

# GENSAT Data Mining

Nick Didkovsky  
Bioinformatics core supervisor  
GENSAT Project at Rockefeller University  
Email: didkovn@mail.rockefeller.edu  
Prepared 3/1/2006

## Disclaimer and attribution

The data are provided without warranty as to the accuracy, completeness or fitness for any specific use. The GENSAT database and its contents may not be redistributed or sold, in whole or in part, without specific written permission.

The recommended citation for the resource is:

The Gene Expression Nervous System Atlas (GENSAT) Project, NINDS Contract # N01NS02331 to The Rockefeller University (New York, NY).

## Introduction

The GENSAT database ([www.gensat.org](http://www.gensat.org)) contains a gene expression atlas of the central nervous system of the mouse based on bacterial artificial chromosomes (BACs). In each of the BAC transgenic vectors, endogenous protein coding sequences have been replaced by sequences encoding the EGFP reporter gene. As in any gene replacement experiment, the stability of the reporter gene can vary somewhat from the endogenous gene. Thus these results measure the relative rates of transcription for each gene; they are not a direct measure of mRNA accumulation or of protein abundance for the endogenous gene products. Furthermore, the enhanced sensitivity of reporter gene assays, particularly in BAC lines carrying multiple copies of the BAC transgene may allow detection of sites of expression that are not evident in situ hybridization experiments. This database contains histological data from given BAC transgenic mouse lines at three developmental stages – embryonic day 15.5 (E15.5), postnatal day 7 (P7) and adult; in all cases the data represent results of multiple transgenic lines. EGFP is visualized by staining with an anti-EGFP antibody using the DAB method, or by confocal microscopy of unstained tissue sections. Protocols for the modification of BACs, BAC transgenesis production and histology are provided.

## Data Mining

We recognize that some researchers need to mine GENSAT data with extended methods that go beyond the search engines that are available at [www.gensat.org](http://www.gensat.org). For example, searches that require multiple stages of filtering are not currently available with our online search engines (eg. where the researcher is interested in genes whose expression level is low in the cortex at age E15.5, strong in the cortex at P7, and undetectable in the adult cortex). While we cannot provide users with a direct database connection to our server, **we are now making our data available in a format that can be used to recreate the database on a researcher's own computer.**

## Database Engine

The GENSAT Project at Rockefeller University uses MySQL as its database engine. MySQL is a freely available database server available at [www.mysql.org](http://www.mysql.org)

Note that MySQL ships with Mac OSX, so if you are using a Mac you may already have MySQL installed.

### **Where to find GENSAT public data**

A MySQL dump of public GENSAT data is available at the following URL:  
[http://www.gensat.org/gensat\\_for\\_public.zip](http://www.gensat.org/gensat_for_public.zip)  
This file is refreshed daily at 6am EST  
You might consider automating a daily download & local build, as GENSAT data changes daily.

### **Overview of GENSAT production**

A thumbnail sketch of the GENSAT production process is outlined here, to illuminate where production intersects with the database

A gene is chosen and entered into the database's **gene table**. A short name, long name, and aliases are entered. The official Entrez Gene ID is associated with it as well. This Entrez ID should be considered the absolute cross reference, taking precedence over any text names that are found, since names change frequently. Since gene to alias is a one-to-many relationship, you will find various aliases for a gene in the **gene\_alias table**.

A BAC is produced for the gene, and is entered into the **bac table**. A bac record points to a gene id, and includes info like Abox primer.

The BAC is injected into mice with the intent of producing PCR positive pups (ie mice in whom the gene will fluoresce). These pups are entered into the **founder table**. Their offspring are entered into the **organism table**. An organism record represents a mouse that was sent to histology for sectioning and imaging.

High resolution images are taken of brain sections. Images are uploaded in batches representing all the images for one gene/age set (for example aarp P7). Image data is entered into the **image table**, where each record corresponds to a single image. An image record links back to the organism it came from (and via the organism record, back to the founder). An image record also holds pointers to the gene, age, acquisition technique (brightfield or confocal), and includes the section. A section is a value 01..11 representing where in the brain the image was taken from. For confocal images, the section field will hold a three character abbreviation like "Hip" to indicate where the high magnification image lies (see the **structure table** for these abbreviations). An image record also points to its batch so you can see which images were uploaded together as a batch and on what date they were uploaded.

