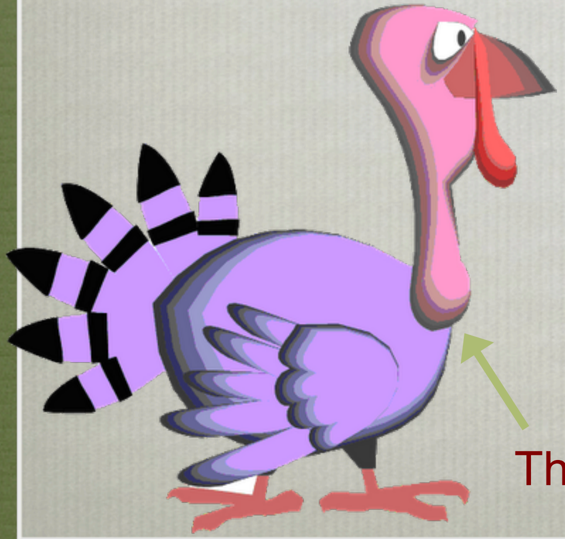
Help! My Software Has Turned Into A Techno Turkey!

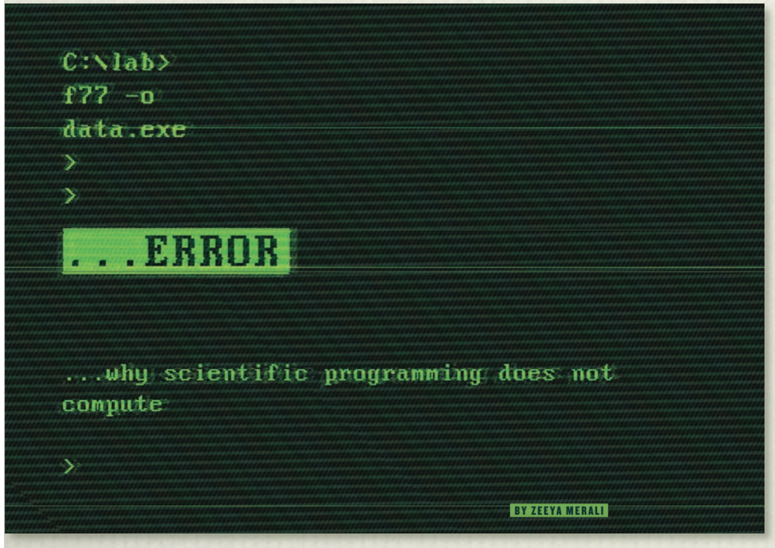


This is not Bruce.

G. Bruce Berriman

GRITS, June 17, 2011

Who Are Your Influences?



```
C:\lab>
f77 -o
data.exe
>
>
...ERROR


...why scientific programming does not
compute

>
```

BY ZEEYA MERALI

“Why Scientific Programming Does Not Compute.”

Zeeya Merali.
Nature, 467, 777. October 2010

May 3-5, 2011. NRAO, Green Bank, West Virginia.
http://www.nrao.edu/meetings/bigdata/
Workshop On How To Process and Analyze PB-scale data sets.

A Little Knowledge …

- Many scientists in all fields have little formal training in software development and software maintenance.
- Many of us learn from our peers or by modifying existing code.
- Many of us learn just enough to be dangerous …

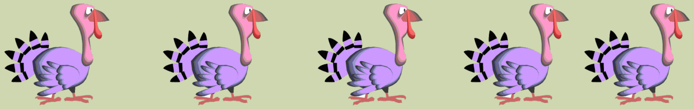
EXAMPLE: Removing a record from a database

Instead of using a simple SQL command to do this

```
DELETE FROM table1 WHERE field1=value1
```

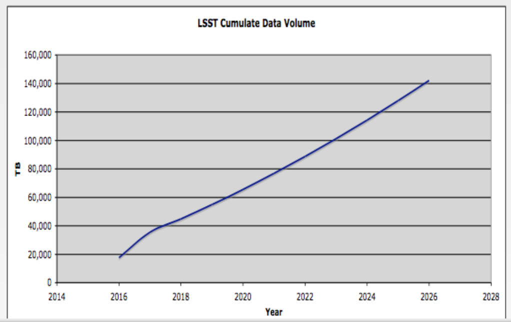
The project dumps the database as a file, uses Unix commands to identify and remove the offending record, then reload the file into the database

