Epigenome Data BrowseR Documentation

Release 1.3 beta

Shraddha Pai, Kelly Sunderland

Apr 12, 2017

Contents

CHAPTER 1

Visit the online demo

Get **source code** from the EDB github repo.

Install instructions

Install environment

This step only needs to be done initially by a web administrator setting up the application. It does not have to be repeated by all end users. EDB should be installed on a machine on which you can run a web server accessible to all your users.

All EDB development has been done on Unix machines, but installation should work on any machine that can run Shiny server.

The recommended machine configuration is

- OS: the EDB has been tested on Debian, Ubuntu 10.04, 12.04 and 14.04, but any machine that runs Shiny server should work.
- > 7 Gb RAM
- > 2 Gb disk; 2.0G is the size of all software dependencies and the demo dataset.

These instructions are for a machine running Ubuntu 14.04 (LTS). I have also successfully set up the app on Debian Wheezy LTS. Each package will prompt saying something along the lines of "This package with require XYZ space. Continue [Y/n]?". Type "Y".

Automatic install using docker

Docker is a hugely popular platform for deploying software in "containers". EDB can be easily installed with a custom dockerfile in the EDB repo.

ocker

If you haven't already done so, first install docker on your target machine. Then build at command line.

Note: The `build` itself takes ~15-20 minutes, as it is installing R, the shiny server, associated BioConductor and Shiny dependences. As part of this process, the dockerfile also downloads the demo dataset (~286Mb).

So take a break and do something else while it's running.

```
git clone␣
↪https://github.com/shraddhapai/shiny-data-browseR.git
cd shiny-data-browseR
docker build docker_edb
```

When the build is complete, you should be provided with an image identifier (e.g. `af9ca39a5c48`).

Now start a new container from the image:

```
docker run -p 3838:3838 -d af9ca39a5c48
```

At this point you should be able to open a web browser on your target machine to `http://localhost:3838/EDB`, or if you're using boot2docker, to `http://192.168.59.103:3838/EDB`.

Manual install

Clone the EDB repo

Ignore the first line below if your machine already has git.

```
sudo apt-get install git
git clone␣
↪https://github.com/shraddhapai/shiny-data-browseR.git
```

Install R (>=3.1.0)

```
sudo add-apt-repository ppa:marutter/rrutter
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install r-base r-base-dev
```

The following will set text-encoding preferences so your R error/status messages don't contain strange symbols. In ~/.profile add the following line

```
export LC_ALL=C
```

Then execute ~/.profile:

```
. ~/.profile
```

Install shiny server

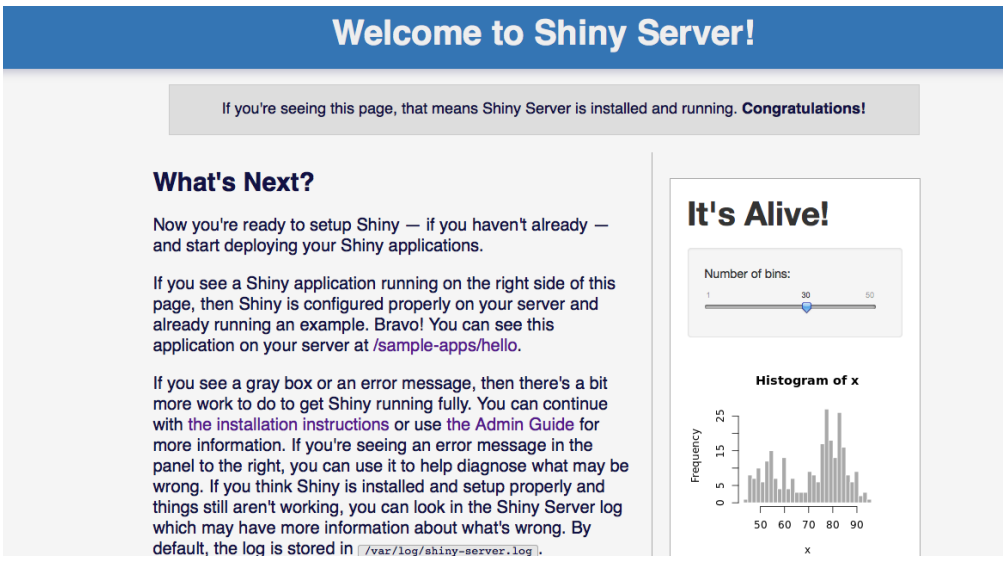
First install the R shiny package:

```
sudo su - \
-c "R -e \"install.packages(
↪'shiny', repos='http://cran.rstudio.com/')\""
```

Now download and install shiny-server:

```
sudo apt-get install gdebi-core
wget http://download3.rstudio.org/
↪ubuntu-12.04/x86_64/shiny-server-1.2.1.362-amd64.deb
sudo gdebi shiny-server-1.2.1.362-amd64.deb
```

At this point, if you point your web browser to <myIpAddress>:3838, you should see a page like this. If you don't see this, shiny server has not been correctly installed. Debug this issue first.



