Tutorial Chado 2013

(Redirecionado do [Tutorial Chado](http://gmod.org/mediawiki/index.php?title=Chado_Tutorial&redirect=no) )

Este tutorial [Chado](http://gmod.org/wiki/Chado" \o "Chado) foi apresentado por [Scott Cain](http://gmod.org/wiki/User:Scott" \o "Usuário: Scott) como parte da [2013 GMOD Summer School](http://gmod.org/wiki/2013_GMOD_Summer_School" \o "Escola de verão GMOD 2013) .

Chado é o esquema de banco de dados do projeto GMOD. Esta sessão apresenta os conceitos de banco de dados e, em seguida, fornece uma visão geral do design e da arquitetura do Chado e, em seguida, entra em detalhes sobre como usar um banco de dados Chado.

**Conteúdo**

 [ [esconder](http://gmod.org/wiki/Chado_Tutorial) ]

* [1 teoria](http://gmod.org/wiki/Chado_Tutorial#Theory)
  + [1.1 Introdução](http://gmod.org/wiki/Chado_Tutorial#Introduction)
    - [1.1.1 Terminologia do Banco de Dados](http://gmod.org/wiki/Chado_Tutorial#Database_Terminology)
      * [1.1.1.1 O que é um banco de dados?](http://gmod.org/wiki/Chado_Tutorial#What.27s_a_database.3F)
      * [1.1.1.2 SQL](http://gmod.org/wiki/Chado_Tutorial#SQL)
  + [1.2 Por que Chado?](http://gmod.org/wiki/Chado_Tutorial#Why_Chado.3F)
  + [1.3 Arquitetura Chado: Módulos](http://gmod.org/wiki/Chado_Tutorial#Chado_Architecture:_Modules)
    - [1.3.1 Extensível](http://gmod.org/wiki/Chado_Tutorial#Extensible)
    - [1.3.2 Plus](http://gmod.org/wiki/Chado_Tutorial#Plus)
    - [1.3.3 Advertências do Módulo](http://gmod.org/wiki/Chado_Tutorial#Module_Caveats)
  + [1.4 Explorando o esquema](http://gmod.org/wiki/Chado_Tutorial#Exploring_the_schema)
    - [1.4.1 Módulo de Sequência](http://gmod.org/wiki/Chado_Tutorial#Sequence_Module)
      * [1.4.1.1 Recursos](http://gmod.org/wiki/Chado_Tutorial#Features)
    - [1.4.2 Módulo CV (vocabulários controlados)](http://gmod.org/wiki/Chado_Tutorial#CV_.28Controlled_Vocabularies.29_Module)
      * [1.4.2.1 Vocabulários controlados](http://gmod.org/wiki/Chado_Tutorial#Controlled_Vocabularies)
      * [1.4.2.2 Ontologias](http://gmod.org/wiki/Chado_Tutorial#Ontologies)
      * [1.4.2.3 CVs e ontologias no Chado](http://gmod.org/wiki/Chado_Tutorial#CVs_and_Ontologies_in_Chado)
        + [1.4.2.3.1 Integridade de Dados](http://gmod.org/wiki/Chado_Tutorial#Data_Integrity)
        + [1.4.2.3.2 Portabilidade e padronização de dados](http://gmod.org/wiki/Chado_Tutorial#Data_Portability_and_Standardization)
        + [1.4.2.3.3 Complexidade](http://gmod.org/wiki/Chado_Tutorial#Complexity)
    - [1.4.3 Abrindo nosso banco de dados de amostra](http://gmod.org/wiki/Chado_Tutorial#Opening_our_sample_database)
    - [1.4.4 Nosso primeiro exemplo de consulta](http://gmod.org/wiki/Chado_Tutorial#Our_first_example_query)
    - [1.4.5 Módulo Geral](http://gmod.org/wiki/Chado_Tutorial#General_Module)
      * [1.4.5.1 IDs](http://gmod.org/wiki/Chado_Tutorial#IDs)
        + [1.4.5.1.1 IDs públicos](http://gmod.org/wiki/Chado_Tutorial#Public_IDs)
        + [1.4.5.1.2 IDs privados](http://gmod.org/wiki/Chado_Tutorial#Private_IDs)
      * [1.4.5.2 IDs no Chado](http://gmod.org/wiki/Chado_Tutorial#IDs_in_Chado)
    - [1.4.6 Propriedades](http://gmod.org/wiki/Chado_Tutorial#Properties)
      * [1.4.6.1 Relacionamentos](http://gmod.org/wiki/Chado_Tutorial#Relationships)
      * [1.4.6.2 Locais](http://gmod.org/wiki/Chado_Tutorial#Locations)
        + [1.4.6.2.1 Coordenadas Interbase](http://gmod.org/wiki/Chado_Tutorial#Interbase_Coordinates)
        + [1.4.6.2.2 Cadeias de Localização](http://gmod.org/wiki/Chado_Tutorial#Location_Chains)
        + [1.4.6.2.3 Tabela de recursos](http://gmod.org/wiki/Chado_Tutorial#featureloc_Table)
      * [1.4.6.3 Exemplo: Gene](http://gmod.org/wiki/Chado_Tutorial#Example:_Gene)
