Fast, named capture regular expressions in R 2.14

Toby Dylan Hocking
toby.hocking@inria.fr
http://cbio.ensmp.fr/~thocking/

16 August 2011
Example: extract developer and project data from HTML

Developers:
- Toby Dylan Hocking
- Claudia Beleites
- Julien Moey
- Keith Ponting
- Philippe Grosjean
- Thomas Wutzler

[View Members]
How to extract user ids and names from HTML?

Data:

- <a href="https://r-forge.r-project.org/users/tdhock/">Toby Dylan Hocking</a>
- <a href="https://r-forge.r-project.org/users/kmpont/">Keith Ponting</a>

...  

Want: table of extracted information.

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>tdhock</td>
<td>Toby Dylan Hocking</td>
</tr>
<tr>
<td>kmpont</td>
<td>Keith Ponting</td>
</tr>
</tbody>
</table>

...
Solution: extract data using capturing regular expressions

<a href="https://r-forge.r-project.org/users/tdhock/">Toby Dylan Hocking</a>

Capturing regular expression:

<code>\([^<]+\)</code></a>

Named capture regular expression:

<code>\(<?id>[^<>]+\)/</code></a>

<table>
<thead>
<tr>
<th></th>
<th>R 2.13</th>
<th>R 2.13</th>
<th>R 2.14</th>
</tr>
</thead>
<tbody>
<tr>
<td>whole match</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>capture</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>fast C code</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>named capture</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Introduction: regular expressions in R 2.13 give you the position and length of the entire match, not groups!

```r
> u <- "http://r-forge.r-project.org/projects/inlinedocs"
> html <- paste(readLines(u), collapse="\n")
> pattern <-
+  paste('<a href="https://r-forge.r-project.org/users/",
+     '([^-]+)/">', # capture group for user id
+     '([^-]+)', # capture group for user name
+     '</a>', sep="")
> gregexpr(pattern, html)[[1]]

[1] 14241 14372 14455 14531 14608 14693
attr("match.length")
[1] 76 77 70 71 79 77

> named.p <-
+  paste('<a href="https://r-forge.r-project.org/users/",
+     '(?<id>[^/]+)/">', # named capture group
+     '(?<name>[^<]+)', # named capture group
+     '</a>', sep="")
```
Perl-Compatible Regular Expressions in R 2.14

> gregexpr(pattern,html,perl=TRUE)[[1]]

[1] 14241 14372 14455 14531 14608 14693
attr("match.length")
[1] 76 77 70 71 79 77
attr("capture.start")

[,1] 14286 14295
[,2] 14417 14429
[,3] 14500 14509
[,4] 14576 14585
[,5] 14653 14666
[,6] 14738 14752
attr("capture.length")

[,1] 6 18
[,2] 9 16
[,3] 6 12
[,4] 6 13
[,5] 10 17
[,6] 11 14
attr("capture.names")
Capture names can be used to identify groups

```r
> gregexpr(named.p,html,perl=TRUE)[[1]]

[1] 14241 14372 14455 14531 14608 14693

attr(,"match.length")
[1] 76 77 70 71 79 77

attr(,"capture.start")

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14286 14295</td>
</tr>
<tr>
<td>2</td>
<td>14417 14429</td>
</tr>
<tr>
<td>3</td>
<td>14500 14509</td>
</tr>
<tr>
<td>4</td>
<td>14576 14585</td>
</tr>
<tr>
<td>5</td>
<td>14653 14666</td>
</tr>
<tr>
<td>6</td>
<td>14738 14752</td>
</tr>
</tbody>
</table>

attr(,"capture.length")

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6 18</td>
</tr>
<tr>
<td>2</td>
<td>9 16</td>
</tr>
<tr>
<td>3</td>
<td>6 12</td>
</tr>
<tr>
<td>4</td>
<td>6 13</td>
</tr>
<tr>
<td>5</td>
<td>10 17</td>
</tr>
<tr>
<td>6</td>
<td>11 14</td>
</tr>
</tbody>
</table>
```
stringr::str_match_all extracts groups using R code

```r
> str_match_all(html,pattern)[[1]]

[,1]
[1,] "<a href="https://r-forge.r-project.org/users/tdhock/"
[2,] "<a href="https://r-forge.r-project.org/users/cbeleites/"
[3,] "<a href="https://r-forge.r-project.org/users/jmoeys/"
[4,] "<a href="https://r-forge.r-project.org/users/kmpont/"
[5,] "<a href="https://r-forge.r-project.org/users/phgrosjean/"
[6,] "<a href="https://r-forge.r-project.org/users/tomaschwutz/"

[,2] [,3]
[1,] "tdhock" "Toby Dylan Hocking"
[2,] "cbeleites" "Claudia Beleites"
[3,] "jmoeys" "Julien Moeys"
[4,] "kmpont" "Keith Ponting"
[5,] "phgrosjean" "Philippe Grosjean"
[6,] "tomaschwutz" "Thomas Wutzler"
```
A function based on the new C code in R 2.14

```r
> str_match_all_perl(html,
+    named.p)[[1]]

[1,] "<a href="https://r-forge.r-project.org/users/tdhock/"
[2,] "<a href="https://r-forge.r-project.org/users/cbeleites/"
[3,] "<a href="https://r-forge.r-project.org/users/jmoeys/"
[4,] "<a href="https://r-forge.r-project.org/users/kmpont/"
[5,] "<a href="https://r-forge.r-project.org/users/phgrosjean/"
[6,] "<a href="https://r-forge.r-project.org/users/tomaschwutz/"

id       name
[1,] "tdhock"  "Toby Dylan Hocking"
[2,] "cbeleites" "Claudia Beleites"
[3,] "jmoeys"   "Julien Moeys"
[4,] "kmpont"  "Keith Ponting"
[5,] "phgrosjean" "Philippe Grosjean"
[6,] "tomaschwutz" "Thomas Wutzler"
```
The new group parsing in C is 10x faster!

```r
> system.time(replicate(1000,{
+   str_match_all(html,pattern)
+ })))

  user  system elapsed
6.290   0.020   6.315

> system.time(replicate(1000,{
+   str_match_all_perl(html,pattern)
+ })))

  user  system elapsed
0.460   0.010   0.472
```
New group extraction is 10x faster than existing methods for extracting the first substring!

