

Using the BrainStars package

Itoshi Nikaido^{‡*} Takeya Kasukawa^{‡†}

March 2, 2012

[‡]The RIKEN Center for Developmental Biology

Contents

1	Overview of BrainStars package	2
2	Overview of BrainStars database	2
3	Getting Started using BrainStars package	2
3.1	Getting expression profiles by ProbeSet IDs	2
3.2	Getting figures	3
3.3	Keyword search	6
3.4	Getting a list of genes	6
3.4.1	Markers	6
3.4.2	Multistate and onestate genes	6
3.4.3	Retrieve a gene list by genes by gene family / categories	7
3.4.4	Inferred connections among CNS regions by neurotransmitter/neurohormone	7
4	BrainStars Web API	8
4.1	Search API	8
4.1.1	Samples	8
4.2	List API	8
4.2.1	Marker gene candidates (marker)	9
4.2.2	Multi-state genes (multistate)	9
4.2.3	One-state genes (onestate)	9
4.2.4	Inferred connections among CNS regions by neurotransmitter/neurohormone (ntnh)	10
4.2.5	Gene family / categories (genefamily)	10
4.2.6	Samples	11

*dritoshi@gmail.com

†kasukawa@gmail.com

4.3	Entry API	11
4.3.1	Samples	11
4.4	Content type	11
5	Conclusion	12

1 Overview of BrainStars package

The BrainStars package can search and get gene expression data and some kind of plots from BrainStars (B*). BrainStars is a quantitative expression database of the adult mouse brain. The database has genome-wide expression profile at 51 adult mouse CNS regions. The BrainStars database has a REST API to query gene expression data, annotations and some kind of figures written by Dr. Takeya Kasukawa. This package is wrapper for BrainStars REST API in R. BrainStars gene expression data, images and texts (excluding ABA data and images) are licensed under a Creative Commons Attribution 2.1 Japan License.

2 Overview of BrainStars database

BrainStars is a quantitative expression database of the adult mouse brain. The database has genome-wide expression profile at 51 adult mouse CNS regions.

For 51 CNS regions, slices (0.5-mm thick) of mouse brain were cut on a Mouse Brain Matrix, frozen, and the specific regions were punched out bilaterally with a microdissecting needle (gauge 0.5 mm) under a stereomicroscope. For each region, we took samples every 4 hours, starting at ZT0 (Zeitgeber time 0; the time of lights on), for 24 hours (6 time-point samples for each region), and we pooled the samples from the different time points. We independently sampled each region twice (n=2).

These samples were purified their RNA, and measured with Affymetrix GeneChip Mouse Genome 430 2.0 arrays. Expression values were then summarized with the RMA method. After several analysis with the expression data, the data and analysis results were stored in the BrainStars database.

3 Getting Started using BrainStars package

BrainStars package provides two major functions, `getBrainStars` and `getBrainStarsFigure`. The `getBrainStars` function is wrapper function of all BrainStars API, list, search and entry API. The `getBrainStarsFigure` is convenient function to get high-resolution plots.

3.1 Getting expression profiles by ProbeSet IDs

Getting expression data from BrainStars is easy. `getBrainStarsExpression` or `getBrainStars` requires gene expression profile and then parse the data into *ExpressionSet* object.

Usage is quite simple:

```
> library("BrainStars")
```

This loads the BrainStars library.

```
> my.eset <- getBrainStars(query = "1439627_at", type = "expression")
```

or

```
> my.eset <- getBrainStarsExpression("1439627_at")
```

Now, `my.eset` contains the R data structure (of class *ExpressionSet*) that represents the entry 1439627_at from BrainStars.

The function `getBrainStarsExpression` accepts vector of multiple probeset IDs.

```
> ids <- c("1439627_at", "1439631_at", "1439633_at")
```

```
> my.esets <- getBrainStarsExpression(ids)
```

```
> my.esets
```

```
ExpressionSet (storageMode: lockedEnvironment)
```

```
assayData: 3 features, 102 samples
```

```
  element names: exprs
```

```
protocolData: none
```

```
phenoData: none
```

```
featureData: none
```

```
experimentData: use 'experimentData(object)'
```

```
Annotation:
```

You can access the matrix of expression in the following method:

```
> my.mat <- exprs(my.esets)
```

3.2 Getting figures

We can retrieve some kind of barplots and brain map images from BrainStars.

```
> getBrainStarsFigure("1439627_at", "exprgraph", "png")
```

Save figure file at 1439627_at.exprgraph.png

```
> getBrainStarsFigure("1439627_at", "exprmap", "png")
```

Save figure file at 1439627_at.exprmap.png

```
> getBrainStarsFigure("1439627_at", "switchgraph", "pdf")
```

Save figure file at 1439627_at.switchgraph.pdf

```
> getBrainStarsFigure("1439627_at", "switchhist", "png")
```

Save figure file at 1439627_at.switchhist.png

You can find "id.type.png" files in local current directory. See detail of image file type in next section.