      * [1.4.6.4 Exemplo: Análise Computacional](http://gmod.org/wiki/Chado_Tutorial#Example:_Computational_Analysis)
      * [1.4.6.5 Outras anotações de recursos](http://gmod.org/wiki/Chado_Tutorial#Other_Feature_Annotations)
      * [1.4.6.6 Extensão do Chado: tabelas de propriedades e novos módulos](http://gmod.org/wiki/Chado_Tutorial#Extending_Chado:_Properties_tables_and_new_modules)
* [2 treinos](http://gmod.org/wiki/Chado_Tutorial#Practice)
  + [2.1 Pré-requisitos](http://gmod.org/wiki/Chado_Tutorial#Prerequisites)
    - [2.1.1 PostgreSQL](http://gmod.org/wiki/Chado_Tutorial#PostgreSQL)
      * [2.1.1.1 Editar arquivos de configuração](http://gmod.org/wiki/Chado_Tutorial#Edit_config_files)
      * [2.1.1.2 Criar um usuário gmod](http://gmod.org/wiki/Chado_Tutorial#Create_a_gmod_user)
    - [2.1.2 BioPerl](http://gmod.org/wiki/Chado_Tutorial#BioPerl)
    - [2.1.3 Vamos lá!](http://gmod.org/wiki/Chado_Tutorial#Let.27s_Go.21)
    - [2.1.4 Variáveis ​​de Ambiente](http://gmod.org/wiki/Chado_Tutorial#Environment_Variables)
  + [2.2 Instalando o Chado](http://gmod.org/wiki/Chado_Tutorial#Installing_Chado)
    - [2.2.1 Salvando seu progresso até este ponto](http://gmod.org/wiki/Chado_Tutorial#Saving_your_progress_to_this_point)
    - [2.2.2 Uma Nota sobre Refazer](http://gmod.org/wiki/Chado_Tutorial#A_Note_about_Redos)
  + [2.3 Preparando dados GFF para carregamento](http://gmod.org/wiki/Chado_Tutorial#Preparing_GFF_data_for_loading)
    - [2.3.1 Trabalhando com arquivos GFF grandes](http://gmod.org/wiki/Chado_Tutorial#Working_with_Large_GFF_files)
    - [2.3.2 Carregando GFF3](http://gmod.org/wiki/Chado_Tutorial#Loading_GFF3)
    - [2.3.3 Capturando a saída para verificar se há problemas](http://gmod.org/wiki/Chado_Tutorial#Capturing_the_output_to_check_for_problems)
    - [2.3.4 Realmente carregando dados](http://gmod.org/wiki/Chado_Tutorial#Really_loading_data)
      * [2.3.4.1 Tente novamente](http://gmod.org/wiki/Chado_Tutorial#Try_again)
  + [2.4 Carregando outros dados](http://gmod.org/wiki/Chado_Tutorial#Loading_other_data)
* [3 Chado para expressão, genótipo, fenótipo e diversidade natural](http://gmod.org/wiki/Chado_Tutorial#Chado_for_Expression.2C_Genotype.2C_Phenotype.2C_and_Natural_Diversity)
  + [3.1 Expressão](http://gmod.org/wiki/Chado_Tutorial#Expression)
    - [3.1.1 O que define um padrão de expressão?](http://gmod.org/wiki/Chado_Tutorial#What_defines_an_expression_pattern.3F)
    - [3.1.2 Como Chado lida com esta variedade?](http://gmod.org/wiki/Chado_Tutorial#How_does_Chado_deal_with_this_variety.3F)
    - [3.1.3 Exemplo FlyBase](http://gmod.org/wiki/Chado_Tutorial#FlyBase_Example)
    - [3.1.4 Chado permite](http://gmod.org/wiki/Chado_Tutorial#Chado_Allows)
    - [3.1.5 Tabela: expressão](http://gmod.org/wiki/Chado_Tutorial#Table:_expression)
    - [3.1.6 Tabela: expression\_cvterm](http://gmod.org/wiki/Chado_Tutorial#Table:_expression_cvterm)
  + [3.2 Genótipo](http://gmod.org/wiki/Chado_Tutorial#Genotype)
    - [3.2.1 Tabela: genótipo](http://gmod.org/wiki/Chado_Tutorial#Table:_genotype)
    - [3.2.2 Tabela: feature\_genotype](http://gmod.org/wiki/Chado_Tutorial#Table:_feature_genotype)
  + [3.3 Meio Ambiente](http://gmod.org/wiki/Chado_Tutorial#Environment)
    - [3.3.1 Tabela: ambiente](http://gmod.org/wiki/Chado_Tutorial#Table:_environment)
    - [3.3.2 Tabela: environment\_cvterm](http://gmod.org/wiki/Chado_Tutorial#Table:_environment_cvterm)
  + [3.4 Fenótipo, Diversidade Natural e Suporte Atlas](http://gmod.org/wiki/Chado_Tutorial#Phenotype.2C_Natural_Diversity_and_Atlas_Support)

Teoria

**Introdução**

**Terminologia de banco de dados**

Ou seis anos de escola em 15 minutos ou menos.

