Data Analysis using the R Project for Statistical Computing

Daniela Ushizima

NERSC Analytics

Lawrence Berkeley National Laboratory

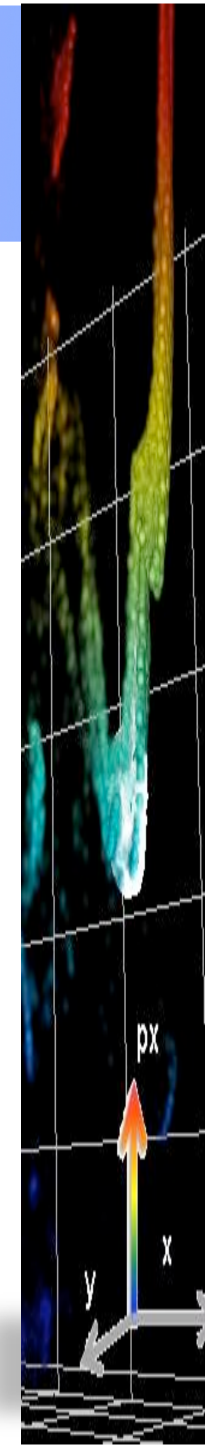
Outline

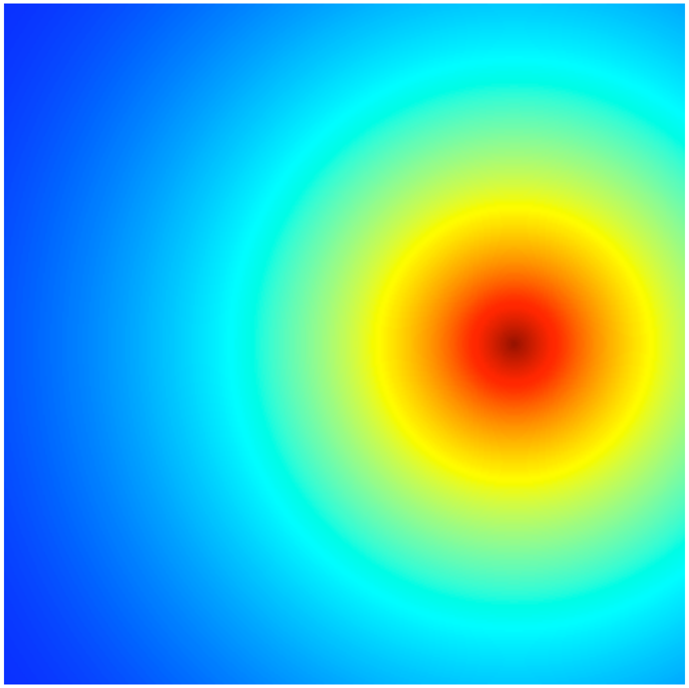
I. R-programming
   - Why to use R
   - R in the scientific community
   - Extensible
   - Graphics
   - Profiling

II. Exploratory data analysis
   - Regression
   - Clustering algorithms

III. Case study
   - Accelerated laser-wakefield particles

IV. HPC
   - State-of-the-art





Packages, data visualization and examples

R-PROGRAMMING

The New York Times

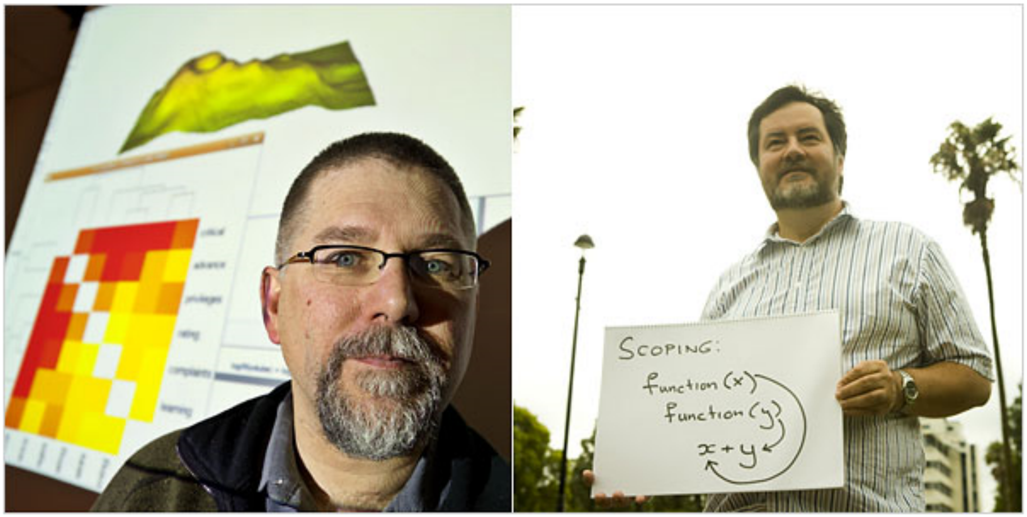
Business Computing

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION

Search Technology Go

Inside Technology Internet | Start-Ups | Business Computing

Data Analysts Captivated by R's Power



Stuart Isett for The New York Times

R first appeared in 1996, when the statistics professors Robert Gentleman, left, and Ross Ihaka released the code as a free software package.

By ASHLEE VANCE
Published: January 6, 2009

To some people R is just the 18th letter of the alphabet. To others, it's the rating on racy movies, a measure of an attic's insulation or what pirates in movies say.

Related

Bits: R You Ready for R?

The R Project for Statistical Computing

R is also the name of a popular programming language used by a growing number of data analysts inside corporations and academia. It is becoming their lingua franca partly

is a language and environment for statistical computing and graphics, a GNU project.
R provides a wide variety of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible.

Download:

http://www.r-project.org

Recommended tutorial:

http://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf

1.Why to use R?

- Open-source, multiplatform, extensible;
- Easy on users with familiarity with S/S+, Matlab, Python or IDL;
- Active and growing community:

– Google, Pfizer, Merck, Bank of America, Boeing, the InterContinental Hotels Group and Shell.

2.R in the scientific community

- Google summer of code and projects using R-project to mine large datasets:

  http://www.r-project.org/SoC08/ideas.html

- With Pfizer:
  - predict the safety of compounds, specifically carcinogenic side effects in potential drugs.
  - models eliminate the expensive and time-consuming process of studying a large number of potential compounds in the physical laboratory…"

  http://www.bio-medicine.org/medicine-news-1/Pfizer-Partners-with-REvolution-Computing-to-Improve-Medicine-Production-Pipeline-17917-2/

2.1. You R with NERSC

- Get started with R on DaVinci:

> module load R

> R

>help()

>demo()

>help.start()

>source('your_function.R')

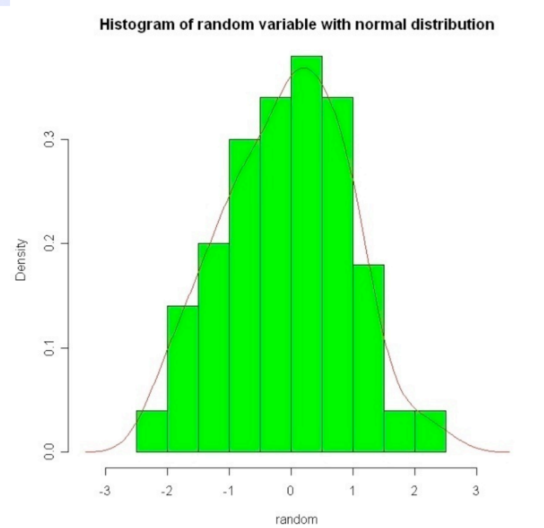
>library(*package_name*)