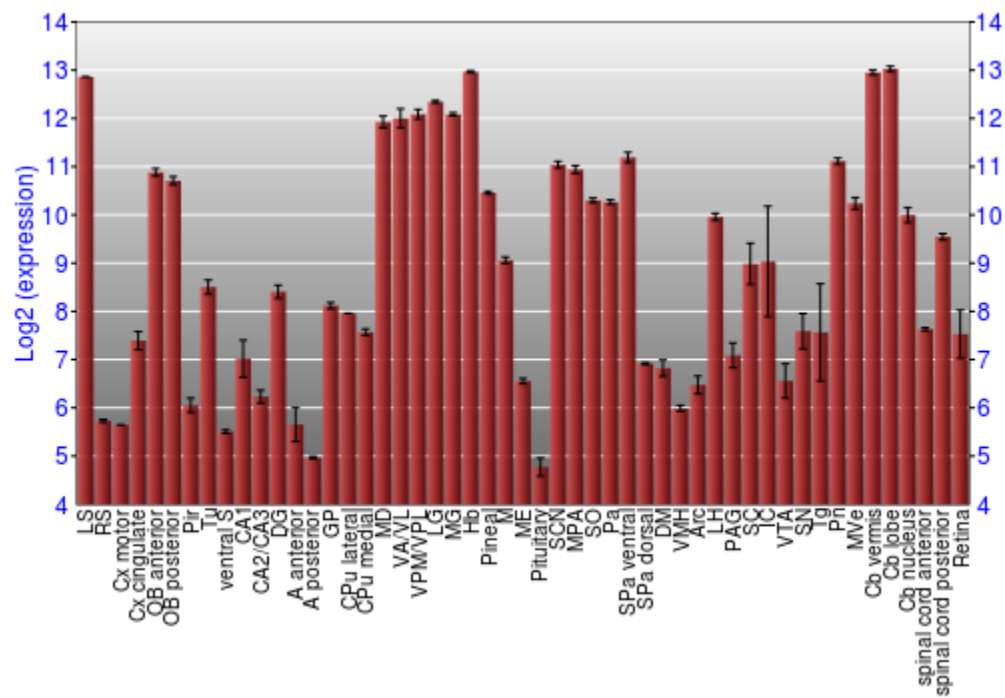


Figure 1: exprgraph

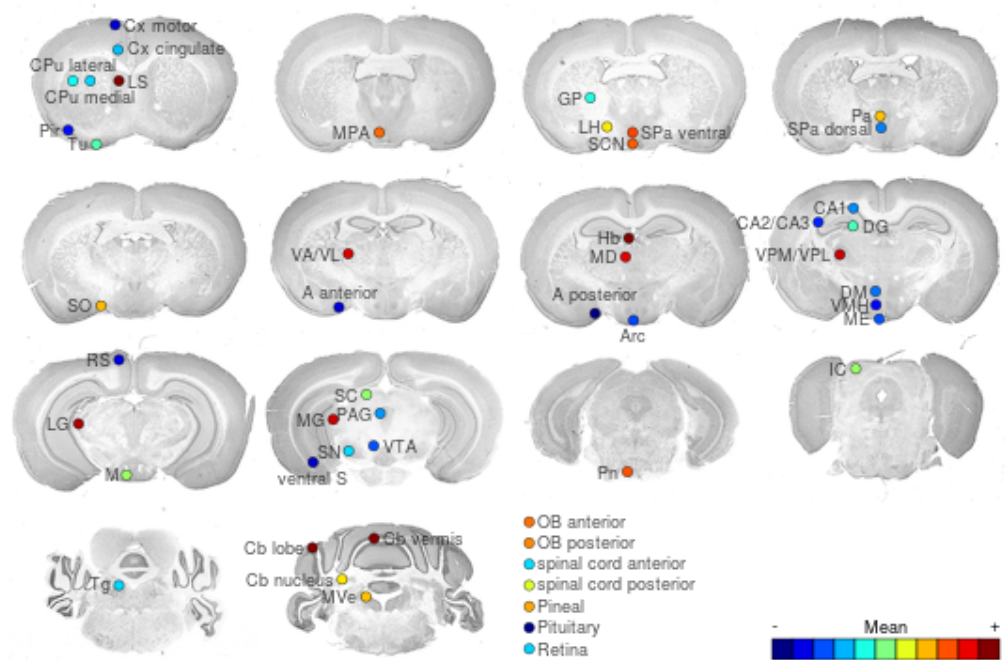


Figure 2: exprmap

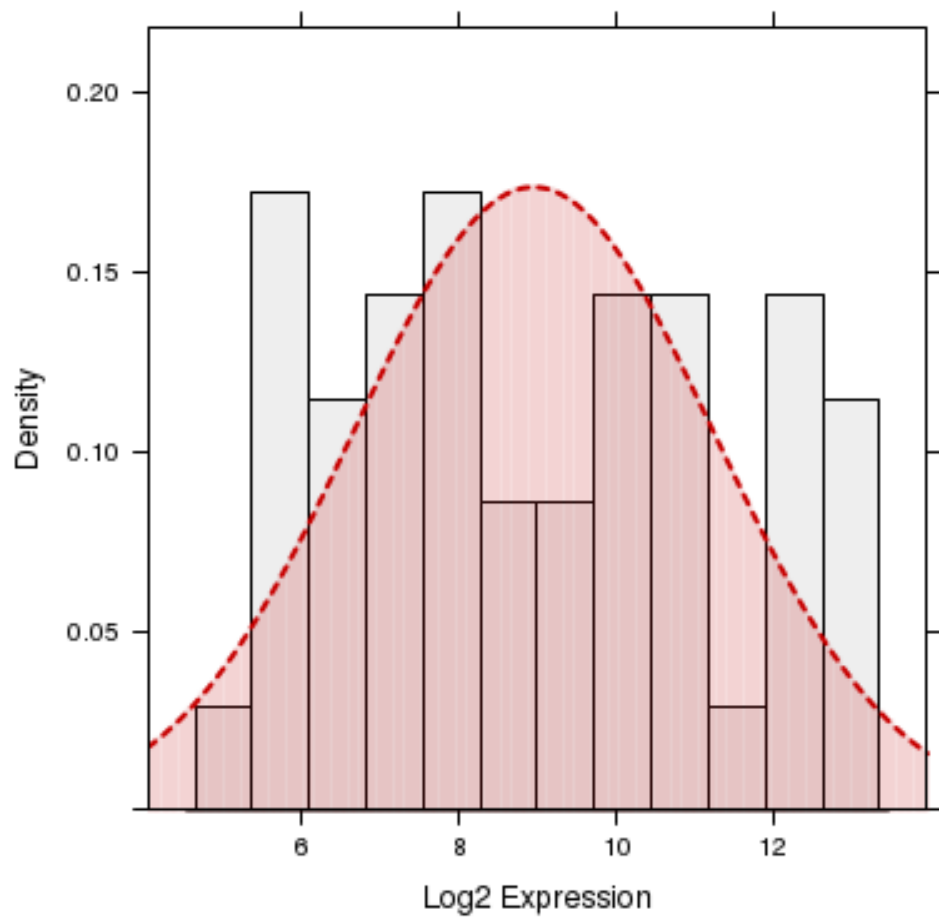


Figure 3: switchhist

3.3 Keyword search

The function `getBrainStars` provides keyword search in BrainStars. This function retrieves a gene list or count of genes.

```
> recep.mat <- getBrainStars(query = "receptor/10,5", type = "search")
> recep.count <- getBrainStars(query = "receptor/count", type = "search")
```

The useful ready-to-made functions were provided to extract "gene names", "gene symbols" and "probeset ids" from *RJSON* character object.

```
> recep.json <- getBrainStars(
+   query = "receptor/10,5",
+   type = "search",
+   output = "json"
+ )
> recep.gns <- geneNames(recep.json)
> recep.gss <- geneSymbols(recep.json)
> recep.ids <- probeSetIDs(recep.json)
```

See detail about query format in next section.

3.4 Getting a list of genes

BrainStars has results of some higher-order biological analysis. You can retrieve these information in following functions.