* [Um breve guia para bancos de dados](http://gmod.org/wiki/A_Brief_Guide_to_Databases)
* [Mesas Chado](http://gmod.org/wiki/Chado_Tables)

**O que é um**[**banco de dados**](http://gmod.org/wiki/Databases_and_GMOD)**?**

* Chado é um [esquema](http://gmod.org/wiki/Glossary#Schema) , um projeto de banco de dados - um projeto para um banco de dados contendo dados genômicos
* Diferente de
  + [Sistema de gerenciamento de banco de dados (DBMS)](http://gmod.org/wiki/Glossary#Database_Management_System)
    - Sistema de software para armazenamento de bancos de dados
    - *por exemplo,* Oracle, [PostgreSQL](http://gmod.org/wiki/PostgreSQL" \o "PostgreSQL) , [MySQL](http://gmod.org/wiki/MySQL" \o "MySQL)
  + Banco de dados, um termo muito vago
    - Qualquer conjunto de dados organizados que podem ser lidos por um computador
    - Um site com conteúdo baseado em banco de dados, por exemplo, FlyBase
    - Esquema + DBMS + Dados

**SQL**

SQL é uma linguagem de consulta padronizada para definir e manipular bancos de dados. Chado usa. SQL é suportado por todos os principais DBMSs.

[As tabelas de mapeamento de campo FlyBase](http://gmod.org/wiki/FlyBase_Field_Mapping_Tables) mostram alguns exemplos de SQL que consultam o banco de dados FlyBase Chado. (Advertência: às vezes o FlyBase usa o Chado de maneiras que nenhuma outra organização faz.)

**O SQL estará em teste?**

Não, não vamos ensinar SQL em profundidade neste curso, mas vamos usá-lo em exemplos e mostrar como escrever consultas no Chado.

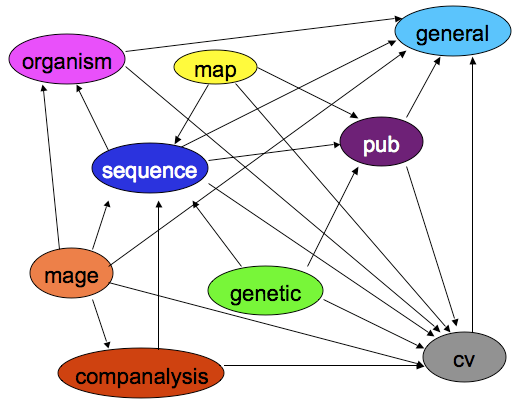
Você *pode* fazer o básico com o Chado sem saber SQL. Muitas tarefas comuns já possuem scripts escritos para elas. No entanto, à medida que você começa a usar o Chado, descobrirá que é necessário um conhecimento prático de SQL.

**Por que Chado?**

* **Integração**
  + Suporta muitos tipos de dados, integra-se com muitas ferramentas
* **Modular**
  + Use apenas o que você precisa, ignore o resto
* **Extensível**
  + Escreva seus próprios módulos e propriedades
* **Amplamente utilizado**
  + [FlyBase](http://gmod.org/wiki/Category:FlyBase) - Chado começou aqui, grande e diversificado conjunto de dados e organização
  + [Xenbase](http://xenbase.org/) - menor, mas com várias equipes de TI
  + [ParameciumDB](http://gmod.org/wiki/ParameciumDB) - ainda menor, loja GMOD completa, incluindo Chado
  + [IGS](http://www.igs.umaryland.edu/) - Anotação em grande escala / dados comparativos no Chado, mais de uma dúzia de desenvolvedores ativos
  + Além de AphidBase, BeeBase, BeetleBase, BovineBase, ...
* **Grande Comunidade de Apoio**
  + [Lista de mala direta](http://gmod.org/wiki/GMOD_Mailing_Lists)

**Arquitetura Chado: Módulos**

O esquema Chado é construído com um conjunto de *módulos* . Um módulo Chado é um conjunto de tabelas de banco de dados e relacionamentos que armazenam informações sobre uma área bem definida da biologia, como sequência ou atribuição.

[](http://gmod.org/wiki/File:ChadoModules.png)

(Também disponível como [animação em PowerPoint](http://gmod.org/mediawiki/images/3/36/ChadoModules.ppt) )

As setas são dependências entre os módulos. Dependências indicam um ou mais módulos de ligação de *chaves estrangeiras* .

* [Geral](http://gmod.org/wiki/Chado_General_Module) - Identificar coisas dentro do banco de dados para o mundo externo e identificar coisas de outros bancos de dados.
* [Vocabulário controlado (cv)](http://gmod.org/wiki/Chado_CV_Module) - Vocabulário controlado e ontologias
* [Publicação (pub)](http://gmod.org/wiki/Chado_Publication_Module) - Publicações e atribuição
* [Organismo](http://gmod.org/wiki/Chado_Organism_Module) - descreve espécies; bem simples. O módulo de filogenia armazena relacionamentos.
* [Sequência](http://gmod.org/wiki/Chado_Sequence_Module) - características genômicas e coisas que podem ser vinculadas ou descendentes de características genômicas.
* [Mapa](http://gmod.org/wiki/Chado_Map_Module) - Mapas sem sequência
* [Genética](http://gmod.org/wiki/Chado_Genetic_Module) - Dados genéticos e genótipos
* [Companalysis](http://gmod.org/wiki/Chado_Companalysis_Module) - Armazenamento de análise de sequência computacional. O conceito-chave é que os resultados de uma análise computacional podem ser interpretados ou descritos como um recurso de sequência.

