Linking R and Qt: a new graphics subsystem

Deepayan Sarkar

Computational Biology
Fred Hutchinson Cancer Research Center

July 13, 2009

R Graphics

- Long history
- Widely used and tested
- Cross-platform, supports multiple output backends
  (screen, PDF, SVG, pixmap, etc.)
- Grid graphics provides enormous flexibility

Drawbacks of the R graphics model

- Limited by ink-on-paper model
  - Logical graphical primitives not retained at the device level
  - E.g., a "+" plotting character is drawn as two line segments
  - Things can be added to a plot but not deleted
  - Interaction difficult
- Grid works around this by keeping track of everything it draws
- But,
  - Grid is slow
  - Underlying limitations of the graphics engine remain
  - E.g., removing a point essentially redraws the whole graph
  - Doesn't look pretty, because grid is slow!

Qt

- Powerful "Application and UI framework" written in C++
- Widely used and tested
- Cross-platform, supports multiple output backends
  (screen, PDF, SVG, pixmap, etc.)
- Qt's Graphics View framework enables flexible graphics
  > "...provides a surface for managing and interacting with a large number of custom-made 2D graphical items, and a view widget for visualizing the items..."
- Can we use Qt to move beyond the limitations of R graphics?

Qt

- Powerful "Application and UI framework" written in C++
- Widely used and tested
- Cross-platform, supports multiple output backends
  (screen, PDF, SVG, pixmap, etc.)
- Qt's Graphics View framework enables flexible graphics
  > "...provides a surface for managing and interacting with a large number of custom-made 2D graphical items, and a view widget for visualizing the items..."
- Can we use Qt to move beyond the limitations of R graphics?

Approach 1

- Implement an R graphics device using the Graphics View framework
- Use Qt's capabilities to enhance functionality
- But still limited: items in the scene cannot in general be mapped back to the data

Approach 2

- A completely independent graphics subsystem
  - Not a novel idea: GGobi, iPlots
- My main interest: implement a Trellis-like system
- Interface should be similar for the end-user
- Should make interaction and dynamic manipulation easier

Why grid?

- What grid features do we really need?
  - Ability to draw arbitrary things in rectangular viewports
  - Ability to create a layout and put things in it
  - Control over layout row and column expansion (e.g., panels vs labels)
  - Basic elements (typically text labels) should know the minimum size needed to display themselves
  - Complex elements (e.g., legends) made up of simpler elements placed in a layout should also know their minimum size (*"frameGrob"* in grid).
- All are features that a GUI toolkit excels at
- In fact, the design of grid was inspired by GUI toolkits
- So, why not just use an actual toolkit?

Why grid?

- What grid features do we really need?
  - Ability to draw arbitrary things in rectangular viewports
  - Ability to create a layout and put things in it
  - Control over layout row and column expansion (e.g., panels vs labels)
  - Basic elements (typically text labels) should know the minimum size needed to display themselves
  - Complex elements (e.g., legends) made up of simpler elements placed in a layout should also know their minimum size (*"frameGrob"* in grid).
- All are features that a GUI toolkit excels at
- In fact, the design of grid was inspired by GUI toolkits
- So, why not just use an actual toolkit?

Mosaiq

- A high-level lattice-like package
- Implemented using the qtpaint API
- Makes extensive use of Qt layouts
  (both Graphics View layouts and widget layouts)
- Opportunity to clean up API based on lessons from lattice

API highlights

- A Trellis-style graph, variables (terms) can be
  - Conditioning variables: used to define subsets of data
  - Panel variables: appropriate subsets used within panel display
  - E.g., `densityplot(~ x | a, groups = g, weights = w)`
  - Finer distinctions exist in the "Grammar of Graphics" worldview, but not in the "panel function" philosophy of Trellis
- The classic Trellis formula API
  - Formula defines conditioning and *some* panel variables
  - Others specified using non-standard evaluation paradigm
- Problems:
  - Limits code re-use;
    special features need to be handled in each high-level function
  - Difficult to write wrappers/methods;
    needs `match.call()`, careful handling

API highlights

- A Trellis-style graph, variables (terms) can be
  - Conditioning variables: used to define subsets of data
  - Panel variables: appropriate subsets used within panel display
  - E.g., `densityplot(~ x | a, groups = g, weights = w)`
  - Finer distinctions exist in the "Grammar of Graphics" worldview, but not in the "panel function" philosophy of Trellis
- The classic Trellis formula API
  - Formula defines conditioning and *some* panel variables
  - Others specified using non-standard evaluation paradigm
- Problems:
  - Limits code re-use;
    special features need to be handled in each high-level function
  - Difficult to write wrappers/methods;
    needs `match.call()`, careful handling

API highlights: plans

- Separate out specification of "conditioning variables" and "panel variables"
- Packets defined solely by conditioning variables (subscripts); the only thing that differs between panel function calls
- Panel variables represented as expressions, passed on to panel function directly.
- Use methods to provide more familiar formula interface
- Provide more control over evaluation
  - Define a new generic function
    `evaluate(e, data, subset, enclos)`
  - Dispatch (at least) on both `e` and `data`
  - Supporting new data types could be as simple as writing a new method
  - Important for high-throughput Bioinformatics data with complex structures

Still to do

- Legends: should be easy, just not done yet
- Aspect ratio: same holds
- Some clipping issues
- PDF outout: currently not vector output for complex widgets
- Mathematical annotation (plotmath): not going to happen, but should be able to embed R graphics
- Interaction model? Will likely involve layers, but needs thought