Images are annotated. Annotations are records in the **annotation table**. One image will have numerous annotations: one for each structure (Olfactory Bulb, Thalamus, Cerebral cortex, etc). The annotation indicates the level and pattern of gene expression and can optionally identify those cell types that express the gene if they are confidently identified in the image.

Before they are uploaded to gensat.org, the confidence that the gene expression in the images is correct is reflected in a veracity value (see **veracity table**). Each image record stores a veracity\_id which points to a value such as Confirmed or To Be Confirmed. See the section on veracity for definitions of these terms.

## Brief Overview of Annotation Data

The main data that the GENSAT Project produces are images showing gene expression at various stages of mouse brain development. Image info is stored in a table called image. However, an image is annotated by a group of anatomists at Rockefeller. Each image has multiple annotations associated with it, one for each of the brain structures found in the image (that is, there is a one-to-many relationship between image and annotation). An **annotation** specifies the expression level and the expression pattern for a particular structure found in a particular image. Therefore, to mine expression levels from GENSAT data, it is important that you understand the annotation structure.

Most of your queries will be mine annotation data, as it is here where you can derive what gene is expressed strongly in what structure, and in what cell types and subtypes. We describe the annotation table here.

### **An annotation record consists of the following fields:**

#### **id**

This is the annotation's "primary key" and is a unique numeric identifier.

#### **Gene id**

This id points to a record in the gene table. Note that a gene record includes a short name, a long name, and most importantly an Entrez Gene id that anchors it to the NCBI database (names tend to change occasionally, so use the Entrez ID as the ultimate determinant).

#### **Mouse age id**

This will be either 1, 5, 8, or 9 for ages Adult, E10.5, E15.5, or P7 (see mouseage table)

#### **Structure id**

This is the id of the brain structure this annotation describes (see structure table, where the id of thalamus is 5, the id of cerebellum is 10, etc)

#### **Expressionlevel id**

This points to one of three expression levels (undetectable, weak signal, moderate to strong signal), and indicates how strongly the gene is expressed in this image, relative to other levels of expression *in the same image* (that is, it is not an absolute quantifier across images and across other genes).

#### **Expressionpattern id**

This points to one of five expression pattern descriptions (region-specific, ubiquitous, scattered, region-specific and scattered, and widely expressed). It describes the clustering or scattering of the cells found to be expressing this gene.

#### **Other**

Occasionally, our annotators will determine that there is interesting expression in a structure that is not listed in our fixed table of 68 structures (see comments on structure id above). In these cases, the annotator will specify id 13 for structure\_id field, and type text into the 'other' field. Current examples of text found in other include "Mesencephalic trigeminal nucleus" and "Meninges."

#### **image id**

This is the unique id of the image the annotator was describing with this annotation. Remember that image-to-annotation is a one-to-many relationship, so one image will have multiple annotations, one each for cortex, cerebellum, etc.

### Technical Instructions

This document assumes you know how to create a MySQL database, create new users, and set permissions. It assumes you know how to log into a MySQL command line and perform SQL queries. It is beyond the scope of this document to provide a MySQL tutorial. Please note that we do not have the resources to provide technical assistance for MySQL related questions. For MySQL related help, ask your system administrator or seek the help available at <http://dev.mysql.com/support/>

#### \* Create your database

Do the following **once** at a mysql prompt. It will create an empty database called `gensat_for_public`

```
mysql> CREATE DATABASE gensat_for_public;
```

Creating the database need only happen ONCE. From then on, you will populate it from the daily GENSAT data dump.

#### \* Download [http://www.gensat.org/gensat\\_for\\_public.zip](http://www.gensat.org/gensat_for_public.zip)

Unzip the archive, and you will have a file called **gensat\_for\_public.sql**

This file is produced by the mysqldump utility. It can be imported directly into mysql with the following command line:

```
mysql -p -u username gensat_for_public < gensat_for_public.sql
```

There's a lot of data here, so this will take a few moments. Be patient.

