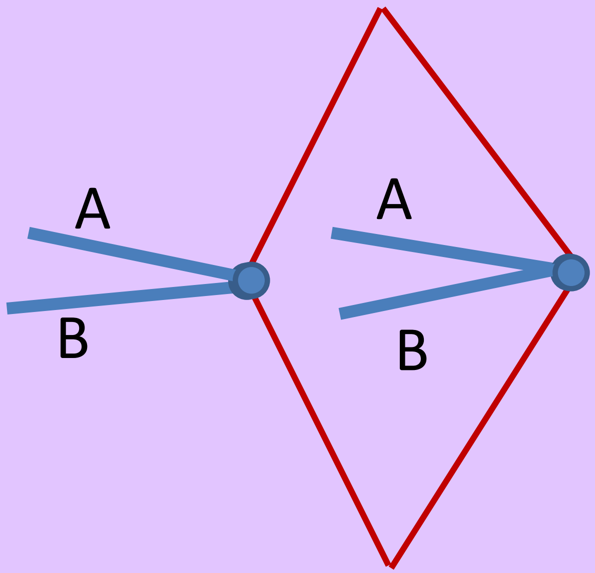


● = centromere
— = chromosome arm
╱ = mitotic spindle

Double reduction

- In tetrasomic ABCD, expected gametes are AB, AC, AD, BC, BD, CD
- More rarely, you can get AA, BB, CC, or DD from double reduction
- Most analysis has “no double reduction” assumption

In metaphase of meiosis II:



● = centromere

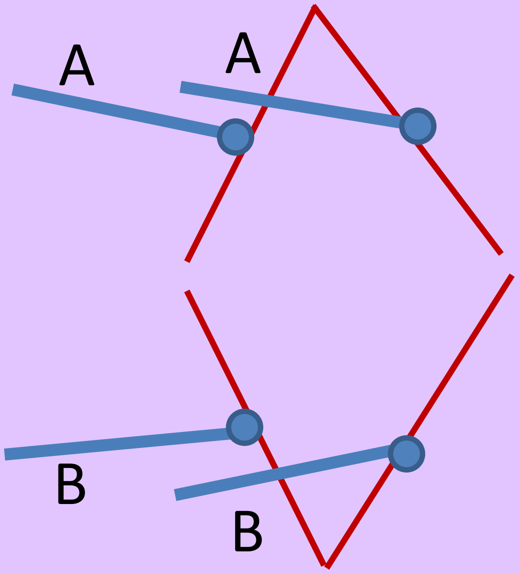
━ = chromosome arm

╱ = mitotic spindle

Double reduction

- In tetrasomic ABCD, expected gametes are AB, AC, AD, BC, BD, CD
- More rarely, you can get AA, BB, CC, or DD from double reduction
- Most analysis has “no double reduction” assumption

In anaphase of meiosis II:



● = centromere

━ = chromosome arm

― = mitotic spindle

Using dominant markers in a polyploid

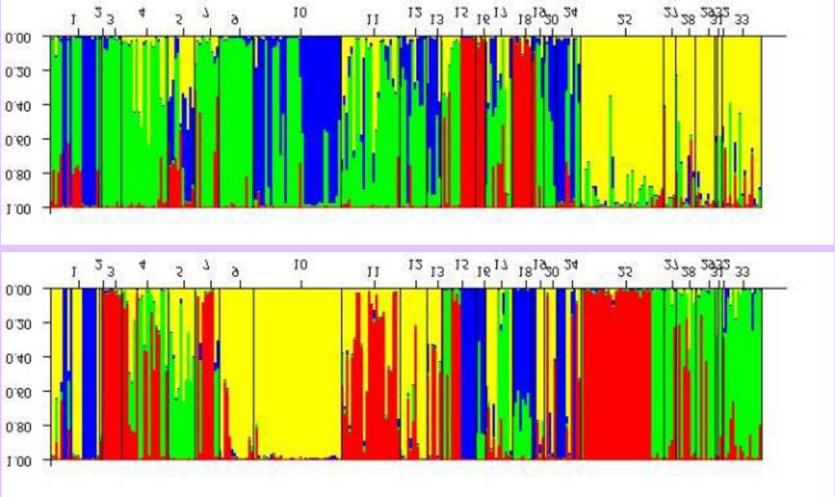
- Homozygous recessive individual is going to be more rare under polysomic inheritance
- If q is the frequency of the recessive allele:
- In diploid, expect absence of band at frequency of q^2
- In autotetraploid, absence of band is at a frequency of q^4
- See http://cran.r-project.org/web/packages/polySegratio/

Using codominant markers in a polyploid

- Allele copy number ambiguity in partial heterozygotes
- You have a tetraploid with alleles A and B at one locus
- Is it AABB, AAAB, or ABBB?
- Use of peak height or band intensity to distinguish copy number is complicated by PCR amplification efficiency

Available Software - STRUCTURE

- STRUCTURE now allows polyploid genotypes with allele copy number ambiguity!
- It is **only** for autopolyploid data
- I have had issues with reproducibility, although I was trying to fudge allopolyploid data into STRUCTURE.

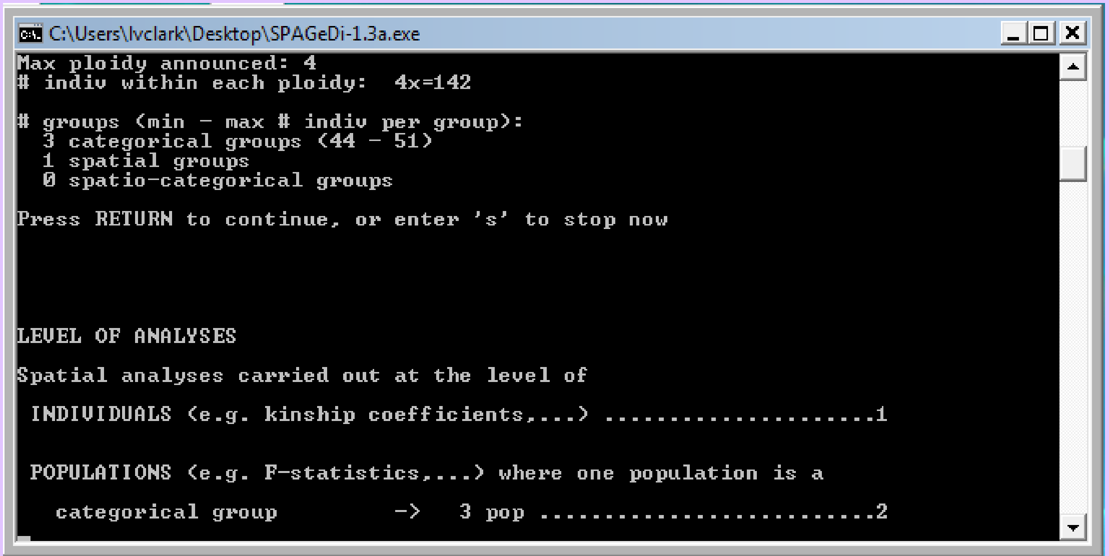


My results at 400,000 reps

http://pritch.bsd.uchicago.edu/structure.html

Available Software - SPAGeDi

- Inter-individual and inter-population distances, analysis involving geographic data
- Only for autopolyploids, and treats every allele as having equal chance of being present in more than one copy (an oversimplifying assumption).