**Extensível**

Esses módulos foram contribuídos para o Chado por usuários que os desenvolveram.

* [Mage](http://gmod.org/wiki/Chado_Mage_Module) - Dados de microarray
* [Estoque](http://gmod.org/wiki/Chado_Stock_Module) - Espécimes e coleções biológicas
* [Diversidade Natural](http://gmod.org/wiki/Chado_Natural_Diversity_Module_Working_Group) - geolocalização, fenótipo, genótipo
* Além de tabelas de propriedades em muitos módulos.

**Mais**

* [Auditoria](http://gmod.org/wiki/Chado_Audit_Module) - trilha de auditoria de banco de dados
* [Expressão](http://gmod.org/wiki/Chado_Expression_Module) - Resumos de RNA e expressão de proteínas
* [Biblioteca](http://gmod.org/wiki/Chado_Library_Module) - Descrições de bibliotecas moleculares
* [Fenótipo](http://gmod.org/wiki/Chado_Phenotype_Module) - dados fenotípicos
* [Filogenia](http://gmod.org/wiki/Chado_Phylogeny_Module) - Organismos e árvores filogenéticas

**Advertências de Módulo**

Todos os módulos são *abençoados* , mas alguns módulos são *mais abençoados do* que outros.

Os módulos Geral, CV, Publicação, Organismo, Seqüência e Análise de Companheirismo são amplamente usados ​​e projetados de forma limpa. Depois disso, os módulos se tornam menos usados ​​(Stock, Expression, Phenotype, Mage). Além disso, vários módulos não são separados de forma tão clara como gostaríamos que fossem. Os dados fenotípicos estão espalhados por vários módulos. Sobreposição de organismo e filogenia. [CMap](http://gmod.org/wiki/CMap" \o "CMap) tem tudo a ver com mapas, mas não usa o módulo Mapa.

De Jeff Bowes, em XenBase:

Quanto ao Chado, somos mais Chadoish do que exatamente Chado. Usamos os módulos principais com poucas alterações - feature, cv, general, analysis. Embora eu prefira adicionar colunas às tabelas quando for razoável e limitar o uso de tabelas de propriedades (muitas junções externas esquerdas). Usamos uma versão ligeiramente modificada do módulo de filogenia. Desenvolvemos módulos completamente diferentes para comunidade, literatura, anatomia e expressão gênica. Se houver uma solução de fenótipo de Chado compatível com PATO, preferimos ir com ela. Embora, possa causar problemas que tenhamos um módulo de anatomia separado em vez de usar o cvterm para armazenar anatomia.

Em outras palavras, o ideal é bom, mas a implementação e o uso são desiguais. Veja a [pesquisa da comunidade GMOD de 2008 para saber o](http://gmod.org/extras/2008GMODCommunitySurvey.html#Chado) que é usado.

**Explorando o esquema**

Em vez de simplesmente listar os módulos e o que está armazenado neles, vamos ter uma visão centrada em dados e imaginar o que queremos armazenar em nosso banco de dados, então aprender a maneira Chado de armazená-lo.

Durante este curso, você trabalhará com dados de anotação do genoma do [MAKER](http://gmod.org/wiki/MAKER) . Vamos simplificar isso e começar considerando que temos anotações nos cromossomos que queremos armazenar em nosso banco de dados. Estes são os tipos de coisas que queremos armazenar:

* cromossomos
* genes
* predições de genes
* tRNAs
* BLAST partidas

Você pode ter trabalhado com bancos de dados no passado, onde cada tipo de coisa que deseja armazenar tem sua própria tabela. Ou seja, teríamos uma tabela para genes, uma para cromossomos, tRNAs etc. O problema com esse tipo de design é que, conforme você encontra novos tipos de coisas, é necessário criar novas tabelas para armazená-los. Além disso, muitas dessas tabelas de 'coisas' serão muito parecidas.

Chado é um esquema *genérico* que, na verdade, significa que os dados são abstraídos sempre que possível para evitar a duplicação no design e nos próprios dados. Portanto, em vez de uma mesa para cada tipo de 'coisa', temos apenas uma mesa para armazenar 'coisas', independentemente de seus tipos. No mundo Chado, eles são conhecidos como 'recursos'. (Este padrão de design de banco de dados é chamado de modelo [Entidade-Atributo-Valor](https://en.wikipedia.org/wiki/Entity-attribute-value_model) .)

Isso nos leva ao Módulo de Sequência, que contém a tabela de recursos central.

**Módulo de Sequência**

[Módulo de Sequência](http://gmod.org/wiki/Chado_Sequence_Module)

O módulo de sequência é usado para gerenciar recursos genômicos.

