Graphical Interaction in Spectroscopic Data Analysis

Sebastian Mellor[1], Claudia Beleites[2,*], Colin Gillespie[1], Christoph Krafft[2], and Jürgen Popp[2,3]

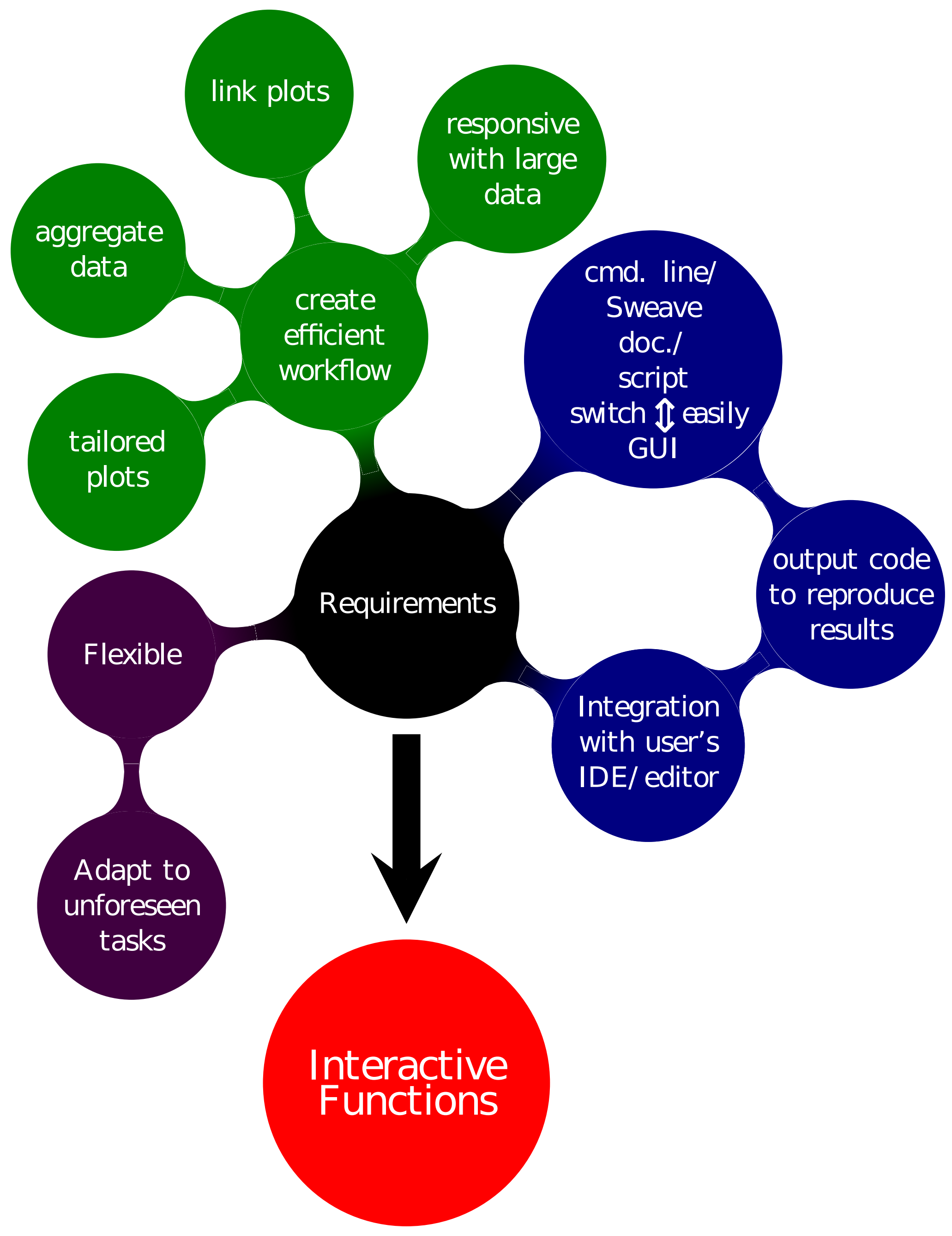
[1] School of Mathematics & Statistics, Newcastle University, Newcastle/UK.
[2] Institute of Photonic Technology, Jena/Germany.
[3] Institute of Physical Chemistry and Abbe Center of Photonics, University Jena/Germany.
[*] Corresponding author: Claudia.Beleites@ipht-jena.de

