

# Chado on Rails

a framework to simplify development on  
the Chado schema

Justin Reese / Chris Childers

# Some links:

## These slides:

`http://tinyurl.com/chadoonrails`

## Source code, have a look:

`svn co http://chadoonrails.rubyforge.org/svn/trunk`

# The Big Idea

Chado database  
(de facto standard in bioinformatics)



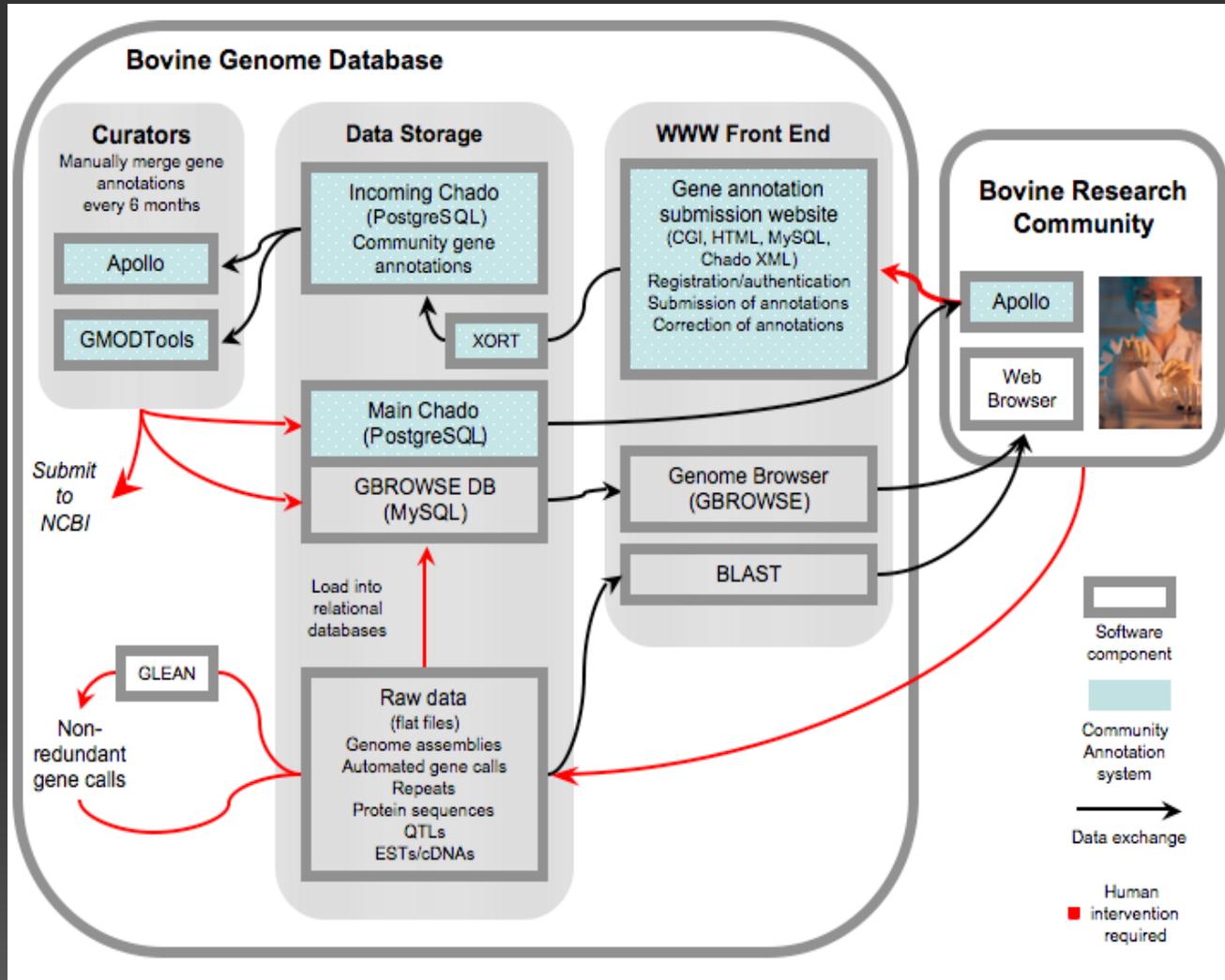
Chado on Rails



Your application  
(gene pages, GFF loader, anything that needs to talk to Chado)

# This time last year...

## Bovine Genome Database status talk:



no gene pages!