**Características**

Chado define uma *característica* como uma região de um polímero biológico (normalmente um DNA, RNA ou uma molécula de polipeptídeo) ou um agregado de regiões neste polímero. Uma região pode ser um cromossomo inteiro ou uma junção entre duas bases. Os recursos são digitados de acordo com a [Ontologia de Seqüência (SO)](http://www.sequenceontology.org/) , podem ser localizados em relação a outros recursos e podem formar parte-todo e outras relações com outros recursos.

Os recursos são armazenados na [tabela de recursos](http://gmod.org/wiki/Chado_Sequence_Module#Table:_feature) .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela: recurso** | | | | | |
| **feature\_id** | **nome** | **Nome único** | **type\_id** | **is\_analysis** | **...** |

Dentro desta tabela de recursos, podemos armazenar todos os tipos de recursos e controlar seu tipo com o campo type\_id. É concebível armazenar o valor nomeado de cada tipo neste campo, como 'gene', 'tRNA', etc., mas isso estaria sujeito a coisas como erros de grafia, para não mencionar a discordância da definição de alguns desses termos.

Então resolva isso, todos os recursos estão vinculados a um tipo específico em um vocabulário ou ontologia controlada. Eles são armazenados no módulo cv.

**Módulo de CV (vocabulários controlados)**

[Tabelas do Módulo de Vocabulário Controlado](http://gmod.org/Chado_CV_Module)

O módulo CV implementa *vocabulários controlados* e seus primos mais complexos, *ontologias* .

**Vocabulários controlados**

Um *vocabulário controlado* (CV) é uma lista de termos dos quais um valor deve vir. Os currículos são amplamente usados ​​em todas as bases de dados, não apenas nas biológicas. Os menus suspensos são freqüentemente usados ​​para apresentar currículos aos usuários em interfaces de consulta ou anotação.

|  |  |
| --- | --- |
| [**Tipo de**](http://zfin.org/)**ensaio CV**[**da ZFIN**](http://zfin.org/) | [Tipo de ensaio CV da ZFIN](http://gmod.org/wiki/File:ZfinAssayTypePullDown.png) |

**Ontologias**

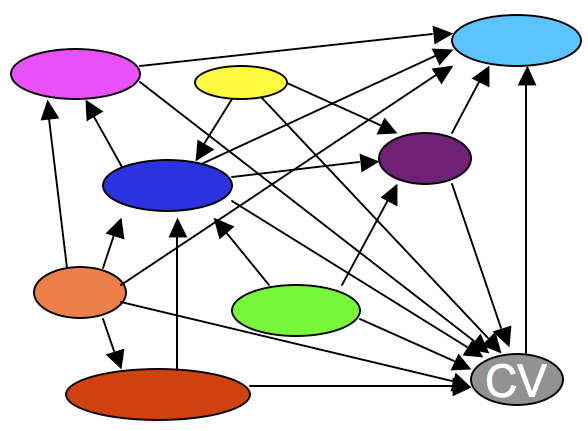
Os vocabulários controlados são listas simples de termos. *Ontologias* são termos mais regras e relacionamentos entre os termos. A Ontologia Genética (GO) e a Ontologia Sequencial (SO) são as duas ontologias mais conhecidas, mas existem muitas outras disponíveis na [OBO](http://www.obofoundry.org/) .

Ontologias podem ser incrivelmente complexas com muitos relacionamentos entre termos. Representá-los e raciocinar com eles não é trivial, mas o módulo CV ajuda em ambos.

|  |
| --- |
| **Visualizador de termo CV [FlyBase](http://flybase.org/) mostrando o termo GO "regeneração de tecido"** |
| [FlyBaseCVTermViewer.png](http://gmod.org/wiki/File:FlyBaseCVTermViewer.png) |

**CVs e ontologias no Chado**

[Módulo Chado CV](http://gmod.org/wiki/Chado_CV_Module)

[](http://gmod.org/wiki/File:ChadoModulesCV.png)

(Consulte a [tabela CVTerm que faz](http://gmod.org/wiki/index.php/Chado_CV_Module#Table:_cvterm) referência à lista de tabelas.)

Todos os outros módulos dependem do módulo CV. Currículos e ontologias são centrais para a filosofia de design do Chado. Porque?

**Integridade de dados**

Usar currículos (e impor seu uso como o Chado faz) garante que seus dados permaneçam consistentes. Por exemplo, no caso mais simples, impede que seu banco de dados use vários valores diferentes para significar a mesma coisa (por exemplo, "desconhecido", "não especificado", "ausente", "outro", "", ...), e evita erros de ortografia ("sagital" em vez de "sagital") e erros de digitação.

**Portabilidade e padronização de dados**

Se você está estudando processos de desenvolvimento e usa os termos de *processo biológico* da Gene Ontology, seus dados podem ser facilmente compartilhados *e integrados* com dados de outros pesquisadores. Se você criar seu conjunto de termos ou apenas inserir um texto livre (egads!), Será necessária muita intervenção humana para converter seus dados em uma nomenclatura padrão para que possam ser integrados com outros.

Usar uma ontologia estabelecida quando uma existe geralmente envolve alguns compromissos, mas aumenta muito a usabilidade de seus dados para outras pessoas (e para você mesmo).