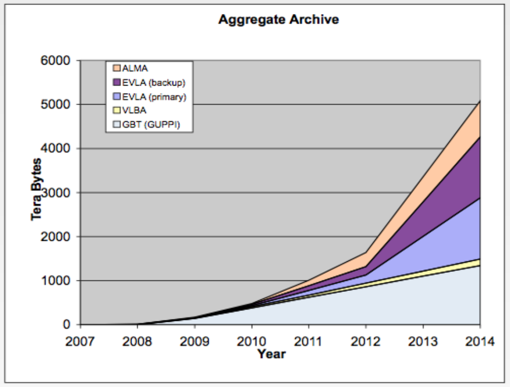
```
pg dump -t table1 mydb | grep -v value1 | pg restore -c mydb
```



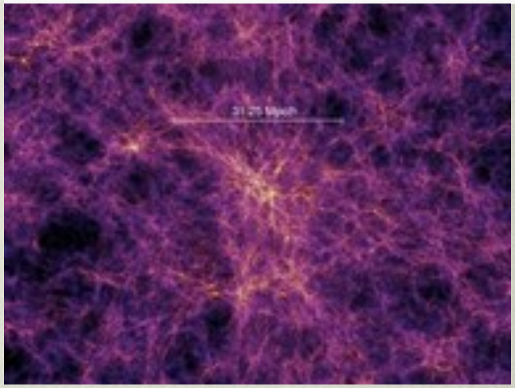
Chilingarian and Zolotukhin, 2011

A New Business Model For Astronomical Computing





- Astronomy is already a data intensive science
  - Over 1 PB of data served electronically through data centers and archives. Growing at 0.5 PB/yr, and accelerating.
  - ALMA, LOFAR, LSST, SKA, EVLA… all will produce **PB-scale data sets**
    - LSST alone may have 60 PB data by 2020



- Simulations for design of observing programs and confrontation with data
  - Millennium Simulation - N-body simulation, 10 billion particles trace the evolution of the matter distribution in a cubic region of the Universe over 2 billion light-years on a side.
- Astro2010 Decadal Survey recognized that future research will demand high performance computing

Practicing Safe Software

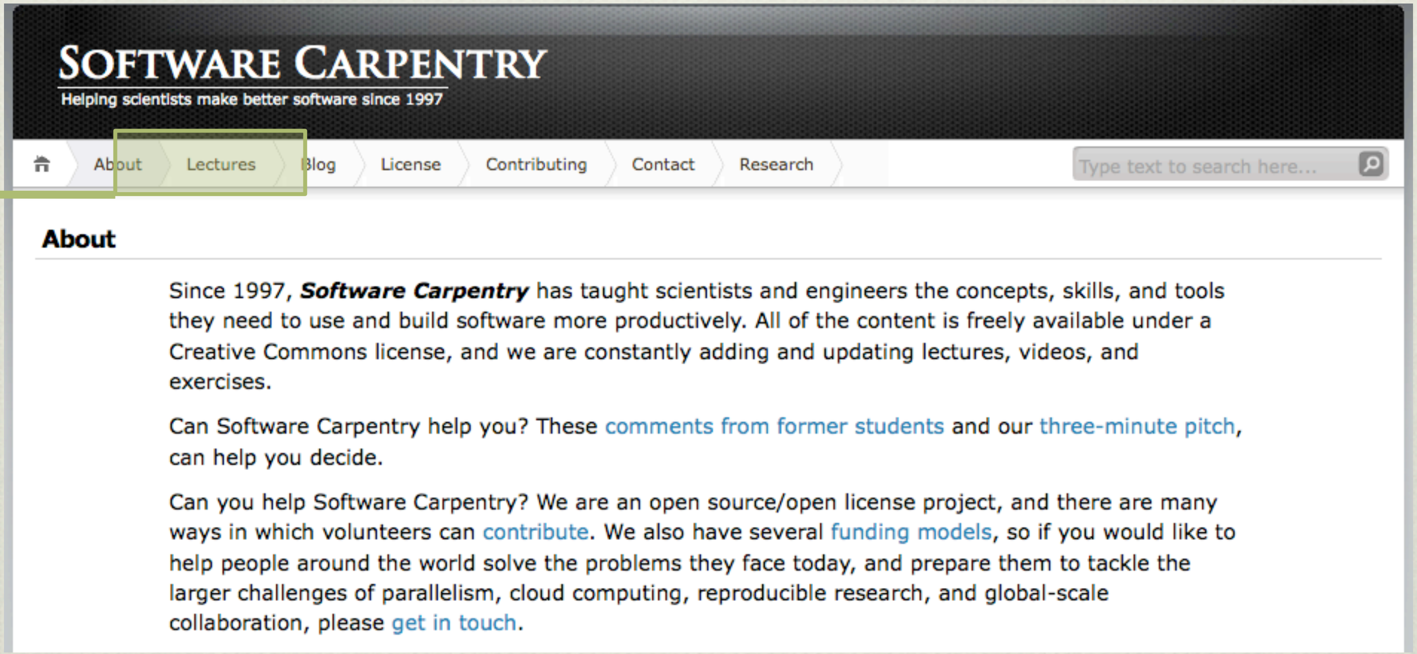
- Respondents spent an average of 30% of work time developing S/W.
- 45% spend much more time developing S/W five years ago.
- 97% said informal self-study important.
- 26% thought formal S/W education important.
- 8% used a high performance platform.