# How to implement gene pages?

- couldn't find existing tool that fit our needs
- why not roll our own?
- why not use Ruby on Rails?

# The case for Ruby on Rails

- implement object-relational mapping for Chado database schema as legacy schema

- what we then get for free:

  - Mature MVC framework for fast web development on top of Chado

  - Rake tasks (a powerful tool for automating tasks)

  - Capistrano (another powerful tool for deployment, automating tasks)

  - Powerful debugging (stop request in stack wherever you want)

  - Sophisticated testing suite

  - Much more...

- BGD, Beebase, HymenopteraBase, other groups can build on this

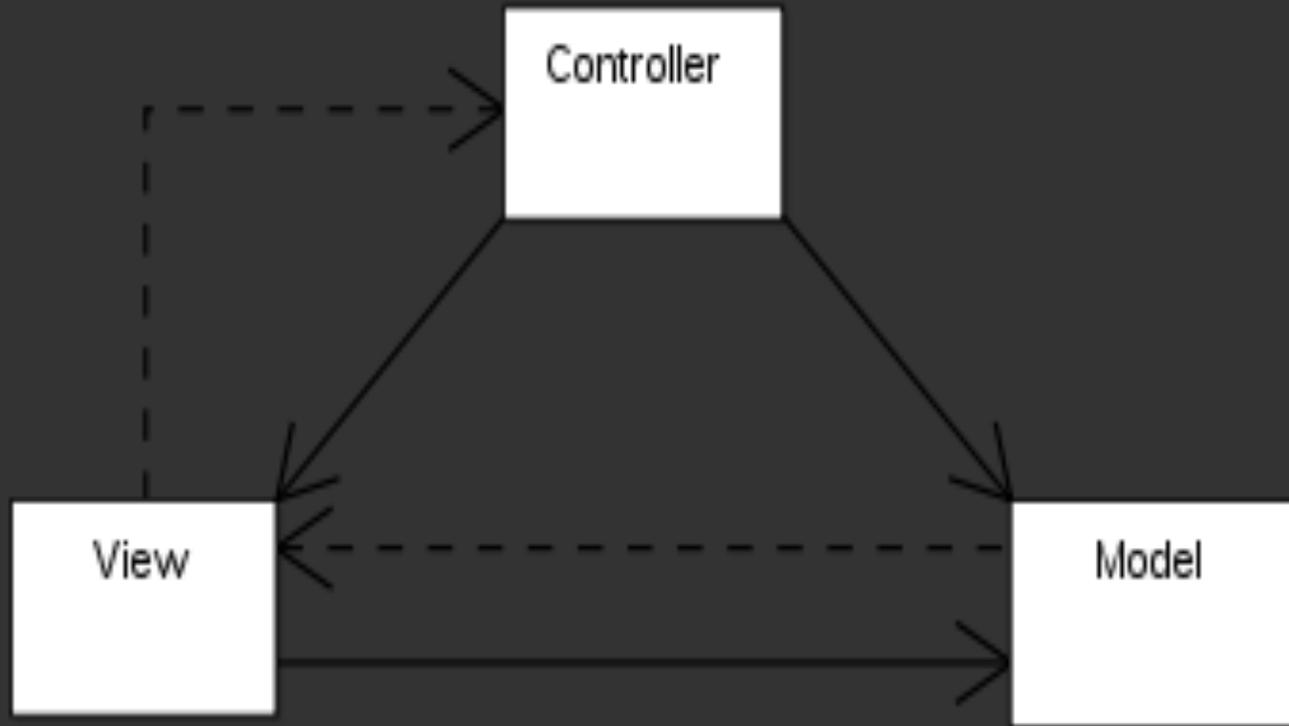
# Ruby

- "Perl's younger prettier sister"
- syntactic sugar, easy metaprogramming, pure object-oriented language, other sundry awesomeness
- A great 15 minute tutorial:  
<http://tryruby.org/>
- A funny, longer tutorial:  
<http://mislav.uniqpath.com/poignant-guide/>

