

Ruby on Rails +10 のアイデア

--- KEGG on Rails の経験から得られた DB 構築のノウハウを中心に

癸生川 絵里

サイエンス・テクノロジー・システムズ株式会社

片山 俊明

東京大学 医科学研究所 ヒトゲノム解析センター ゲノムデータベース分野



は使いにくい

GLYCOLYSIS

こんなのを

Nucleotide sugars metabolism

Pentose and glucuronate interconversions

Starch and sucrose metabolism

2.7.1.41

3.1.3.10

α -D-Glucose-1P

5.4.2.2

DBGET Result: C.intestinalis 289557

http://www.genome.jp/dbget-bin/www_bget? C.intestinalis:289557

Kanehisa Lab. (Tokyo) Google ニュース (2587) アップル 便利 research hobby person

DBGET Result: COMPOUND C00566

http://www.genome.jp/dbget-bin/www_bget? C00566

KEGG *Ciona intestinalis* (sea squirt): 289557

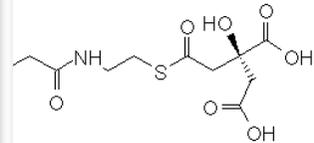
Entry	289557	CDS	C.intestinalis
Definition			
KO	KO: K01679 fumarate hydratase		
Pathway	PATH: dcin00020 Citrate cycle (TCA cycle) PATH: dcin00720 Reductive carboxylate cycle		
Class	Gene catalog		
Other DBs	JGI: 289557		
LinkDB	PDB All DBs		
AA seq	502 aa AA seq DB search		
MIGAPFTVVRTSQIVTGFCSRKPIFNRIITMFGANTARSLMNFKIGGPGDKMPIEVIHAFALFAGDLDHFPLVIWQTGSGTQSNMINEVLANFDTYPTAMHIGVARSITNVLLPGLRTLETALC SGYVQVKNGIARVESCLPRLYELAAGGTAVCPNKFEALAAHDALVEASGAFNTVACSLMKIAPGKVNPTQCEAITMVCAQVMGNHVAVSIGGSSFTFENCVNGITANEDRISLLHESMLVLTALNLLLTAEQDFDMVVKPEDMLGPK			
NT seq	1506 nt NT seq +upstream 0		
atgattggtgctccgctttacagttagaacttcagaaaacctatattcaatagaattctcactatggttttcgtatagaactgacagcttttggggggagcaaacactgcaagatcgctcatgaactccaattgaagtgatccacgcttttgcgattctgaatttggcttctgctgagttggctaatgcgctggagatttggactcacattcccgcctgttctaatatgaacataaacgaagtcttagcaaaccttggaagcaaaatccagttcatccaaatgagatacacaacacagcaatgcacattggagtcctggcttaagaacattggaacccgctttggagttaagatcgccgctacgcacacacaagatgc			

KEGG KO: K00024

Entry	K00024	KO
Name	E1.1.1.37	
Definition	malate dehydrogenase	
Class	Metabolism; Carbohydrate Metabolism; Citrate cycle (TCA cycle) [PATH:ko00020] Metabolism; Carbohydrate Metabolism; Pyruvate metabolism [PATH:ko00620] Metabolism; Carbohydrate Metabolism; Glyoxylate and dicarboxylate metabolism [PATH:ko00630] Metabolism; Energy Metabolism; Carbon fixation [PATH:ko00710] Metabolism; Energy Metabolism; Reductive carboxylate cycle (CO2 fixation) [PATH:ko00720]	
Other DBs	RN: R00342 EC: 1.1.1.37 GO: 0030060	
Genes	SCE: YDL078C(MDH3) YOL126C(MDH2) TBR: Tb10.61.0980 Tb10.70.5110 Tb10.70.5120 Tb11.01.3040 TCR: 507883.100 LMA: LmjF34.0130 ECO: b2020(hisD) HSO: HS_1055(mdh) VFI: VF1252 PEN: PSEEN2669 FTF: FTF0535c(mdh) HCH: HCH_01978(mdh) ABO: ABO_1248(mdh) BXE: Bxe_B2898 BCN: Bcen_4435 RFR: Rfer_1803 Rfer_2403 POL: Bpro_3598 HHE: HH1571(mdh) MXA: MXAN_3538(mdh) ERU: Erum4380 RET: RHE_PF00428 RDE: RD1_1617(mdh) PGE: PGE7434(mdh)	

KEGG COMPOUND: C00566

Entry	C00566	Compound
Chemical structure: <chem>CC(=O)NCCSC(=O)C[C@H](O)C(=O)O</chem>		
Pathway: Citrate cycle (TCA cycle)		
EC: 6.2.1.18		



[GenomeNet](#)

espu00010 - Strongylocentrotus purpuratus (ムラサキウニ) (EST)

http://dev.kegg.jp:9064/pathway/show/espu00010#

バスイ>代謝>糖質代謝

解糖系・糖新生系 - Strongylocentrotus purpuratus (ムラサキウニ) (EST)

[Pathway menu | Pathway help | Ortholog table]

Pathways Organisms

GLYCOLYSIS

α-D-Glucose-1P

α-D-Glucose

D-Glucose 6-sulfate

β-D-Glucose

Arbutin (extracellular)

Salicin (extracellular)

Carbon fixation in photosynthetic organisms

Nucleotide sugars metabolism

Pentose and glucuronate interconversions

Starch and sucrose metabolism

Galactose metabolism

Fructose and mannose metabolism

Pentose phosphate pathway

cpd:C00103

NAME - D-Glucose 1-phosphate
- alpha-D-Glucose 1-phosph
- Corl ester
- D-Glucose alpha-1-phosp

espu:13141

NAME
DEFINITION
PATHWAYS path: espu00010 Glycolysis / Gluconeogenesis
path: espu00052 Galactose metabolism
path: espu00500 Starch and sucrose metabolism

cpd:C00267

NAME
DEFINITION
PATHWAYS
COMMENT
MASS 180.063

ec:2.7.1.69

NAME protein-N(pi)-phosphohistidine:sugar N(pi)-phosphotransferase
DESCRIPTION protein-N(pi)-phosphohistidine-sugar phosphotransferase

cpd:C06189

NAME
DEFINITION
PATHWAYS
COMMENT
FORMULA C3H6O9P2

Select organism:

- 真核生物 (85)
 - 動物 (25)
 - 脊椎動物 (13)
 - ホヤ類 (3)
 - 棘皮動物 (1)
 - espu Strongylocentrotus purpuratus (ムラサキウニ) (EST)
 - 昆虫類 (5)
 - dme Drosophila melanogaster (キロショウジョウバエ)
 - ddpo Drosophila pseudoobscura
 - daga Anopheles gambiae (ハマダラカ)
 - dame Apis mellifera (ミツバチ)
 - dbmo Bombyx mori (カイコ)
 - 線虫類 (2)
 - 扁虫類 (1)
- 植物 (28)
- 菌類 (20)
- 原生生物 (12)
- 原核生物 (411)
 - 真正細菌 (382)
 - プロテオバクテリア (200)
 - アシドバクテリア (2)
 - グラム陽性菌 (95)
 - 放線菌 (27)
 - フソバクテリア (1)

こんな感じにしたい

E-Cell 3D には遠く及びませんが

KEGG on Rails のできるまで

レシピ

- 今回作るもの
 - KEGG on Rails
- 用意するもの
 - <ftp://ftp.genome.jp/pub/kegg/> のミラー
 - Ruby on Rails
 - BioRuby
 - 癸生川さん
- できあがったものが、ここにありm、前のプレゼンです。
 - ♪ 3分間クッキング → 明日のハンズオンで試してみてください

KEGG on Rails 特徴概要

- KEGG を少しでも使いやすくするためのプロトタイプ（日本語化含む）
- FTP で公開されている KEGG のデータを利用
 - 公開データから KEGG の機能を再現
- Rails を利用し全データを正規化
 - 今後の開発のインフラ
- 特にパスウェイ部分について UI の改善
 - RJS による AJAX 実験場
 - ポップアップ
 - 階層型データ

Rails によるサイト構築の流れ

- DB 設計
 - スキーマを migration という仕組みで定義
 - 初期データをロード
 - タブ切り、CSV、fixture、一個ずつ手で、などなど
 - モデルを生成
 - ビューを生成 (HTML のテンプレート)
 - コントローラを生成 (テンプレートに埋め込む値、ヘルパー)
 - 見た目 (CSS) や UI (AJAX) などに凝る
- ```
% rails kegg
% vi config/database.yml
% mysqladmin -u root create database kegg_development
% ./script/generate migration create_kegg_tables
% rake db:migrate
% mysqlimport -u root kegg_development `pwd`/linkdb.txt
% ./script/generate scaffold pathway
% vi app/models/pathway.rb
% vi app/views/pathway/index.rhtml
% vi app/controllers/pathway_controller.rb
% vi public/stylesheets/kegg.css
```

# 実際は、

---

- ftp にすぎ一時間がかかる
- 各データベースのフォーマットがうぜー
- えーっと habtm が多くて正規化できませんよ？
- きゃー Rails でイメージマップからのポップアップ、誰かやってないの？

rails ajax イメージマップ ポップアップ の検索結果 約 45 件中 1 - 10 件目 (0.16 秒)

- ぐはあ、ツリー型のデータを閉じたり開いたりとか標準装備じゃないの？
- 日本語化っていうか UTF8 にしないと化けるし、

# ftp 問題

---

- KEGG のディレクトリ構成は取捨選択しないとミラーしにくい
  - 古いリリースがサブディレクトリに全部あったり
  - サブデータベースごとにファイル名のポリシーが違うとか
  - KGML はディレクトリ名にバージョン番号がついてたり
  - genome とか、ただのファイルが tarfiles/ の下にあったり
  - size0 というサイズ 497801 bytes のファイルが転がっていたり
- ともあれ、丸1日～数日くらいかければ取れます
  - 選べば半日、必要なものを知っていればすぐかも

```
% lftp ftp://ftp.genome.jp/pub/kegg/
ftp> mirror -e ligand
ftp> mirror -e pathways
ftp> mirror -e genomes
ftp> mirror -e brite
ftp> lcd xml
ftp> cd xml
ftp> mirror -e KGML_v0.6.1
```

# KEGG 独自フォーマット問題

---

- かたくなに、伝統的なフラットファイル形式を採用
  - 毎週の会議でタグが増えたり減ったりフォーマットはコロコロ変わる
  - HTML ファイルにまで DBGET 用のヘッダがついていたり

- とりあえず BioRuby でがんばる
  - KGML
  - GENES
  - GENOME
  - ENZYME
  - COMPOUND ...

```
<html>
<!--
ENTRY zmo03070
DEFINITION Type III secretion system - Zymomo
---->
<head>
 :
</body>
<!--
///
---->
</html>
```

# habtm 問題

---

- has\_and\_belongs\_to\_many
  - ある COMPOUND は複数の ENZYME に代謝される
  - ある ENZYME は複数の COMPOUND を代謝する