NOTE: The command above assumes that you have already created an empty database called `gensat_for_public`

### Test your GENSAT database

Let's run a couple of queries to verify that your GENSAT database is operational.

At a shell prompt, log in to the database like so:

```
mysql -p -u username gensat_for_public
```

Now that you are in the mysql command line and you have logged into `gensat_for_public`, let's run a couple of queries:

**Type** the following SELECT command:

```
mysql> select * from celltype;
```

MySQL should respond with the following:

```
+-----+-----+-----+
| id | name          | orderby |
+-----+-----+-----+
```

```

| 1 | ND          | 0 |
| 2 | neuron      | 1 |
| 3 | glial cell  | 2 |
| 7 | vascular cell | 6 |
| 6 | other       | 7 |
| 4 | not clear   | 8 |
| 9 | choroid plexus | 4 |
| 10 | ependymal cell | 5 |
| 11 | pial cell   | 3 |
+-----+
9 rows in set (0.00 sec)

```

**Type the following SELECT command:**  
mysql> select \* from subtype;

MySQL should respond with the following (which we truncate here for brevity)

```

+-----+-----+-----+
| id | name                | listid |
+-----+-----+-----+
| 1 | ND                  | 0      |
| 2 | interneuron         | 2      |
| 3 | pyramidal cell     | 2      |
| 4 | Purkinje cell      | 2      |
| 5 | granule cell       | 2      |
| 6 | astrocyte          | 3      |
| 7 | Bergmann glia      | 3      |
(etc)
44 rows in set (0.00 sec)

```

**Type the following SELECT command:**  
mysql> select \* from structure;

MySQL should respond with the following (which we truncate here for brevity)

```

+-----+-----+-----+-----+
| id | name                | orderby | fibertract | abbr |
+-----+-----+-----+-----+
| 1 | Olfactory Bulb     | 20      | N          | OLF  |
| 2 | Cerebral cortex    | 30      | N          | CEX  |
| 5 | Thalamus           | 160     | N          | THA  |
| 6 | Hypothalamus       | 150     | N          | HYP  |
| 7 | Hippocampus        | 170     | N          | HIP  |
| 8 | Amygdala           | 70      | N          | AMY  |
| 9 | Pons+Medulla       | 230     | N          | PAM  |
| 10 | Cerebellum         | 240     | N          | CER  |
| 22 | Midbrain           | 190     | N          | MID  |
| 20 | Rhombencephalon    | 350     | N          | ---  |
(etc)
+-----+-----+-----+-----+
68 rows in set (0.00 sec)

```

Now let's do something useful, and find all the genes where stellate cell (subtype id 9) are expressed strongly (expressionlevel id 4) in the cerebellum (structure id 10)

Type (or copy and paste) the following SELECT query:

```
SELECT DISTINCT gene.shortname, entrez_id FROM annotation,
gene, annotation_subtype WHERE gene.id=annotation.gene_id
AND annotation.expressionlevel_id=4 AND
annotation.structure_id=10 AND
annotation_subtype.annotation_id=annotation.id AND
annotation_subtype.subtype_id=9;
```

MySQL will respond with the following six genes:

```
+-----+-----+
| shortname | entrez_id |
+-----+-----+
| Lamb3     | 16780    |
| Chrn4     | 108015   |
| Lhx5      | 16873    |
| Drd3      | 13490    |
| Pak1      | 18479    |
| Nos1      | 18125    |
+-----+-----+
6 rows in set (0.01 sec)
```

Note that gene symbols like the ones shown above occasionally change. Users should rely on the `entrez_id` for an unambiguous reference.