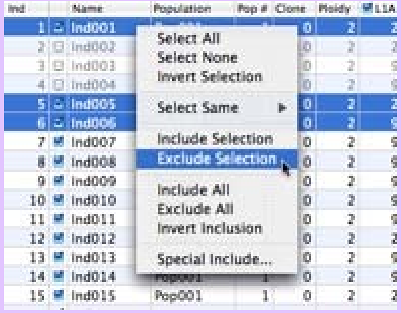


```
Max ploidy announced: 4
# indiv within each ploidy:  4x=142

# groups (min - max # indiv per group):
   3 categorical groups (44 - 51)
   1 spatial groups
   0 spatio-categorical groups

Press RETURN to continue, or enter 's' to stop now




LEVEL OF ANALYSES

Spatial analyses carried out at the level of

 INDIVIDUALS (e.g. kinship coefficients,...) .....................1


 POPULATIONS (e.g. F-statistics,...) where one population is a

   categorical group         ->   3 pop .........................2
```

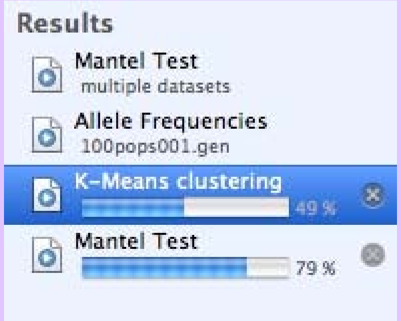
http://ebe.ulb.ac.be/ebe/Software.html

Available Software - GenoDive

- A variety of tools available, particularly dealing with asexual reproduction in polyploids. User-friendly interface.
- Old, limited version is Mac/PC
- Newer version is Mac only. Unpublished, with brief documentation.



Ind		Name	Population	Pop #	Clone	Ploidy	☑ L1A
1	☑	Ind001			0	2	2
2	☐	Ind002			0	2	2
3	☐	Ind003			0	2	5
4	☐	Ind004			0	2	5
5	☑	Ind005			0	2	2
6	☑	Ind006			0	2	5
7	☑	Ind007			0	2	5
8	☑	Ind008			0	2	5
9	☑	Ind009			0	2	5
10	☑	Ind010			0	2	5
11	☑	Ind011			0	2	9
12	☑	Ind012			0	2	2
13	☑	Ind013			0	2	5
14	☑	Ind014	Pop001	1	0	2	5
15	☑	Ind015	Pop001	1	0	2	2



Results

Mantel Test
multiple datasets

Allele Frequencies
100pops001.gen

K-Means clustering
49 %

Mantel Test
79 %

http://www.bentleydrummer.nl/software/software/GenoDive.html

Available software - others

- Tetrasat, Tetra, and Atetra for allotetraploid analysis, although they still don’t deal with issue of assigning alleles to isoloci
- POPDIST – distances between populations, deals with allele copy number ambiguity like SPAGeDi

How this led me to write my own

- Wanted to perform calculations that weren’t readily available in other software
- Wanted an easy and accurate way to convert my data to different formats (Structure, presence-absence)
- (Most available conversion software is just for diploids!)

So I learned R and started writing functions

- Introduction to R: http://cran.r-project.org/doc/manuals/R-intro.html
- Data Import/Export: http://cran.r-project.org/doc/manuals/R-data.html
- R listserves: Google will often find answers to your questions there
- Experiment with code to see if it works the way you think it does!

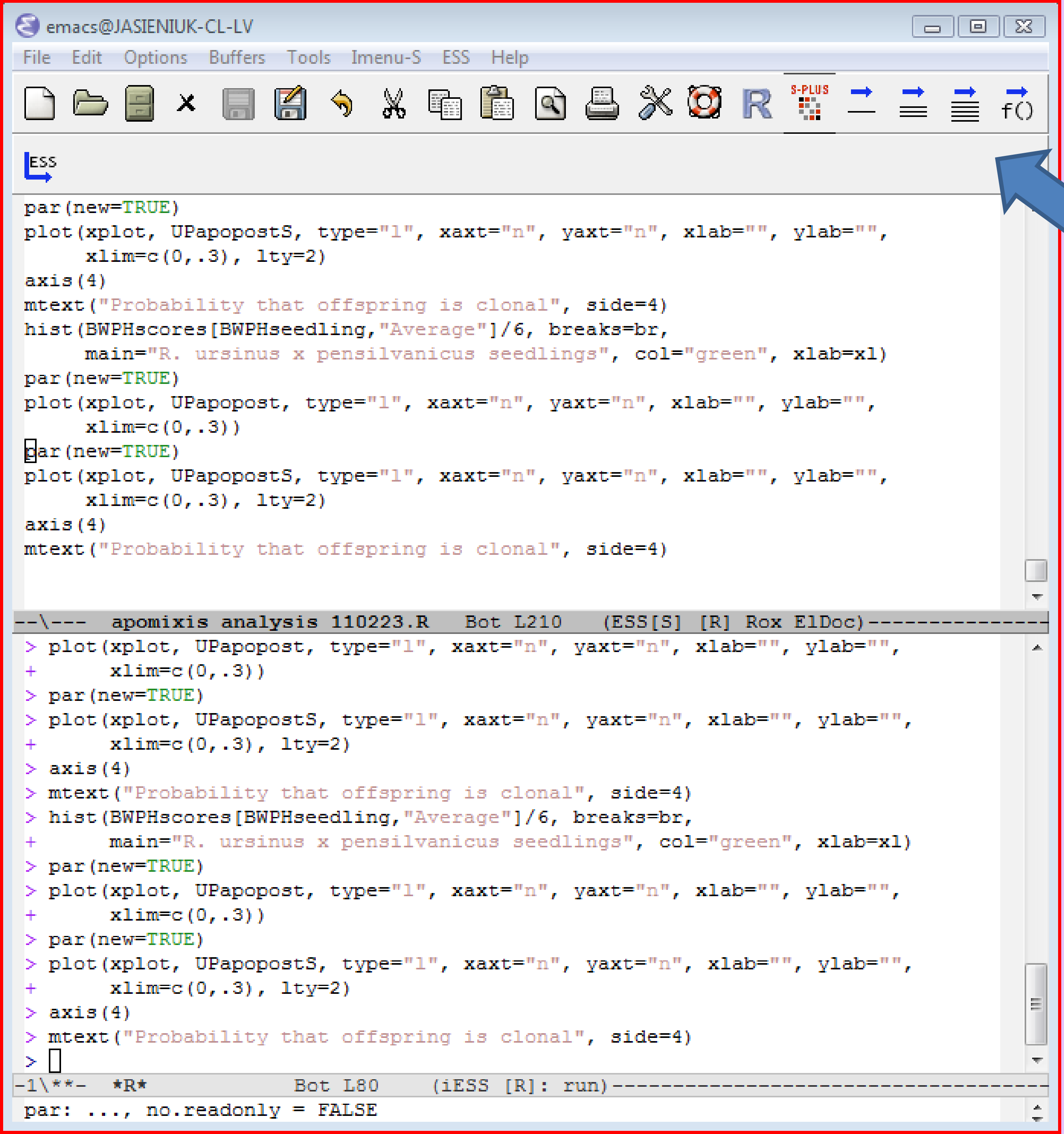
Example R code to show in R GUI

- 2 + 2
- X <- 2 + 2
- X
- Y <- c(1, 5, 9)
- Y
- mean(Y)
- Div2plus1 <- function(v) return(v/2 + 1)
- Div2plus1(Y)

Resources for R Users

- CRAN
  - R download
  - Manuals
  - Packages – on web or through R GUI
  - Genetics view, including Pop Gen stuff
- Bioconductor: more for genomics
- Emacs Speaks Statistics – text editor with utilities for R