GUIs and Spectroscopic Data

- data are complex
- variety of statistical models used
- are often large: typically 10^2 - 10^5 spectra x 10^2 - 10^3 wavelengths, can reach hundreds of GB.

+ some tasks need visual interaction, see the example of spike filtering.
+ good for exploring complex data and models

Interactive Functions


link plots
responsive with large data
aggregate data
create efficient workflow
tailored plots
cmd. line/ Sweave doc./ script switch ⇕ easily GUI
output code to reproduce results
Requirements
Integration with user's IDE/ editor
Flexible
Adapt to unforeseen tasks
Interactive Functions

Obtaining hyperSpec+GUI

? homepage:
⇨ *hyperspec.r-forge.r-project.org*

? hyperSpec:
- latest stable from CRAN
  ⇨ install.packages ("hyperSpec")
- latest nightly build (dev. version)
  ⇨ `install.packages("hyperSpec", repos="http://R-Forge.R-project.org")`

? hyperSpecGUI:
- latest nightly build (dev. version)
  ⇨ `install.packages("hyperSpecGUI", repos="http://R-Forge.R-project.org")`

References

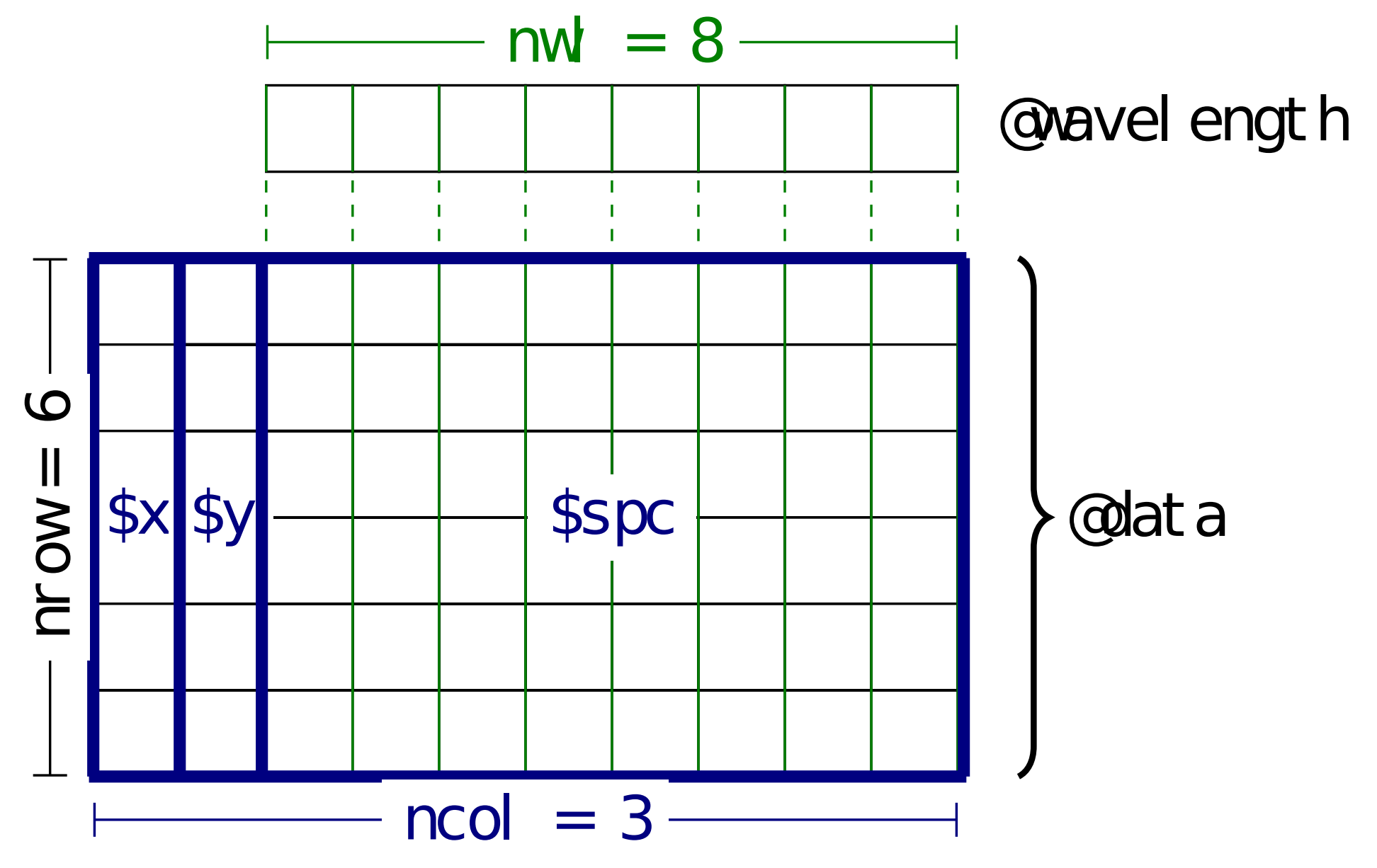
[1] Beleites, C. hyperSpec (*http://hyperspec.r-forge.r-project.org/*)