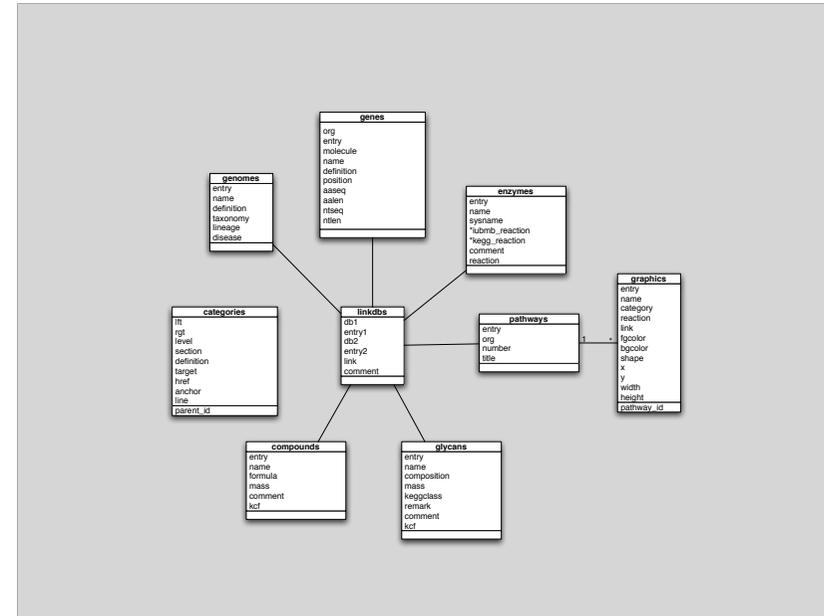
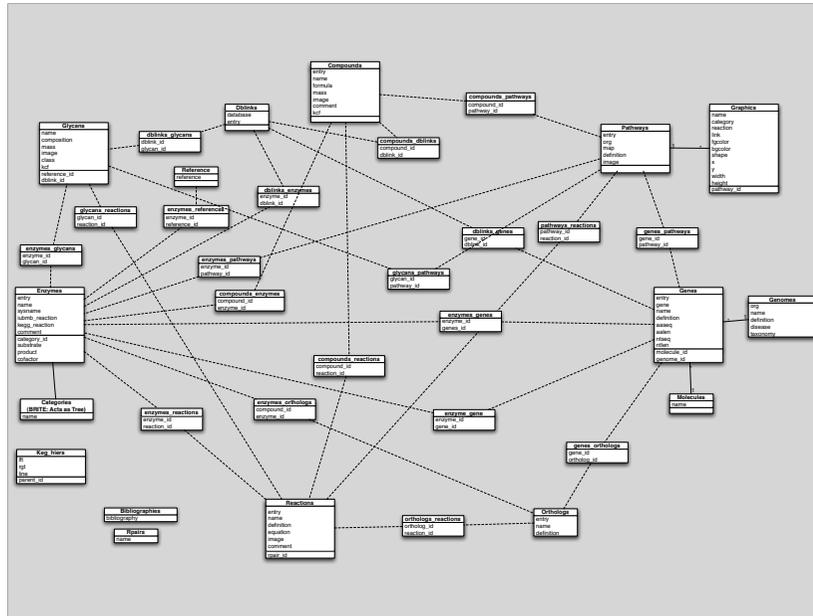
```
>> Compound.dbletry("cpd:C00005").enzymes
=> ["ec:1.13.11.20", "ec:1.14.99.3", "ec:1.14.99.9"]
```

```
>> Enzyme.dbletry("ec:1.13.11.20").compounds
=> ["cpd:C00097", "cpd:C00007", "cpd:C00606", "cpd:C00004", "cpd:C00005", "cpd:C00020"]
```

- どちらのデータベースから先にロードすればいいのでしょうか

```
>> enz = Enzyme.create(:entry => "ec:1.13.11.20") ← 新しい ENZYME を保存するとき
>> cpd = Compound.find_by_entry("C00005") ← すでに COMPOUND があれば
>> enz.compounds << cpd ← 探してきて関係を追加できるが、
>> enz.save
```

- しかも↑これ、実はこのまま頑張ってもすげー遅い



LinkDB 使用前・使用後（新旧スキーマ図）

-- 複雑すぎる habtm はデータのインポートがかなり遅くなる

得られたもの

## ☪メニュー (ウェブなどで参考例を見つけきれなかったテクあり☑)

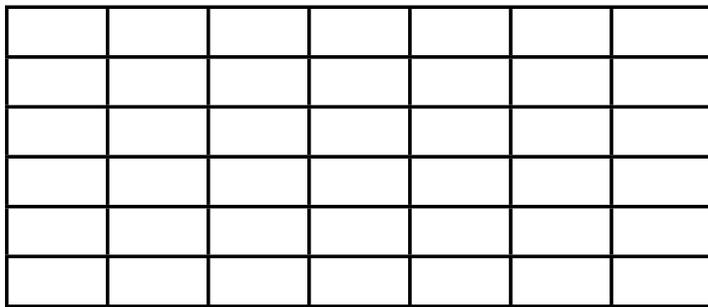
---

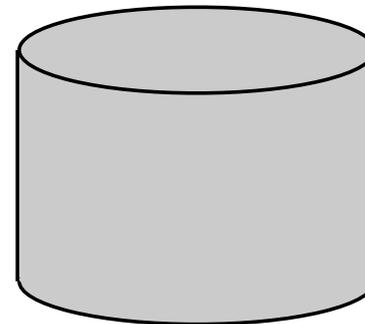
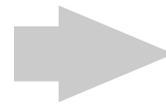
- タブ区切りファイルのロード
  1. 第1カラム目がIDではないタブ切りファイルをロードする技
  2. セルに改行が含まれているエクセルファイルをやっつける技
- Rails + AJAX でポップアップ
  3. イメージマップを AJAX 呼び出しに変更しポップアップを生成する技
  4. ポップアップの表示位置を画像からの相対座標で指定する技
  5. ポップアップを半透明やドラッグ可能にする技
- Rails + AJAX でツリー構造を扱う
  6. `acts_as_nested_set` を `acts_as_tree` としても使いたい
  7. サブツリーの開閉を AJAX で行う方法
- 定型のタスクは rake に集約
  8. タスクを namespace に分ける方法
  9. 複数のタスクに共通の処理はメソッドに
- ActiveRecord のモデルは Ruby のクラスでもある
  10. モデルにもガシガシと都合の良いメソッドを追加する

# タブ区切りファイルのロード

---

- よくある Excel で作られた表のデータベース化の依頼とか
- 2000万件 SQL 発行するより mysqlimport (LOAD LOCAL) の方が速いとか



# 1. 第1カラム目がidではないタブ切りファイル

---

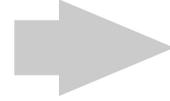
- Rails のデフォルトはテーブルの第1カラムがid という名前の auto\_increment
  - でも、ふつー身の回りのタブ切りテキストにそんなものはない！
- mysqlimport する前にデータをいじって空の1カラム目を増やす（ありがち）

```
% perl -pe 's/^/t/' data/tabfile.txt > data/hoges.txt
% mysqladmin -u root create hogedb
% script/generate migration hoges_table
% vi db/migrate/001_hoges_table.rb
def self.up
 create_table :hoges do |t|
 t.column :foo, :string
 t.column :bar, :integer
 t.column :gee, :text
 end
end
% rake db:migrate
% mysqlimport -u root hogedb `pwd`/data/hoges.txt
```