Emacs Speaks Statistics!



```
par(new=TRUE)
plot(xplot, UPapopostS, type="l", xaxt="n", yaxt="n", xlab="", ylab="",
     xlim=c(0,.3), lty=2)
axis(4)
mtext("Probability that offspring is clonal", side=4)
hist(BWPHscores[BWPHseedling,"Average"]/6, breaks=br,
     main="R. ursinus x pensilvanicus seedlings", col="green", xlab=xl)
par(new=TRUE)
plot(xplot, UPapopost, type="l", xaxt="n", yaxt="n", xlab="", ylab="",
     xlim=c(0,.3))
par(new=TRUE)
plot(xplot, UPapopostS, type="l", xaxt="n", yaxt="n", xlab="", ylab="",
     xlim=c(0,.3), lty=2)
axis(4)
mtext("Probability that offspring is clonal", side=4)

--\---  apomixis analysis 110223.R   Bot L210    (ESS[S] [R] Rox ElDoc)---------------
> plot(xplot, UPapopost, type="l", xaxt="n", yaxt="n", xlab="", ylab="",
+      xlim=c(0,.3))
> par(new=TRUE)
> plot(xplot, UPapopostS, type="l", xaxt="n", yaxt="n", xlab="", ylab="",
+      xlim=c(0,.3), lty=2)
> axis(4)
> mtext("Probability that offspring is clonal", side=4)
> hist(BWPHscores[BWPHseedling,"Average"]/6, breaks=br,
+      main="R. ursinus x pensilvanicus seedlings", col="green", xlab=xl)
> par(new=TRUE)
> plot(xplot, UPapopost, type="l", xaxt="n", yaxt="n", xlab="", ylab="",
+      xlim=c(0,.3))
> par(new=TRUE)
> plot(xplot, UPapopostS, type="l", xaxt="n", yaxt="n", xlab="", ylab="",
+      xlim=c(0,.3), lty=2)
> axis(4)
> mtext("Probability that offspring is clonal", side=4)
> 
-1\**-  *R*          Bot L80    (iESS [R]: run)----------------------------------------
par: ..., no.readonly = FALSE
```

Buttons to send your code to R

A text file of R code

R

Packages with formats for DNA marker data

- **genetics** – “genotype” class - diploid codominant ; stores two alleles + locus info
- **adegenet** – “genind” class - one ploidy, no allele copy ambiguity, dominant/codominant, stores all genotypes + population identities
- **polysat**- “genambig” class – mixed ploidy, codominant, allele copy ambiguity, stores all genotypes + pop id + ploidies + SSR repeat type

Making an R package

- The language for using R is the exact same language for programming R
- (R packages also allow some C code)
- Simply create text files of R code
- Documentation needs to follow a specific format, but there are functions to automatically generate template files

Resources for R Programmers

- “Writing R Extensions” manual: http://cran.r-project.org/doc/manuals/R-exts.html
- *Software for Data Analysis: Programming with R* by John M. Chambers, 2008. DOI: 10.1007/978-0-387-75936-4

Making code human-friendly

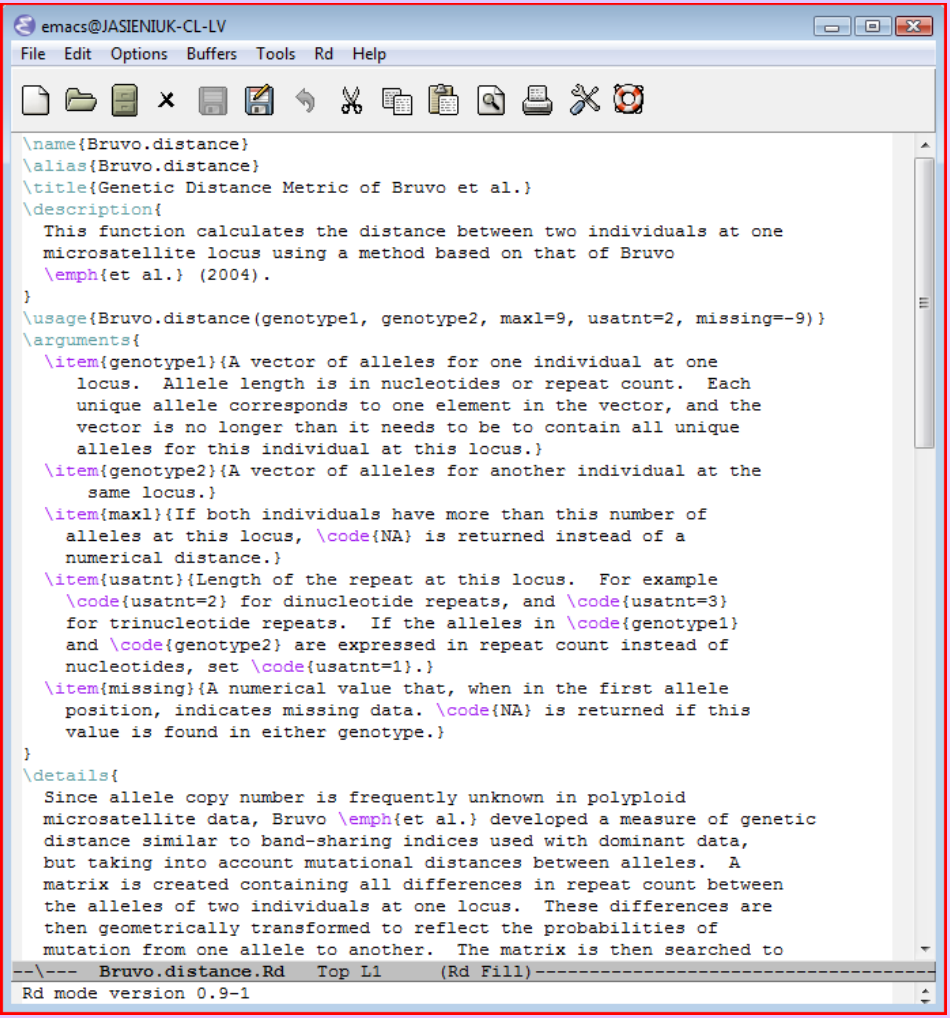
- In any programming language, always “comment” your code: make lines that aren’t read by the computer, but explain in English what the code is doing.
- Also write a separate document explaining what the code does and how to use it.
- Do this immediately; I regularly use my own documentation because I don’t have my software memorized.

