R/qtl2
rewrite of a very old R package

Karl Broman
Biostatistics & Medical Informatics, U Wisconsin–Madison

kbroman.org
github.com/kbroman
@kwbrman
Slides: bit.ly/rstudio2019
Intercross

P₁ \times P₂

F₁ \times F₁

F₂
QTL mapping

![QTL mapping graph](image-url)
19 years of R/qtl

Year

Lines of code


idea svn git

R

C
doc
Good things
Good things

- some of the code
- basics of the user interface
- diagnostics and data visualization
- quite comprehensive
- quite flexible
Bad things
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>liver</td>
<td>spleen</td>
<td>sex</td>
<td>pgm</td>
<td>D1Mit18</td>
<td>D1Mit80</td>
<td>D1Mit17</td>
<td>D2Mit379</td>
<td>D2Mit75</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>27.3</td>
<td>51.4</td>
<td>110.4</td>
<td>38.3</td>
<td>48.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>61.92</td>
<td>153.16</td>
<td>m</td>
<td>1</td>
<td>BB</td>
<td>SB</td>
<td>SB</td>
<td>SB</td>
<td>SB</td>
</tr>
<tr>
<td>5</td>
<td>88.33</td>
<td>178.58</td>
<td>m</td>
<td>1</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>BB</td>
<td>BB</td>
</tr>
<tr>
<td>6</td>
<td>58</td>
<td>131.91</td>
<td>m</td>
<td>1</td>
<td>BB</td>
<td>SB</td>
<td>SB</td>
<td>SB</td>
<td>SB</td>
</tr>
<tr>
<td>7</td>
<td>78.06</td>
<td>126.13</td>
<td>m</td>
<td>1</td>
<td>SB</td>
<td>SB</td>
<td>BB</td>
<td>SS</td>
<td>SS</td>
</tr>
<tr>
<td>8</td>
<td>65.31</td>
<td>181.05</td>
<td>m</td>
<td>1</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>SB</td>
<td>SB</td>
</tr>
<tr>
<td>9</td>
<td>59.26</td>
<td>191.54</td>
<td>m</td>
<td>1</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>SS</td>
<td>SS</td>
</tr>
<tr>
<td>10</td>
<td>59.47</td>
<td>154.88</td>
<td>m</td>
<td>1</td>
<td>BB</td>
<td>BB</td>
<td>BB</td>
<td>SB</td>
<td>SB</td>
</tr>
<tr>
<td>11</td>
<td>65.63</td>
<td>184.12</td>
<td>m</td>
<td>1</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>SB</td>
<td>SB</td>
</tr>
<tr>
<td>12</td>
<td>38.64</td>
<td>133.05</td>
<td>m</td>
<td>1</td>
<td>SB</td>
<td>BB</td>
<td>SB</td>
<td>SB</td>
<td>SB</td>
</tr>
<tr>
<td>13</td>
<td>60.94</td>
<td>275.63</td>
<td>m</td>
<td>1</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>SB</td>
<td>BB</td>
</tr>
<tr>
<td>14</td>
<td>51.48</td>
<td>395.25</td>
<td>m</td>
<td>1</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>SB</td>
<td>BB</td>
</tr>
<tr>
<td>15</td>
<td>47.12</td>
<td>260.45</td>
<td>m</td>
<td>1</td>
<td>BB</td>
<td>SB</td>
<td>SB</td>
<td>BB</td>
<td>BB</td>
</tr>
</tbody>
</table>
Stupidest code ever

```
n <- ncol(data)
temp <- rep(FALSE, n)
for(i in 1:n) {
  temp[i] <- all(data[2,1:i]=="")
  if(!temp[i]) break
}
if(!any(temp)) stop("...")
n.phe <- max((1:n)[temp])
```
Open source means everyone can see my stupid mistakes
Open source means everyone can see my stupid mistakes

Version control means everyone can see every stupid mistake I’ve ever made
More typically bad code

The `scantwo()` function is 1446 lines long.

The related C code is 20% of the C code in R/qtl.
Baroque data structures

attr(mycross$geno[["X"]])$probs, "map")
Inconsistencies

plotPXG(mycross, marker="D5M250")

effectplot(mycross, mname1="D5M250")
Useless warning messages

X'X is singular.
Tests
I don’t need tests; I have users.

– Me
If you use software that lacks automated tests, you are the tests.

– Jenny Bryan
It’s not that we don’t test our code, it’s that we don’t store our tests so they can be re-run automatically.

– Hadley Wickham
How did this happen?

- I didn’t know any better
- Not enough re-writing
- Not enough planning
- Personal weaknesses
- Trade-offs (e.g. between interface and data structures)
Documentation

Tailored tutorials

User guide

Examples

Formal documentation
User support

- No question is too insignificant
- Treat each question like it is the first
- Take a breath before responding
- “Send me your data, and I’ll try to figure out what’s going on.”
QTL mapping

![QTL mapping graph]

- **Chromosome**
- **LOD score**
- BB, BR, RR

Legend:
- LOD score range from 0.8 to 1.1
- Chromosome numbers from 1 to X
Congenic line
Advanced intercross lines

P
A
B

F2
F3
F4
F7
F10
Heterogeneous stock
R/qtl2
Now in 3D
High-density genotypes
High-dimensional phenotypes
Multi-parent populations
Linear mixed models
R/qtl2: Let’s not make the same mistakes

- C++ and Rcpp
- Roxygen2 for documentation
- Unit tests with testthat
- A single “switch” for cross type
R/qtl2: Let’s not make the same mistakes

- C++ and Rcpp
- Roxygen2 for documentation
- Unit tests with testthat
- A single “switch” for cross type
- Yet another data input format
- Flatter data structures, but still complex
Why devote such effort to R/qtl?

- For my own research
- So that others will use my methods ideas
- Has led to numerous collaborations
- I like to be useful
Acknowledgments

Danny Arends
Gary Churchill
Nick Furlotte
Dan Gatti
Ritsert Jansen
Pjotr Prins
Śaunak Sen
Petr Simecek
Artem Tarasov
Hao Wu
Brian Yandell

Robert Corty
Timothée Flutre
Lars Ronnegard
Rohan Shah
Laura Shannon
Quoc Tran
Aaron Wolen

NIH/NIGMS
R/qtl users
Slides: bit.ly/rstudio2019

kbroman.org

kbroman.org/qtl2

github.com/kbroman

@kwbrnoman