The Gaggle

Paul Shannon

October 3, 2007

The day-to-day practice of molecular biology requires the exploration of many kinds of data. No single software tool, web site, or combination of Bioconductor packages, can do justice to these data. Furthermore – and despite significant effort having been devoted to integrating many kinds of data within single programs and web sites in recent years– the challenge presented by heterogeneity of biological data is only likely to increase, as are the the number of useful programs and web sites for exploring that data.

The Gaggle (Shannon et al. 2006) tackles this heterogeneity by providing a simple mechanism for broadcasting data among properly *gaggled* programs. And, contrary to expectation, careful semantic mapping is *not* required for these broadcasts to be useful.

In the Gaggle, the data types are distilled versions of data types commonly used in bioinformatics (and, indeed, in many other scientific fields). They are essentially free of biological semantics, but they take on rich semantics when they are interpreted by the receiving program. For instance: a simple list of (gene) names may be used to select rows of a matrix in R, nodes in a Cytoscape network, metabolic pathways in KEGG, and protein-protein associations in EMBL's STRING. This works equally well for the other data types – matrices, networks, and associative arrays (about which more below).

The Gaggle is open source and written in Java. We rely upon (and are grateful for) the R package *rJava* for Java/R integration. Further information about the Gaggle may be found at **http://gaggle.systemsbiology.org** and in the references.

The current vignette illustrates the Gaggle with two simple examples. In the first – and assuming you have installed the gaggle toolbar Firefox extension, for which we provide instructions – you will broadcast four Entrez GeneIDs to KEGG. In the second exapmle, you will create a random edge graph in R, broadcast it to Cytoscape for display, followed by broadcasting selected node names back and forth.

In the first example, you can link your R analysis to a polished and useful website. In the second, you can visualize and make selections upon data using an open source bioinformatics program. While these two particular 'geese' may not answer your current analytical needs, we hope that they demonstrate how loose-coupling and semantic flexiblity can add extra capabilities to Bioconductor.

The Gaggle's four **data types** are translated into R as follws:

- name list (mapped to an R character list)

1

- matrices (R matrix)

- networks (GraphNEL object)

- associative arrays (R environment)

'***Geese***' are programs or web resources adapted to run the Gaggle. They are typically written independent of the Gaggle (as with R), and then adapted to the Gaggle with a modest amount of programming. (See the paper and website for more details.) Some current geese are:

- the '*Firegoose*' – the popular web browser extended with a gaggle toolbar and broadcast capabiliites

- Cytoscape (see http://www.cytoscape.org)

- TIGR Mev (see http://www.tm4.org/mev.html)

- a variety of name translators

We encourage you to visit this vignette's companion website (see url just below). It presents more background, several demos beyond the one presented here, and Java Web Start links from which you can (with one click) download and run all of the necessary geese. This includes the **Gaggle Boss**, which must always be started **first** in any Gaggle session you run.

<div align="center">

http://gaggle.systemsbiology.org/R/vignettes/1

</div>

We have recently (Spring of 2007) released a gaggle toolbar extension to the popular Firefox web browser. Once installed (see link below for installation details) you can broadcast Entrez GeneID's to the browser, and thence to KEGG, Entrez Gene, EMBL STRING and DAVID websites; other websites can added with with little difficulty. In selected cases (KEGG and STRING) information may be broadcast back to R.

<div align="center">

http://gaggle.systemsbiology.org/docs/geese/firegoose

</div>

1   Technical Background and Notes

The **Gaggle** is a simple, open-ended collection of RMI-linked Java programs, which broadcast selected data to each other at the user's behest. These broadcasts are managed by a simple RMI server, the **Gaggle Boss**. Though it is not stricly necessary, we find it very convenient to launch most geese via Java Web Start links in a web page. The **R goose**, of course, is an exception: simply install the gaggle package as you would any other R or Bioconductor package. Please see the companion website for a generally useful set of Java Web Start links.

You **must** always start a Gaggle Boss on your computer before you start any geese. Every goose registers with this boss as it starts up; if it isn't running, you get no gaggle capabilities.

Upon receiving a broadcast, each goose interprets the data according to its own, local semantics. (This strategy of **semantic flexibility** is discussed at length in the Gaggle paper.) When the gaggle-enabled Firefox broadcasts a list of gene names to **KEGG** for example, after having received those names from **R**, KEGG responds to by displaying a list of the metabolic pathways to which they have been annotated; having no sensible interpretation of matrices or networks, KEGG will simply ignores those broadcasts. **STRING** presents a web page for the discovery of protein associations; the resulting network may broadcast back to the Gaggle for further analysis.