Documenting an R package

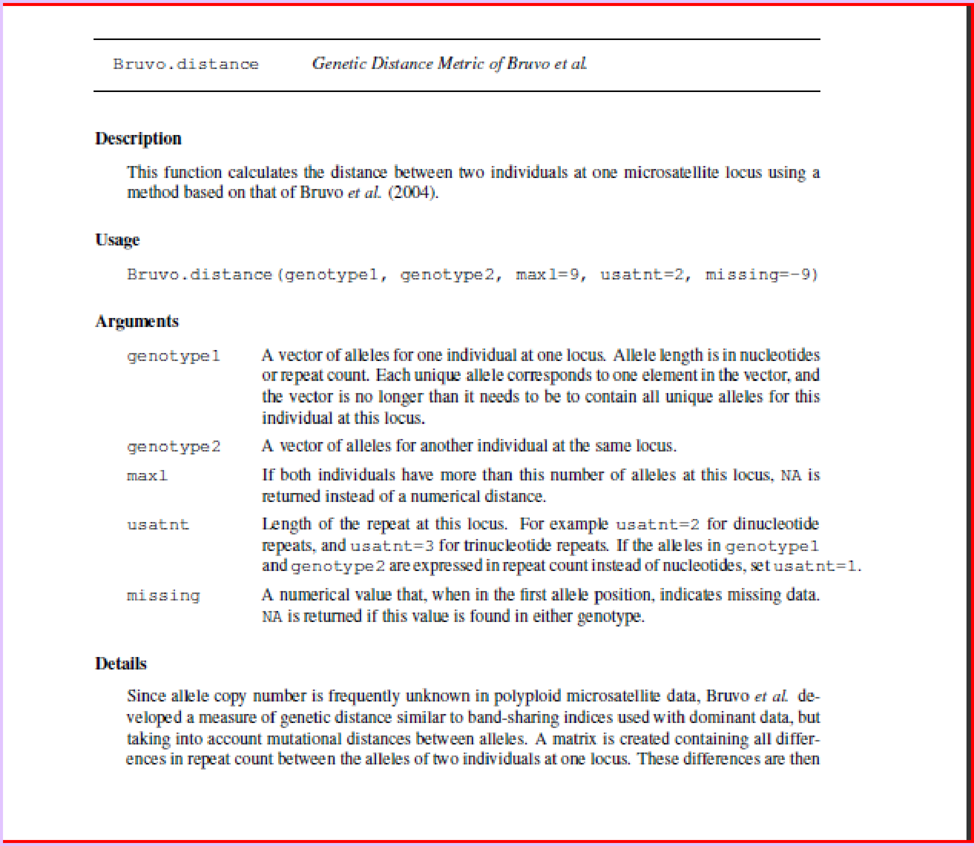
- Rd files: required for each function, follow a specific format, and are used to generate html, pdf, and text versions of help files.
- Optional free-form documentation
  - PDF that you make with MS Word
  - TeX/LaTeX: markup language to make professional-looking PDFs, great for mathematical formulas
  - Sweave: incorporate R code into a LaTeX document.

.Rd file

PDF version, made automatically



```
\name{Bruvo.distance}
\alias{Bruvo.distance}
\title{Genetic Distance Metric of Bruvo et al.}
\description{
  This function calculates the distance between two individuals at one
  microsatellite locus using a method based on that of Bruvo
  \emph{et al.} (2004).
}
\usage{Bruvo.distance(genotype1, genotype2, maxl=9, usatnt=2, missing=-9)}
\arguments{
  \item{genotype1}{A vector of alleles for one individual at one
     locus.  Allele length is in nucleotides or repeat count.  Each
     unique allele corresponds to one element in the vector, and the
     vector is no longer than it needs to be to contain all unique
     alleles for this individual at this locus.}
  \item{genotype2}{A vector of alleles for another individual at the
      same locus.}
  \item{maxl}{If both individuals have more than this number of
    alleles at this locus, \code{NA} is returned instead of a
    numerical distance.}
  \item{usatnt}{Length of the repeat at this locus.  For example
    \code{usatnt=2} for dinucleotide repeats, and \code{usatnt=3}
    for trinucleotide repeats.  If the alleles in \code{genotype1}
    and \code{genotype2} are expressed in repeat count instead of
    nucleotides, set \code{usatnt=1}.}
  \item{missing}{A numerical value that, when in the first allele
    position, indicates missing data. \code{NA} is returned if this
    value is found in either genotype.}
}
\details{
  Since allele copy number is frequently unknown in polyploid
  microsatellite data, Bruvo \emph{et al.} developed a measure of genetic
  distance similar to band-sharing indices used with dominant data,
  but taking into account mutational distances between alleles.  A
  matrix is created containing all differences in repeat count between
  the alleles of two individuals at one locus.  These differences are
  then geometrically transformed to reflect the probabilities of
  mutation from one allele to another.  The matrix is then searched to
--\---  Bruvo.distance.Rd    Top L1     (Rd Fill)------------------------------------
Rd mode version 0.9-1
```



`Bruvo.distance` *Genetic Distance Metric of Bruvo et al.*

Description

This function calculates the distance between two individuals at one microsatellite locus using a method based on that of Bruvo *et al.* (2004).

Usage

`Bruvo.distance(genotype1, genotype2, maxl=9, usatnt=2, missing=-9)`

Arguments

`genotype1` A vector of alleles for one individual at one locus. Allele length is in nucleotides or repeat count. Each unique allele corresponds to one element in the vector, and the vector is no longer than it needs to be to contain all unique alleles for this individual at this locus.

`genotype2` A vector of alleles for another individual at the same locus.

`maxl` If both individuals have more than this number of alleles at this locus, `NA` is returned instead of a numerical distance.

`usatnt` Length of the repeat at this locus. For example `usatnt=2` for dinucleotide repeats, and `usatnt=3` for trinucleotide repeats. If the alleles in `genotype1` and `genotype2` are expressed in repeat count instead of nucleotides, set `usatnt=1`.

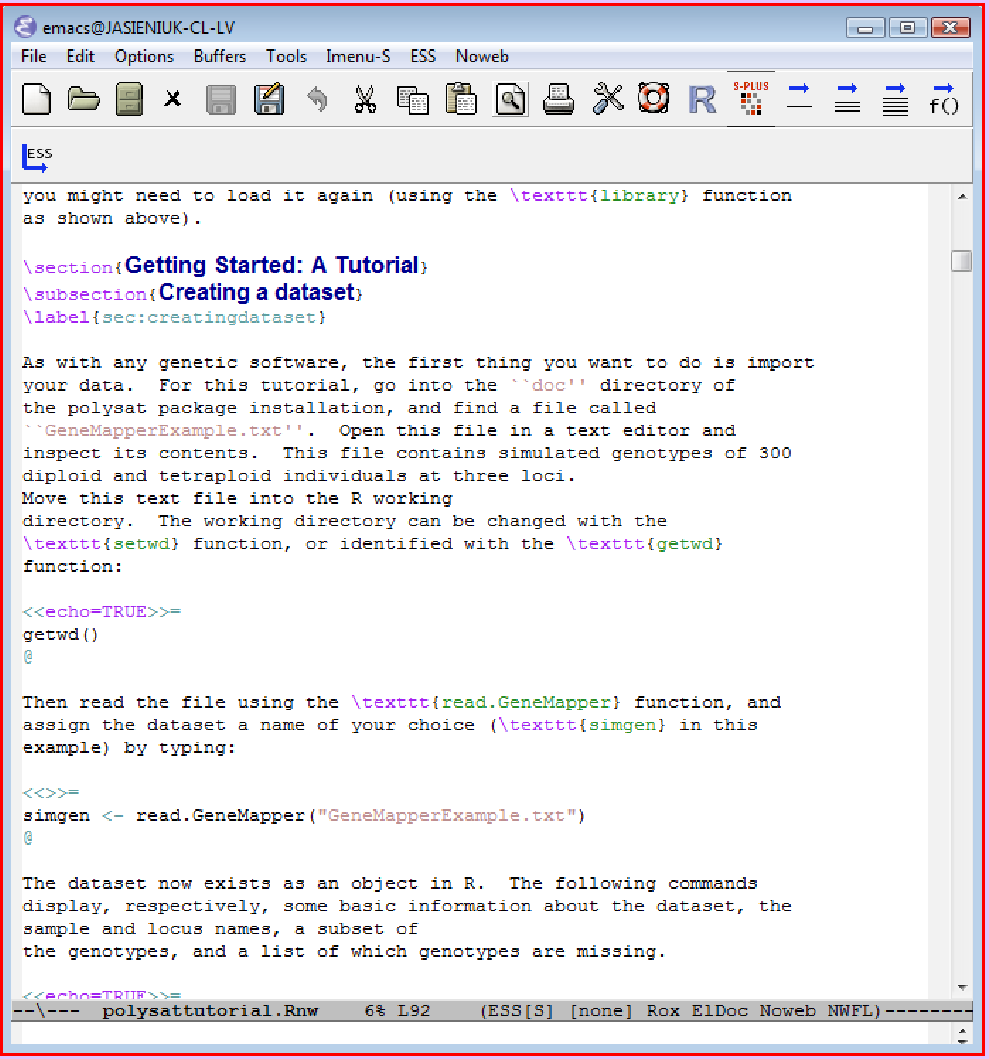
`missing` A numerical value that, when in the first allele position, indicates missing data. `NA` is returned if this value is found in either genotype.