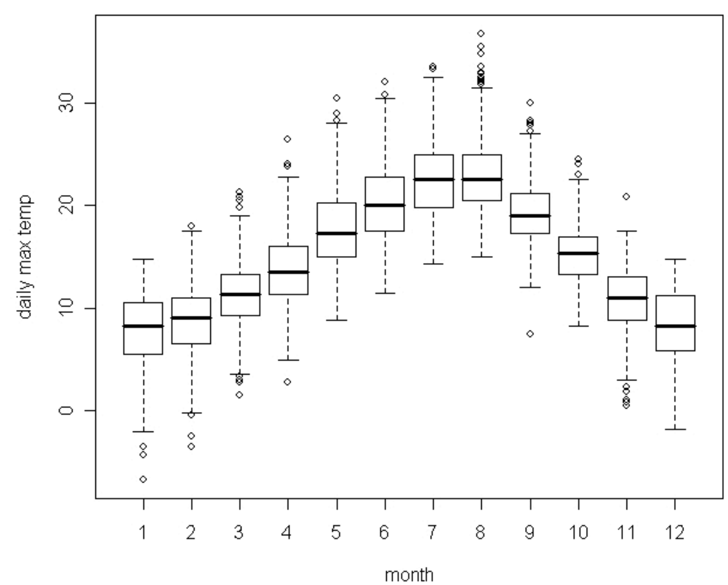
3.Extensible

- Add-on packages:
  - Data input/output: hdf5, Rnetcdf, DICOM, etc.
  - Graphics: trellis, gplot, RGL, fields, etc.
  - Multivariate analysis: MASS, mclust, ape, etc.
  - Other languages: Rcpp, Rpy, R.matlab, etc.

4.Statistical analysis and graphs

- Histogram
- Density
- Boxplot
- Multivariate plot
- Conditioning plot
- Contour plot

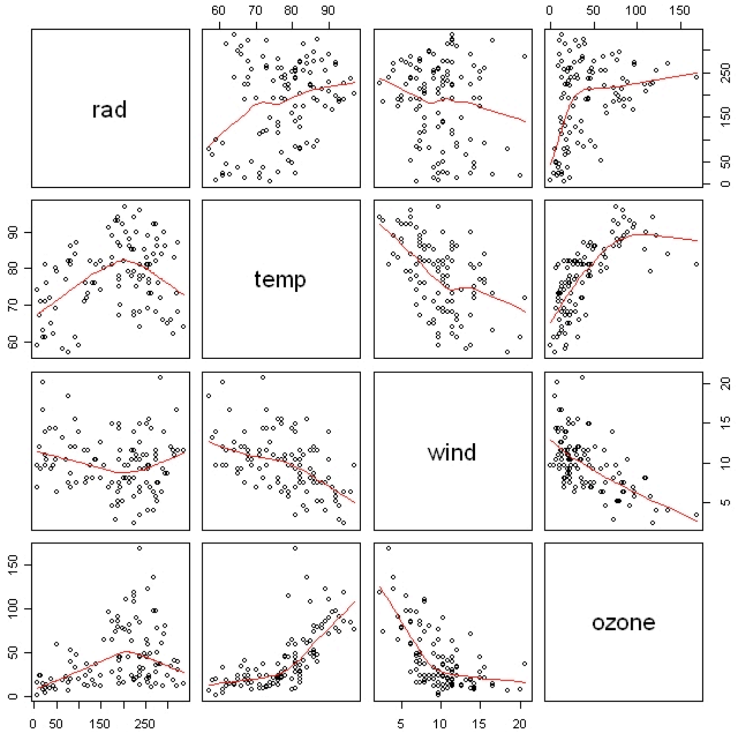




4.1.Multivariate plots

Ex: Explanatory variables: solar radiation, temperature, wind and the response variable ozone;

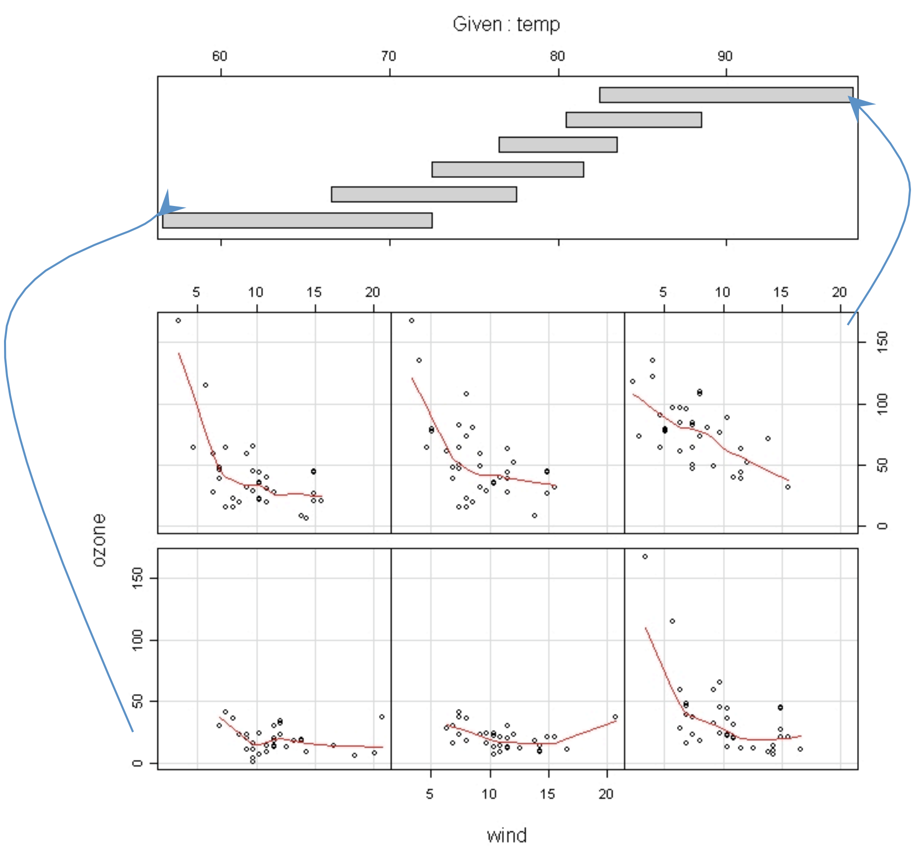
- use of pairs() with dataframes to check for dependencies between the variables.




```
> data=read.table('ozone.data.txt',
  header=T)
> names(data)
[1] "rad"   "temp"  "wind"  "ozone“
> pairs(data,panel.smooth)
```

#panel.smooth = locally-weighted polynomial regression

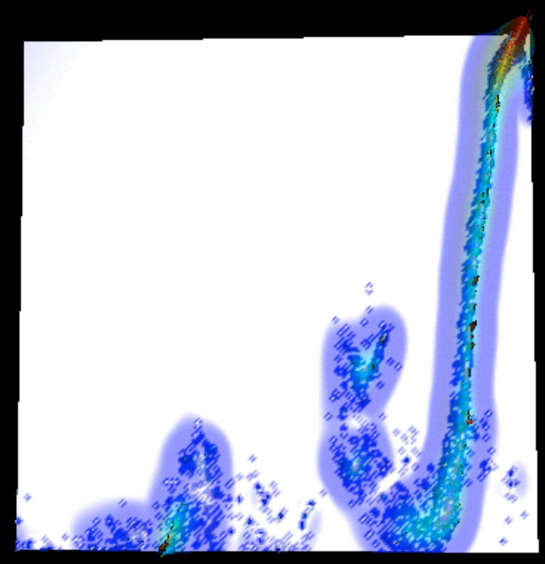
4.2.Conditional plots



- Check the relation of the two explanatory variables *wind, temp* and the response variable *ozone;*

>coplot(ozone~wind|
    temp,panel=panel.smooth)