# 1. 第1カラム目がidではないタブ切りファイル

- 巨大であるとかの理由でデータファイルをいじりたくない場合もある

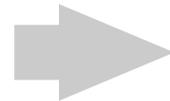
```
def self.up
 create_table :hoges do |t|
 t.column :foo, :string
 t.column :bar, :integer
 t.column :gee, :text
 end
end
```



```
mysql> desc hoges;
+-----+-----+
| id | primary_key |
| foo | string |
| bar | integer |
| gee | text |
+-----+-----+
```

- 単純に id を最後のカラムにしたらいんじゃないんじゃん？

```
def self.up
 create_table :hoges do |t|
 t.column :foo, :string
 t.column :bar, :integer
 t.column :gee, :text
 end
 remove_column :hoges, :id
 add_column :hoges, :id, :primary_key
end
```



```
mysql> desc hoges;
+-----+-----+
| foo | string |
| bar | integer |
| gee | text |
| id | primary_key |
+-----+-----+
```

注：ただし、db/migrate/\*.rb が db/schema.rb にまとめられる際には技が消える orz

# 1. 第1カラム目がidではないタブ切りファイル

---

- :id => false にしておいて add\_column でいいのかも by 中尾さん

```
def self.up
 create_table :hoges, :id => false do |t|
 t.column :foo, :string
 t.column :bar, :integer
 t.column :gee, :text
 end
 add_column :hoges, :id, :primary_key
end
```

## 2. セルに改行が含まれる Excel ファイル

- 送られてきた Excel ファイルが、セル内に改行を含む場合
  - タブ切りで保存しても、通常の 1 行ずつ読み込む処理をするとコケる

1	カテゴリ	名称	組織	Availability	URL	キーワード
2	パラメトリック連鎖解析 ノンパラメトリック連鎖解析 罹患同胞対解析 TDT	GENEHUNTER	Fred Hutchinson Cancer Research Center	Download	Broad Institute http://www.broad.mit.edu/ftp/distribution/software/genehunter/	パラメトリック連鎖解析 ノンパラメトリック連鎖解析 罹患同胞対解析 QTL解析 TDT
3	パラメトリック連鎖解析	LINKAGE	Lab of Statistical Genetics-Rockefeller Univ	Download	Rockefeller Univ. http://linkage.rockefeller.edu/	パラメトリック連鎖解析
4	パラメトリック連鎖解析	FASTLINK	the National Center for Biotechnology Informa	Download	NCBI http://www.ncbi.nlm.nih.gov/	パラメトリック連鎖解析
5	パラメトリック連鎖解析	LIPED	Lab of Statistical Genetics-Rockefeller Univ	Download	Rockefeller Univ. ftp://linkage.rockefeller.edu/liped/	パラメトリック連鎖解析
6	連鎖マップ作成	MAP+	the Genetic Epidemiology Research Group-Univ	Download	Southampton Univ.  http://cedar.bham.ac.uk/	連鎖マップ作成
7	連鎖マップ作成	MAPL	東京大学植物育種情報研究所	Download	ISGI http://lbn.ab.a.u-tokyo.ac.jp/	連鎖マップ作成 QTL解析
8	パラメトリック連鎖解析	MAPMAKER/EXP	Whitehead Institute/MIT Center for Genome Res	Download	MIT http://ftp.genome.wi.mit.edu/pub/	パラメトリック連鎖解析

## 2. セルに改行が含まれる Excel ファイル

- 最後の列にダミーのありえない文字列 %dummy% を入れておく

	H	I	J	K	L	M	N	O	P
1	URL	キーワード	特徴	論文名	引用数	引用数取得日	ツールアップデート日	%dummy%	
2	Broad Institute http://www.broad.mit.edu/ftp/distribution/software/genhunter/	パラメトリック連鎖解析 ノンパラメトリック連鎖解析 罹患同胞対解析 QTL解析 TDT	比較的小さな家系。 多数座位の取扱が可能。 (Lander-Green-Kruglyakアルゴリズム)	Parametric and nonpa	1729	10.05.06	05.07.01	%dummy%	
3	Rockefeller Univ. http://linkage.rockefeller.edu/	パラメトリック連鎖解析	家系について汎用。 比較的少ない座位。(5座位程度まで) (Elston-Stewartアルゴリズム) 複数のプログラムからなるパッケージ。 汎用バージョンとCEPH用バージョンがある。	Easy calculations of lo	745	10.05.06	05.31.03	%dummy%	
4	NCBI http://www.ncbi.nlm.nih.gov/	パラメトリック連鎖解析	家系について汎用。 比較的少ない座位。(5座位程度まで) (Elston-Stewartアルゴリズム) 複数のプログラムからなるパッケージ。	Faster sequential gene	832	10.05.06	10.07.05	%dummy%	
5	Rockefeller Univ. http://linkage.rockefeller.edu/	パラメトリック連鎖解析	家系について汎用。2座位。	Estimation of the recor	241	10.05.06	08.21.02	%dummy%	
6	Southampton Univ.  http://cedar.bath.ac.uk/	連鎖マップ作成	多数マーカー間の組み換え	Counting algorithms fo	1	10.05.06	03.15.01	%dummy%	
7	ISGI http://bm.ab.a.u-tokyo.ac.jp/	連鎖マップ作成	多数マーカー間の組み換え	-	0		03.15.04	%dummy%	
8	MIT http://ftp.genome.wi.mit.edu/	パラメトリック連鎖解析	多数マーカー間の組み換え	MARMAKER: an intera	2418	10.05.06	02.11.02	%dummy%	

## 2. セルに改行が含まれる Excel ファイル

---

- タブ切りテキストで保存して置換する

```
% perl -pe 's/[%r%n]/;/g; s/%dummy%/%n/g' tabfile.txt > hoges.txt
% mysqlimport -u root hogedb `pwd`/data/hoges.txt
```