# Rails legacy database support

- Rails paradigm: "convention over configuration"  
e.g. Model name "Feature" == table name "featuresu"
- Chado doesn't follow Rails convention  
e.g. Feature model == Chado table name "feature"
- Legacy support for Chado means overriding these conventions, explaining to Rails how to talk to Chado

# MVC paradigm



**Model** - the representation of your data (doesn't change much after development)

**Controller** - "business logic", i.e. the logic to retrieve and process information coming and going from view (changes more seldom)

**View** - presentation of information to user (changes every 3 seconds)

# Quick rails anatomy lesson

**Model files live in:**

`trunk/app/models/[model_name].rb`

**Controller files live in:**

`trunk/app/controllers/[model_name]_controller.rb`

**View files live in:**

`trunk/app/views/[model_name]/`

**One important config file:**

`trunk/config/database.yml` - describes db conn.

# Legacy Chado support - things to change

"Feature" model as an example:

```
in trunk/app/models/feature.rb :
```

This line tells Chado to look for features in table "feature", not table "features":

```
set_table_name "feature" #default is "features"
```

This line tells Chado to use "feature\_id" not "id" as primary key:

```
set_primary_key "feature_id" # default is "id"
```

Repeat for each Chado tables we will support

# Hooking up tables to each other ("associations")

Feature -- belongs to --> Cvterm

Cvterm <-- has many -- features

Easy:

in `trunk/app/models/feature.rb`, add this:

```
belongs_to :cvterm
```

in `trunk/app/models/cvterm.rb`, add this:

```
has_many :features, :foreign_key => "type_id"
```

# We now have a powerful Chado console

Example: show me the name of the CV for the last thing loaded into Chado feature table

(try out by starting up `trunk/script/console`)

Before, using SQL:

```
SELECT cv.name FROM cv, cvterm, feature WHERE  
cv.cv_id = cvterm.cv_id AND cvterm.cvterm_id =  
feature.type_id ORDER BY feature_id DESC LIMIT  
1;
```

After, using Rails:

```
Feature.find(:last).cvterm.cv.name
```

# Example app: 5 min. Chado gene pages

model: done already

controller: make controller, write method to load up gene of interest into variable:

```
[~/chado_on_rails/trunk] jtr4v$ ./script/generate controller gene
exists app/controllers/
exists app/helpers/
create app/views/gene
exists test/functional/
exists test/unit/helpers/
create app/controllers/gene_controller.rb
create test/functional/gene_controller_test.rb
create app/helpers/gene_helper.rb
create test/unit/helpers/gene_helper_test.rb
```

then add this to `trunk/app/controllers/gene_controller.rb`:

```
def show
  @gene = Feature.find(params[:id])
end
```

view: add `trunk/app/views/gene/show.html.erb`, present some gene info to user (next slide)

# Simple view for gene page

```
<%= render :partial => 'header' %>

<% if @gene.nil? %>
There is no record for feature id <%= params[:id] %>

<% else %>
<ul>
<li>Name: <%= @gene.name %></li>
<li>Uniquename: <%= @gene.uniquename %></li>
<li>Feature id: <%= @gene.feature_id %></li>
<li>Cvterm: <%= @gene.cvterm.name %></li>
<li>Organism: <%= "#{@gene.organism.genus} #{@gene.organism.
species}" %></li>
<li>Time last modified: <%= @gene.timelastmodified %></li>
</ul>

<% end %>

<br/>
<%= render :partial => 'footer' %>
```

# Chado on Rails + Chado db + 19 lines of code = gene pages!

<http://0.0.0.0:3000/gene/show/17030045>



## **BOVINEGENOME.ORG** THE BOVINE GENOME DATABASE



- Name: FGPP\_ORF\_314\_from\_Chr1\_1
- Uniquename: FGPP\_ORF\_314\_from\_Chr1\_1
- Feature id: 17030045
- Cvterm: gene
- Organism: Bos taurus
- Time last modified: 2007-08-03 10:36:01 UTC

*Photos courtesy of USDA NRSC and Mike MacNeil, USDA/ARS Fort Keogh LARRL.*

The Bovine Genome Database is supported by the USDA CSREES, the Kleberg Foundation and the Texas Agricultural Experiment Station and is hosted at [Georgetown University](#). If you have comments or if you wish to report a problem, please contact the [Bovine Genome Database Administrator](#).