**Complexidade**

Os vocabulários controlados não são particularmente complexos - são apenas listas de termos. Ontologias, no entanto, podem ser muito complexas, como mostrado pelo exemplo GO acima. Essa complexidade pode ser ignorada. Você poderia, por exemplo, converter GO para um vocabulário controlado - uma lista muito longa de termos. Você ainda teria integridade de dados e portabilidade, e não seria tão complexo.

Também não seria tão poderoso. Ontologias suportam raciocínio sobre os termos nelas e isso pode ser muito útil. Com GO, por exemplo, você pode perguntar

Mostre-me todos os genes envolvidos no *desenvolvimento da estrutura anatômica*

e obter de volta os genes diretamente marcados com o *desenvolvimento da estrutura anatômica* , além de quaisquer genes marcados com qualquer um dos subtermos desse termo, desde o *desenvolvimento* do *órgão* até a *regulação da regeneração do músculo esquelético* . Se você converter GO para apenas uma lista de termos, não poderá mais responder a essa pergunta.

O Módulo Chado CV suporta tais consultas complexas com ontologias, pré-calculando o [*fechamento transitivo*](http://gmod.org/wiki/Chado_CV_Module#Transitive_Closure) de todos os termos em uma ontologia. Há uma grande explicação sobre o fechamento transitivo na [página do Módulo Chado CV](http://gmod.org/wiki/Chado_CV_Module#Transitive_Closure) . Veja também a descrição dessas 3 tabelas:

* [cvterm](http://gmod.org/wiki/Chado_CV_Module#Table:_cvterm)
* [cvterm\_relationship](http://gmod.org/wiki/Chado_CV_Module#Table:_cvterm_relationship)
* [cvtermpath](http://gmod.org/wiki/Chado_CV_Module#Table:_cvtermpath)

Não entraremos em mais detalhes sobre isso aqui.

**Abrindo nosso banco de dados de amostra**

$ **psql drupal**

**Nosso primeiro exemplo de consulta**

Até este ponto, vimos como armazenar recursos usando a tabela de recursos, bem como definir rigidamente que tipos de coisas eles estão usando as tabelas do módulo cv. Aqui está um exemplo de SQL de como consultar algumas informações básicas sobre todos os recursos genéticos em nosso banco de dados:

Gene **SELECT** . feature\_id , gene . nome único , gene . nome

**FROM** gene recurso

**JOIN** cvterm c **ON** gene . type\_id = c . cvterm\_id

**ONDE** c . nome = 'gene' **AND** organism\_id = 13 ;

Isso deve retornar algo como:

feature\_id | uniquename | nome

------------ + ------------------------------------- ----------- + -------------------------------------- ----------

405 maker-scf1117875582023-snap-gene-0.0 | maker-scf1117875582023-snap-gene-0.0

409 maker-scf1117875582023-snap-gene-0.3 | maker-scf1117875582023-snap-gene-0.3

415 genemark-scf1117875582023-abinit-gene-0.42 | genemark-scf1117875582023-abinit-gene-0.42

1011 maker-scf1117875582023-snap-gene-1.0 | maker-scf1117875582023-snap-gene-1.0

1018 | maker-scf1117875582023-snap-gene-1.4 | maker-scf1117875582023-snap-gene-1.4

1022 maker-scf1117875582023-snap-gene-1.1 | maker-scf1117875582023-snap-gene-1.1

1027 maker-scf1117875582023-snap-gene-1.2 | maker-scf1117875582023-snap-gene-1.2

1032 maker-scf1117875582023-snap-gene-1.7 | maker-scf1117875582023-snap-gene-1.7

1038 maker-scf1117875582023-snap-gene-1.5 | maker-scf1117875582023-snap-gene-1.5

1698 | snap\_masked-scf1117875582023-abinit-gene-2.4 | snap\_masked-scf1117875582023-abinit-gene-2.4

...

Digite q para escapar da lista.

Listas de coisas são ótimas, mas vamos precisar fazer muito mais com nossos dados genômicos do que manter listas de recursos. Primeiro, precisamos ser capazes de identificá-los adequadamente. Isso pode parecer simples, mas criar uma coluna por tipo de ID no recurso seria uma má ideia, já que qualquer recurso específico poderia ter dezenas de identificadores diferentes de fontes de dados diferentes. O Módulo Geral ajuda a resolver isso.

**Módulo Geral**

O módulo Geral é sobre como identificar coisas dentro deste banco de dados para o mundo externo e identificar coisas do mundo externo (ou seja, outros bancos de dados) dentro deste banco de dados.

**IDs**

[Tabelas de módulos gerais](http://gmod.org/wiki/Chado_General_Module)

Os bancos de dados biológicos têm IDs *públicos* e *privados* e geralmente são coisas diferentes.

**IDs públicos**

Eles são mostrados em páginas da web e em publicações. Também são conhecidos como *números de acesso* .

|  |
| --- |
| **GO + 0043565 = GO: 0043565** |
| **InterPro + IPR001356 = InterPro: IPR001356** |
| **YourDB + qualquer coisa = YourDB: qualquer coisa** |

Os IDs públicos tendem a ser *chaves alternativas* dentro do banco de dados: eles identificam objetos no banco de dados de maneira exclusiva.