- ま、中身いじらないなら CSV のまま突っ込んでもいいわけですが、  
ちまたのデータは他にもいろいろ前処理が必要だったりします
  - 日付のフォーマットレコードごとに 2002.8.21 とか 05.31.93 とか  
2005/2 とか、妙にバラバラだったり
  - セルごとに漢字コードが違っていたり
    - どーやって作ったのか？！

# Rails + AJAX でポップアップ

---

- HTML に隠し <div> で埋め込んでいて JavaScript でやればいいじゃん
  - <div> が巨大になると効率が悪い
  - JavaScript を書きたくない (^^;
  - Rails では RJS を使うと細かいチューニングも簡単
- AJAX (Asynchronous JavaScript and XML)
  - ページ (DOM) の一部を JavaScript の非同期通信で更新
  - ウェブの操作性を一変 - アプリ化 (Google maps, docs, Yahoo pipes, etc.)
- RJS Template (Rails JavaScript Template?)
  - Ruby のコードから JavaScript を自動生成、誰でも AJAX が書ける



### 3. イメージマップから AJAX でポップアップ

---

ビュー

```
<div id="pathway">

<map name="mapdata">
<area shape="rectangle" coords="100,100,200,200" href="#"
 onclick="<%= remote_function(:url => {:action => 'popup', :entry => 'エントリID'}) %>"
</map>
 "new Ajax.Updater('popup', '/pathway/popup?entry=hsa%3A1737', {asynchronous:true, evalScripts:true})"
</div>
```

コントローラ

```
def popup
 @data = "ポップアップに表示するデータを用意"
 @popup_id = "エントリID"
 render :update do |page|
 page.insert_html :bottom, "pathway", :partial => "popup"
 page[@popup_id].set_style :left => "X座標", :top => "Y座標"
 end
end
```

パーシヤル

app/views/pathway/\_popup.rhtml に  
@popup\_id と data を変数として渡す

## 4. 表示位置を相対座標で指定する技

- Prototype.js の `makePositioned` を利用



```
<div id="parent">
 <div id="child">hoge</div>
</div>
```

```
render :update do |page|
 page["parent"].make_positioned
 page.insert_html :bottom, "parent", :partial => "child"
 page["child"].set_style :position => "absolute"
 page["child"].set_style :left => "#{x}px", :top => "#{y}px"
end
```

## 5. ポップアップを半透明やドラッグ可能にする技

- RJS なら <div> 単位で不透明度やドラッグ可能などの指定がたったの 1 行



```
<div id="child">
 moinmoin
</div>

render :update do |page|
 page["child"].set_opacity 0.9
 page.draggable "child"
end
```

注：ただし、Firefox や IE (Vistaも) では移動後不透明になるっぽい orz

# Rails + AJAX でポップアップ まとめ

## ビュー

```
<div id="pathway">

<map name="mapdata">
<%- @graphics.each do |g| -%>
<area
 shape=<%= make_shape(g) %>
 coords=<%= make_coords(g) %>
 href="#"
 onclick="<%= remote_function(
 :update => 'popup',
 :url => {:action => 'popup',
 :entry => g.name,
 :x => g.x,
 :y => g.y,
 :link => g.link}) %>">
<%- end -%>
</map>
</div>
```

Rails でのマウス座標の取り方、誰か教えてください。。

## コントローラ

```
def popup
 @data = "ポップアップに表示するデータを用意"
 @popup_id = "ポップアップごとにユニークなIDをアサイン"

 render :update do |page|
 page['pathway'].make_positioned
 x = params[:x].to_i
 y = params[:y].to_i

 page.insert_html :bottom, "pathway",
 :partial => "popup"
 page[@popup_id].set_style :position => "absolute"
 page[@popup_id].set_style :left => "#{x}px",
 :top => "#{y}px"
 page[@popup_id].set_opacity 0.9
 page.draggable @popup_id
 end
end
```

## パーシャル

```
<div id="<%= @popup_id %>">
 * 閉じるボタンを付ける
 * @data を表にレンダリングする
</div>
```

# Rails + AJAX でツリー構造を扱う

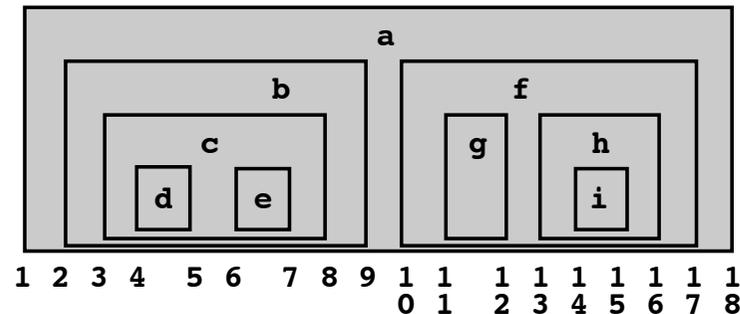
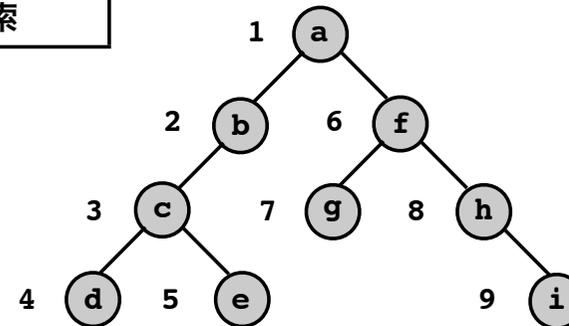
## 階層型データの3通りの表現方法

- path /a/f/g
  - text
- link a←f←g
  - acts\_as\_tree
- nested set
  - acts\_as\_nested\_set

	○	×
path	全文検索	データ量
link	探索	クエリ量
nest	クエリ量	探索

キホンの的にRDBとツリーは相性が悪い

id	parent_id	data	path	lft	rgt
1	null	a	/a	1	18
2	1	b	/a/b	2	9
3	2	c	/a/b/c	3	8
4	3	d	/a/b/c/d	4	5
5	3	e	/a/b/c/e	6	7
6	1	f	/a/f	10	17
7	6	g	/a/f/g	11	12
8	6	h	/a/f/h	13	16
9	8	i	/a/f/h/i	14	15