[2] Verzani, J. gWidgets (*http://cran.r-project.org/web/packages/gWidgets/index.html*)

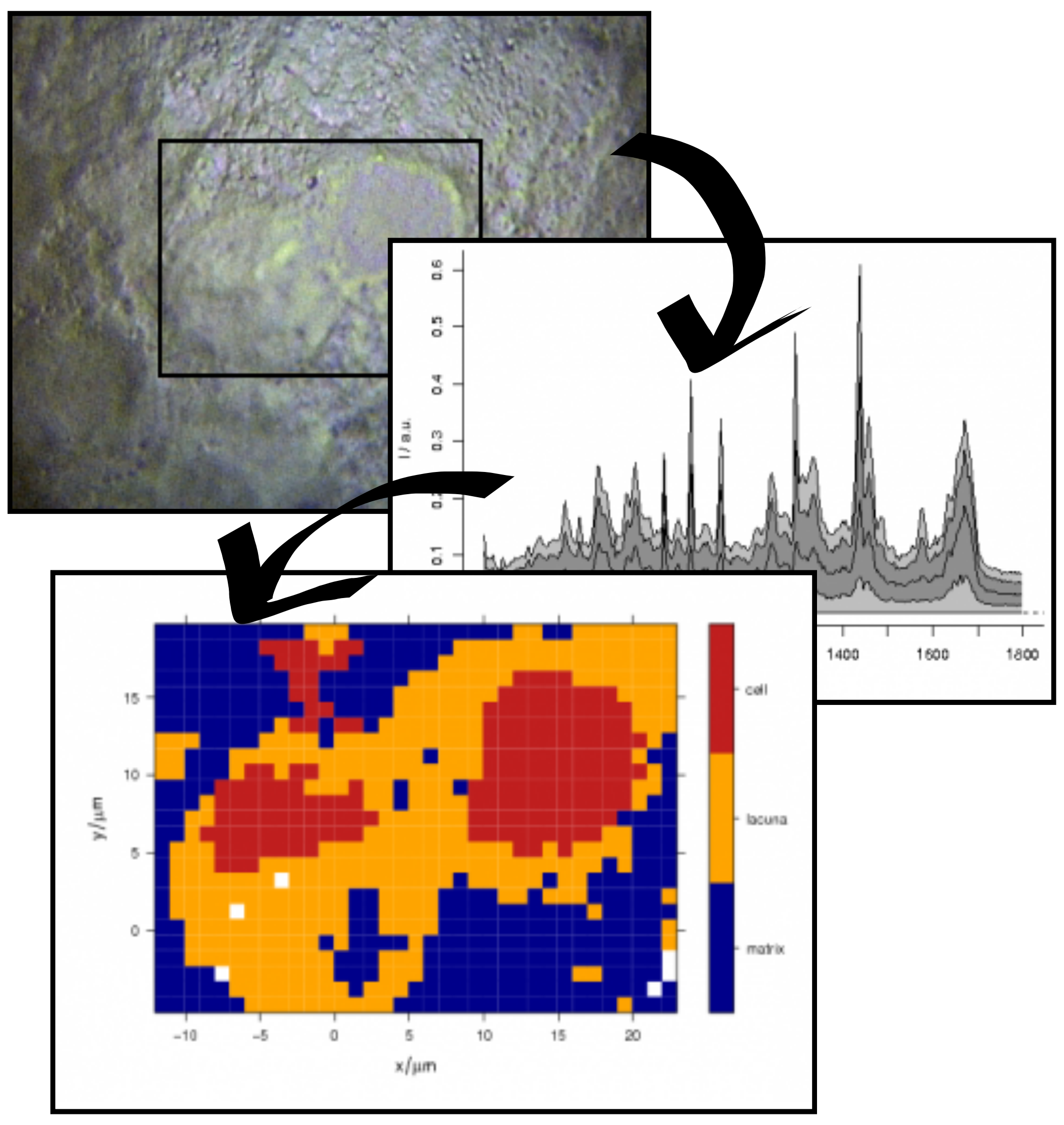
hyperSpecGUI can also be found at *http://hyperspec.r-forge.r-project.org/*.