**IDs privados**

Eles são usados ​​dentro do banco de dados e não devem ser exibidos ou publicados. Tendem a ser inteiros longos. Existem muito mais IDs privados do que públicos.

IDs privados são usados ​​para *chaves primárias* e *chaves estrangeiras* .

A maioria dos DBMSs tem mecanismos integrados para gerar IDs privados.

**IDs no Chado**

O [módulo Geral](http://gmod.org/wiki/Chado_General_Module) define IDs *públicos* de

* itens definidos neste banco de dados, e
* itens definidos em outros bancos de dados, que são usados ​​ou referenciados neste banco de dados.

Na verdade, essas duas classes de IDs são definidas exatamente da mesma maneira, na [tabela dbxref](http://gmod.org/Chado_General_Module#Table:_dbxref) .

No Chado, cada tabela (em cada módulo) define seus próprios IDs privados.

**Propriedades**

Até agora, vimos apenas informações muito básicas para cada recurso armazenado. Como a tabela de recursos foi projetada para ser muito genérica e armazenar todos os tipos de recursos, os atributos específicos de apenas alguns tipos não podem ser armazenados lá. Não faria sentido, por exemplo, ter uma coluna chamada 'gene\_product\_name', uma vez que essa coluna estaria vazia para todos os recursos, como cromossomos, que não são produtos gênicos.

Esses atributos específicos de recursos são conhecidos como 'propriedades' de um recurso no Chado. Eles são armazenados em uma tabela chamada [featureprop](http://gmod.org/wiki/index.php/Chado_Sequence_Module" \l "Table:_featureprop) .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tabela: featureprop** | | | | |
| **featureprop\_id** | **feature\_id** | **type\_id** | **valor** | **classificação** |

Você deve ter notado que a tabela featureprop compartilha uma coluna 'type\_id' com a tabela de recursos. As propriedades dos recursos são digitadas de acordo com um vocabulário controlado, assim como os próprios recursos. Isso ajuda a garantir que qualquer pessoa que use os mesmos vocabulários codifique suas asserções de propriedade da mesma maneira.

Usando propriedades de recursos, podemos agora descrever nossos recursos tão detalhadamente quanto necessário, mas nossos recursos ainda são independentes uns dos outros. Como muitos dos recursos que estamos armazenando têm algum tipo de relação uns com os outros (como genes e seus produtos finais polipeptídicos), nosso esquema precisa acomodar isso.

**Relacionamentos**

Os relacionamentos entre os recursos são armazenados na [tabela feature\_relationship](http://gmod.org/wiki/index.php/Chado_Sequence_Module#Table:_feature_relationship) .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela: feature\_relationship** | | | | | |
| **feature\_relationship\_id** | **subject\_id** | **object\_id** | **type\_id** | **classificação** | **...** |

As características podem ser organizadas em gráficos, por exemplo, "exon *part\_of* transcript *part\_of* gene"; Se o tipo for pensado como um verbo, então cada arco ou aresta faz uma declaração:

* Assunto *verbo* Objeto, ou
* Pai *verbo* filho , ou
* Contém *verbo* Container, ou
* Subfeature *verbo* Recurso

Novamente, observe o uso de vocabulários controlados (type\_id) para definir a relação entre os recursos.

Pode-se pensar em uma relação entre um gene e um cromossomo como 'localizado\_on' e armazenar isso como uma entrada feature\_relationship entre os dois. Se você fez isso, dê um tapinha nas costas por seu pensamento chado, mas há uma maneira melhor de lidar com recursos localizáveis, uma vez que uma entrada de relacionamento sozinha não diria ONDE o recurso foi localizado, apenas que estava.

**Localizações**

A localização descreve de onde um recurso é / vem em relação a outro recurso. Algumas características, como cromossomos, não são localizadas em coordenadas de sequência, embora contigs / assemblies que compõem os cromossomos possam estar.

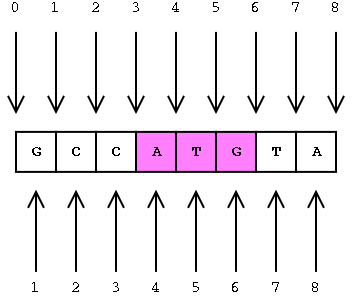
Os locais são armazenados na [tabela de recursos](http://gmod.org/wiki/index.php/Chado_Sequence_Module#Table:_featureloc) e um recurso pode ter zero ou mais registros de recursos. Os recursos terão

* um registro featureloc, para recursos localizados para os quais a localização é conhecida, ou
* registros de zero featureloc, para recursos não localizados, como cromossomos, ou para recursos para os quais a localização ainda não é conhecida, como um gene descoberto usando técnicas clássicas de genética.
* Os recursos com vários recursos são explicados a seguir.

(Para obter uma boa explicação de como os recursos estão localizados no Chado, consulte [Locais dos recursos](http://gmod.org/wiki/index.php/Chado_Sequence_Module#Feature_Locations) . Esta explicação foi extraída disso.)