Hanney et al. ***How Do Scientists Develop And Use Software?*** *Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering.*

- Use version control
- Track materials
- Write testable software
  - **Build code in chunks**
- .. And test it! And get someone else to use it.
- Share software and **build a user community** where feasible.

Software Carpentry

http://software-carpentry.org/

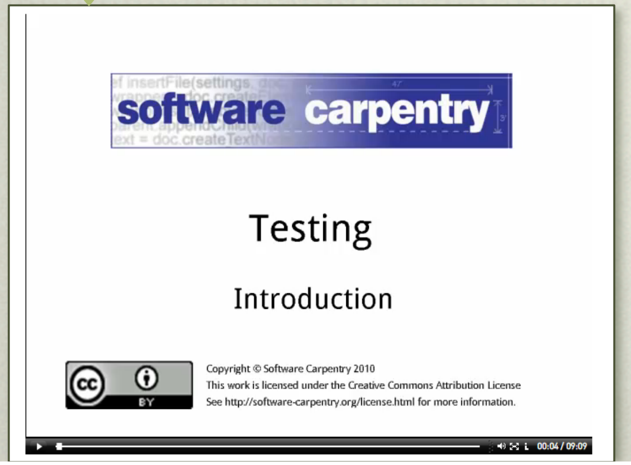

SOFTWARE CARPENTRY
Helping scientists make better software since 1997
About
Lectures
Blog
License
Contributing
Contact
Research
Type text to search here...

About

Since 1997, ***Software Carpentry*** has taught scientists and engineers the concepts, skills, and tools they need to use and build software more productively. All of the content is freely available under a Creative Commons license, and we are constantly adding and updating lectures, videos, and exercises.

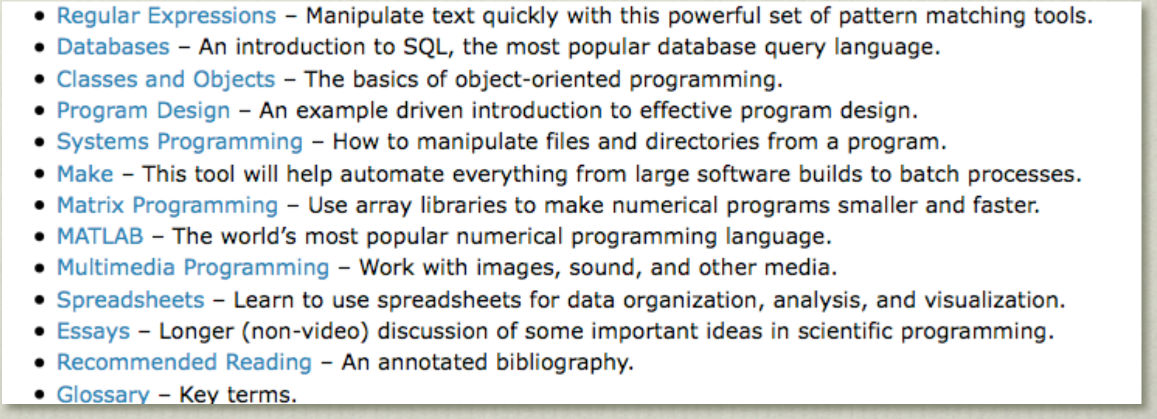
Can Software Carpentry help you? These comments from former students and our three-minute pitch, can help you decide.