# Mature BGD genes pages

<http://genomes.arc.georgetown.edu/bovine/genepages/genes/BT10609>



**Home** About BGD Tools and Resources Genome Browsers

## BT10609

[Link to gene wiki page](#)

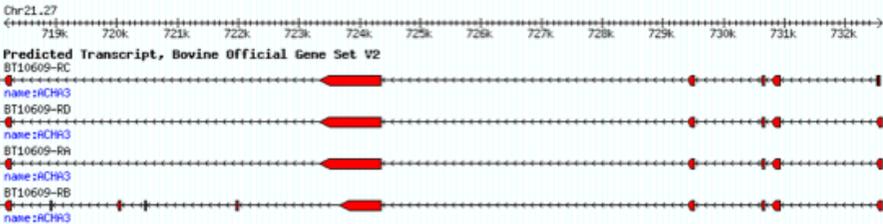
### [-] General Information

Transcripts for **BT10609**: [BT10609-RA](#) [BT10609-RB](#) [BT10609-RC](#) [BT10609-RD](#)

Ensembl Gene Symbol: **ACHA3**

Genomic Location: Chr21.27 718148 .. 732609 [-]

[View Evidence Tracks](#) [View in GBrowse](#)



### GO Annotations for BT10609

Biological Process	Molecular Function	Cellular Component
<a href="#">GO:0006810</a> transport	<a href="#">GO:0004872</a> receptor activity	<a href="#">GO:0005886</a> plasma membrane
<a href="#">GO:0006811</a> ion transport	<a href="#">GO:0004889</a> nicotinic acetylcholine-activated cation-selective channel activity	<a href="#">GO:0005892</a> nicotinic acetylcholine-gated receptor-channel complex
<a href="#">GO:0007165</a> signal transduction	<a href="#">GO:0005216</a> ion channel activity	<a href="#">GO:0016020</a> membrane
<a href="#">GO:0007399</a> nervous system development	<a href="#">GO:0005230</a> extracellular ligand-gated ion channel activity	<a href="#">GO:0016021</a> integral to membrane
<a href="#">GO:0035095</a> behavioral response to nicotine	<a href="#">GO:0015464</a> acetylcholine receptor activity	<a href="#">GO:0030054</a> cell junction
<a href="#">GO:0060084</a> synaptic transmission involved in micturition	<a href="#">GO:0030594</a> neurotransmitter receptor activity	<a href="#">GO:0045202</a> synapse
		<a href="#">GO:0045211</a> postsynaptic membrane

### [+] Transcripts for BT10609

### [+] Sequences

\*The scaffold Min and Max coordinates indicate the minimum and maximum range of the gene on the scaffold, and not the start and end coordinates of the gene.

Photos courtesy of USDA NRSC and Mike MacNeill, USDA/ARS Fort Keogh LARRL.

The Bovine Genome Database is supported by grants 2007-35616-17882 and 2010-65205-20407 from the USDA National Institute of Food and Agriculture and is hosted at Georgetown University. If you have comments or if you wish to report a problem, please contact the Bovine Genome Database Administrator.

# Testing

- Rails ships with sophisticated testing suite (a whole other talk)
- Chado on Rails contains unit tests for associations, other reality checks
- Easy to add more sophisticated tests (functional, etc.)
- `rake test`

# Caching

- Add one line in controller to cache gene pages:  
`cache_page :show`
- More sophisticated caching (expire cache after x days, caching expensive queries, etc.)

# Deployment - a non-issue

- Used to be a major weakness of Rails
- Passenger Phusion + Capistrano provide dead-easy deployment, plays nicely with Apache and Nginx
- Fairly easy to set up staging environments to try out new releases

# Other applications

- Powerful framework for the development of applications on top of Chado
- Ideas for possible applications?
- Help available from me, Chris, Rails community
- <http://rubyforge.org/projects/chadoonrails/>  
(there's another dead project by the same name)
- Justin Reese:  
[justaddcoffee@gmail.com](mailto:justaddcoffee@gmail.com)
- Chris Childers:  
[genetics.guy@gmail.com](mailto:genetics.guy@gmail.com)

# Thanks

Chris Childers

Chris Elsik

Jay Sundaram

Darren Hagen

Monica Muñoz-Torres

Steve Shen

Anna Bennett

Shu Tao

Don Vile