Install Unix dependencies for the R/BioConductor packages

```
sudo apt-get install libcurl4-openssl-dev libxml2-dev
```

Install R and BioConductor packages

Select your local mirror when prompted (e.g. 15 for Toronto)

```
R -e␣
↪"install.packages(c('shinyBS','doMC','RColorBrewer',
↪ 'matrixStats'), repos='http://cran.rstudio.com/'); "
```

Install BioConductor and package dependencies for the modified BioC package, "Gviz".

```
R -e "source('http://bioconductor.org/biocLite.R
↪'); biocLite(c('Biobase','GenomicRanges','rtracklayer
↪','GenomicFeatures','biovizBase','Rsamtools'));"
```

Install development version of Gviz

Exit R and insert the following at command line:

```
    wget http://www.bioconductor.org/
↪packages/devel/bioc/src/contrib/Gviz_1.11.2.tar.gz
R -e "install.packages('Gviz_1.11.2.tar.gz')"
```

Configure EDB

That's it! Installation is done. Now we configure the server.

```
cd /srv/shiny-server
mkdir EDB
sudo cp ~/shiny-data-browseR/*.R .
sudo cp -r ~/shiny-data-browseR/www .
sudo cp -r ~/shiny-data-browseR/data_types .
sudo touch restart.txt
```

Now create a file named "config_location.txt" with the path to all your data configuration files. Save it. There should be no spaces before or after the path.

```
/path/to/config/file/here
```

At this point, you should be able to see the EDB interface in your web browser. In the image below, config_location.txt points to a dummy directory with no config files. Therefore the dropdown box for "Select a dataset" is empty.

Great! Now let's proceed to adding our custom datasets.

Add a dataset

Quickstart: Use demo datasets

Download the demo datasets seen on the live EDB demo. Follow install instructions in the included `README`.

File: EDB_demodata.tar.gz (270Mb)

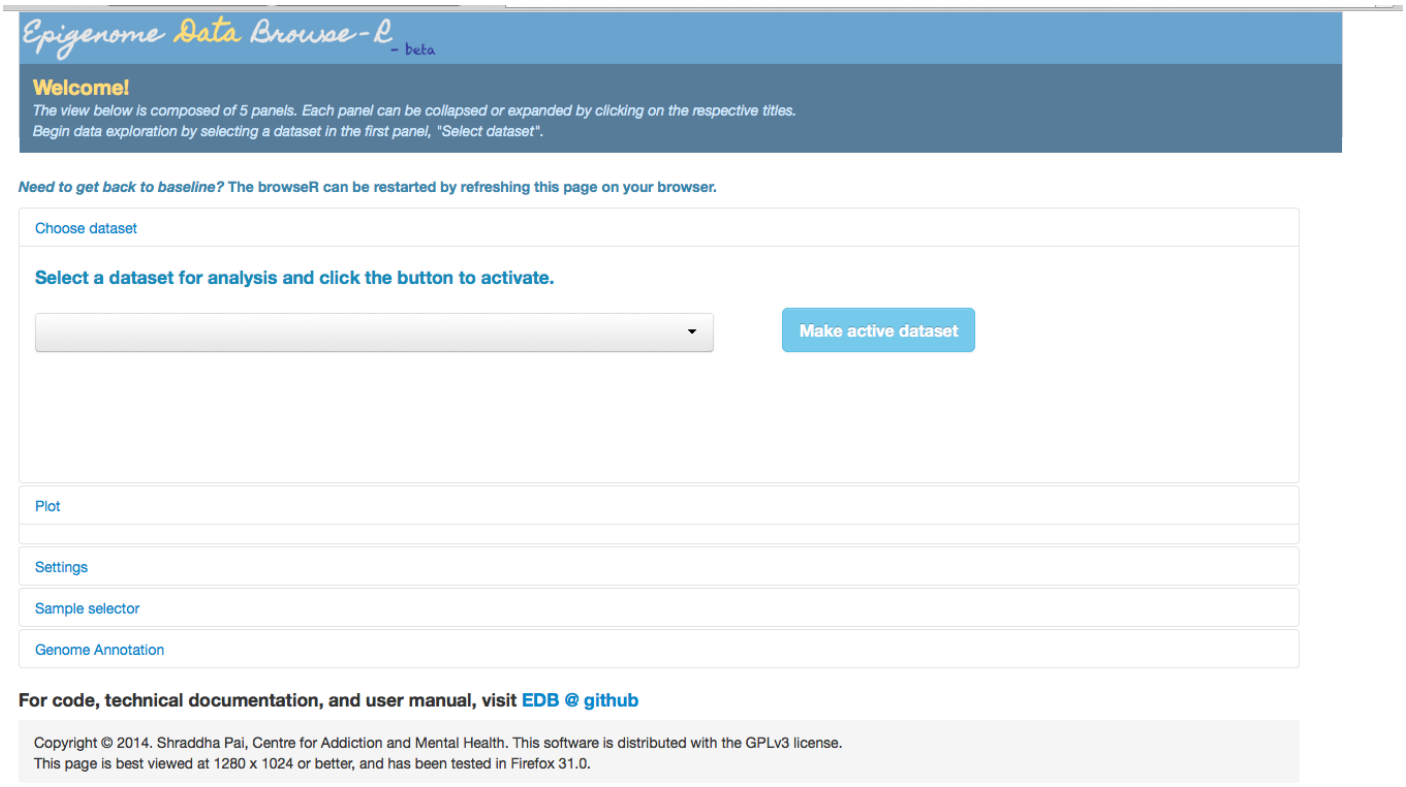
md5sum: `d529107a0ac482f4d39f70ea0e7546aa`

Note: If you installed EDB using the custom dockerfile, you already have the demo data.