> SELECT \* WHERE CHILD.LEFT IS BETWEEN PARENT.LEFT AND PARENT.RIGHT

## 6. acts\_as\_tree と acts\_as\_nested\_set の融合

---

- Rails のプラグイン (標準添付)
  - Linked list のための acts\_as\_tree
  - Nested set のための acts\_as\_nested\_set
- acts\_as\_nested\_set に acts\_as\_tree の機能を追加したい
  - root まで辿る ancestors などの機能がない

### Linked list

acts_as_tree	メソッドの機能
tree.roots	rootノードのリスト
node.parent	親ノード
node.children	子ノードのリスト
node.root	rootノード
node.ancestors	rootノードまでのリスト
node.siblings	兄弟ノードのリスト

### Nested set

acts_as_nested_set	メソッドの機能
node.root?	rootノードかどうか
node.children?	rootノードでないかどうか
node.add_children	子ノードを追加
node.children_count	子ノードの数
node.all_children	全ての子ノードのリスト
node.direct_children	直下の子ノードのリスト

## 6. acts\_as\_tree と acts\_as\_nested\_set の融合

- acts\_as\_nested\_set に acts\_as\_tree の機能を追加したい
  - 自モデルの parent\_id を foreign\_key で参照し、メソッドをコピー

```
class Category < ActiveRecord::Base
 acts_as_nested_set

 belongs_to :parent, :class_name => name, :foreign_key => "parent_id"

 def ancestors
 node, nodes = self, []
 nodes << node = node.parent until node.root?
 nodes.pop
 return nodes
 end
end
```

ココで自分のパスウェイ ID から  
親階層を取るのに使われてたり

hsa00030 - Homo sapiens (ヒト)

http://dev.kegg.jp:9064/pathway/show/hsa00030#

パスウェイ > 代謝 > 糖質代謝

KEGG  
Pentose Phosphate Pathway - Homo sapiens (ヒト)

[ Pathway menu | Pathway help | Ortholog table ]

Pathways

PENTOSE PHOSPHATE PATHWAY

4.2.1.39

## 7. サブツリーの開閉を AJAX で行う方法

- 階層のインデントは CSS で分かりやすくできるかも

```
<div id="category">
 <div id="level1">
 ここは第1階層
 <div id="level2">
 ここは第2階層
 <div id="level3">
 ここは第3階層
 </div>
 </div>
 </div>
</div>
```

```
div.level1 {
 background: #FF5C61;
 margin-left: 20px;
}

div.level2 {
 background: #FFCC5D;
 margin-left: 20px;
}

div.level3 {
 background: #BEFF5C;
 margin-left: 20px;
}
```

```
div.level1 {
 margin-left: 20px;
 border-style: dotted;
 border-width: 0px 0px 0px 1px;
}

div.level2 {
 margin-left: 20px;
 border-style: dotted;
 border-width: 0px 0px 0px 1px;
}

div.level3 {
 margin-left: 20px;
 border-style: dotted;
 border-width: 0px 0px 0px 1px;
}
```



## 7. サブツリーの開閉を AJAX で行う方法

- 開くときは空の `<div>` にサブツリーの内容を `replace_html` する

```
app/controllers/pathway_controller.rb
```

```
def expand
```

```
 @parent = Category.find(params[:id])
```

```
 render :update do |page|
```

```
 @nodes = @parent.direct_children
```

```
 page.replace_html "node_#{@parent.id}", :partial => "expand"
```

```
 page.visual_effect :highlight, "node_#{@parent.id}", :duration => 0.5
```

```
 end
```

```
end
```

```
app/views/pathway/_expand.rhtml
```

```
<%= label, title, entry = label_title_entry(@parent) -%>
```

```
<%= label %>
```

```
<%= link_to_remote(title, :url => {:action => "collapse", :id => @parent.id}) %>
```

```
<%= @nodes.each do |node| -%>
```

```
<div id="node_<%= node.id %>" class="level<%= node.level %>">
```

```
<%= label, title, entry = label_title_entry(node) -%>
```

```
<%= label %>
```

```
<%= if node.direct_children.size == 0 -%>
```

```
<%= link_to_remote(title, :url => {:action => "category_link", :id => entry}) %>
```

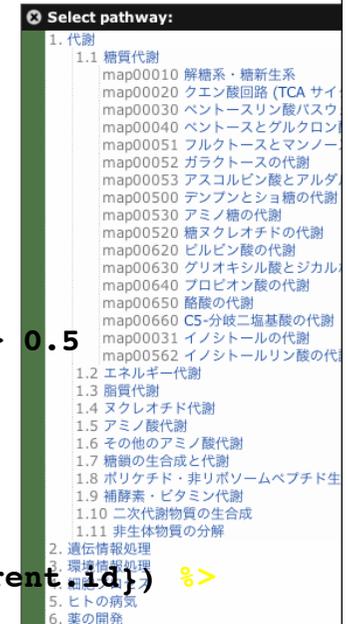
```
<%= else -%>
```

```
<%= link_to_remote(title, :url => {:action => "expand", :id => node.id}) %>
```

```
<%= end -%>
```

```
</div>
```

```
<%= end -%>
```



## 7. サブツリーの開閉を AJAX で行う方法

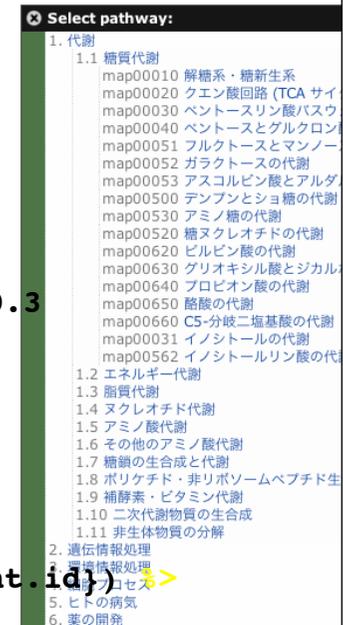
- 閉じるときは <div> の中身を空っぽに replace\_html する

```
app/controllers/pathway_controller.rb
```

```
def collapse
 @parent = Category.find(params[:id])
 render :update do |page|
 page.replace_html "node_#{@parent.id}", :partial => "collapse"
 page.visual_effect :pulsate, "node_#{@parent.id}", :duration => 0.3
 end
end
```

```
app/views/pathway/_collapse.rhtml
```

```
<%= label, title, entry = label_title_entry(@parent) -%>
<%= label %>
<%= link_to_remote(title, :url => {:action => "expand", :id => @parent.id}) %>
```