## Detailed Description of GENSAT Database

Here we document each table and each field in the GENSAT database.

```
CREATE TABLE `acquisitiontechnique` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(20) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

Each image stores an acquisitiontechnique\_id indicating the technique used. Values are:

```
| 1 | brightfield |  
| 2 | confocal   |
```

```
CREATE TABLE `annotation` (  
  `id` int(11) NOT NULL auto_increment,  
  `gene_id` int(11) default NULL,  
  `mouseage_id` int(11) default NULL,  
  `structure_id` int(11) default NULL,  
  `expressionlevel_id` int(11) default '1',  
  `expressionpattern_id` int(11) default NULL,  
  `other` varchar(32) default '-',  
  `image_id` int(11) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

Each image has many annotations. See description of annotation table in “Brief Overview of Annotation Data” above.

```
CREATE TABLE `annotation_celltype` (  
  `id` int(11) NOT NULL auto_increment,  
  `annotation_id` int(11) default NULL,  
  `celltype_id` int(11) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

An annotation record can point to multiple celltypes and subtypes, indicating the type of cells in which the indicated expression is visible. So the relationship of annotation to celltype is one-to-many, as is the relationship of annotation to subtype.

Example query:

```
mysql> select name as celltype from annotation_celltype, celltype where  
celltype.id=annotation_celltype.celltype_id and annotation_id=444910;  
+-----+  
| celltype |  
+-----+  
| neuron   |  
| glial cell |  
| vascular cell |  
+-----+  
3 rows in set (0.02 sec)
```

```

CREATE TABLE `annotation_subtype` (
  `id` int(11) NOT NULL auto_increment,
  `annotation_id` int(11) default NULL,
  `subtype_id` int(11) default NULL,
  PRIMARY KEY (`id`)
) TYPE=MyISAM;

```

See comments above about annotation\_celltype.

Example query, shows that annotation 134421 describes 7 subtypes:

```

mysql> select name as subtype from annotation_subtype, subtype where
subtype.id=annotation_subtype.subtype_id and annotation_id=134421;

```

```

+-----+
| subtype |
+-----+
| other |
| fibers |
| periglomerular cell of MOB |
| periglomerular cell of AOB |
| mitral cell of MOB |
| mitral cell of AOB |
| granule cell of MOB |
+-----+
7 rows in set (0.00 sec)

```

```
CREATE TABLE `bac` (  
  `id` int(11) NOT NULL auto_increment,  
  `address` varchar(32) default '-',  
  `marker_id` int(11) default NULL,  
  `gene_id` int(11) default NULL,  
  `aboxprimer` varchar(255) default "",  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

Numerous BACs may be created for a given gene.

Each BAC record points to the gene for which it was modified.

Each BAC includes a field for the marker id, which can be EGFP, iEGFP, etc (see marker table).

Address field contains values like RP23-75J13.

Aboxprimer contains values like GGTCTCAGCCCCCTGCCCTGC

Note that for a given annotation you can retrieve the BAC via the image id -> organism  
-> founder

```
CREATE TABLE `batch` (  
  `id` int(11) NOT NULL auto_increment,  
  `uploaddate` date default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

Images are uploaded in batches, where one batch holds a group of images from the same mouse brain. A batch corresponds to a set of images uploaded for one gene and one age, from one particular organism. Records in the image table point to the batch they came from.

```
CREATE TABLE `batch_veracitysupport` (  
  `id` int(11) NOT NULL auto_increment,  
  `batch_id` int(11) default NULL,  
  `veracitysupport_id` int(11) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

The confidence that our expression pattern is correct is reflected by one or more veracity support values (Multiple Lines Match, Matches In Situ Data, and Supported by Literature). These are associated with a batch, so for a given image, you can trace back to the batch, and derive the sources that support the expression asserted in the GENSAT database.

```
CREATE TABLE `celltype` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(20) default NULL,  
  `orderby` int(11) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

An annotation may point to one or more cell types found in an image. Celltypes include neuron, glial cell, etc. There are 9 cell types. There are many subtypes (ie Bergmann glia, microglia, etc)

```
CREATE TABLE `expressionlevel` (  
  `id` int(11) NOT NULL auto_increment,  
  `symbol` varchar(10) default NULL,  
  `description` varchar(32) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

An annotation points to an expression level which indicates the level of expression relative to the rest of the image. Possible values include undetectable, weak signal, moderate to strong signal).

```
CREATE TABLE `expressionpattern` (  
  `id` int(11) NOT NULL auto_increment,  
  `description` varchar(32) default NULL,  
  `abbreviation` char(2) default NULL,  
  `orderby` int(11) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

An annotation points to an expression pattern which indicates the distribution of expression. Possible values include region-specific, ubiquitous, scattered, etc

```
CREATE TABLE `founder` (  
  `id` int(11) NOT NULL auto_increment,  
  `founderline` varchar(10) default '-',  
  `bac_id` int(11) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

Each founder was born to a mouse whose eggs were injected with a BAC. Those founders who were PCR+ are used to create offspring (organisms) that are imaged.