4.3. Package RGL for 3D visualization



- OpenGL

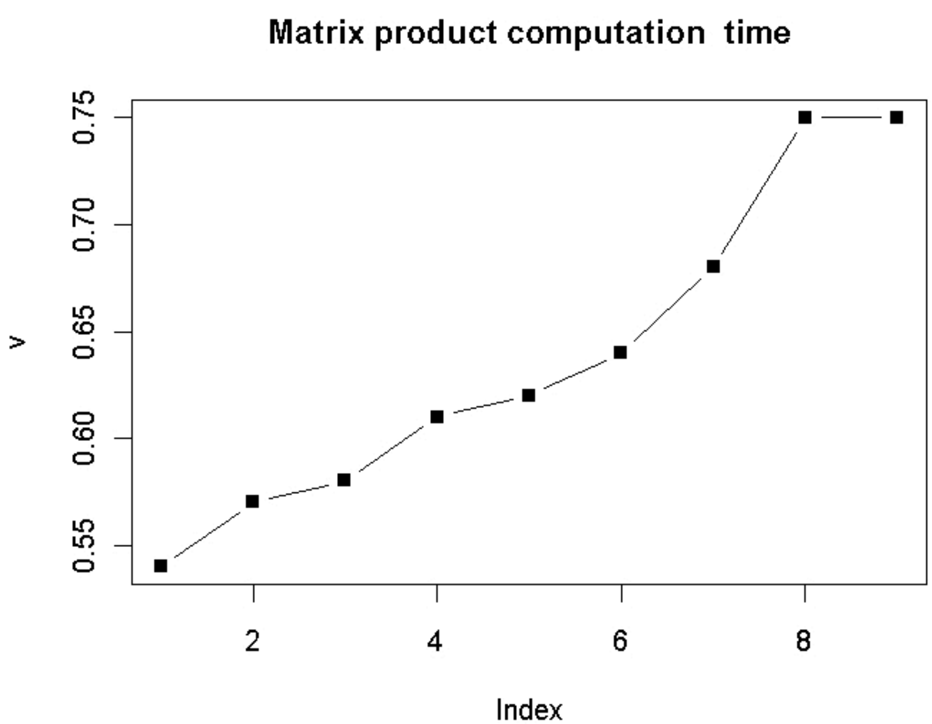
- rgl.demo.lsystem()

- kernel density estimation

Use Visit: https://wci.llnl.gov/codes/visit/

5.Profiling

- Where does your program spend more time?



Also try packages: *profr* and *proftools*

Variable number of arguments

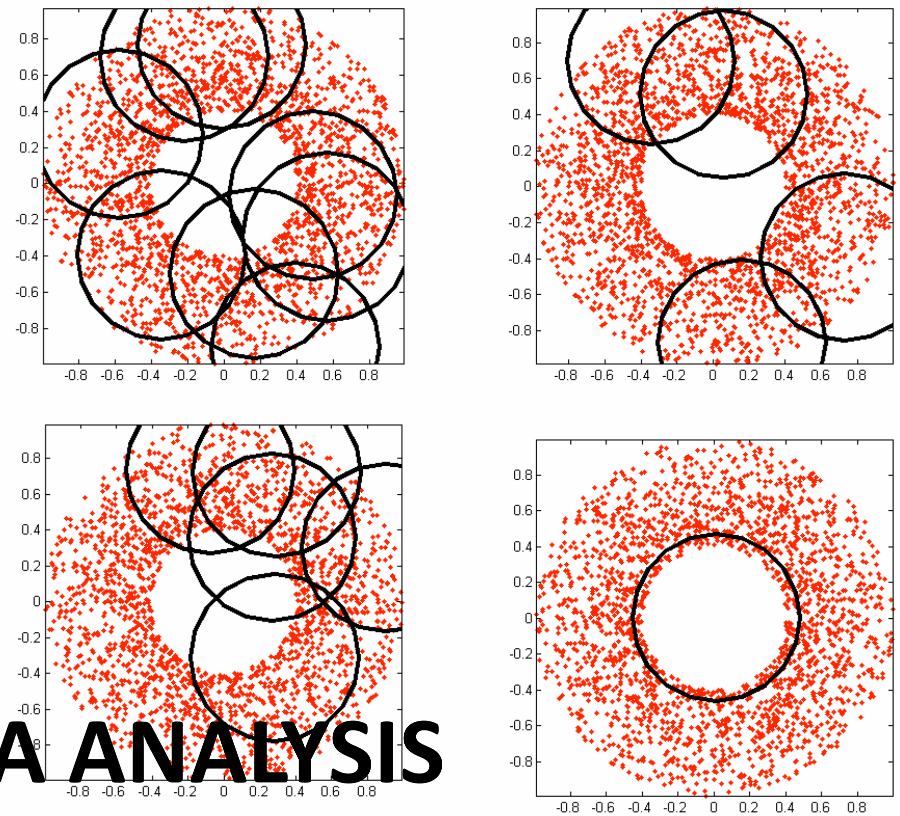
```
several.times <- function (n, f, ...) {
  for (i in 1:n) {
    f(...)
  }
}

matrix.multiplication <- function (s) {
  A <- matrix(1:(s*s), nr=s, nc=s)
  B <- matrix(1:(s*s), nr=s, nc=s)
  C <- A %*% B
}

v <- NULL
for (i in 2:10) {
  v <- append(
    v,
    system.time(
      several.times(
        10000,
        matrix.multiplication,
        i
      )
    ) [1]
  )
}
plot(v, type = 'b', pch = 15,
     main = "Matrix product computation  time")
```

Basics and beyond

EXPLORATORY DATA ANALYSIS

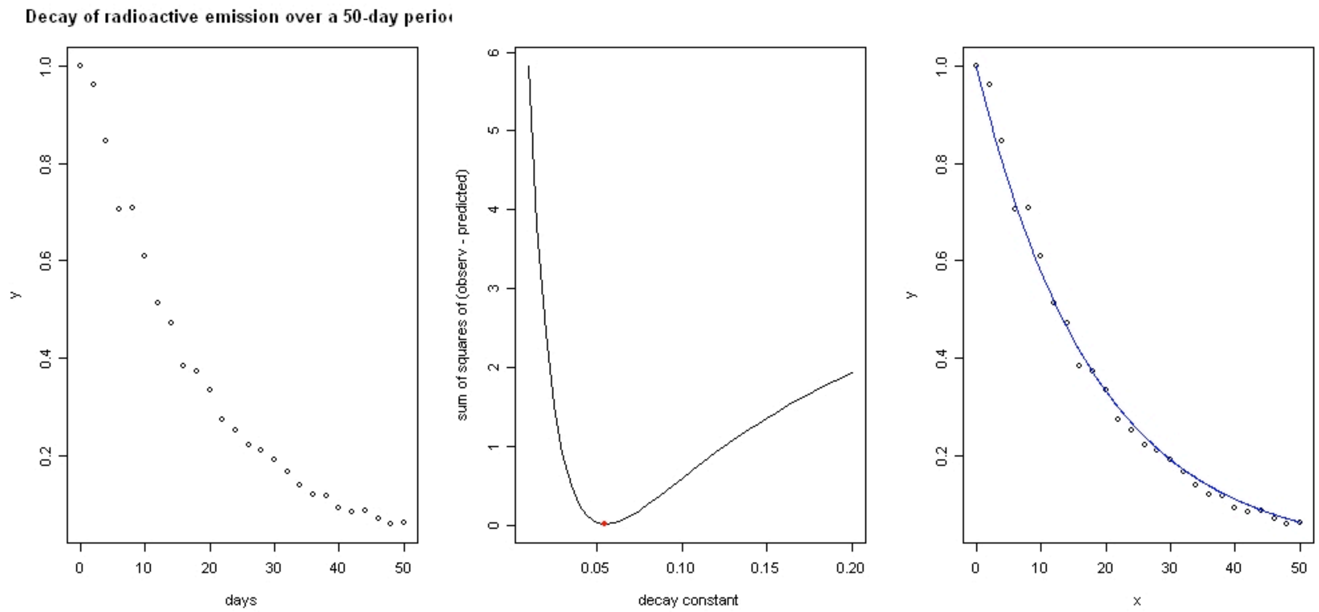


1. Statistical analysis

- Statistical modeling: check for variations in the response variable given explanatory variables;
  - Linear regression
- Multivariate statistics: look for structure in the data;
  - Clustering:
    - Hierarchical
      - Dendrograms
    - Partitioning
      - Kmeans (stats)
      - Mixture-models (mclust)