Details

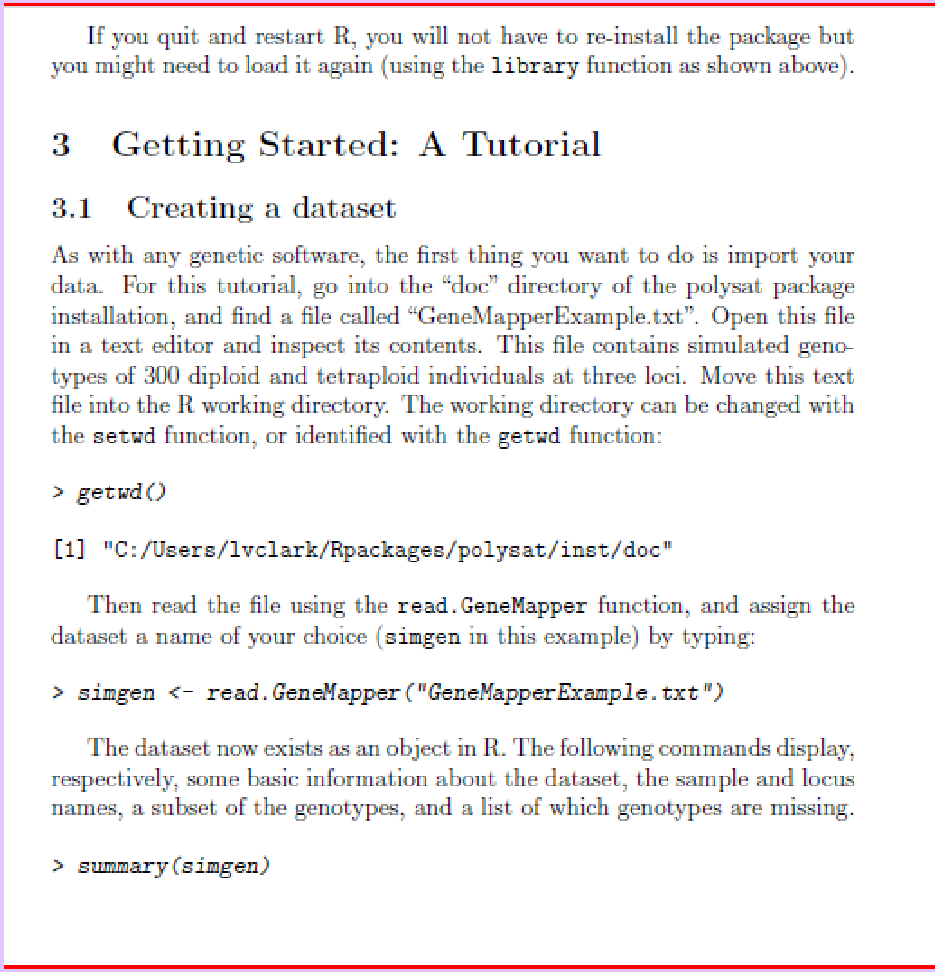
Since allele copy number is frequently unknown in polyploid microsatellite data, Bruvo *et al.* developed a measure of genetic distance similar to band-sharing indices used with dominant data, but taking into account mutational distances between alleles. A matrix is created containing all differences in repeat count between the alleles of two individuals at one locus. These differences are then

Sweave (.Rnw) file

PDF version



```
you might need to load it again (using the \texttt{library} function
as shown above).

\section{Getting Started: A Tutorial}
\subsection{Creating a dataset}
\label{sec:creatingdataset}

As with any genetic software, the first thing you want to do is import
your data.  For this tutorial, go into the ``doc'' directory of
the polysat package installation, and find a file called
``GeneMapperExample.txt''.  Open this file in a text editor and
inspect its contents.  This file contains simulated genotypes of 300
diploid and tetraploid individuals at three loci.
Move this text file into the R working
directory.  The working directory can be changed with the
\texttt{setwd} function, or identified with the \texttt{getwd}
function:

<<echo=TRUE>>=
getwd()
@

Then read the file using the \texttt{read.GeneMapper} function, and
assign the dataset a name of your choice (\texttt{simgen} in this
example) by typing:

<<>>=
simgen <- read.GeneMapper("GeneMapperExample.txt")
@

The dataset now exists as an object in R.  The following commands
display, respectively, some basic information about the dataset, the
sample and locus names, a subset of
the genotypes, and a list of which genotypes are missing.

<<echo=TRUE>>=
--\---  polysattutorial.Rnw    6% L92    (ESS[S] [none] Rox ElDoc Noweb NWFL)--------
```



If you quit and restart R, you will not have to re-install the package but you might need to load it again (using the `library` function as shown above).

3 Getting Started: A Tutorial

3.1 Creating a dataset

As with any genetic software, the first thing you want to do is import your data. For this tutorial, go into the "doc" directory of the polysat package installation, and find a file called "GeneMapperExample.txt". Open this file in a text editor and inspect its contents. This file contains simulated genotypes of 300 diploid and tetraploid individuals at three loci. Move this text file into the R working directory. The working directory can be changed with the `setwd` function, or identified with the `getwd` function:

```
> getwd()

[1] "C:/Users/lvclark/Rpackages/polysat/inst/doc"
```

Then read the file using the `read.GeneMapper` function, and assign the dataset a name of your choice (`simgen` in this example) by typing:

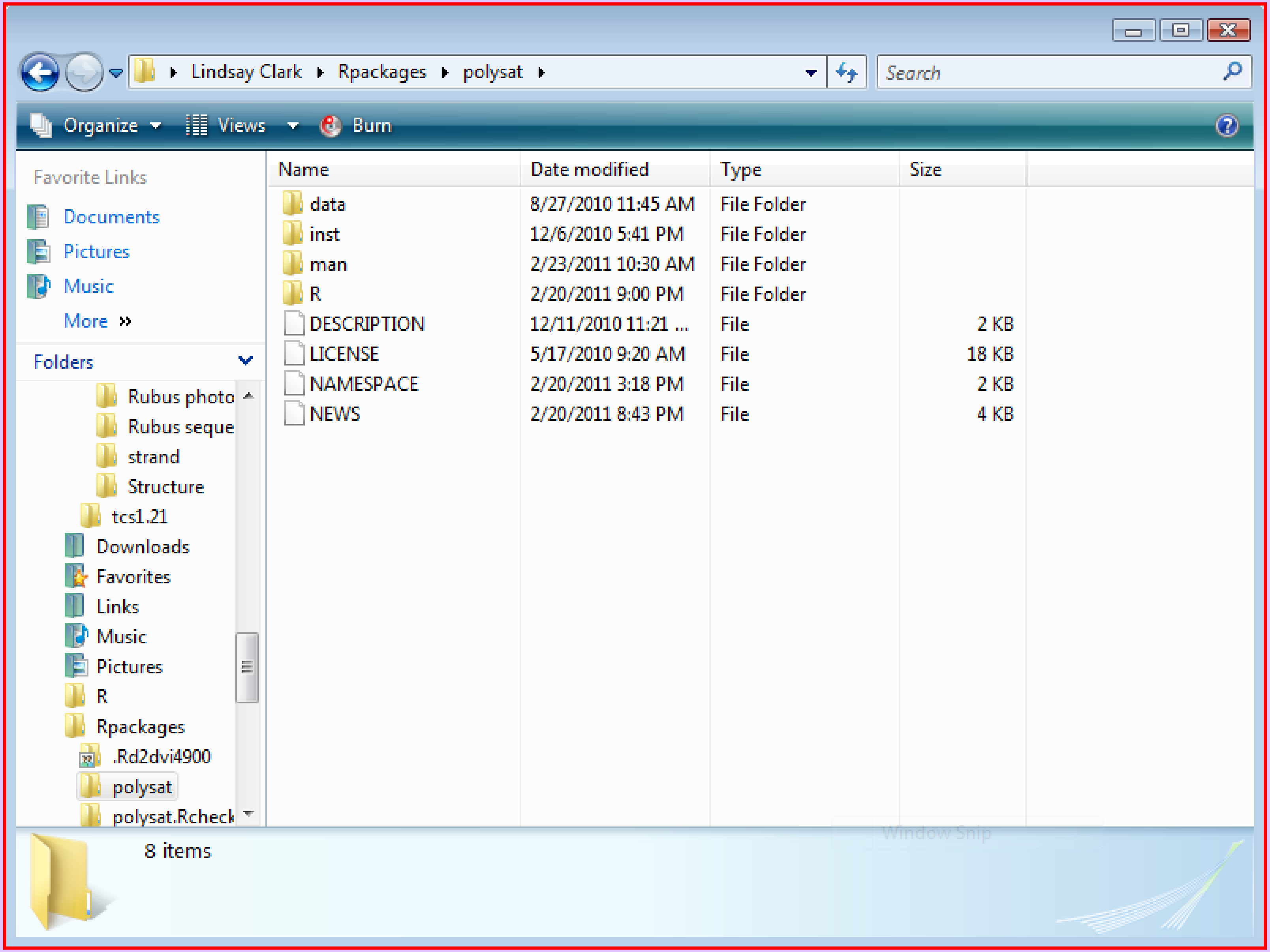
```
> simgen <- read.GeneMapper("GeneMapperExample.txt")
```

The dataset now exists as an object in R. The following commands display, respectively, some basic information about the dataset, the sample and locus names, a subset of the genotypes, and a list of which genotypes are missing.

```
> summary(simgen)
```

Arranging files to make a package

- Make a folder with the name of your package, and put sub-folders into it
- You are really just making folders of text files
- “R” folder: contains text files of R code
- “man” folder: contains .Rd files
- “DESCRIPTION” file
- Others optional, listed in manuals



Name	Date modified	Type	Size
data	8/27/2010 11:45 AM	File Folder	
inst	12/6/2010 5:41 PM	File Folder	
man	2/23/2011 10:30 AM	File Folder	
R	2/20/2011 9:00 PM	File Folder	
DESCRIPTION	12/11/2010 11:21 ...	File	2 KB
LICENSE	5/17/2010 9:20 AM	File	18 KB
NAMESPACE	2/20/2011 3:18 PM	File	2 KB
NEWS	2/20/2011 8:43 PM	File	4 KB

8 items

Making an Installable Package

- Some things must be done from the Unix Shell or Windows Command Prompt
  - *R CMD check*: Looks for problems with code and documentation
  - *R CMD build*: Makes compressed version of package for CRAN upload (or binaries for installation)
  - *R CMD Rd2txt, Rd2dvi*: Preview help files
- See: P. Rossi's "Making R Packages Under Windows: A Tutorial"
- "R Installation and Administration" manual

Benefits of open source

- Any R user can look at the source code for my package and check that it actually does what I claim it does – scientific integrity and openness
- Any R user can build upon what I have written, rather than having to create their own software from scratch

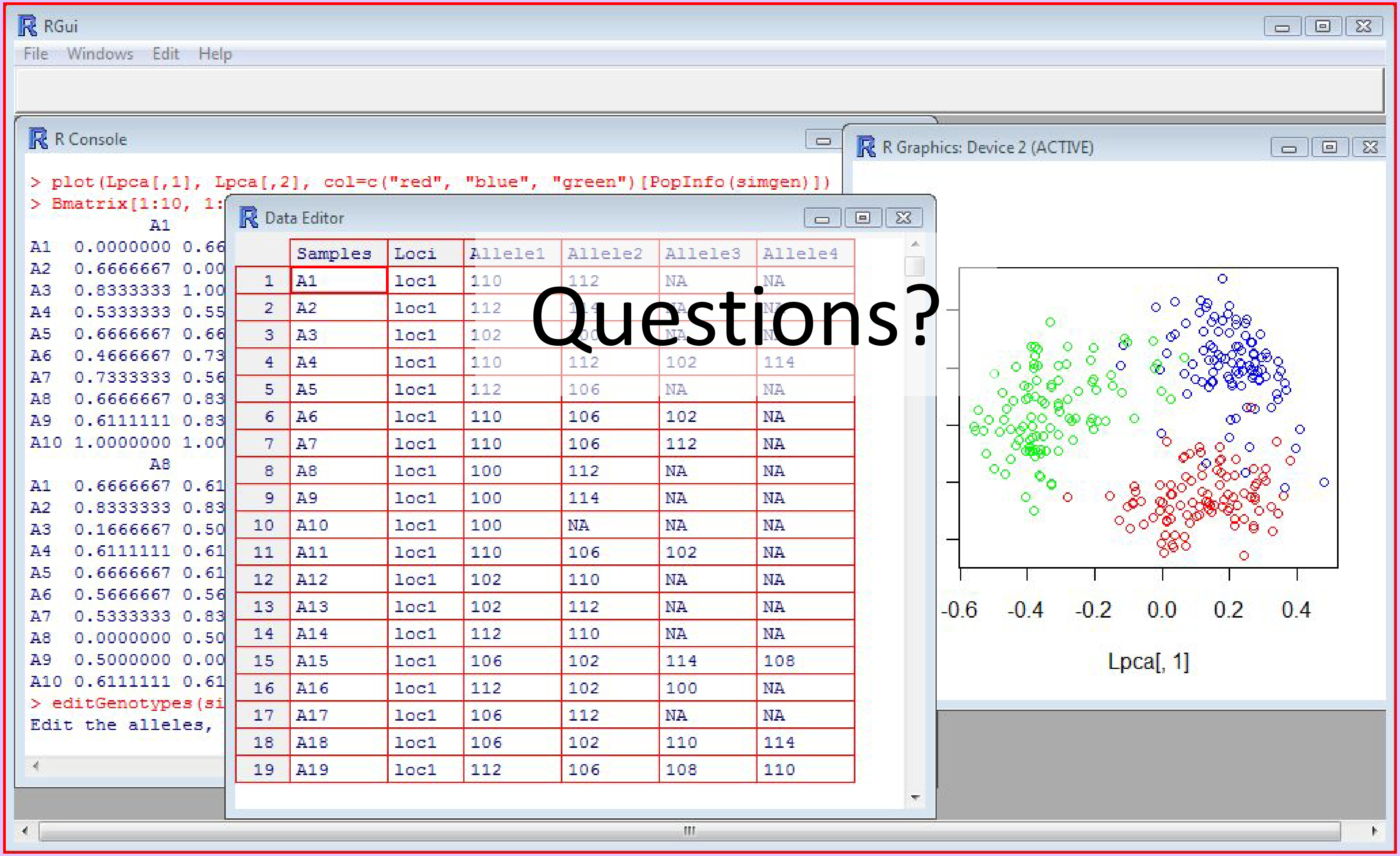
Benefits of archive networks like CRAN

- You do not need to host a website where users can download your software
- The whole archive is duplicated on many servers around the world
- Software is permanently available, including all previous versions
- Software and documentation is free to everyone, including developing world