Can you help Software Carpentry? We are an open source/open license project, and there are many ways in which volunteers can contribute. We also have several funding models, so if you would like to help people around the world solve the problems they face today, and prepare them to tackle the larger challenges of parallelism, cloud computing, reproducible research, and global-scale collaboration, please get in touch.



Testing

Introduction



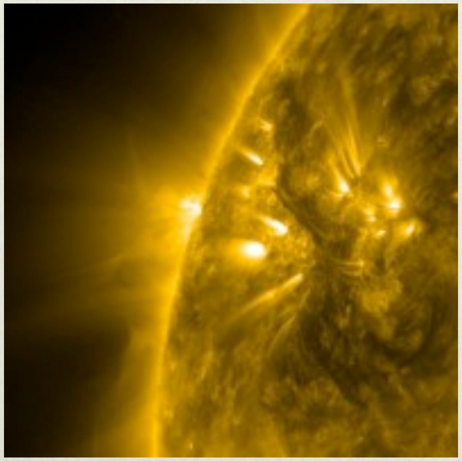
- Regular Expressions – Manipulate text quickly with this powerful set of pattern matching tools.
- Databases – An introduction to SQL, the most popular database query language.
- Classes and Objects – The basics of object-oriented programming.
- Program Design – An example driven introduction to effective program design.
- Systems Programming – How to manipulate files and directories from a program.
- Make – This tool will help automate everything from large software builds to batch processes.
- Matrix Programming – Use array libraries to make numerical programs smaller and faster.
- MATLAB – The world's most popular numerical programming language.
- Multimedia Programming – Work with images, sound, and other media.
- Spreadsheets – Learn to use spreadsheets for data organization, analysis, and visualization.
- Essays – Longer (non-video) discussion of some important ideas in scientific programming.
- Recommended Reading – An annotated bibliography.
- Glossary – Key terms.

Next Steps for Astronomy As A Profession

- Software and computer science as a mandatory part of graduate studies.
- Have scientists work closely with IT professionals
  - Highly successful model used at many organizations, including IPAC since it opened for business
- Greater recognition of the role of software engineering
  - Provide career-paths for IT professionals in astronomy
  - An on-line journal devoted to computational techniques in astronomy.
- Develop “software brain trusts” to share computational knowledge from different fields.