3.4.1 Markers

Marker genes were ones whose levels in a specific CNS region are higher (or lower) than in others.

```
> mk.genes.count <- getBrainStars(query = "high/LS/count", type = "marker")
```

or

```
> mk.genes.count <- getBrainStarsMarker("high/LS/count")
```

3.4.2 Multistate and onestate genes

Multi-state genes are ones with multi-modal expression patterns (i.e. expression of discretized states) or multi-modal patterns. For each probeset, CNS regions were grouped into the several "states" according to their expression values with variational Bayesian inference to fit to a gaussian mixture model. Multi-state genes, which have two or more states, were shown in the list. one-state genes, which have only one state, were not shown.

```
> ms.genes.list <- getBrainStars(query = "low/SCN/all", type = "multistate")
```

or

```
> ms.genes.list <- getBrainStarsMultistate("low/SCN/all")
```

One-state genes are expressed unimodally across the CNS regions, and approximately followed a log-normal distribution. Some one-state genes exhibited stable expression patterns, characterized by the log-normal distribution with a small variance, whereas others had more variable expression and showed the log-normal with a larger variance.

```
> os.genes.count <- getBrainStars(query = "count", type = "onestate")
```

or

```
> os.genes.count <- getBrainStarsOnestate("count")
```

3.4.3 Retrieve a gene list by genes by gene family / categories

The function provides lists of genes in gene families or in a gene categories.

```
> gfc.genes1.count <- getBrainStarsGeneFamCat("tf//count")
> gfc.genes2.list <- getBrainStarsGeneFamCat("tf/terminal/all")
> gfc.genes3.count <- getBrainStarsGeneFamCat("tf/terminal/count")
```

or

```
> gfc.genes1.count <- getBrainStars(query = "tf//count", type = "genefamily")
> gfc.genes2.list <- getBrainStars(query = "tf/terminal/all", type = "genefamily")
> gfc.genes3.count <- getBrainStars(query = "tf/terminal/count", type = "genefamily")
```

3.4.4 Inferred connections among CNS regions by neurotransmitter/neurohormone

In the CNS, various neurohormones (NHs) and neurotransmitters (NTs) are secreted from neurons to convey information among distinct regions. The function get inferred interconnections among CNS regions by expression data (especially multi-state expression) for NH and NT (NH/NT) genes. To analyze the expression patterns of the NH/NT genes, a list that included the genes for the ligands themselves and those for enzymes that were rate-limiting in the biosynthesis of these ligands was first made. Here both of these categories were termed as "ligand" genes. the genes for NH/NT receptor proteins (i.e., "receptor" genes) were also included. Next, a list of connection among CNS regions that a state of a ligand gene of a NH/NT is "on" (or "up") in one region, and a state of a receptor gene of the NH/NT is "on" (or "up") in the other region was made.

```
> os.genes <- getBrainStars(query = "high/SCN/ME/all", type = "ntnh")
```

or

```
> os.genes <- getBrainStarsNtNh("high/SCN/ME/all")
```

4 BrainStars Web API

The BrainStars database has a REST-like Web API interface for accessing from your Web applications. This document shows how to access the database via our Web API. We have the following Web APIs:

- Search API
- List API
- Probe set API

You can get contents in HTML, JSON, YAML, and so on. (See below)

4.1 Search API

Search API is for keyword search and is based on Tokyo Manifesto and TogoWS REST interface. URL for retrieving a list of hit entries:

```
http://brainstars.org/search/(query+string)[/(offset),(limit)]
```

If the result has at least one hit entries, a list of entry IDs is returned in text/plain format (each line corresponds to each hit entry). If not, 404 Not found error code is returned. The output format can be changed to JSON, and so on (see below). "offset,limit" can be used to retrieve a part of hit entries. If "offset,limit" is not given, all hits are returned.

URL for retrieving the count of hit entries:

```
http://brainstars.org/search/(query+string)/count
```

The count of hit entries is returned in text/plain format.

4.1.1 Samples

```
http://brainstars.org/search/receptor
http://brainstars.org/search/receptor/1,5
http://brainstars.org/search/receptor/count
```

4.2 List API

List API is for retrieving a list of entries, such as genes in a specific gene category.

```
http://brainstars.org/(type)[/(category)[/(subcategory)[/...]]/(offset),(limit)
http://brainstars.org/(type)[/(category)[/(subcategory)[/...]]/all
http://brainstars.org/(type)[/(category)[/(subcategory)[/...]]/count
```

You can specify "type" in the following list:

- "marker": marker gene candidates
- "multistate": multi-state genes
- "onestate": one-state genes
- "ntnh": inferred connections among CNS regions by neurotransmitter/neurohormone
- "genefamily": gene family / categories

4.2.1 Marker gene candidates (marker)

List of entries and Count of entries:

`http://brainstars.org/marker/{high,low}/(region)/(offset),(limit)`

`http://brainstars.org/marker/{high,low}/(region)/count`

`{high,low}`

- "high": highly expressed regions
- "low": low expressed regions

`(region)`: CNS region

4.2.2 Multi-state genes (multistate)

List of entries and Count of entries:

`http://brainstars.org/multistate/{high,up,low,down}/(region)/(offset),(limit)`

`http://brainstars.org/multistate/{high,up,low,down}/(region)/count`

`{high,up,low,down}`

- "high": high state regions
- "up": up state regions
- "low": low state regions
- "down": down state regions

`(region)` CNS region

4.2.3 One-state genes (onestate)

List of entries and Count of entries:

`http://brainstars.org/onestate/(offset),(limit)`

`http://brainstars.org/onestate/count`

4.2.4 Inferred connections among CNS regions by neurotransmitter/neurohormone (ntnh)

List of entries and Count of entries:

[http://brainstars.org/ntnh/{high,low}/\(ligand-region\)/\(receptor-region\)/\(offset\),\(limit\)](http://brainstars.org/ntnh/{high,low}/(ligand-region)/(receptor-region)/(offset),(limit))

[http://brainstars.org/ntnh/{high,low}/\(ligand-region\)/\(receptor-region\)/count](http://brainstars.org/ntnh/{high,low}/(ligand-region)/(receptor-region)/count)

{high,low}

- "high": high state regions
- "up": up state regions

(ligand-region) Ligand CNS region

(receptor-region) Receptor CNS region.

4.2.5 Gene family / categories (genefamily)

List of entries and Count of entries:

[http://brainstars.org/genefamily/\(category\)/\(keyword\)/\(offset\),\(limit\)](http://brainstars.org/genefamily/(category)/(keyword)/(offset),(limit))

[http://brainstars.org/genefamily/\(category\)/\(keyword\)/count](http://brainstars.org/genefamily/(category)/(keyword)/count)

(category) gene family / category name

- "tf": transcription factors
- "transmem": transmembrane genes
- "channel": channel genes
- "gpcr": GPCR genes
- "adhesion": cell adhesion genes
- "excellmat": extracellular matrix genes
- "structural": structural protein genes
- "neurogenesis": neurogenesis related genes
- "hox": homeobox genes
- "nuc rept": nuclear receptor genes
- "ntnh": neurotransmitter/neurohormone genes
- "axon": axon guidance genes
- "fox": forkhead genes

(keyword) keyword. If no keyword search required, make this omitted or blank

4.2.6 Samples

```
http://brainstars.org/marker/low/SCN/all
http://brainstars.org/multistate/up/SCN/20,10
http://brainstars.org/onestate/count
http://brainstars.org/ntnh/high/SCN/ME/all
http://brainstars.org/genefamily/tf//1,20
http://brainstars.org/genefamily/tf//count
http://brainstars.org/genefamily/tf/terminal/all
http://brainstars.org/genefamily/tf/terminal/count
```

4.3 Entry API

Entry API is for obtaining information (annotation, expressions, links) about each entry

```
http://brainstars.org/probeset/(id)
```

4.3.1 Samples

```
http://brainstars.org/probeset/1450371_at
http://brainstars.org/probeset/1450371_at?content-type=application/json
```

4.4 Content type

How to designate a content type

You can get information in another format rather than default format (default: text/plain in search/list API; text/html in entry API).

Using a "content-type" parameter in the query string of the URI.

```
http://brainstars.org/marker/high/LS?content-type=application/json
```

Using an "Accept" header of your HTTP request.

```
$ curl -H "Accept: application/json" http://brainstars.org/marker/high/LS
$ wget --header "Accept: application/json" http://brainstars.org/marker/high/LS
```

Supported content type:

- JSON (text/x-json or application/json)
- YAML (text/yaml, text/x-yaml, or application/x-yaml)
- Perl Data::Dumper (text/x-data-dumper)
- Perl Data::Denter (text/x-data-denter)
- Perl Data::Taxi (text/x-data-taxi)

- Perl Storable (application/x-storable)
- Perl FreezeThaw (application/x-freezethaw)
- PHP Serialization (text/x-php-serialization)

5 Conclusion

The *BrainStars* package provides way to retrieve gene expression, annotation and result of higher-order biological data analysis in R from BrainStars database. To perform many useful functions of Bioconductor, *BrainStars* package is adopted Bioconductor data structure, such a *ExpressionSet*. These package will hopefully access BrainStars data more fully to the neuroscience community.