Benefits of being able to write software

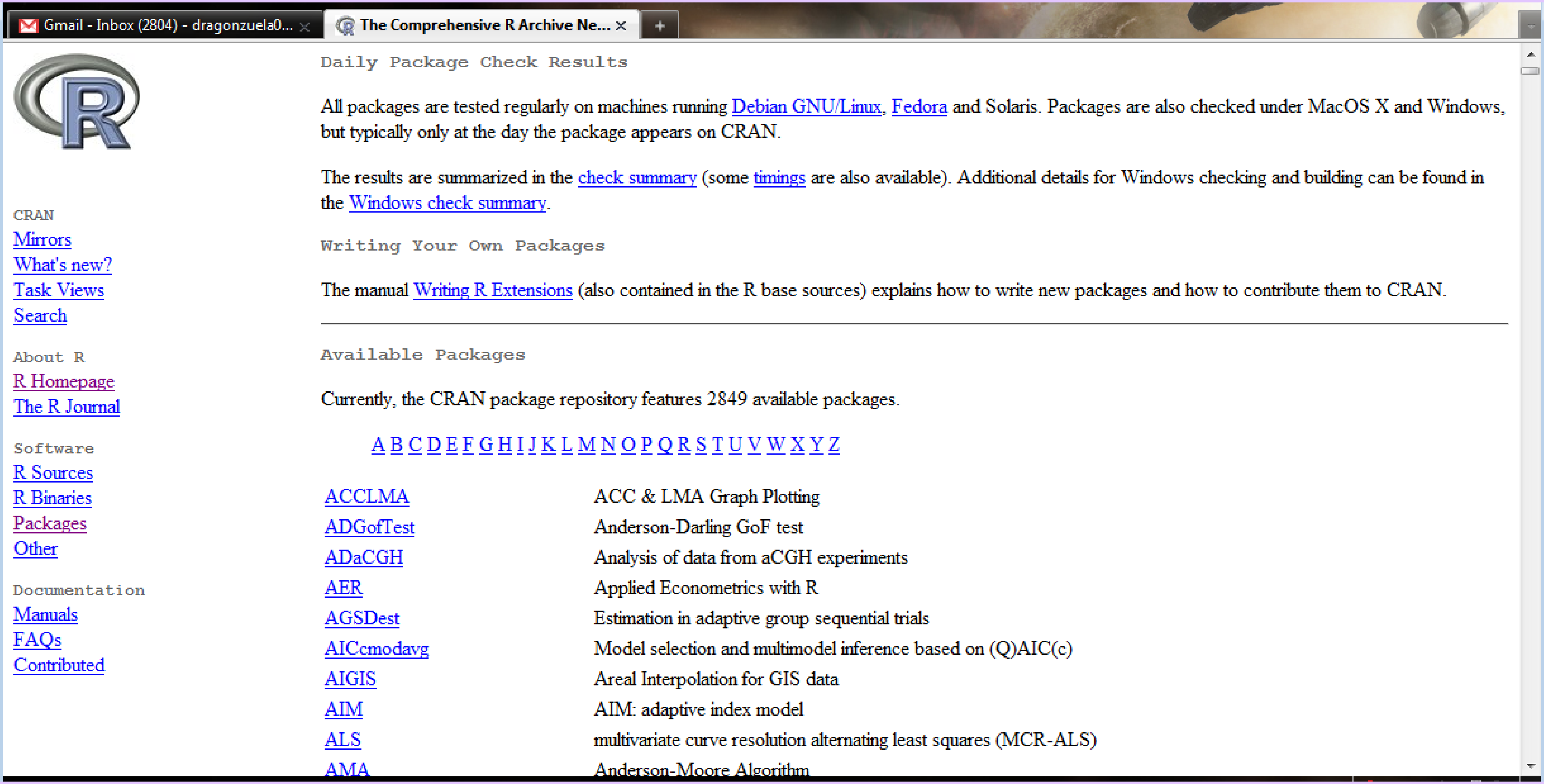
- Automate tasks that might be tedious by hand
- Create software that does exactly what you want it to
  - No hunting around for the right software
  - No issues converting to the right format
  - No compromising with software that doesn’t do exactly what you want
- If you are going to perform an analysis, you should have a deep understanding of how it works anyway

Questions?



```
> plot(Lpca[,1], Lpca[,2], col=c("red", "blue", "green")[PopInfo(simgen)])
> Bmatrix[1:10, 1:
              A1
A1  0.0000000 0.66
A2  0.6666667 0.00
A3  0.8333333 1.00
A4  0.5333333 0.55
A5  0.6666667 0.66
A6  0.4666667 0.73
A7  0.7333333 0.56
A8  0.6666667 0.83
A9  0.6111111 0.83
A10 1.0000000 1.00
              A8
A1  0.6666667 0.61
A2  0.8333333 0.83
A3  0.1666667 0.50
A4  0.6111111 0.61
A5  0.6666667 0.61
A6  0.5666667 0.56
A7  0.5333333 0.83
A8  0.0000000 0.50
A9  0.5000000 0.00
A10 0.6111111 0.61
> editGenotypes(si
Edit the alleles,
```

	Samples	Loci	Allele1	Allele2	Allele3	Allele4
1	A1	loc1	110	112	NA	NA
2	A2	loc1	112	114	NA	NA
3	A3	loc1	102	100	NA	NA
4	A4	loc1	110	112	102	114
5	A5	loc1	112	106	NA	NA
6	A6	loc1	110	106	102	NA
7	A7	loc1	110	106	112	NA
8	A8	loc1	100	112	NA	NA
9	A9	loc1	100	114	NA	NA
10	A10	loc1	100	NA	NA	NA
11	A11	loc1	110	106	102	NA
12	A12	loc1	102	110	NA	NA
13	A13	loc1	102	112	NA	NA
14	A14	loc1	112	110	NA	NA
15	A15	loc1	106	102	114	108
16	A16	loc1	112	102	100	NA
17	A17	loc1	106	112	NA	NA
18	A18	loc1	106	102	110	114
19	A19	loc1	112	106	108	110



Daily Package Check Results

All packages are tested regularly on machines running Debian GNU/Linux, Fedora and Solaris. Packages are also checked under MacOS X and Windows, but typically only at the day the package appears on CRAN.

The results are summarized in the check summary (some timings are also available). Additional details for Windows checking and building can be found in the Windows check summary.