Basic steps

Follow these steps to add a new dataset to the EDB. As an example, we use the demo BS-seq dataset from Lister, Mukamel et al. (2013). Science, which charts DNA methylation over human postnatal brain development.

Let us assume that EDB datasets are all located at `<dataRoot>`.



Step 1: dataset config: Create a *master config file* for the dataset.

Step 2: phenotype matrix: Add a *table specifying sample phenotypes/metadata properties*, and links to the sample-wise data

Step 3: data: Assemble the data files and any required index files (e.g. *tbi* for tabix files).

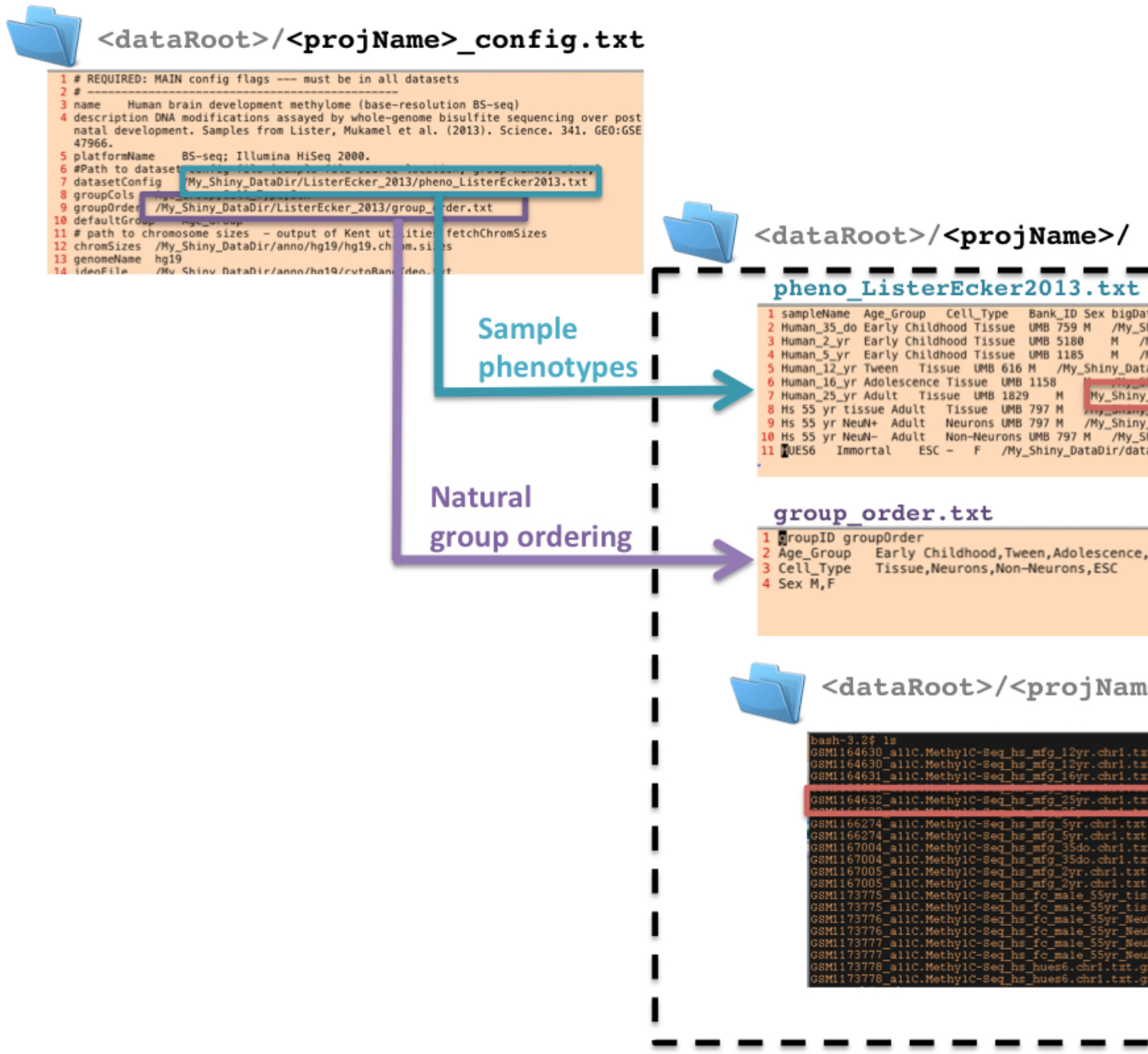
Step 4: group order: Add a file to specify the *ordering of categorical variables*

Example directory structure for datasets

This is what a simple project directory structure should look like when you are done. The example below assumes that all datasets and associated config files are located at `<dataRoot>`. Coloured arrows show how the master config file references the phenotype table and group order table.