hyperSpectral Data = Spectroscopic Data + Extra Dimensions

hyperSpec handles hyperspectral data:



```
R> chondro
hyperSpec object
   875 spectra
   4 data columns
   300 data points / spectrum
wavelength: Delta * tilde(nu)/cm^-1 [numeric] 602 606 ... 1798
data:  (875 rows x 4 columns)
   1. y: y/(mu * m) [numeric] -4.77 -4.77 ... 19.23
   2. x: x/(mu * m) [numeric] -11.55 -10.55 ... 22.45
   3. clusters: clusters [factor] matrix matrix ... lacuna + NA
   4. spc: I / a.u. [matrix300] 501.8194 500.4552 ... 169.2942
```



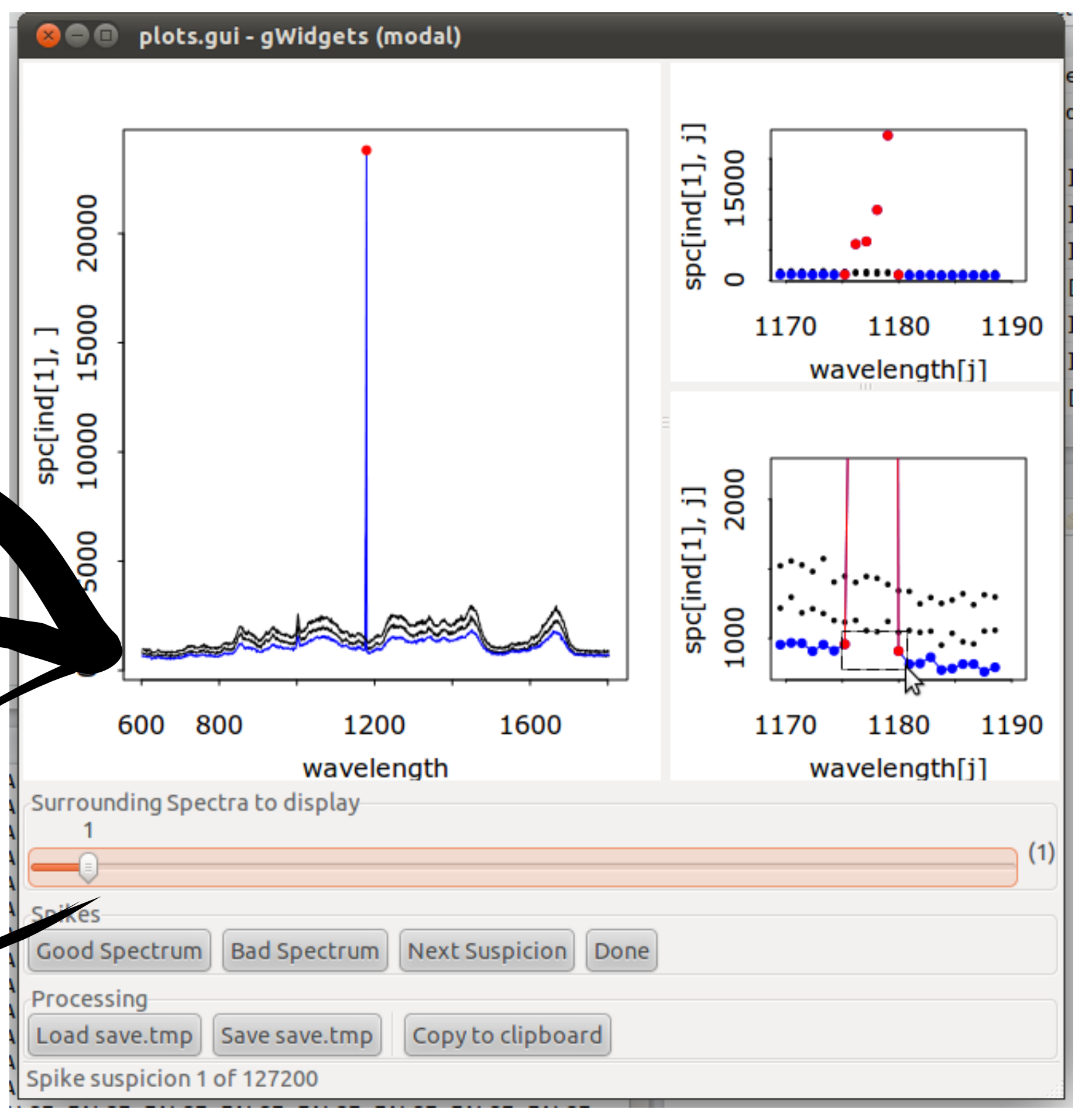
Example: Interactive Spike Filtering

- cosmic rays hit the detector
  ⇨ spikes are observed in the Raman spectra

- several stategies exist for automated detection

+ works well with high, sharp spikes
! broader artifacts confused with sharp Raman bands
! and vice versa
! borders of broader artifacts difficult to detect
⇨ manual control and adjustment necessary