Each record of the founder table points to the bac that it came from. It also includes a text field called founder line which is used to uniquely name the animal.

```
CREATE TABLE `gene` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(250) default NULL,  
  `shortname` varchar(25) default NULL,  
  `entrez_id` varchar(32) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

The gene table contains the genes investigated by the GENSAT Project. Names change unpredictably, so the main field of interest is the `entrez_id` which points to a gene in the Entrez database at NCBI. The user can locate a gene in Entrez Gene by invoking the following URL (using `entrez_id` 68394 as an example):  
[http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene&cmd=Retrieve&dopt=Graphics&list\\_uids=68394](http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene&cmd=Retrieve&dopt=Graphics&list_uids=68394)

```
CREATE TABLE `gene_alias` (  
  `id` int(11) NOT NULL auto_increment,  
  `gene_id` int(11) default NULL,  
  `alias` varchar(150) default "",  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

This table maintains the one-to-many relationship between a gene and the numerous aliases it can be referred to. For example, Aatf has aliases Trb; Che-1; MGC35916; 4933415H02Rik; 5830465M17Rik

```
CREATE TABLE `gene_literature` (  
  `id` int(11) NOT NULL auto_increment,  
  `gene_id` int(11) default NULL,  
  `literature_id` int(11) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

A gene's expression pattern may be reported in literature. This table associates genes with an entry in our literature table (see below). Each record in the literature table includes a PubMed id called pmid.

```
CREATE TABLE `gene_supportcomment` (  
  `gene_id` int(11) default NULL,  
  `comment` text,  
  `id` int(11) NOT NULL auto_increment,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

This is comment entered by GENSAT personnel that summarizes the correlation between GENSAT expression data and expression patterns found in literature, as well as comparing it to the expression pattern found in St. Jude data. Example: "No detailed CNS expression literature found. The Bac data is reproducible and matches Stjude in situ data."

```

CREATE TABLE `image` (
  `id` int(11) NOT NULL auto_increment,
  `gene_id` int(11) default NULL,
  `mouseage_id` int(11) default NULL,
  `orientation_id` int(11) default NULL,
  `section` varchar(10) default NULL,
  `acquisitiontechnique_id` int(11) default NULL,
  `stain_id` int(11) default '1',
  `magnification` varchar(5) default '10X',
  `sectionprocedure_id` int(11) default '1',
  `batch_id` int(11) default NULL,
  `organism_id` int(11) default NULL,
  `veracity_id` int(11) default '1',
  PRIMARY KEY (`id`)
) TYPE=MyISAM;

```

Each image record points to a file on our server. These files are very large, and are not stored as blobs in the database, rather as TIFF images on our servers. Each image points to the gene, the age of the mouse, the sectioning orientation, acquisition technique, magnification, sectioning procedure, and the organism whose tissue it shows. Additionally, the veracity of the expression pattern shown in the image is stored as a pointer to the veracity table (see below).

```
CREATE TABLE `literature` (  
  `id` int(11) NOT NULL auto_increment,  
  `pmid` varchar(32) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

This table stores literature references relevant to the expression pattern found in our images. The pmid field points to a PubMed paper. You may retrieve the paper by constructing a URL like the following (using pmid 11343646 as an example):  
[http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list\\_uids=11343646&dopt=Abstract](http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=11343646&dopt=Abstract)

```
CREATE TABLE `marker` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(20) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

Stores the marker used in the BAC construction. Each BAC record points to a marker.

```
CREATE TABLE `mouseage` (  
  `id` int(11) NOT NULL auto_increment,  
  `age` varchar(10) default NULL,  
  `ageorder` int(11) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

A table of the mouse ages under consideration by the GENSAT project. Ages are adult, P7, and E15.5. A few images with age 10.5 exist.