In this example we have two datasets, one showing epigenetic dynamics in mouse brain development and the other, for human. Here, `config_location.txt` points to a directory which also contains all browser-related metadata and the data files themselves:

```
<config_location_points_here>
        |
        |---- mouseBrainDev_config.txt *symlink*
        |
        |---- humanBrainDNAMethylome_config.txt *symlink*
        |---- mouseBrainDev/
                        |---- config.txt
```


<dataRoot>/<projName>_config.txt
REQUIRED: MAIN config flags --- must be in all datasets
name Human brain development methylome (base-resolution BS-seq)
description DNA modifications assayed by whole-genome bisulfite sequencing over post natal development. Samples from Lister, Mukamel et al. (2013). Science. 341. GEO:GSE 47966.
platformName BS-seq; Illumina HiSeq 2000.
datasetConfig /My_Shiny_DataDir/ListerEcker_2013/pheno_ListerEcker2013.txt
groupOrder /My_Shiny_DataDir/ListerEcker_2013/group_order.txt
defaultGroup Age_Group
path to chromosome sizes - output of Kent utilities fetchChromSizes
chromSizes /My_Shiny_DataDir/anno/hg19/hg19.chrom.sizes
genomeName hg19
<dataRoot>/<projName>/
Sample phenotypes
pheno_ListerEcker2013.txt
sampleName Age_Group Cell_Type Bank_ID Sex bigDa
Human_35_do Early Childhood Tissue UMB 759 M /My_S
Human_2_yr Early Childhood Tissue UMB 5180 M
Human_5_yr Early Childhood Tissue UMB 1185 M
Human_12_yr Tween Tissue UMB 616 M /My_Shiny_Dat
Human_16_yr Adolescence Tissue UMB 1158
Human_25_yr Adult Tissue UMB 1829 M My_Shiny
Hs 55 yr tissue Adult Tissue UMB 797 M
Hs 55 yr NeuN+ Adult Neurons UMB 797 M /My_Shiny
Hs 55 yr NeuN- Adult Non-Neurons UMB 797 M /My_S
HUES6 Immortal ESC - F /My_Shiny_DataDir/dat
Natural group ordering
group_order.txt
groupID groupOrder
Age_Group Early Childhood,Tween,Adolescence,
Cell_Type Tissue,Neurons,Non-Neurons,ESC
Sex M,F
<dataRoot>/<projNam
bash-3.2$ ls
GSM1164630_allC.Methyl-Seq_hs_mfg_12yr.chr1.tx
GSM1164630_allC.Methyl-Seq_hs_mfg_12yr.chr1.tx
GSM1164631_allC.Methyl-Seq_hs_mfg_16yr.chr1.tx
GSM1164632_allC.Methyl-Seq_hs_mfg_25yr.chr1.tx
GSM1166274_allC.Methyl-Seq_hs_mfg_5yr.chr1.txt
GSM1166274_allC.Methyl-Seq_hs_mfg_5yr.chr1.txt
GSM1167004_allC.Methyl-Seq_hs_mfg_35do.chr1.tx
GSM1167004_allC.Methyl-Seq_hs_mfg_35do.chr1.tx
GSM1167005_allC.Methyl-Seq_hs_mfg_2yr.chr1.txt
GSM1167005_allC.Methyl-Seq_hs_mfg_2yr.chr1.txt
GSM1173775_allC.Methyl-Seq_hs_fc_male_55yr_tis
GSM1173775_allC.Methyl-Seq_hs_fc_male_55yr_tis
GSM1173776_allC.Methyl-Seq_hs_fc_male_55yr_Neu
GSM1173776_allC.Methyl-Seq_hs_fc_male_55yr_Neu
GSM1173777_allC.Methyl-Seq_hs_fc_male_55yr_Neu
GSM1173777_allC.Methyl-Seq_hs_fc_male_55yr_Neu
GSM1173778_allC.Methyl-Seq_hs_hues6.chr1.txt.g
GSM1173778_allC.Methyl-Seq_hs_hues6.chr1.txt.g

```
                                |---- pheno.txt
                                |---- group_order.txt
                                |---- data
                                                |---- GSM123456_mm9.bw
                                                |---- GSM654321_mm9.bw
                                                 | ...
                                                 |---- GSM9999_mm9.bw
                |---- humanBrainDNAMethylome/
                                |---- config.txt
                                |---- pheno.txt
                                |---- group_order.txt
                                |---- data
                                                |----
                                                |---- PCW4.bw
                                   |---- midGestation_Cortex.bw
                                                | ...
                  |---- PostPuberty_AnteriorCingulateCortex.bw
    |----anno
```

Dataset config file

The dataset config file contains general metadata for the dataset. It is a tab-delimited file with two columns: a key (controlled word for config parameter) and value.