2.Linear regression

- Ex: Find the equation that best fit the data, given the decay of radioactive emission over a 50-day period



- Linear regression: variables expected to be linearly related;
- Maximum likelihood estimates of parameters = least squares;

2.1.Linear regression

```
data = read.table('sapdecay.txt',header=T)
attach(data)
par(mfrow=c(1,3))
plot(x,y,main='Decay of radioactive emission over a 50-day period',xlab='days')
#  the log(y) gives a rough idea of the decay constant, a, for these data by linear regression of log(y) against x
mylm = lm(log(y)~x)
print(mylm$coefficients)
# sum of squares of the difference between the observed yv and predicted yp values of y, given a specific value of parameter a
sumsq <-function(a,xv=x,yv=y)
{
 yp = exp(-a*xv) #predicted model for y
 sum((yv-yp)^2)
}
a=seq(0.01,0.2,.005)
sq=sapply(a,sumsq)
plot(a,sq,type='l',xlab='decay constant',ylab='sum of squares of (observ - predicted)')
decayK=a[min(sq)==sq] #this is the least-squares estimate for the decay constant
matplot(decayK,min(sq),pch=19,col='red',add=T)
plot(x,y)
days=seq(0,50,0.1)
lines(days,exp(-decayK*days),col='blue')
detach()
```

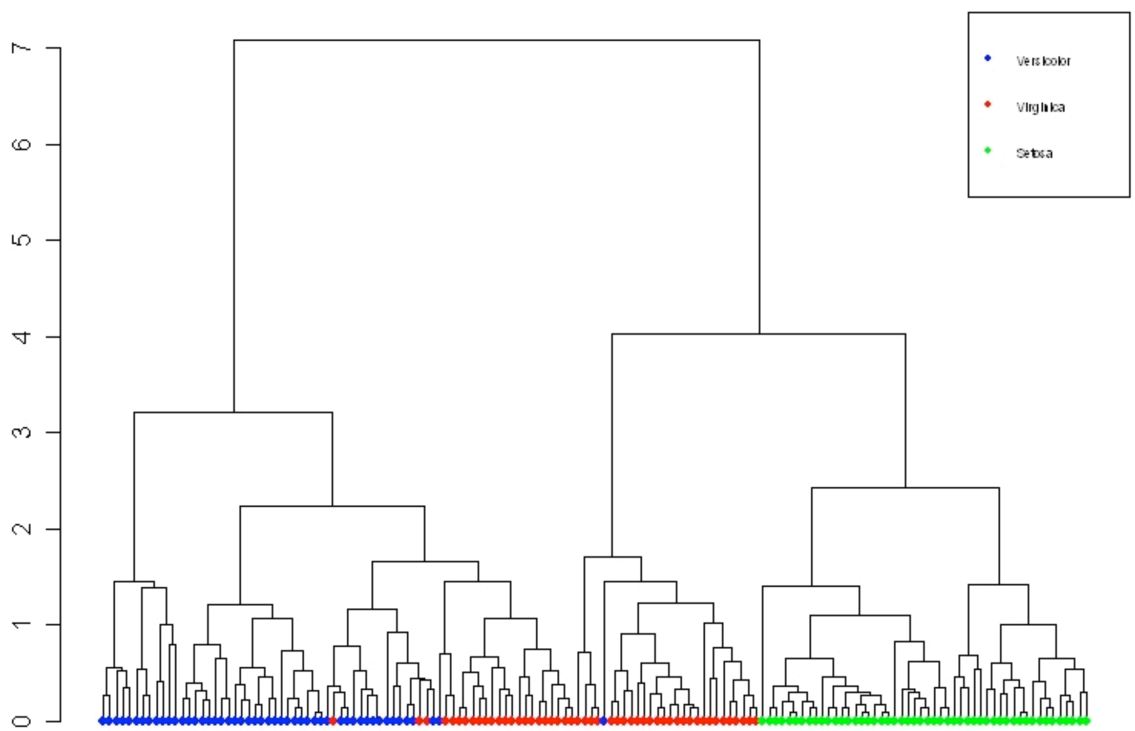
3.Cluster analysis

- Hierarchical
  - dendrogram(stats)
- Partitioning
  - kmeans (stats)
- Mixture-models:
  - Mclust (mclust)

Iris dataset: 150 samples of Iris flowers described in terms of its petal and sepal length and width

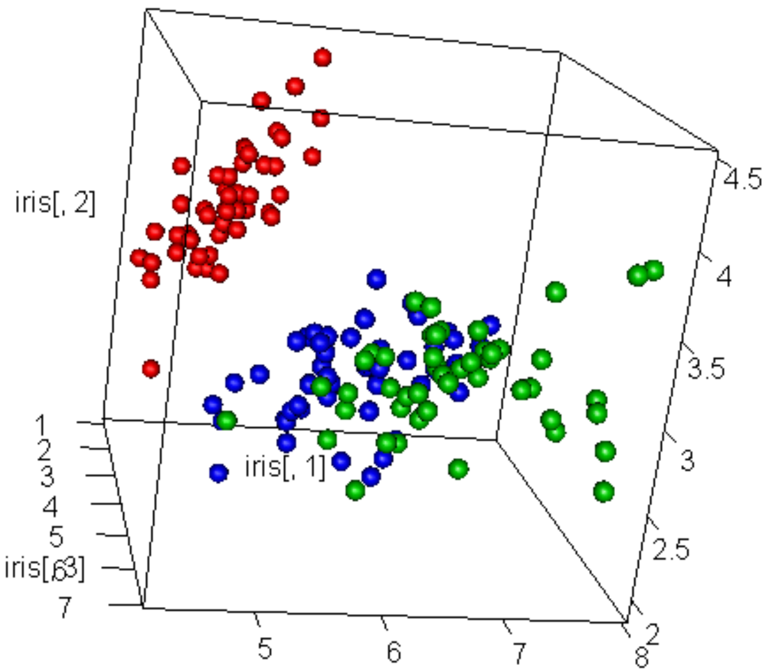
3.1.Hierarchical clustering

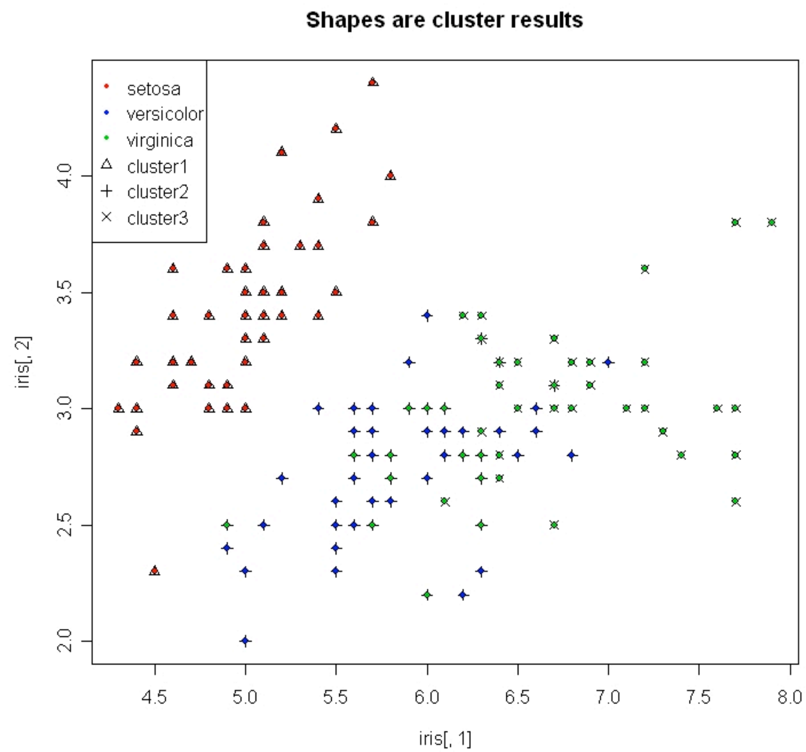
Iris database



- Analysis on a set of dissimilarities, combined to agglomeration methods for analyzing it:
- Dissimilarities: Euclidean, Manhattan, …
- Methods:
  - ward, single, complete, average, mcquitty, median or centroid.