U.K. Software Sustainability Institute

http://www.software.ac.uk

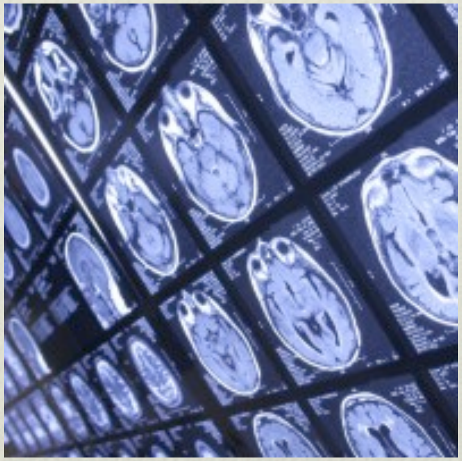


Nuclear Fusion - Culham Centre for Fusion Energy

Pharmacology - DMACRYS

Climate change - Enhancing Community Integrated Assessment

Geospatial Information



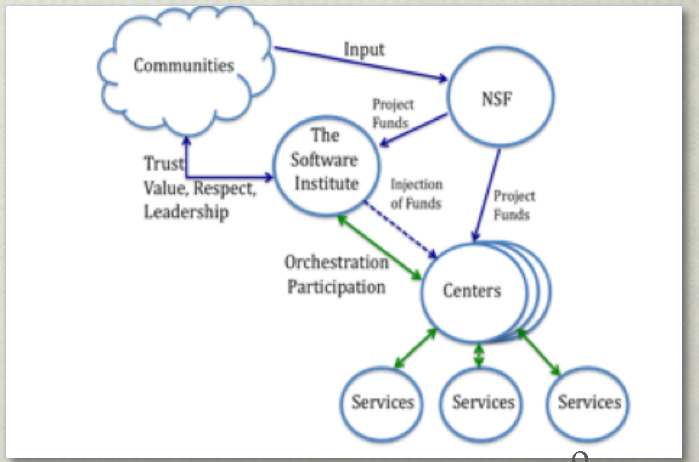
Scottish Brain Imaging Research Centre

Keeping up to date with research

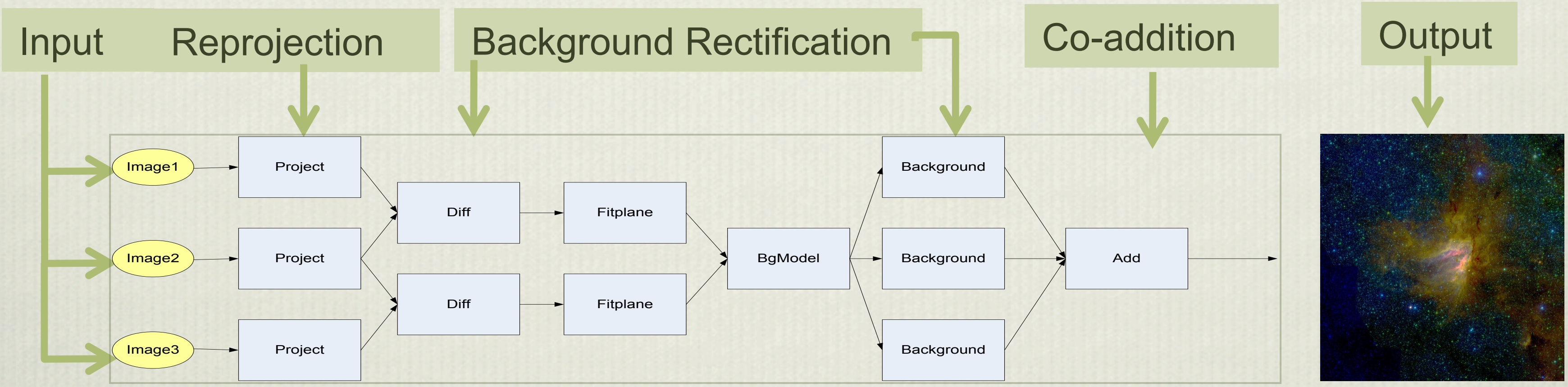
A U.S. Software Sustainability Institute: A Brain Trust For Software

“A US Software Infrastructure Institute that provides **a national center of excellence for community based software architecture, design and production**; expertise and services in support of software life cycle practices; marketing, documentation and networking services; and transformative workforce development activities.”

Report from the *Workshops on Distributed Computing, Multidisciplinary Science, and the NSF’s Scientific Software Innovation Institutes Program* Miron Livny, Ian Foster, Ruth Pordes, Scott Koranda, JP Navarro. August 2010.



Montage: An Example Of Sharable Component Based Software



Montage Workflow

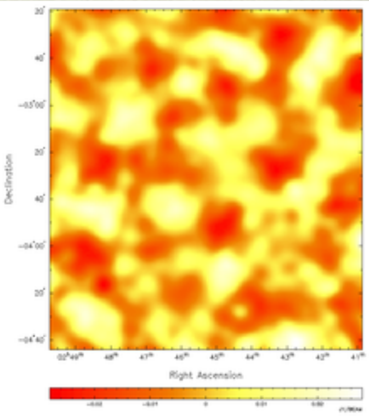
- ❖ Downloaded 5,000 times with wide applicability in astronomy and computer science.
- ❖ Simple to build.
- ❖ Written in ANSI-C for performance and portability.
- ❖ Portable to all flavors of *nix
- ❖ Environment agnostic
- ❖ Technology Agnostic: Supports tools such as Pegasus, MPI, .. Same code runs on all platforms.

See **"Ten Years of Software Sustainability"**. Berriman et al. 2011. Philosophical Transactions of the Royal Society, in press.