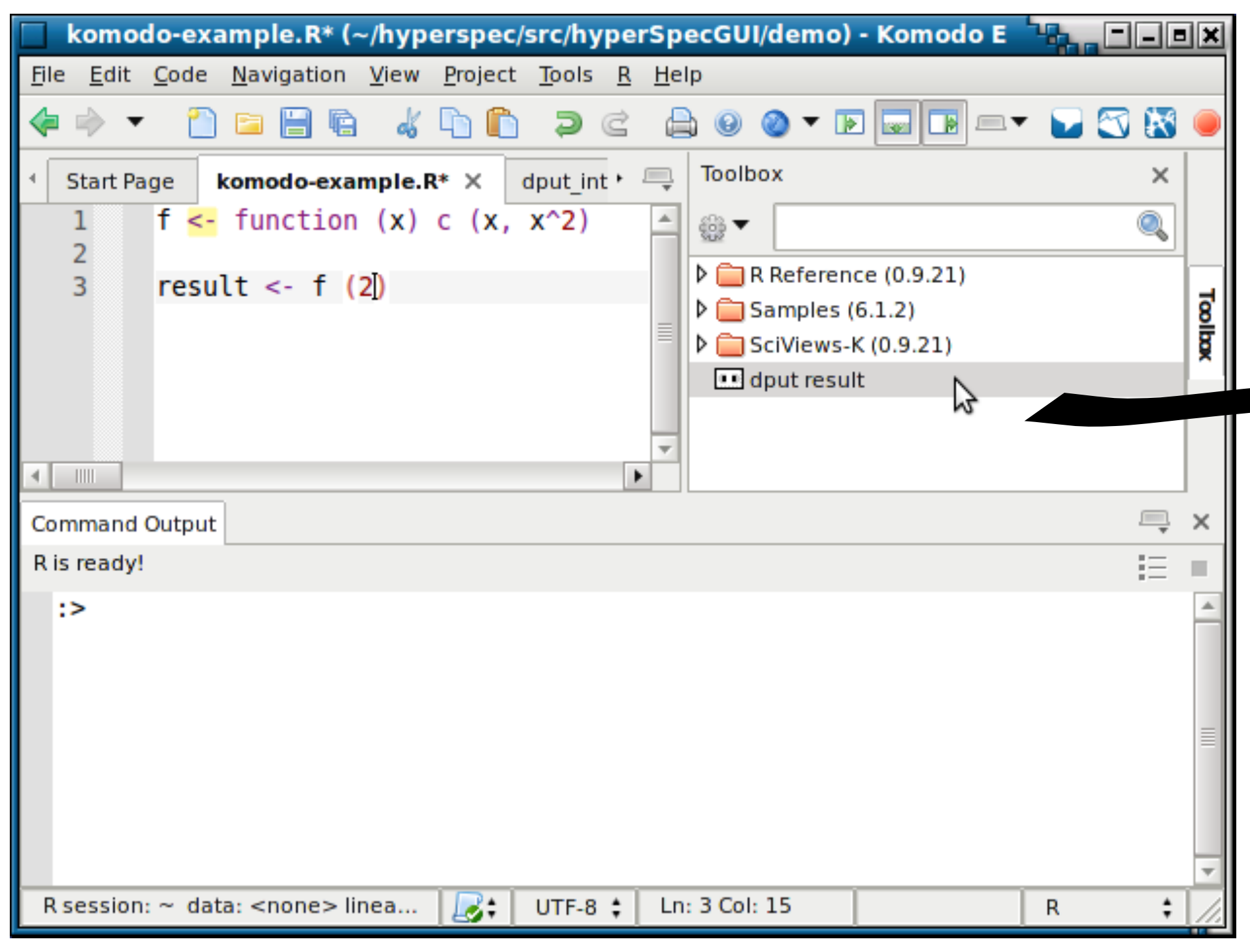
```
R> load("cartilage-raw.RData")
R> tmp <- sweep(cartilage, 1, median, `/`)
R> tmp <- sweep(tmp, 2, median, `-`)
R> scores <- spikefilter2d(spcmatrix=tmp[[]])
R> spikes <- spikes.interactive.GUI(
                  cartilage[1:100],
                  scores[1:100, ])

R> spikes
1175.22 1176.18 1177.13 1178.08 1179.03 1179.98
    577     578     579     580         582
```