Format and required fields

key	description
`name`	Dataset name; human-readable (~70 char)
`description`	Brief description of dataset; human-readable (~170 char)
`platformName`	Short description of platform on which data was generated; human-readable
`datasetConfig`	Path to *sample phenotype table*
`groupCols`	Column names of *sample phenotype table <add-data-pheno>* that EDB shou grouped by; comma-separated list, case-sensitive
`groupOrder`	Path to *group order file*
`defaultGroup`	Name of group that serves as the default value in the *Group by* dropdown bo must be a column name of the *sample phenotype table <add-data-pheno>* ,
`chromSizes`	Path to text file containing genome sequence sizes. This file should be the ou *fetchChromSizes* from Kent utilities at UCSC.
`genomeName`	Name of genome build ; must correspond to a *UCSC release name <https://genome.ucsc.edu/FAQ/FAQreleases.html>*
`ideoFile`	Path to text file corresponding to UCSC `cytoBandIdeo` table. **Header?**
`annoConfig`	Path to config file for genome annotation. **TODO: Add link**
`datatype`	Datatype. Currently, one of `{ bigwig , BSseq }`. Indicates how data s most single continuous traces (e.g. coverage, tiling microarrays) can use the

All the above fields are required and must not contain missing values.

Variables for different datatypes

bigwig:

This datatype has no required parameters.

BSseq:

- `BSseq__COV_POS`: Column index of tabix file which contains base coverage
- `BSseq__M_POS`: Column index of tabix file which contains number of methylated reads at that base ("M" read count)

Example dataset config file

```
# REQUIRED:
↪MAIN config flags --- must be in all datasets
# ----------------------------------------------
name    Human
↪brain development methylome (base-resolution BS-seq)
description DNA modifications
↪assayed by whole-genome bisulfite sequencing
↪over postnatal development. Samples from Lister,
↪ Mukamel et al. (2013). Science. 341. GEO:GSE47966.
platformName    BS-seq; Illumina HiSeq 2000.
#Path to dataset config file
↪(sample file source location, group names, etc.,)
datasetConfig   /
↪<dataRoot>/ListerEcker_2013/pheno_ListerEcker2013.txt
groupCols   Age_Group,Cell_Type,Sex
groupOrder
↪ /<dataRoot>/ListerEcker_2013/group_order.txt
defaultGroup    Age_Group
# path to chromosome
↪sizes  - output of Kent utilities fetchChromSizes
chromSizes  /<dataRoot>/anno/hg19/hg19.chrom.sizes
genomeName  hg19
ideoFile    /<dataRoot>/anno/hg19/cytoBandIdeo.txt
annoConfig  /<dataRoot>/anno/hg19/anno_config.txt
datatype    BSseq
ylabel  % methylation
#----------------------------------------------
# OPTIONAL: DATATYPE-
↪specific; flags must have prefix <datatype>__
# Leave section blank if there are no flags.
#----------------------------------------------
BSseq__COV_POS  6
BSseq__M_POS    5
```

EDB requires that all dataset config files be located at the path specified in the special `config_location.txt` file. Recall that this file is at `<EDBServerRoot>/config_location.txt`.

Back to top

Sample phenotype table

This tab-delimited file contains sample-wise metadata, including locations of data files. Each row should contain data for one sample, and each column should contain a unique type of metadata. Column order is unimportant to the browser. The browser expects the following columns, named exactly in this way:

Format

column name	expected value
`sampleName`	unique identifier, no spaces
`bigDataURL`	absolute path to data source (e.g. *.bw* file)
`...`	all grouping columns as described in the *grouping order* file.

Example phenotype table

```
sampleName␣
↪ Age_Group   Cell_Type   Bank_ID Sex bigDataURL
Human_35_do Early Childhood␣
↪Tissue  UMB 759 M   /My_Shiny_DataDir/data/GSM116700
4_allC.MethylC-Seq_hs_mfg_35do.chr1.txt.gz
Human_2_yr  Early Childhood␣
↪Tissue  UMB 5180    M   /My_Shiny_DataDir/data/
↪GSM1167005_allC.MethylC-Seq_hs_mfg_2yr.chr1.txt.gz
Human_5_yr  Early Childhood␣
↪Tissue  UMB 1185    M   /My_Shiny_DataDir/data/
↪GSM1166274_allC.MethylC-Seq_hs_mfg_5yr.chr1.txt.gz
Human_12_yr␣
↪Tween   Tissue  UMB 616 M   /My_Shiny_DataDir/data/
↪GSM1164630_allC.MethylC-Seq_hs_mfg_12yr.chr1.txt.gz
Human_16_yr Adolescence␣
↪Tissue  UMB 1158    M   /My_Shiny_DataDir/data/
↪GSM1164631_allC.MethylC-Seq_hs_mfg_16yr.chr1.txt.gz
Human_25_yr Adult␣
↪  Tissue  UMB 1829    M   /My_Shiny_DataDir/data/
↪GSM1164632_allC.MethylC-Seq_hs_mfg_25yr.chr1.txt.gz
Hs 55 yr tissue Adult   Tissue␣
↪ UMB 797 M   /My_Shiny_DataDir/data/GSM1173775_
↪allC.MethylC-Seq_hs_fc_male_55yr_tissue.chr1.txt.gz
Hs 55 yr NeuN+  Adult   Neurons␣
↪UMB 797 M   /My_Shiny_DataDir/data/GSM1173776_
↪allC.MethylC-Seq_hs_fc_male_55yr_NeuN_pos.chr1.txt.gz
Hs 55 yr NeuN-  Adult   Non-Neurons␣
↪UMB 797 M   /My_Shiny_DataDir/data/GSM1173777_
↪allC.MethylC-Seq_hs_fc_male_55yr_NeuN_neg.chr1.txt.gz
HUES6   Immortal    ESC -   F   /My_Shiny_DataDir/
↪data/GSM1173778_allC.MethylC-Seq_hs_hues6.chr1.txt.gz
```

Groups and grouping order

The `group_order.txt` file is a tab-delimited file containing a table of two columns: #. groupID: Group name, must match a column name in

the phenotype matrix #. groupOrder: Order in which group members must be shown. Comma-separated collection of values. **All values for a given group must be specified here. The browser will return an error if any additional group members are found in the phenotype table but are not listed here.**

In addition to these groups, the browser allows a non-grouping option - i.e. viewing sample-specific data - with "Grouping: (none)".