2   Demo 1: Broadcast Entrez GeneIDs to KEGG

In this first demonstration, we

- Start the Gaggle Boss (browse to `http://gaggle.systemsbiology.org/R/vignettes/1`, Part 1, for web start links)

- Start R, and load the gaggle package

- Broadcsast a few human geneID's to KEGG using a gaggle-enabled Firefox browser

```
> setSpecies("Homo sapiens")
> geese()
> setTargetGoose("Firegoose")
> ids = c("9475", "2316", "5747", "10298")
> broadcast(ids)
```

In Firefox, do the following:

- In the gaggle toolbar, towards the left, is a region titled *Gaggle Data* with a pulldown menu which, after your broadcast, should read *NameList(4) from Gaggle*.
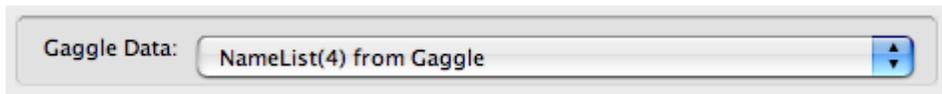
- Also in the toolbar, but towards the right, in a region titled *Target* you will see another pulldown menu. Select *KEGG* and then click on *Broadcast*.
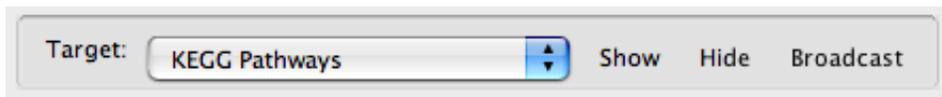
- In a moment, a list of KEGG pathways will appear in your browser

3

3 Demo 2: Broadcast a randomEGraph to Cytoscape for viewing; broadcast selected nodes back to R

Here we

- Start the Gaggle Boss (browse to `http://gaggle.systemsbiology.org/R/vignettes/1`, Part 1, for web start links)

- Start Cytoscape

- Start R, and load the gaggle package

- Create a randomEGraph, broadcast it, and see it displayed in Cytoscape

- Select a few nodes of the graph in Cytoscape, and broadcast them back to R

- Broadcast these selected nodes back to R again, but this time, not as a list of node names, but as a connected subgraph (that is, with edges included)

```
> library(gaggle)
> gaggleInit()
> set.seed(123)
> g = randomEGraph(LETTERS[1:8], edges = 10)
> broadcast(g)
```

You should see an 8-node, 10-edge network appear in cytoscape.

Switch your focus to Cytoscape, and select a few nodes by drag-selecting with your left mouse button. Then look near the top of the Cytoscape window for the broadcast buttons (**S H B N**):
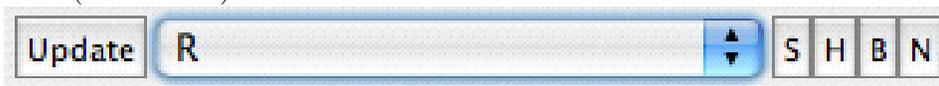
These stand for **S**how, **H**ide, **B**roadcast names, and broadcast **N**etwork, respectively. The *target* of these actions is picked by manipulating the 'goose selection menu' which shows **R** in the illustration. (See below for how to broadcast and select target geese using R function calls.) If the **Boss** is the target, then your broadcasts are sent to *all* of the geese who are registered with the Boss – though one can manipulate the user interface of the Boss so that it only forwards messages to selected geese).

Click the 'Update' button to ensure that your goose has a fresh list of all currently running geese for you to choose from.

Inasmuch as the R goose does not have a full-fledged graphical user interface, function calls must be used instead of buttons and menus to interact with the Gaggle. Here are the relevant commands:

- **geese ()** names of the current geese

- **setTargetGoose (someGooseName)** one of the names returned by 'geese ()'

- **getTargetGoose ()** find out the current setting

- **broadcast (someVariable)** this generic function suffices for name lists, graphs, matrices, environments the broadcast goes to the current targetGoose, or to the Boss by default.

- **showGoose ()** raise the window of the current target goose

- **hideGoose ()** hide the window of the current target goose

When using a new goose with which you are unfamiliar, you can often learn your way around from the tooltips associated with buttons in most GUI geese. These 'flyover' explanations of otherwise tersely named buttons should help you to use the various geese.

Returning, now, to the Cytoscape goose, with at least a few selected nodes, broadcast this selection back to R. There are two kinds of broadcasts available to you: of just the names, or of the selected subgraph. Whichever you choose, your R console will display a message indicating the type and size of the broadcast it has just received. (These messages, unfortunately, are not displayed by the Windows Rgui application.) You must then call one of the following methods in order to assign this data to an R variable:

```
> selectedNodes = getNameList()
> subgraph = getNetwork()
```

You may also select nodes in the Cytoscape goose from R. (You may wish to clear the current selection, if any, in Cytoscape first, using the **Cl** button near the top center of the Cytoscape window). Then, in R:

```
> broadcast(c("B", "E"))
```

4   For More Information

For more information, and for more extensive demonstrations, please visit
http://gaggle.systemsbiology.org/R/vignettes/1
and the Gaggle website:
http://gaggle.systemsbiology.org

5 References

- Shannon P, Reiss DJ, Bonneau R, Baliga NS. The Gaggle: A system for integrating bioinformatics and computational biology software and data sources, *BMC Bioinformatics* 2006, 7:176.