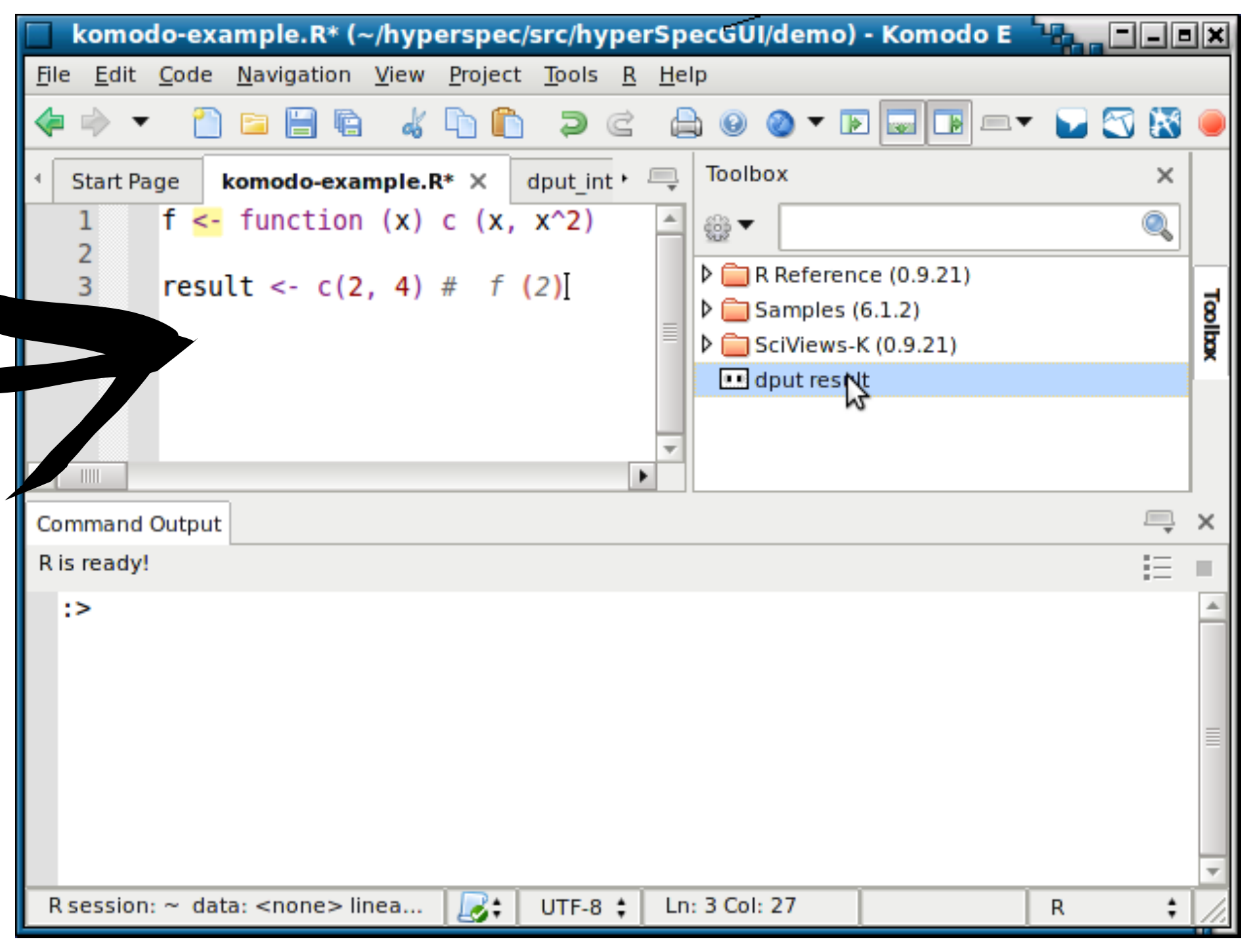
Spike suspicion 1 of 127200

Example: Integration with IDE

- wrapper function replacing function call by "`dput (result)`"



```
f <- function (x) c (x, x^2)

result <- f (2)
```



```
f <- function (x) c (x, x^2)

result <- c(2, 4) #  f (2)
```

⇨ compatible with reproducible research practices
- "poor man's solution": applet copies "`dput (result)`" to clipboard

SM's work is funded by the Google Summer of Code 2011.
CB acknowledges funding by the European Union via the Europäischer Fonds für Regionale Entwicklung (EFRE) and the "Thüringer Ministerium für Bildung, Wissenschaft und Kultur" (Project: B714-07037).