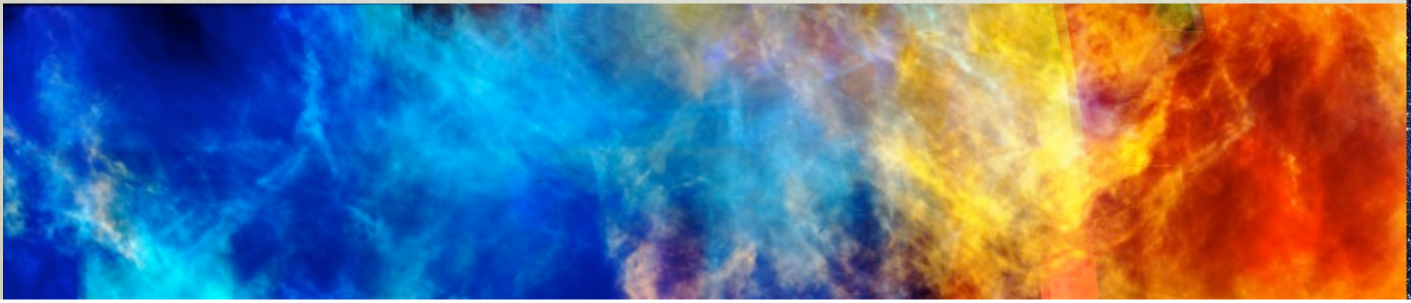
Applications of Montage: Science Analysis

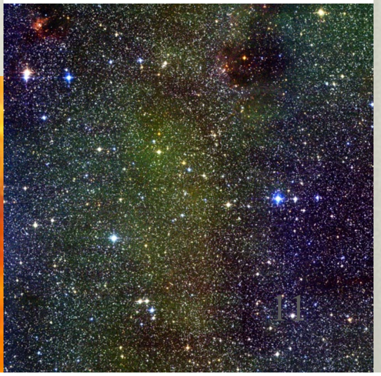
- Desktop research tool – astronomers sharing their scripts
  - Python interface to Montage (Tom Robitaille)
  - C-shell scripts to produce 2MASS, SDSS, DSS mosaics (Colin Aspin)

- Incorporation into pipelines
  - Cosmic Background Imager
  - ALFALFA
  - BOLOCAM

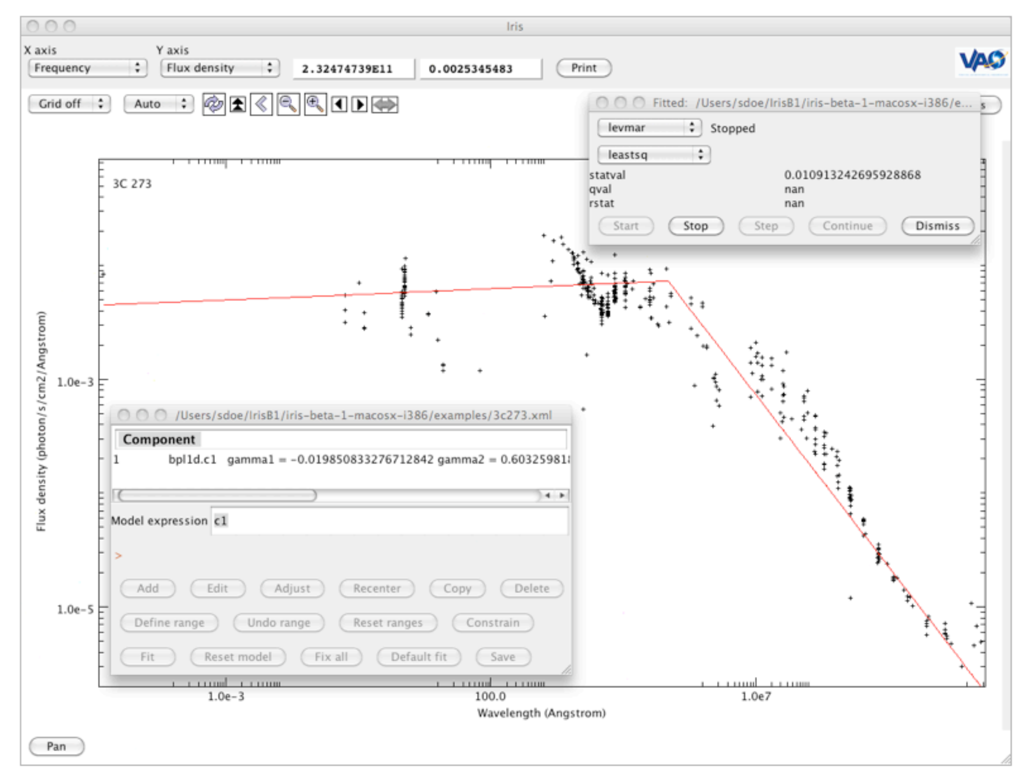


1,500-square-degree-equal-area Aitoff projection mosaic, of HI observed with (ALFALFA) survey near the North Galactic Pole (NGP). *Dr Brian Kent*