```
mysql> select * from mouseage order by ageorder;
```

```
+----+-----+-----+  
| id | age   | ageorder |  
+----+-----+-----+  
|  1 | adult |         1 |  
|  9 | P7    |         2 |  
|  8 | E15.5 |         4 |  
|  5 | E10.5 |         6 |  
+----+-----+-----+
```

```
CREATE TABLE `organism` (  
  `id` int(11) NOT NULL auto_increment,  
  `taxon` varchar(12) default 'Mus musculus',  
  `mouseage_id` int(11) default '1',  
  `sacdate` datetime default NULL,  
  `sex_id` int(11) default '1',  
  `gene_id` int(11) default NULL,  
  `geneticbackground` varchar(32) default 'FVBN/SW',  
  `sectioningdate` datetime default NULL,  
  `founder_id` int(11) default '2447',  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

An organism is an offspring from a PCR+ founder. An organism is sectioned and imaged. Each image points to an organism record. Each organism points to the founder it came from.

```
CREATE TABLE `orientation` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(12) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

Describes orientation of sectioning for an image

```
mysql> select * from orientation;
```

```
+-----+-----+  
| id | name          |  
+-----+-----+  
|  1 | Sagittal     |  
|  2 | Coronal      |  
|  3 | not used     |  
|  4 | Whole Mount  |  
|  5 | Horizontal   |  
+-----+-----+
```

```
CREATE TABLE `sectionprocedure` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(10) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

Contains a number of sectioning procedures. All GENSAT images currently point to the same procedure: cryostat (id 2)

```
CREATE TABLE `sex` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(10) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

Each organism has a sex\_id field.

```
mysql> select * from sex;
```

```
+----+-----+  
| id | name  |  
+----+-----+  
|  1 | male  |  
|  2 | female|  
|  3 | UD    |  
+----+-----+
```

```
CREATE TABLE `stain` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(12) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

Describes which stain was used:

```
mysql> select * from stain;  
+----+-----+  
| id | name          |  
+----+-----+  
|  1 | DAB           |  
|  2 | TSA           |  
|  3 | LACZ          |  
|  4 | EGFP          |  
|  5 | CresylViolet |  
|  6 | none          |  
+----+-----+
```

```
CREATE TABLE `structure` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(80) default NULL,  
  `orderby` int(11) default NULL,  
  `fibertract` char(1) default 'N',  
  `abbr` char(3) default '---',  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

Describes brain structures annotated by GENSAT annotators. Each annotation record points to a structure.

```
CREATE TABLE `subtype` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(32) default NULL,  
  `listid` int(11) default '2',  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

A listing of specific cell types annotated by GENSAT's annotators. An annotation record may site more than one subtype has having the indicated expression level and pattern. This one-to-many relationship is stored in the annotation\_subtype table.

```
CREATE TABLE `veracity` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(20) default NULL,  
  `definition` text,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

One of: NA, Confirmed, To be confirmed, and Anomalous. Definitions are included in the table, and are reprinted below. Each image points to one of these values, reflecting the veracity of the expression pattern it exhibits.

Confirmed refers to genes in which multiple lines yield matching datasets that agree with the available literature, or in which a single line has been produced that matches the available literature. In those cases where there a discrepancy between the literature and the reporter gene data (missing or additional sites of expression), this is noted in the database. |

To Be Confirmed refers to genes in which multiple lines produce matching datasets, but for which no independent expression data is available.

Anomalous refers to data collected from BAC reporter constructs producing different expression patterns in different lines (i.e. position effects).

```
CREATE TABLE `veracitysupport` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(25) default NULL,  
  PRIMARY KEY (`id`)  
) TYPE=MyISAM;
```

The veracity of the expression displayed by a batch of images may be supported by up to three sources:

- Multiple Lines Match
- Matches In Situ Data
- Supported by Literature

This one-to-many relationship is stored in the batch\_veracitysupport table. Follow an image to its batch via the batch\_id and query for the support for a particular image;