3.2.K-means

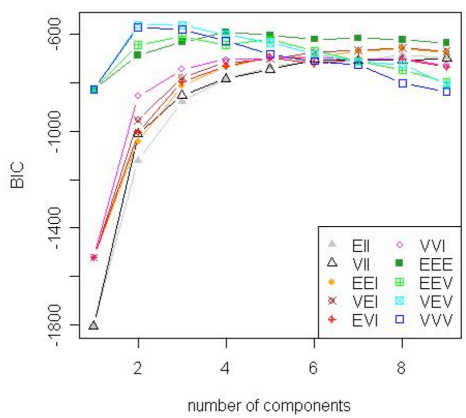


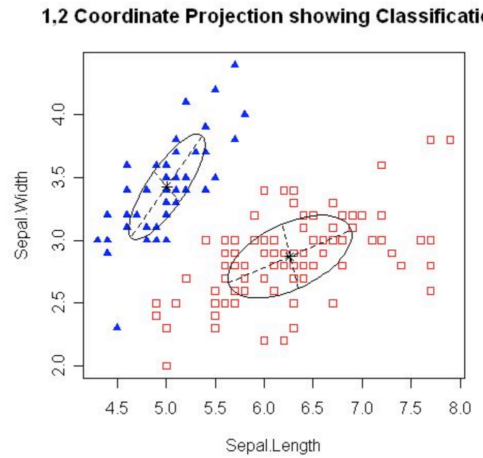


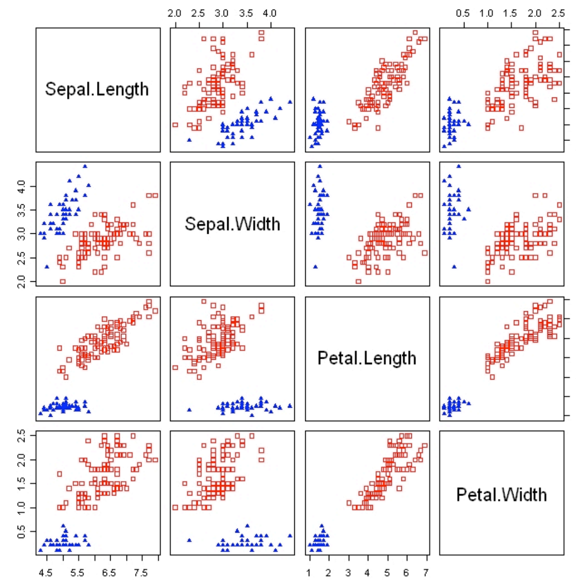
- Split *n* observations into *k* clusters;
  – each observation belongs to the cluster with the nearest mean.

	setosa	versicolor	virginica
1	0	48	14
2	0	2	36
3	50	0	0

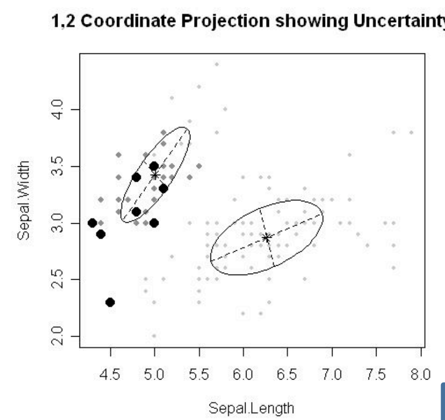
3.3. Model-based clustering

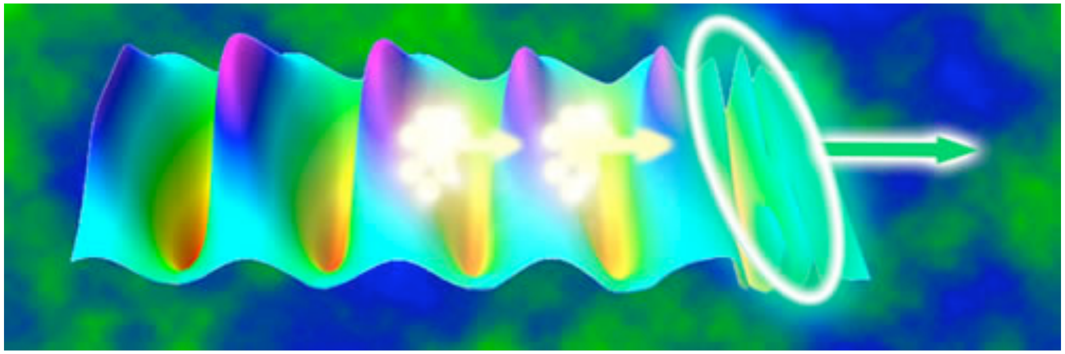






- Mixture Models
  - Each cluster is mathematically represented by a parametric distribution;
  - Set of k distributions is called a *mixture*, and the overall model is a *finite mixture model*;
  - Each probability distribution gives the probability of an instance being in a given cluster.



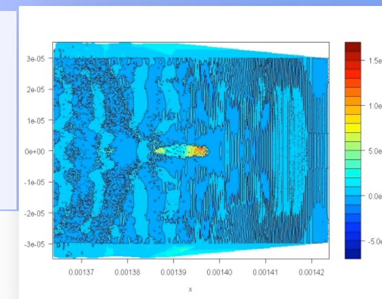


http://www.lbl.gov/publicinfo/newscenter/features/2008/apr/af-bella.html

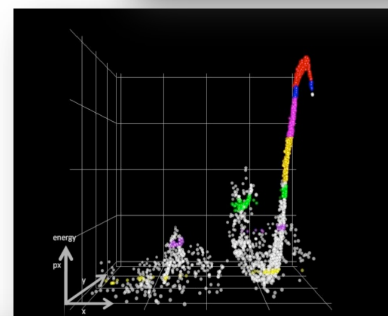
Accelerated laser-wakefield particles

Case study

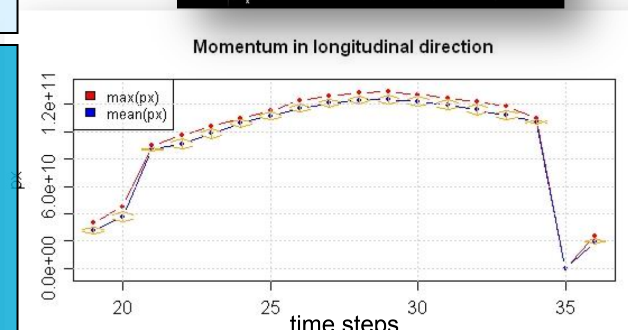
Knowledge discovery in LWFA science via machine learning



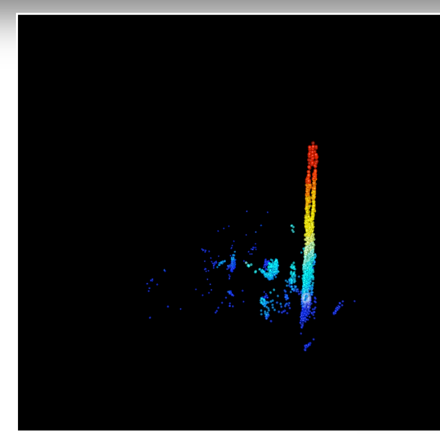
- PI: C. Geddes (LBNL) in SciDAC COMPASS project, Incite.