Example group order file

```
groupID groupOrder
Tissue  Brain,Sperm
Diagnosis       Control,Schizophrenia,Bipolar disorder
TimeOfSampling␣
↪ Before_Treatment,During_Treatment,After_Treatment
```

For this dataset, the browser would show 4 grouping options: *Tissue*, *Diagnosis*, *TimeOfSampling*, *(none)*.

The *(none)* option is automatically added, and allows samples to be inspected individually instead of being grouped.

Back to top

Adding demo datasets

This section is added for completeness. Demo datasets have not yet been made publicly available. - SP 5 Sep 2014.

In this example, the demo data and config files are contained in a directory named "Shiny_BrowseR_Github_data".

From the source machine, rsync the datasets to the machine where the EDB instance will be hsoted. The data directory should be in a location readable by the user shiny, which will run the EDB app.

```
rsync -avL␣
↪--progress Shiny_BrowseR_Github_data $DEST_MACHINE:/.
```

On the destination machine, we update the path in the metadata files to reflect the path on our new machine. **Replace the substitution sed command with one relevant to your source and destination paths.**

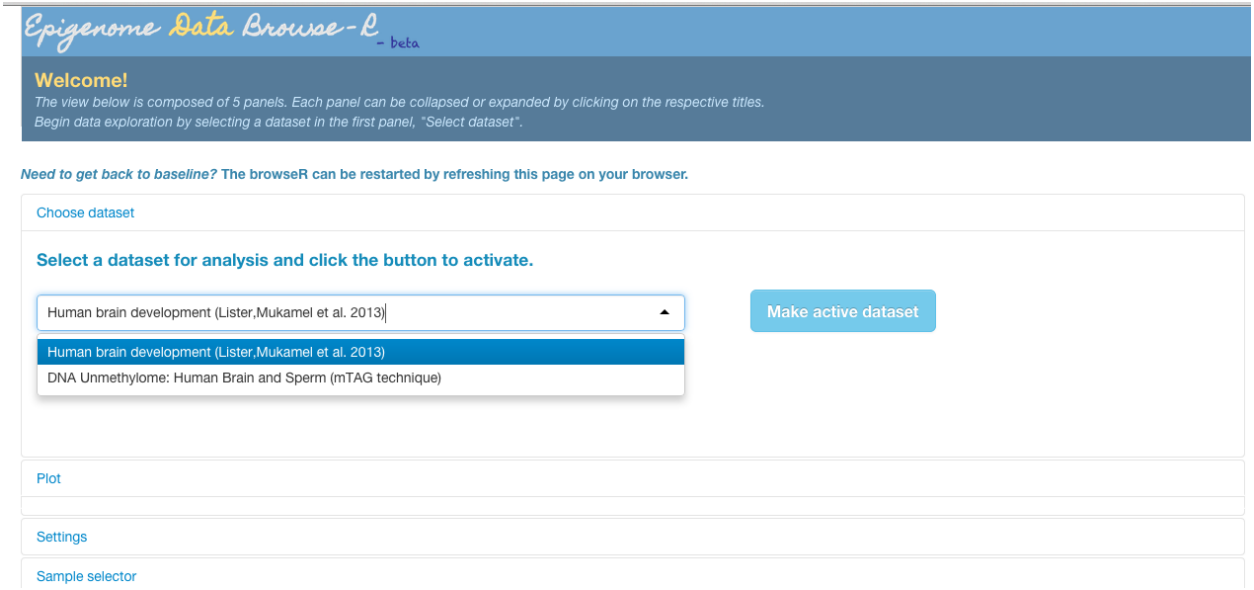
Below we update the paths in the data dirs as well as the annotation directory (anno/hg19).

```
cd /Shiny_BrowseR_Github_data
rm *_config.txt
cd mTAG_BrainSperm
sed -i 's/\/src\/path\/dir/\/dest\/path/g' *.txt
cd ../ListerEcker_2013
sed -i 's/\/src\/path\/dir/\/dest\/path/g' *.txt
cd ../anno/hg19
sed -
↪i 's/\/src\/path\/dir/\/dest\/path/g' anno_config.txt
```

Create a symlink to dataset-specific config files in the data directory

We change `/srv/shiny-server/EDB/config_location.txt` to point to our new data directory path: `/Shiny_BrowseR_Github_data`

At this point, refresh the EDB. If the "Choose dataset" dropdown box is populated, shiny can see the datasets.



If not, stop here and check the following:

- Is the data root directory in a location with read permissions for user "shiny"?
- Have the paths been correctly updated for all dataset directories?
- Is config_location.txt pointing to the correct data directory?