# 定型のタスクは rake に集約

---

- なんども実行する作業は rake で実行
  - Ruby スクリプトで Makefile を書くことができる make のような
  - Rails の定型作業は rake 化されている
    - スキーマ変更、テスト、tmp, log などの掃除 etc.

```
% rails hoge
```

```
% cd hoge
```

```
% rake -T
```

```
rake db:migrate # Migrate the database through scripts in db/migrate. Target specific version with VERSION=x
rake db:schema:dump # Create a db/schema.rb file that can be portably used against any DB supported by AR
rake db:schema:load # Load a schema.rb file into the database
rake db:sessions:clear # Clear the sessions table
rake db:sessions:create # Creates a sessions table for use with CGI::Session::ActiveRecordStore
rake db:structure:dump # Dump the database structure to a SQL file
rake doc:app # Build the app HTML Files
rake doc:plugins # Generate documation for all installed plugins
rake log:clear # Truncates all *.log files in log/ to zero bytes
rake rails:update # Update both configs, scripts and public/javascripts from Rails
rake stats # Report code statistics (KLOCs, etc) from the application
rake test # Test all units and functionals
rake tmp:cache:clear # Clears all files and directories in tmp/cache
rake tmp:clear # Clear session, cache, and socket files from tmp/
rake tmp:create # Creates tmp directories for sessions, cache, and sockets
rake tmp:pids:clear # Clears all files in tmp/pids
rake tmp:sessions:clear # Clears all files in tmp/sessions
rake tmp:sockets:clear # Clears all files in tmp/sockets
```

## 8. タスクを namespace に分ける方法

---

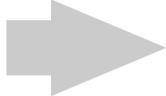
- namespace ブロックで囲むだけ

- lib/tasks/load\_kegg\_genes.rake

```
desk "Load KEGG GENES database"
task :load_kegg_genes => :environment do
 Bio::FlatFile.open(genes_file).each do |entry|
 Gene.create(gene2hash(entry))
 end
end
```

- lib/tasks/kegg/load\_genes.rake (ディレクトリも分けるとキレイ)

```
namespace "kegg" do
 desk "Load KEGG GENES database"
 task :load_genes => :environment do
 Bio::FlatFile.open(genes_file).each do |entry|
 Gene.create(gene2hash(entry))
 end
 end
end
end
```



```
% rake -T
rake kegg:load_all # Load all KEGG databases
rake kegg:load_category # Load KEGG category (.keg) file
rake kegg:load_compound # Load KEGG COMPOUND database
rake kegg:load_enzyme # Load KEGG ENZYME database
rake kegg:load_genes # Load KEGG GENES database
rake kegg:load_genome # Load KEGG GENOME database
rake kegg:load_glycan # Load KEGG GLYCAN database
rake kegg:load_kegg # Load all KEGG databases
rake kegg:load_pathway # Load KEGG PATHWAY database from KGML
rake kegg:show_config # Show configurations of KEGG files
```

## 9. 複数のタスクに共通の処理はメソッドに

---

- rake ファイルの中にメソッド定義できます (make ではキビシイ)

```
namespace "kegg" do
 def save_link(db1, entry1, db2, entry2, link = nil)
 #save_line_in_linkdb(db1, entry1, db2, entry2, link)
 save_line_in_tabfile(db1, entry1, db2, entry2, link)
 STDERR.print "."
 end

 def save_line_in_linkdb(db1, entry1, db2, entry2, link)
 hash = {
 :db1 => db1,
 :entry1 => entry1,
 :db2 => db2,
 :entry2 => entry2,
 :link => link,
 }
 Linkdb.create(hash)
 end

 def save_line_in_tabfile(db1, entry1, db2, entry2, link)
 puts ['', db1, entry1, db2, entry2, link].map{|x| x.to_s}.join("\t")
 end
end
```

## 9. 複数のタスクに共通の処理はメソッドに

---

- メソッド化しておくことで「DB 間のリンクを保存する」を抽象化
  - Linkdb.create で毎回 SQL 保存するか、タブ切りファイルに保存するか

### COMPOUND

```
entry.glycans.each do |name|
 save_link("cpd", cpd, "gl", name)
end

entry.reactions.each do |name|
 save_link("cpd", cpd, "rn", name)
end

entry.pathways.each do |line|
 if name = line[/PATH:\s*(\S+)/, 1]
 save_link("cpd", cpd, "path", name)
 end
end
```