- Accomplishments:
  - Described compact electron clouds using minimum enclosing ellipsoids;
  - Developed algorithms to adapt mixture model clustering to large datasets;

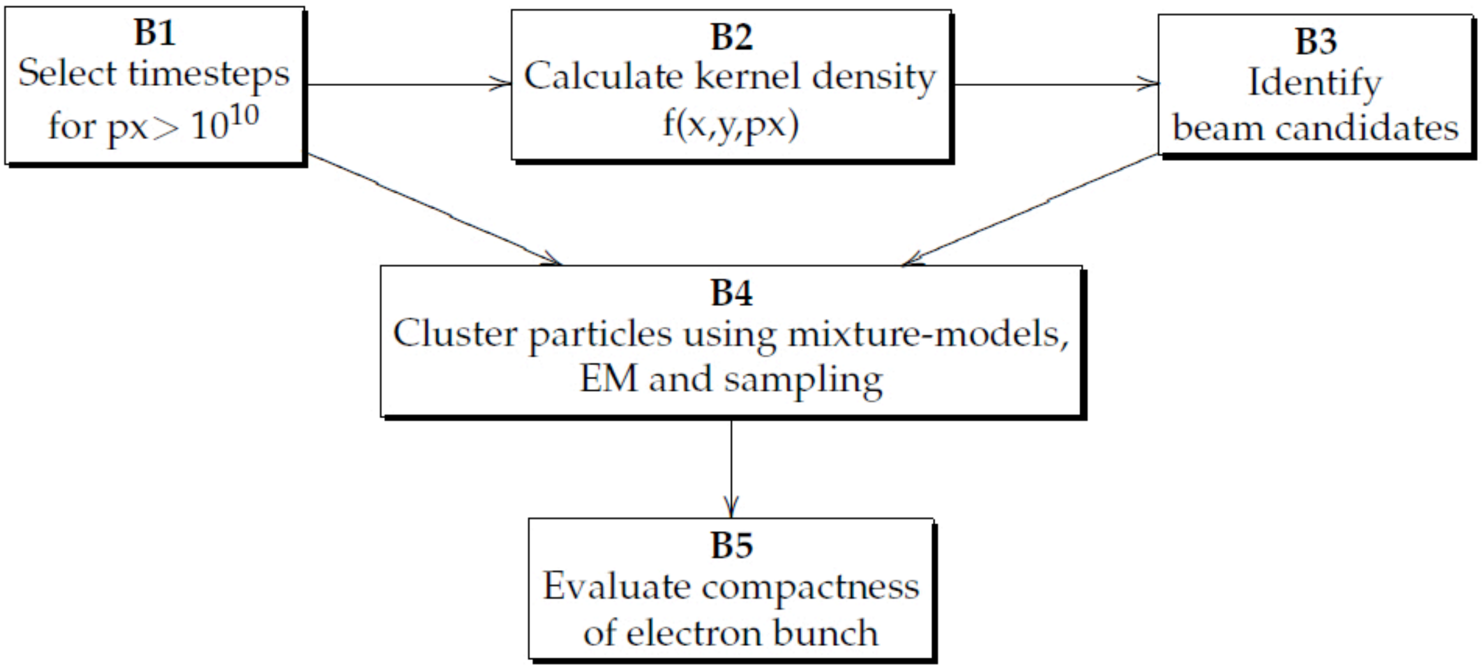


- Science Impact:
  - Automated detection and analysis of compact electron clouds;
  - Derived dispersion features of electron clouds;
  - Extensible algorithms to other science problems;



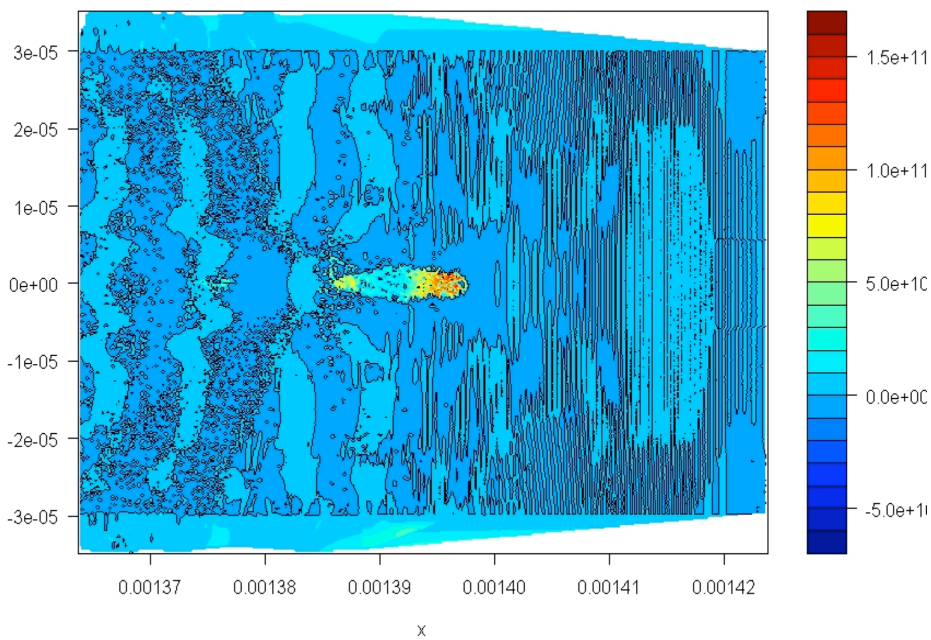
- Collaborators:
  - Tech-X
  - Math Group, LBNL
  - UCDavis, University of Kaiserlautern

Framework



- Goal: automate the analysis of electron bunches by detecting compact groups of particles, subjected to similar momentum and spatio-temporal coherence.

B1. Select relevant particles



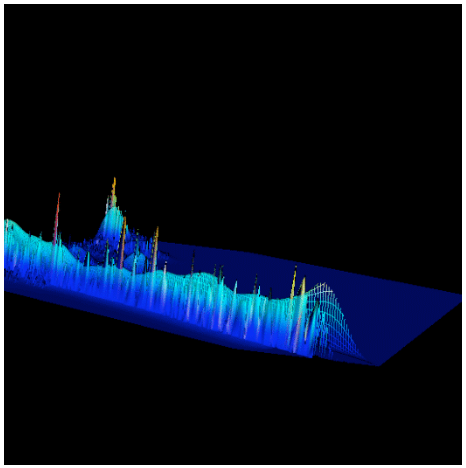
Representation of particle momentum in one time step: spline interpolation onto a grid for visualization of irregularly spaced input data.