Writing Your Own Packages

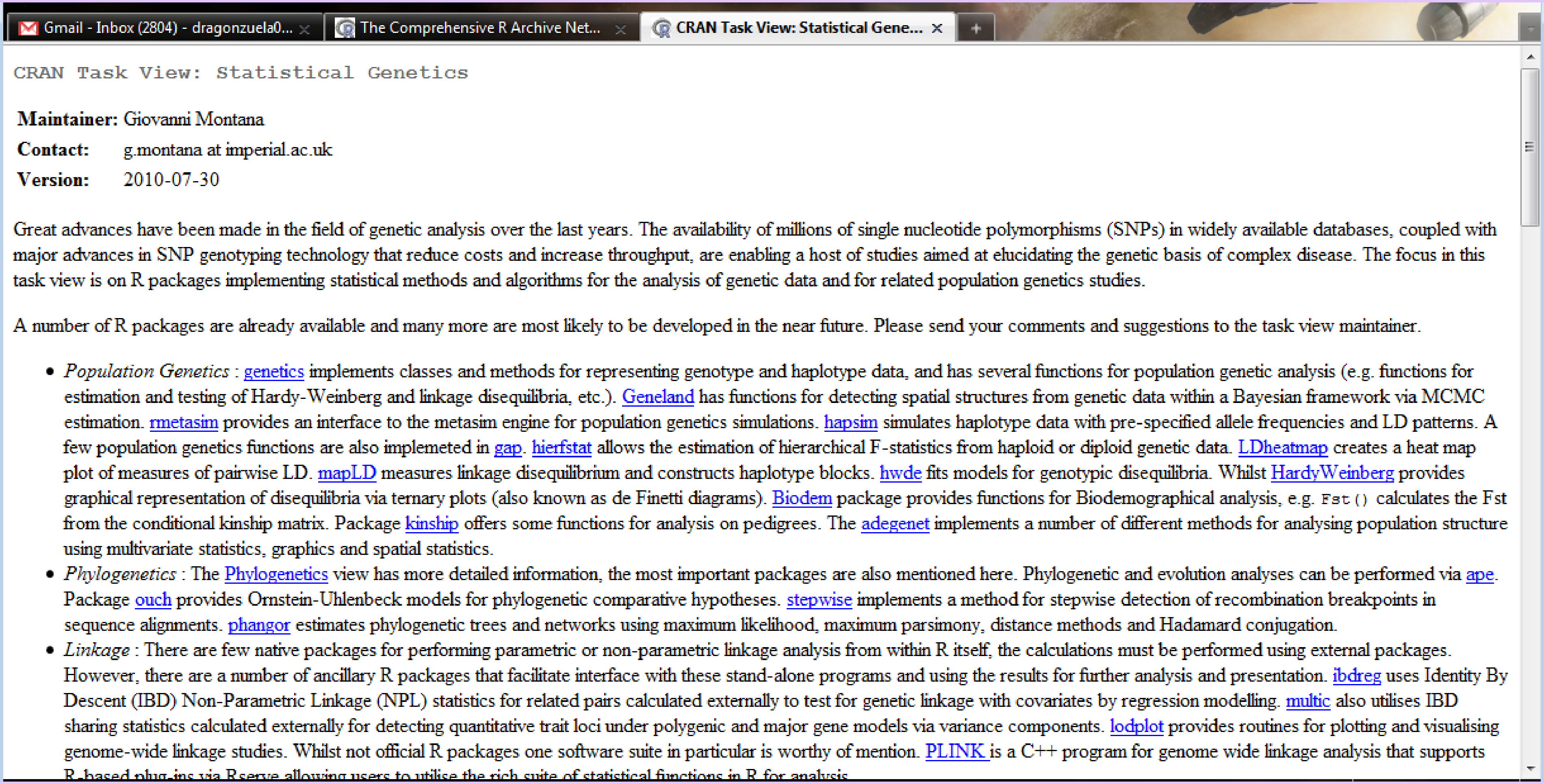
The manual Writing R Extensions (also contained in the R base sources) explains how to write new packages and how to contribute them to CRAN.

Available Packages

Currently, the CRAN package repository features 2849 available packages.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

ACCLMA	ACC & LMA Graph Plotting
ADGofTest	Anderson-Darling GoF test
ADaCGH	Analysis of data from aCGH experiments
AER	Applied Econometrics with R
AGSDest	Estimation in adaptive group sequential trials
AICcmodavg	Model selection and multimodel inference based on (Q)AIC(c)
AIGIS	Areal Interpolation for GIS data
AIM	AIM: adaptive index model
ALS	multivariate curve resolution alternating least squares (MCR-ALS)
AMA	Anderson-Moore Algorithm


Gmail - Inbox (2804) - dragonzuela0...
The Comprehensive R Archive Net...
CRAN Task View: Statistical Gene...

CRAN Task View: Statistical Genetics

Maintainer: Giovanni Montana
Contact: g.montana at imperial.ac.uk
Version: 2010-07-30

Great advances have been made in the field of genetic analysis over the last years. The availability of millions of single nucleotide polymorphisms (SNPs) in widely available databases, coupled with major advances in SNP genotyping technology that reduce costs and increase throughput, are enabling a host of studies aimed at elucidating the genetic basis of complex disease. The focus in this task view is on R packages implementing statistical methods and algorithms for the analysis of genetic data and for related population genetics studies.

A number of R packages are already available and many more are most likely to be developed in the near future. Please send your comments and suggestions to the task view maintainer.

- *Population Genetics* : genetics implements classes and methods for representing genotype and haplotype data, and has several functions for population genetic analysis (e.g. functions for estimation and testing of Hardy-Weinberg and linkage disequilibria, etc.). Geneland has functions for detecting spatial structures from genetic data within a Bayesian framework via MCMC estimation. rmetasim provides an interface to the metasim engine for population genetics simulations. hapsim simulates haplotype data with pre-specified allele frequencies and LD patterns. A few population genetics functions are also implemeted in gap. hierfstat allows the estimation of hierarchical F-statistics from haploid or diploid genetic data. LDheatmap creates a heat map plot of measures of pairwise LD. mapLD measures linkage disequilibrium and constructs haplotype blocks. hwde fits models for genotypic disequilibria. Whilst HardyWeinberg provides graphical representation of disequilibria via ternary plots (also known as de Finetti diagrams). Biodem package provides functions for Biodemographical analysis, e.g. `Fst()` calculates the Fst from the conditional kinship matrix. Package kinship offers some functions for analysis on pedigrees. The adegenet implements a number of different methods for analysing population structure using multivariate statistics, graphics and spatial statistics.
- *Phylogenetics* : The Phylogenetics view has more detailed information, the most important packages are also mentioned here. Phylogenetic and evolution analyses can be performed via ape. Package ouch provides Ornstein-Uhlenbeck models for phylogenetic comparative hypotheses. stepwise implements a method for stepwise detection of recombination breakpoints in sequence alignments. phangor estimates phylogenetic trees and networks using maximum likelihood, maximum parsimony, distance methods and Hadamard conjugation.
- *Linkage* : There are few native packages for performing parametric or non-parametric linkage analysis from within R itself, the calculations must be performed using external packages. However, there are a number of ancillary R packages that facilitate interface with these stand-alone programs and using the results for further analysis and presentation. ibdreg uses Identity By Descent (IBD) Non-Parametric Linkage (NPL) statistics for related pairs calculated externally to test for genetic linkage with covariates by regression modelling. multic also utilises IBD sharing statistics calculated externally for detecting quantitative trait loci under polygenic and major gene models via variance components. lodplot provides routines for plotting and visualising genome-wide linkage studies. Whilst not official R packages one software suite in particular is worthy of mention. PLINK is a C++ program for genome wide linkage analysis that supports R-based plug-ins via Rserve allowing users to utilise the rich suite of statistical functions in R for analysis.