Adding custom annotation tracks

Annotation files are expected to live under `<dataRootDir>/anno` where `<dataRootDir>` is the directory to which `config_location.txt` points. Sources are organized by genome build as in the example below. EDB uses the BioConductor Gviz package to construct annotation objects. Refresh the EDB browser page and reload dataset to see the listing of new annotation sources.

Directory structure for custom annotation

```
<dataRootDir>/
     anno/
       |------ hg19/
                        |------ cpgIslandExt.txt
                                                  ␣
↪                                |------ cytoBandIdeo.txt
```

```
                                                                    ␣
↪       |------ TxDb.Hsapiens.UCSC.hg19.refGene.sqlite
                                        |------ anno_config.txt
|------ mm9/
                                |------ cpgIslandExt.txt
                                        |------ LAD_NPC_mm9.txt
                                        |------ anno_config.txt
```

anno_config.txt

`anno_config.txt` s used by EDB to get a listing of all available annotation for a genome build. EDB gets the genome build for the current dataset as the value of the `genomeName` variable in the dataset config file. It then refers to `<dataRootDir>/<genomeName>/anno_config.txt` for a list of all annotation available for that genome build. `anno_config.txt` is expected to be a tab-delimited file with rows representing each annotation source, and the following columns:

column	description
trackName	one-word unique identifier for track
name	title of track as it would appear in EDB (<50 char)
description	(currently unused)
trackType	See allowed values below
defaultView	`[dense|squish|full]`. Similar to UCSC tracks.
bigDataURL	absolute path to source file
color	(currently unused)
format	See values in table below.
sizes	number between 0 and 1. determines the height of the track. See `plotTra` Gviz

Columns except `description` and `color` must not have missing values.

EDB currently supports the following file formats. Behaviour is undefined if the `format-trackType` combinations below are not respected.

EDB `format`	input file format / object	EDB `trackType`
`tabix`	tabix (*.gz,.gz.tbi*)	`AnnotationTrack`
`bigwig`	bigwig (*.bw*)	`AnnotationTrack`
`txdb`	BioC TranscriptDB object (*.sqlite*)	`GeneRegionTrack`

Visit these pages to learn more about the tabix , bigwig and BioC TranscriptDB format/objects.

Sample `anno_config.txt` file

```
trackName   name    description trackType␣
↪  defaultView bigDataURL  color   format  sizes
cpgIslands  CpG Islands CpG Islands␣
↪AnnotationTrack dense   /home/docker/EDB_demodata/
↪anno/hg19/cpgIslandExt.bed.gz green   tabix   0.15
refGene RefSeq␣
↪genes    RefSeq genes    GeneRegionTrack squish␣
↪ /home/docker/EDB_demodata/anno/hg19/TxDb.Hsapiens.
↪UCSC.hg19.refGene.sqlite  mediumblue  TxDb    0.1
```

Back to top

Navigating the Browse-R

This section provides a tour of the capabilities of the EDB.

Select a dataset

Pick a dataset. Click "make active dataset". If you don't see a dataset, add one (see section on adding dataset).

"Plot settings" panel

The "Plot Settings" panel is the master panel to determine what region of the genome to view, how to bin/smooth/baseline samples, how to group/colour-code samples, and set axes.

Here are the current available options:

Genome Location (left panel)

After updating these values, click the refresh icon for changes to take effect.

field name	purpose
sequence	select from available chromosomes
x-range, from/to	coordinates for x-axis units in bp, commas not currently supported
Number of Bins:	specifies the number of bins in which the current x-axis is broken up
Smooth bw (bp):	smoothing bandwidth (uses Gaussian kernel)

Data and Grouping

Some of these options result in an automatic refresh. Manually refresh for other options.

field name	purpose
Group samples by	values are those speciifed in da EDB adds additional value of "(showing sample- level data
Baseline trends	sets whether samples are shown vided by another value. Options • Sample-wise data (no ba presented as-is • Sample/Mean-of-all: eac by the mean of all traces • Sample/Mean-of-baseline vided by the same trace se Depending on the value of G each trace is either a sample-le group-level trace