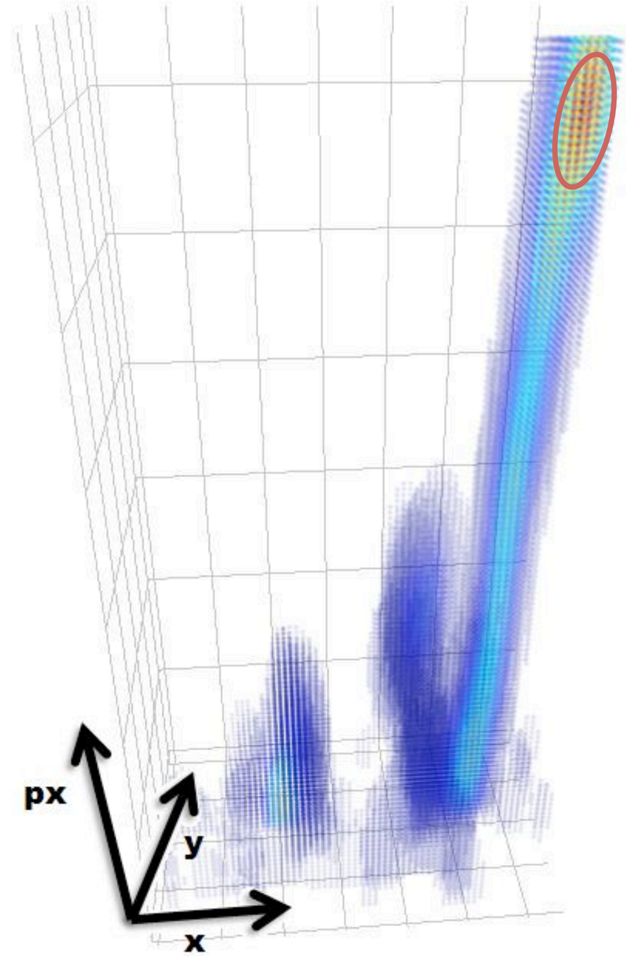
- Beams of interest are characterized by high density of high-energy particles:

1. Elimination of low energy particles (px<1e10)
   - Wake oscillation: px<=1e9
   - Excludes particles of the background
2. Calculation of the simulation average number of particles (μ_s);
3. Elimination of timesteps with number of particles inferior to μ_s;

Packages:
akima, hdf5, fields

B2.Kernel-based estimation

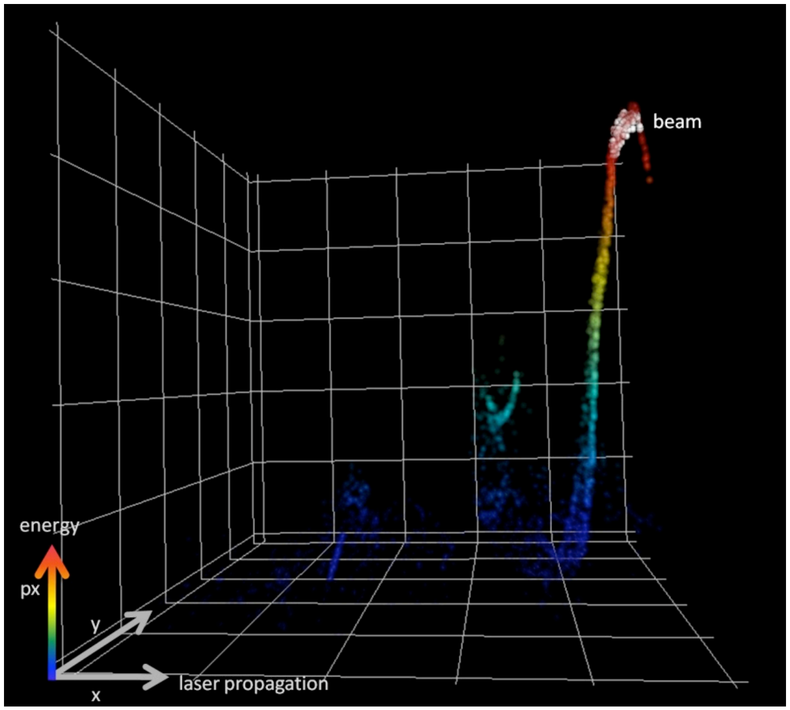


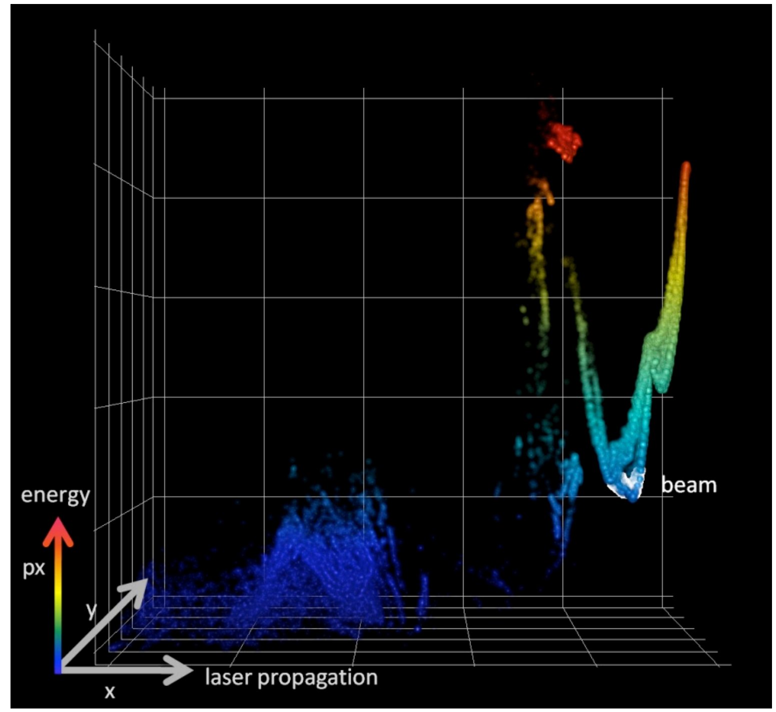


- Kernel density estimators are less sensitive to the placement of the bin edges;
- Goal: retrieve a dense group of particles with similar spatial and momentum characteristics:
  - argmax f(x,y,px),
  - Neighborhood: 2 μm

Packages:
misc3d, rgl, fields

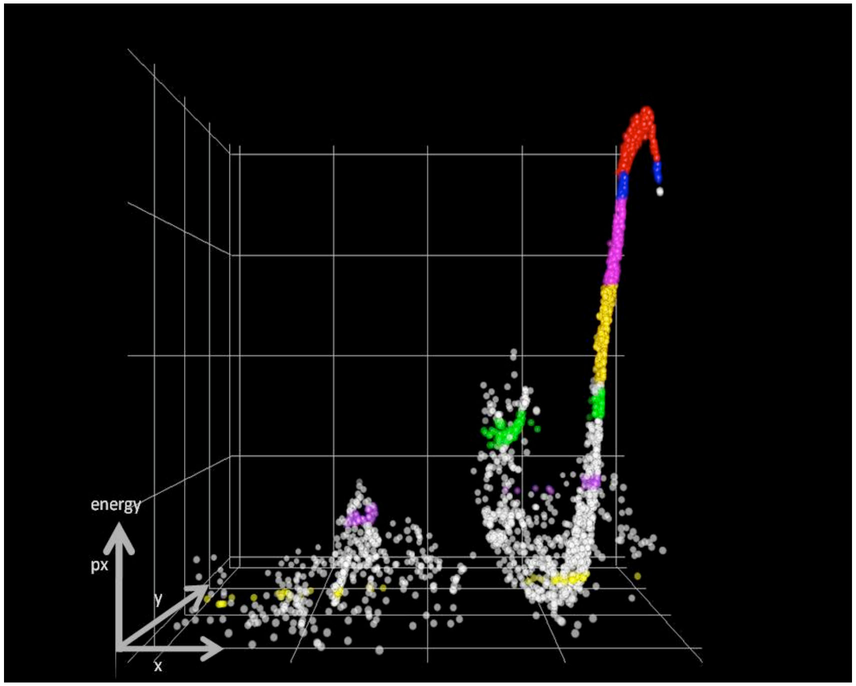
B3. Identify beam candidates





- Detection of compact groups of particles independent of being a maximum in one of the variables;