Plugging Together Applications



Iris

X axis Frequency

Y axis Flux density

2.32474739E11 0.0025345483

Fitted: /Users/sdoe/IrisB1/iris-beta-1-macosx-i386/e...

levmar Stopped

leastsq

```
statval                     0.010913242695928868
qval                        nan
rstat                       nan
```

/Users/sdoe/IrisB1/iris-beta-1-macosx-i386/examples/3c273.xml

Component
1 bpl1d.c1 gamma1 = -0.019850833276712842 gamma2 = 0.60325981

Model expression c1

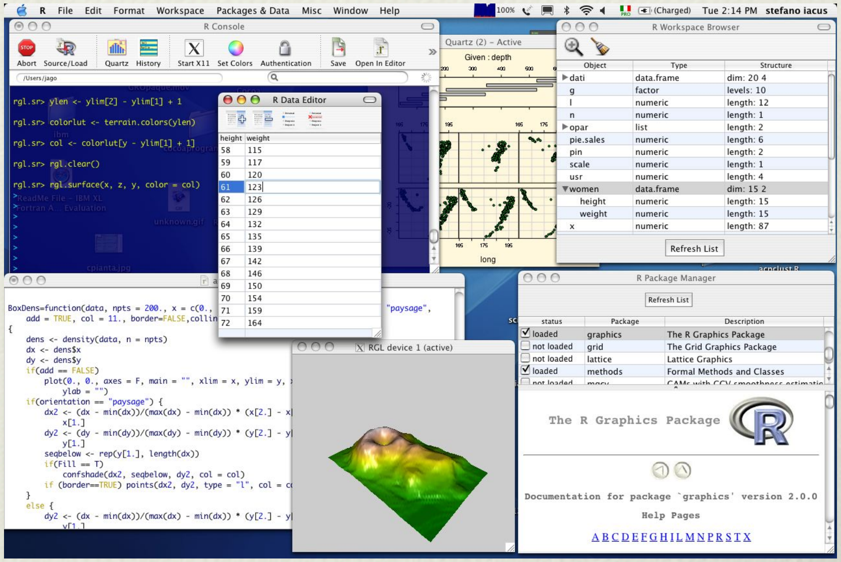
Plugged together using the
Simple Access Messaging Protocol

- VAO Spectral Energy Distribution Builder
- First Release Aug 2011
- Plugs together SpecView and Sherpa
- SpecView: Interactive Visualization of Spectra (STScI)
- Sherpa: Modeling and fitting (Chandra)

Code Sharing and Building Communities

The R Project for Statistical Computing

- “An environment where statistical techniques are implemented and extended”


R File Edit Format Workspace Packages & Data Misc Window Help
100% (Charged) Tue 2:14 PM stefano iacus
R Console
Abort Source/Load Quartz History Start X11 Set Colors Authentication Save Open In Editor
/Users/jago
Quartz (2) – Active
Given : depth
long
R Data Editor
R Workspace Browser
Refresh List
R Package Manager
RGL device 1 (active)

```
rgl.sr> ylen <- ylim[2] - ylim[1] + 1

rgl.sr> colorlut <- terrain.colors(ylen)

rgl.sr> col <- colorlut[y - ylim[1] + 1]

rgl.sr> rgl.clear()

rgl.sr> rgl.surface(x, z, y, color = col)
>
>
>
>
>
>
>
>
```

height	weight
58	115
59	117
60	120
61	123
62	126
63	129
64	132
65	135
66	139
67	142
68	146
69	150
70	154
71	159
72	164