Text to extract:

<a href="https://r-forge.r-project.org/users/tdhock/">Toby Dylan Hocking</a>

Registered: 2009-07-29 14:37

time.method("users/","[^]/+")

<table>
<thead>
<tr>
<th>seconds</th>
<th>result</th>
<th>seconds</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>stringr</td>
<td>3.252</td>
<td>tdhock</td>
<td>stringr</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2009-07-29</td>
</tr>
<tr>
<td>gsub</td>
<td>0.761</td>
<td>tdhock</td>
<td>gsub</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2009-07-29</td>
</tr>
<tr>
<td>lookbehind</td>
<td>0.806</td>
<td>tdhock</td>
<td>lookbehind</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2009-07-29</td>
</tr>
<tr>
<td>R.2.14</td>
<td>0.078</td>
<td>tdhock</td>
<td>R.2.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2009-07-29</td>
</tr>
</tbody>
</table>
Efficient algorithms crucial for processing more data

![Graph showing the total number of projects on R-Forge from 2007 to 2011. The graph shows a significant increase in the number of projects after the R-Forge announcement at useR 2008 on 30 May 2011. The total number of projects at the announcement was 1017.]
Extracted developer and project data shows collaboration frequency in R-Forge projects

<table>
<thead>
<tr>
<th>Project</th>
<th>Developers</th>
<th></th>
<th>Developers</th>
<th>Number of projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>cvt</td>
<td>25</td>
<td>1</td>
<td>25</td>
<td>1</td>
</tr>
<tr>
<td>rmetrics</td>
<td>22</td>
<td>2</td>
<td>22</td>
<td>2</td>
</tr>
<tr>
<td>phyloc</td>
<td>22</td>
<td>1</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>phylobase</td>
<td>16</td>
<td>1</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>phylohelper</td>
<td>13</td>
<td>1</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>mlr</td>
<td>12</td>
<td>2</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>genabel</td>
<td>12</td>
<td>3</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>yuima</td>
<td>11</td>
<td>4</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>rsiena</td>
<td>11</td>
<td>4</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>flr</td>
<td>10</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>distr</td>
<td>10</td>
<td>20</td>
<td>6</td>
<td>20</td>
</tr>
<tr>
<td>blotter</td>
<td>10</td>
<td>34</td>
<td>5</td>
<td>34</td>
</tr>
<tr>
<td>sedar</td>
<td>9</td>
<td>54</td>
<td>4</td>
<td>54</td>
</tr>
<tr>
<td>diseasemapping</td>
<td>9</td>
<td>114</td>
<td>3</td>
<td>114</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>254</td>
<td>2</td>
<td>254</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>513</td>
<td>1</td>
<td>513</td>
</tr>
</tbody>
</table>
Use regular expressions for fast and easy text processing!

Example to match:

```html
<a href="https://r-forge.r-project.org/users/tdhock/">
Toby Dylan Hocking </a>
```

Named capture regular expression:

```html
<a href="https://r-forge.r-project.org/users/(?<id>[^/]+)/\(?<name>[^<]+\)"/>
```

Available R functions:

<table>
<thead>
<tr>
<th></th>
<th>R 2.13</th>
<th>R 2.13</th>
<th>R 2.14</th>
</tr>
</thead>
<tbody>
<tr>
<td>gregexpr()</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>str_match_all()</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>whole match</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>groups</td>
<td>✓</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>fast C code</td>
<td>✓</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>named groups</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Conclusion: faster, easier text processing in R 2.14

► Before the 2.14 release, you can download and compile

to get access to the new gregexpr().

► After: str_match_all_perl() function in the stringr package?

► Slides and Sweave source available on my web page:
http://cbio.ensmp.fr/~thocking/

► Questions? Contact me directly: toby.hocking@inria.fr