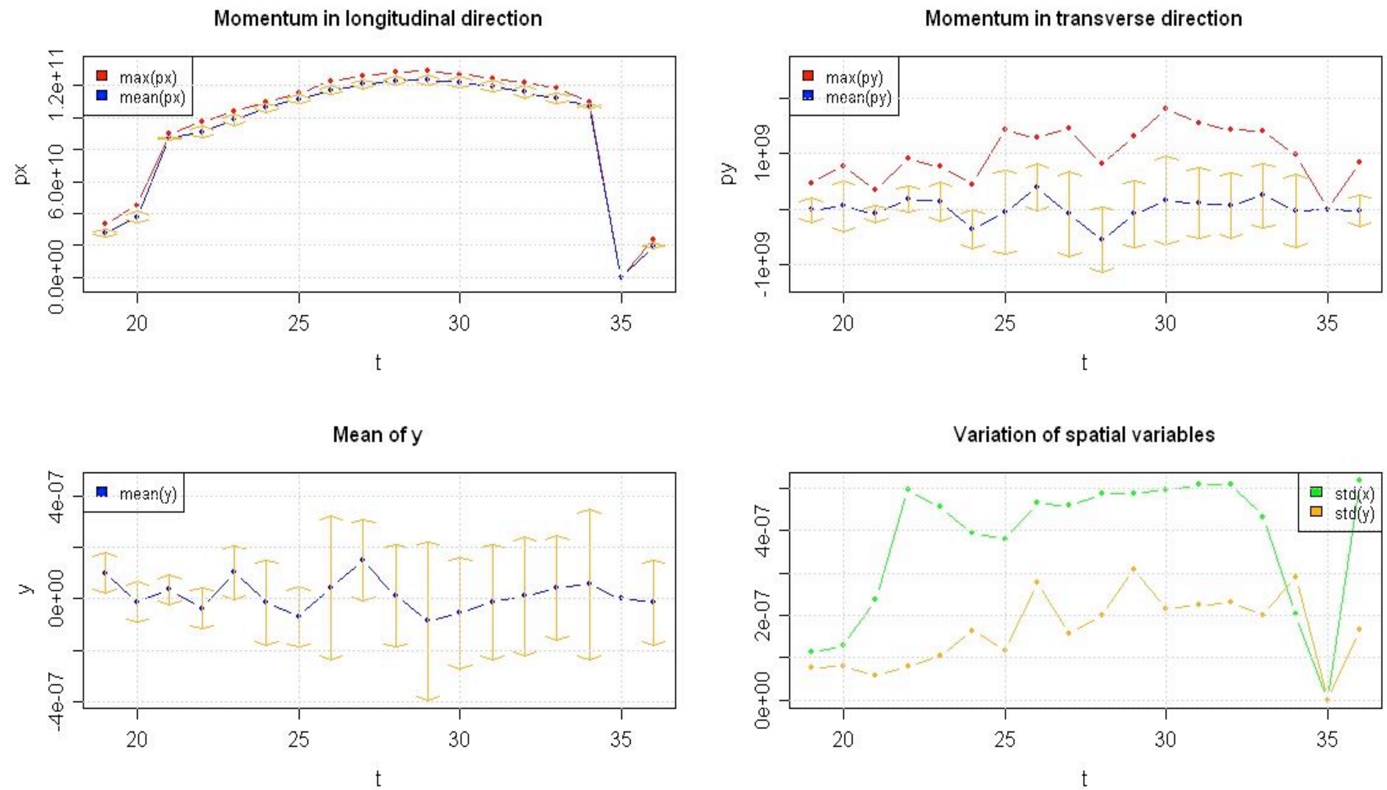
B4. Cluster using mixture models



- Model and number of clusters can be selected at run time (*mclust*);
- Partition of multidimensional space;
- Assume that the functional form of the underlying probability density follows a mixture of normal distributions;

Packages:
mclust, rgl

B5. Evaluation of compactness



- Bunches of interest move at speed ≈ c, hence are nearly stationary in the moving simulation window;
- Moving averages smoothes out short-term fluctuations and highlights longer-term trends.

Packages, challenges and new businesses

High performance computing

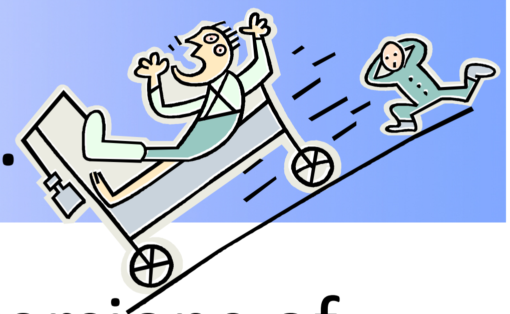
1. Improve performance/reusability

- Good coding: avoid loops, vectorization;
- Extend R using compiled code:
  - packages: Rcpp, inline
- Recycle your Python codes:
  - Package: Rpython
- Parallelism:
  - Explicit: packages Rmpi, Rpvm, nws
  - Implicit: packages pnmath, pnmath0 for multithreaded math functions
- Use out-of-memory processing with
  - packages bigmemory and ff

2. What is going on HPC in R?

- Parallelism:
  - Multicore: multicore, pnmath, …
  - Computer cluster: snow, Rmpi, rpvm, …
  - Grid computing: GRIDR, …
- GPU:
  - gputools: parallel algorithms using CUDA + CUBLAS
- Extremely large data:
  - ff: memory mapped pages of binary flat files.

3. Nothing is perfect…



- Limits on individual objects: on all versions of R, the maximum number of elements of a vector is 2^31 – 1;
- R will take all the RAM it can get (Linux only);
- More information, type:

```
>help('Memory-limits')
>gc() #garbage collector
>object.size(your_obj) #size of your object
```

Take home

- **Everything is an object**. This means that your variables are objects, but so are output from analyses. Everything that can possibly be an object by some stretch of the imagination… is an object.
- **R works in columns, not rows**. R thinks of variables first, and when you line them up as columns, then you have your dataset. Even though it seems fine in theory (we analyze variables, not rows), it becomes annoying when you have to jump through hoops to pull out specific rows of data with all variables.
- **R likes lists**. If you aren't sure how to give data to an R function, assume it will be something like this: c("item 1", "item 2") meaning "concatenate into a list the 2 objects named Item 1, Item 2". Also, "list" is different to R from "vector" and "matrix" and "dataframe" etc.
- **Its open source**. It won't work the way you want. It has far too many commands instead of an optimized core set. Multiple ways to do things, few of them really complete. People on the mailing lists revel in their power over complexity, lack of patience, and complete inability to forgive a novice.

Source: http://www.nettakeaway.com/tp/R/129/understanding-r

References

- Michael J. Crawley. *Statistics: An Introduction using R*. Wiley, 2005. ISBN 0-470-02297-3.
  - data: http://www.bio.ic.ac.uk/research/mjcraw/therbook/
- Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications With R Examples*. Springer, New York, 2006. ISBN 978-0-387-29317-2

- Basics
  - *http://cran.r-project.org/doc/contrib/Short-refcard.pdf*
  - *http://cran.r-project.org/doc/contrib/refcard.pdf*

    Cheat sheets
  - http://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf
  - *http://www.manning.com/kabacoff/Kabacoff_MEAPCH1.pdf*
- *Intermediate*
  - *http://math.acadiau.ca/ACMMaC/Rmpi/basics.html*
  - *User-lists*

Acknowledgements

NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

ERNEST ORLANDO LAWRENCE BERKELEY NATIONAL LABORATORY

Visualization Group

BERKELEY LAB

AFRD
ACCELERATOR & FUSION RESEARCH DIVISION

TECH

Lawrence Berkeley National Lab
Mathematics Department

http://www.sciviews.org/Tinn-R/

Office of Science
U.S. DEPARTMENT OF ENERGY