Object	Type	Structure
▶ dati	data.frame	dim: 20 4
g	factor	levels: 10
l	numeric	length: 12
n	numeric	length: 1
▶ opar	list	length: 2
pie.sales	numeric	length: 6
pin	numeric	length: 2
scale	numeric	length: 1
usr	numeric	length: 4
▼ women	data.frame	dim: 15 2
height	numeric	length: 15
weight	numeric	length: 15
x	numeric	length: 87

```
BoxDens=function(data, npts = 200., x = c(0.,                "paysage",
    add = TRUE, col = 11., border=FALSE,collin
{
    dens <- density(data, n = npts)
    dx <- dens$x
    dy <- dens$y
    if(add == FALSE)
        plot(0., 0., axes = F, main = "", xlim = x, ylim = y,
            ylab = "")
    if(orientation == "paysage") {
        dx2 <- (dx - min(dx))/(max(dx) - min(dx)) * (x[2.] - x
            x[1.]
        dy2 <- (dy - min(dy))/(max(dy) - min(dy)) * (y[2.] - y
            y[1.]
        seqbelow <- rep(y[1.], length(dx))
        if(Fill == T)
            confshade(dx2, seqbelow, dy2, col = col)
        if (border==TRUE) points(dx2, dy2, type = "l", col = c
    }
    else {
        dy2 <- (dx - min(dx))/(max(dx) - min(dx)) * (y[2.] - y
```

status	Package	Description
☑ loaded	graphics	The R Graphics Package
□ not loaded	grid	The Grid Graphics Package
□ not loaded	lattice	Lattice Graphics
☑ loaded	methods	Formal Methods and Classes
□ not loaded	mgcv	GAMs with GCV smoothness estimatio

The R Graphics Package

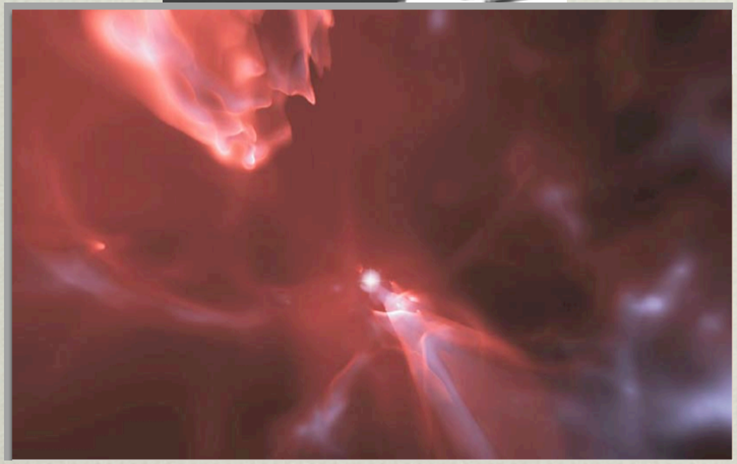
Documentation for package `graphics' version 2.0.0

Help Pages

A B C D E F G H I L M N P R S T X

http://www.r-project.org/index.html

enzo



ENZO

- Adaptive mesh refinement (AMR), grid-based hybrid code (hydro + N-Body) for cosmological simulations

http://code.google.com/p/enzo/

Conclusions

- Massive data sets are driving a new business model for scientific computing.

- The computationally self-taught scientist working at a desktop will be at a big disadvantage in this new world.

- Software components that are portable and scalable will have a much bigger role to play in the future.

- I think we need more formal computer education, and a cultural change to reward computational skills..

Where Can I Learn More?

- **ERROR … Why Scientific Programming Does Not Compute.** 2010. Zeeya Merali. Nature, 467, 775.

- **Articles on the Software Carpentry Site:**
  How Do Scientists Really Use Computers?
  How Do Scientists Develop and Use Scientific Software
  Those Who Will Not Learn From History
  Getting Scientists to Write Better Code To Make Them More Productive
  Where’ s the Real Bottleneck in Scientific Computing?

- **Ten Years of Software Sustainability**. G. B. Berriman et al. 2011. Philosophical Transactions of the Royal Society A, in press.

- **The True Bottleneck of Modern Scientific Computing in Astronomy**. 2011**.** Igor Chilingarian and Ivan Zolotukhin. ADASS XX, 471. http://arxiv.org/abs/1012.4119v1.

- Bruce Berriman’ s blog, “Astronomy Computing Today,” at **http://astrocompute.wordpress.com**