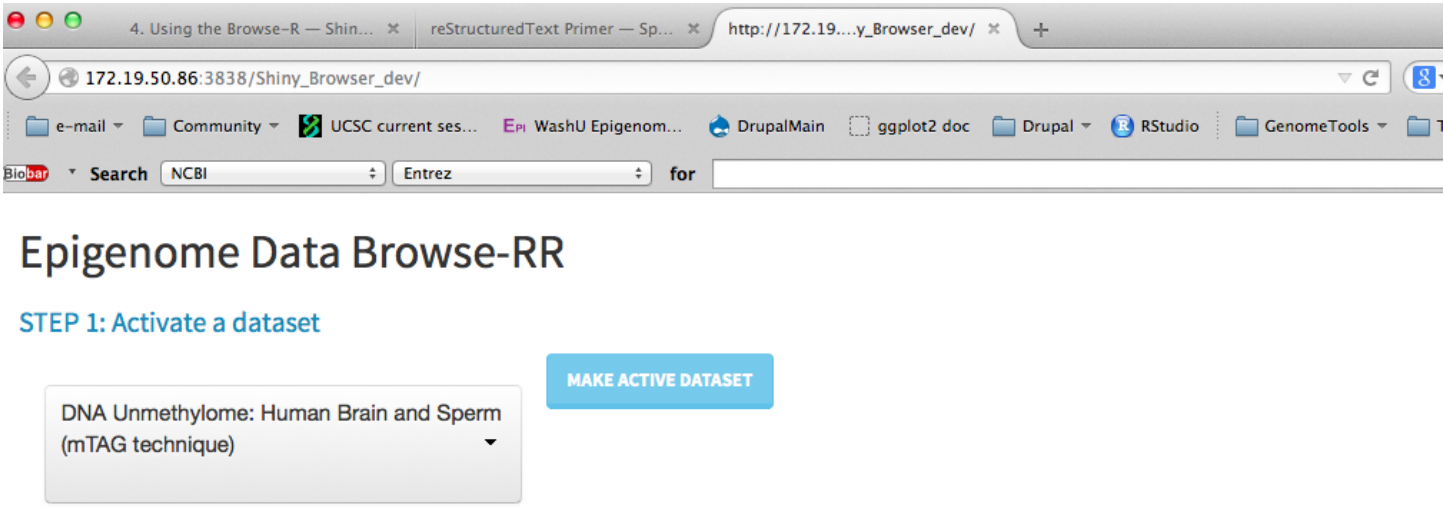


Fig. 1.1: Select a dataset

Plot Options

These changes result in an automatic refresh and a manual refresh should not be necessary.

field name	purpose
Plot type	Data representation. Options are • Mean+CI: Group average fidence interval bands. On `samples by` is not (no • Points+Lines: Line show along with individual data • Points: Individual data poi • Lines: Line showing raw t • Smooth: Line showing sm
Color by	Setting to color traces. Option by which samples can be group `group_order.txt`). This m `Group samples by` is set to
Color scheme	Color palette for determining tra from RColorBrewer, which in tu Brewer. Refer to ColorBrewer t ing nature of data series (e.g. s gent) *Note: A number of palettes fea for the first item in the series. It different palettes to make sure al
Y-axis	Determines if the y-axis limi EDB (`Default bounds`) or `bounds`). When `Custom b` slider appears to let the user set
show legend	(checkbox) Toggles the legend shows either sample or group nam individual traces are showing.

Sample selector

This panel provides a view of the *sample phenotype table* as provided to EDB.

Rows can be sorted by any column, by clicking the column header at the top.

At present, the sample table only allows selecting of samples, and not deselection. This will change in future versions of EDB.

Samples can be selected using Ctrl-click (or Cmd+<space> in OS X). After selection, click the refresh icon to update the plot view.

Genome Annotation

This panel shows a list of custom genome annotation available for the dataset genome. Refer to this section for instructions to *add custom an-*

notation.

*Note: The rendering of gene models by Gviz can slow down browser refreshes considerably. EDB automatically switches gene view from ''full'' to ''dense'' mode depending on width of viewed region. However, users are advised to turn on the gene models track, e.g. tracks like *RefSeq genes*,* only when coordinates are reasonably refined.*

.._add_datatype:

Extending Browse-R capabilities

Add a datatype handler

The Shiny Browse-R is easily customizable to show data from a new platform; all that is required is the addition of a platform-specific .R file in the code directory. For illustration, let us suppose we want to add a new datatype of 450K microarrays.

These steps will allow a new 'datatype' value to be specified in a *dataset config* file. In our example, let that datatype be "FourFiftyK".

1. Create FourFiftyK.R in the `data_types` directory; the latter is at the same level with ui.R and server.R)
2. In FourFiftyK.R, create an R function, fetchData_base() with the following signature:

```
fetchData_base <- function
(
pheno,          ##<<(data.frame) phenotype matrix
selRange,       ##<<(GRanges)␣
↪range being viewed on browser [start,end] - length 1
bin_GR,
↪          ##<<(GRanges) ranges of individual data bins
numBins,        ##<<(integer) num. bins
aggFUN=mean     ##<<(function) aggregating function
) {
        # fetch code goes here.

### (list with two keys):
### 1)␣
↪coords: data.frame with three columns corresonding␣
↪to the chromosome, start and end coordinate
### 2) values: sample-wise values. Row order should␣
↪correspond to coords and column order to samples.
```

}

Current datatypes

bigwig

Used for any datatype which can be represented with a single column containing a continuous value.

BSseq

Used for bisulfite-seq data. Usually has two columns, M and COV, which are combined into a %methylation over an arbitrary genomic interval (e.g. in 2Kb bins, or over a gene).

Example view of input file:

Custom columns are: # CHROM_POS - columm # of sequence name in tabix file # START_POS - columm # of position start in tabix file # END_POS - column # of position end in tabix file # STRAND_POS - column # of strand # M_POS - column # of num. methylated cytosine (M) # COV_POS - column # of position coverage # minCov - minimum coverage to use

Computation is: (M/COV)

Future version will incorporate non-conversion rate subtraction capability.

Please send feedback to Shraddha [dot] Pai [at] camh [dot] ca. This website best viewed at 1280x800 resolution or better.