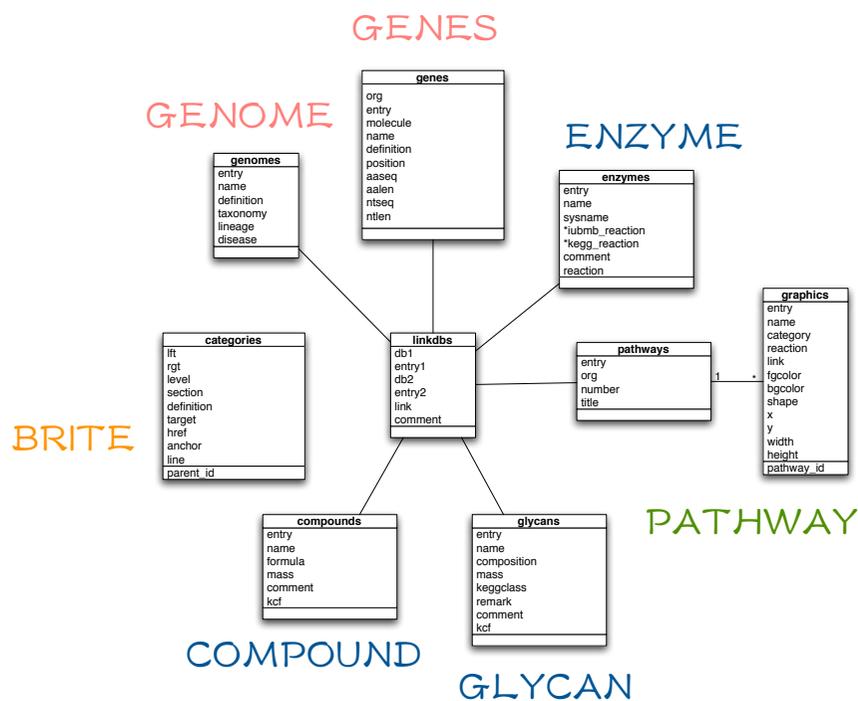
### GENES

```
entry.motif.each do |dbname, list|
 case dbname
 when "Pfam"
 db = "pf"
 when "PROSITE"
 db = "ps"
 end
 list.each do |name|
 save_link(org, gene, db, name, "motif")
 end
end

entry.orthologs.each do |line|
 if name = line[/KO:\s*(\S+)/, 1]
 save_link(org, gene, "ko", name)
 end
end
```

# ActiveRecord のモデルは Ruby のクラスでもある

- モデルは単なる O/R マッピングではない
  - R → O マッピングされたオブジェクトに機能（メソッド）を追加
    - 先の Category クラスに nested\_set や tree の機能を追加する例など



## LinkDB

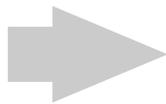
Rails 標準の habtm を使うのはあきらめたので、LinkDB を利用した代替メソッドを、各モデルにインプリする必要がある。

## 10. モデルにもガシガシとメソッドを追加する

---

- Linkdb (db1:entry1 → db2:entry2 の関係を保持) クラスの拡張

```
class Linkdb < ActiveRecord::Base
 def self.targets(dbentry, db2 = nil, link = nil)
 list = []
 db1, entry1 = dbentry.split(':')
 if link
 if db2
 list = self.find_all_by_db1_and_entry1_and_db2_and_link(db1, entry1, db2, link)
 else
 list = self.find_all_by_db1_and_entry1_and_link(db1, entry1, link)
 end
 else
 if db2
 list = self.find_all_by_db1_and_entry1_and_db2(db1, entry1, db2)
 else
 list = self.find_all_by_db1_and_entry1(db1, entry1)
 end
 end
 return list.map {|x| "#{x.db2}:#{x.entry2}"}
 end
end
```



```
def self.pathways(dbentry)
 targets(dbentry, "path")
end
```

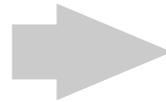
```
def self.enzymes(dbentry)
 targets(dbentry, "ec")
end
```

## 10. モデルにもガシガシとメソッドを追加する

---

- Linkdb を他のモデルでも使う

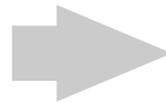
```
class Linkdb < ActiveRecord::Base
 def self.pathways(dbentry)
 targets(dbentry, "path")
 end
end
```



```
Linkdb.pathways("cpd:C00023")
```

```
class Compound < ActiveRecord::Base
 def self.dbentry(dbentry)
 db, entry = dbentry.split(':')
 self.find_by_entry(entry)
 end
```

```
 def dbentry
 "cpd:#{self.entry}"
 end
```



```
cpd = Compound.dbentry("cpd:C00023")
cpd.pathways
```

```
 def pathways
 Linkdb.pathways(dbentry)
 end
end
```

おまけ

# インデックス重要

- schema で hogehoge\_id となっているカラムには add\_index すべし！

```
class Graphic < ActiveRecord::Base class Pathway < ActiveRecord::Base
 belongs_to :pathway has_many :graphics
end end
```

```
% ./script/generate migration add_index_to_pathway_id_on_graphics
```

```
class AddIndexToPathwayIdOnGraphics < ActiveRecord::Migration
```

```
 def self.up
 add_index :graphics, :pathway_id
 end
```

```
 def self.down
 end
```

```
end
```

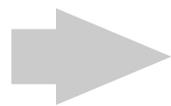
```
% RAILS_ENV=production rake db:migrate
```

```
== AddIndexToPathwayIdOnGraphics: migrating =====
-- add_index(:graphics, :pathway_id)
-> 215.8169s
== AddIndexToPathwayIdOnGraphics: migrated (215.8171s) =====
```

```
mysql> select count(*) from graphics;
```

```
+-----+
| count(*) |
+-----+
| 4179674 |
+-----+
```

```
1 row in set (8.96 sec)
```



```
mysql> select count(*) from graphics;
```

```
+-----+
| count(*) |
+-----+
| 4179674 |
+-----+
```

```
1 row in set (0.25 sec)
```

```
mysql> desc graphics;
```

Field	Type	Null	Key	Default	Extra
id	int(11)		PRI	NULL	auto_increment
entry	int(11)	YES	MUL	NULL	
name	text	YES		NULL	
category	varchar(255)	YES		NULL	
reaction	varchar(255)	YES		NULL	
link	varchar(255)	YES		NULL	
fgcolor	varchar(255)	YES		NULL	
bgcolor	varchar(255)	YES		NULL	
shape	varchar(255)	YES		NULL	
x	int(11)	YES		NULL	
y	int(11)	YES		NULL	
width	int(11)	YES		NULL	
height	int(11)	YES		NULL	
pathway_id	int(11)	YES	MUL	NULL	

# Rails による DB 構築の利点

---

- ActiveRecord で RDBMS に依存しない開発が可能
  - MVC モデルの Model 部分を ActiveRecord モデルにより実装
  - MySQL, PostgreSQL, SQLite3 などに対応
  - スキーマの変更が migration の仕組みにより容易
- SQL の代わりに ./script/console が便利
- RJS でわりと簡単にイマドキな AJAX ができる
- AWS でウェブサービス化も簡単にできる
- ていうか、普通の人には Rails なしにここまでキレイに作り込めない、という



KEGG on Rails  
公開してみますので  
プラグインなど  
開発して下さい