KNIME and R
The best of two worlds

For questions and suggestions please contact education@knime.com
Overview

Meet: R (Interactive)

The new nodes

Integration with KNIME

A real world application
Why use KNIME and R?

**R**
- Powerful statistics
- Leading edge algorithms
- Powerful/flexible graphics
- Widely accepted language

**KNIME**
- Powerful user interface
- Designed for big data
- Integrates com and org tools
- Enterprise grade solutions

- Open source analytics
- Cross platform
- Vibrant communities
R in KNIME: 3 ways to play...

- Community (RServe Integration)
- Core (Deprecated soon)
- R Interactive (Today's topic)
Overview of R (Interactive)

- Different input and output options
- Grey ports enable workspace branching
The Interactive Editor

- Columns
- Variables
- Code Editor
- Workspace Overview
- Console
Templates

Dialog - 0.58 - R Snippet (Cumulative Sum)

Categories:
- Column List
- Description
- Flow Variable List

Column List:
- Sepal.Length
- Sepal.Width
- Petal.Length
- Petal.Width
- Species

Description:
Use built-in functions to generate cumulative statistics (cumulative sum, cumulative min, cumulative max etc.) for a KNIME Table column.

Flow Variable List:
- knime workspace

R Script:
1. Some examples for how to generate cumulative metrics for a column of data.
2. Reference a column in your table here.
3. Column = knime.in"<Column Name>"
4. Add the cumulative sum of all values in your column:
   knime.in"cumsum" <- cumsum(column)
5. The cumulative product of all values in your column:
   knime.in"cumprod" <- cumprod(column)
6. Add the min for all values in your column from the start of the vector up to the current position:
   knime.in"cummin" <- cummin(column)
7. Add the max for all values in your column from the start of the vector up to the current position:
   knime.in"cummax" <- cummax(column)
8. knime.out <- knime.in

Copyright © 2013 by KNIME.com AG. All rights reserved.
Node: R Source

- Get data from an R data frame
- Assign output to knime.out
- Use with foreign, RCurl, or ...
Node: R Snippet

```
# Reference a column in your table here.
column = knime.in$"Sepal.Length"

data = knime.in

# Add the cumulative sum of all values in your column
data$"cumsum(Sepal.Length)" <- cumsum(column)

knime.out <- data
```

- Generic data manipulation
- Derive knime.out from knime.in
- Use with grep(), plyr, or ...
Nodes: R Mining

- Use R models in KNIME
- Learner & Predictor motif
- PMML support for portability
Nodes: R View

- Generic R plots
- Plot(knime.in)
- Use with many packages including ggplot2
Metanodes and R: Quickforms

R Script
1. require(ggplot2)
2. require(grid)
3. 
4. # Select data according to quickforms here.
5. x = knime.in[knime.flow.in["x"]]
6. y = knime.in[knime.flow.in["y"]]
7. pcolor = knime.in[knime.flow.in["color"]]
8. pshape = knime.in[knime.flow.in["shape"]]
9. title = knime.Flow.in["title"]
10. x_title = knime.Flow.in["x"]
11. y_title = knime.Flow.in["y"]
12. 
13. # Retype for ggplot2
14. xn = as.numeric(as.matrix(x))
15. yn = as.numeric(as.matrix(y))
16. colorf = as.factor(as.matrix(pcolor))
17. shapef = as.factor(as.matrix(pshape))

18. 
19. # define a plot theme
20. 21. clean_theme = theme(panel.background = element_blank(),
22. panel.border = element_rect(color = "black", linetype = "solid", fill = "transparent"),
23. axis.title.x = element_text(size=14, face="italic", colour = "black"),
24. axis.title.y = element_text(size=14, face="italic", colour = "black"),
25. legend.text = element_text(size=12, face="italic", colour = "black"),
26. panel.grid = element_blank())
27. 
28. # Generate a plot and apply the theme
29. ggplot(xn, yn, color = colorf, shape = shapef, xlab = x_title, ylab = y_title, main = title) + clean_theme
Metanodes and R: Deployment

- Configure yields simple dialog
- Share (Email/TeamSpace/Server)
- Deploy to web (KNIME Webportal)
Embedding plots in BIRT

- Generate plots in R
- Send to BIRT
EQPOL Data with Bioconductor I

- External Quality Assurance Program Oversight Laboratory
- NIH, NIAID, DAIDS program for QA of HIV/AIDS research
- Can machine learning automate some manual analysis?
- Problem: Lots of real data (~100,000,000 rows)
- Bioconductor provides flowCore to make this easier

R Script

```r
library(flowCore)

# Read data
fcs = read.FCS(knime.flow.in[['Location']])
labels = read.csv(knime.flow.in[['File path']], header = FALSE)

# An exotic transform:
# Estimate parameters
lgcl = estimateLogicle(fcs,c("FITC", "PE", "APC", "PerCP"))

# Apply
logiclefcs = transform(fcs,lgcl)

f = as.data.frame(exprs(fcs))
# Format and send data to knime.out
def <- cbind(f,labels)
knome.out <- df
```
EQPOL Data with Bioconductor II
Thank you

education@knime.com