**Coordenadas Interbase**

Isso é abordado com mais detalhes no [site](http://gmod.org/wiki/Introduction_to_Chado#Interbase_Coordinates) do [GMOD](http://gmod.org/wiki/Introduction_to_Chado#Interbase_Coordinates) .

[](http://gmod.org/wiki/File:Interbase.png)

Um registro de featureloc especifica um intervalo nas *coordenadas da sequência interbase* , delimitado pelas colunas fmin e fmax, cada uma representando a posição linear inferior e superior da *fronteira* entre as bases ou pares de bases (com a direcionalidade indicada pela coluna da fita).

As coordenadas do Interbase foram escolhidas em vez do sistema de coordenadas orientado para a base porque a matemática é mais fácil e suporta recursos de comprimento zero, como locais de emenda e pontos de inserção.

**Cadeias de localização**

Chado oferece suporte a *cadeias de localização* . Por exemplo, localizar um exon em relação a um contig que está localizado em relação a um cromossomo. A maioria das instâncias do Chado não exigirá essa flexibilidade; características são normalmente localizadas em relação aos cromossomos ou braços de cromossomos.

A capacidade de armazenar tais redes de localização ou gráficos de localização pode ser útil para genomas inacabados ou partes de genomas, como heterocromatina, em que é desejável localizar recursos em relação a contigs estáveis ​​ou andaimes, que são eles próprios localizados em uma montagem instável para cromossomos ou braços cromossômicos.

As cadeias de localização não abrangem necessariamente apenas conjuntos - os domínios da proteína podem ser localizados em relação às características do polipeptídeo, eles próprios localizados em um transcrito (ou no genoma, como é mais comum). As cadeias também podem abranger alinhamentos de sequência.

**Tabela featureloc**

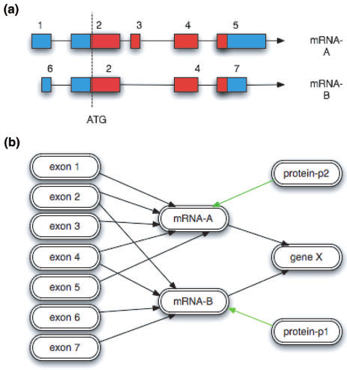
As informações de localização do recurso são armazenadas na [tabela de recursos](http://gmod.org/wiki/index.php/Chado_Sequence_Module#Table:_featureloc) .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela: featureloc** | | | | | |
| **featureloc\_id** | **feature\_id** | **srcfeature\_id** | **fmin** | **classificação** | **...** |

**Exemplo: Gene**

**Nota:** Este exemplo e algumas das figuras foram extraídos do [*estudo de caso A Chado: um esquema modular baseado em ontologia para representar informações biológicas associadas ao genoma*](http://gmod.org/wiki/index.php/Image:A_Chado_Case_Study.pdf) , por Christopher J. Mungall, David B. Emmert e FlyBase Consortium (2007)

Como um gene de "dogma central" é representado no Chado?

[](http://gmod.org/wiki/File:CentralDogma.png)

Como representamos esses exons, mRNAs, proteínas e as relações entre eles?

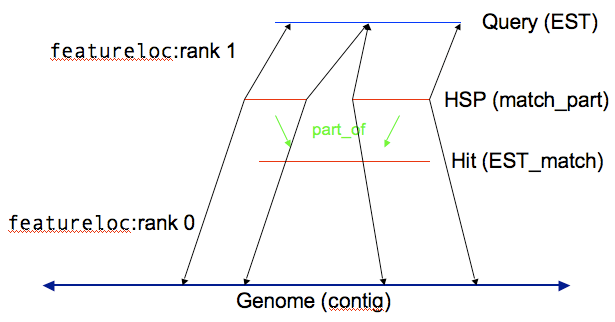
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Tabela: recurso** | | | | | | | **feature \_id** | **nome** | **Nome único** | **type\_id** | **is\_ analysis** |  | | 10456 | exon 1 | genex\_exon1 | (exon) | falso |  | | 10457 | exon 2 | genex\_exon2 | (exon) | falso |  | | 10458 | exon 3 | genex\_exon3 | (exon) | falso |  | | 10459 | exon 4 | genex\_exon4 | (exon) | falso |  | | 10460 | exon 5 | genex\_exon5 | (exon) | falso |  | | 10461 | exon 6 | genex\_exon6 | (exon) | falso |  | | 10462 | exon 7 | genex\_exon7 | (exon) | falso |  | | 10463 | mRNA-A | genex\_mRNA-A | (mRNA) | falso |  | | 10464 | mRNA-B | genex\_mRNA-B | (mRNA) | falso |  | | 10465 | proteína-p2 | genex\_protein-p2 | (polipeptídeo) | falso |  | | 10466 | proteína-p1 | genex\_protein-p1 | (polipeptídeo) | falso |  | | 10467 | gene X | genex | (polipeptídeo) | falso |  | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Tabela: feature\_relationship** | | | | | | | **feature\_ relacionamento \_id** | **subject\_id** | **object\_id** | **type\_id** | **classificação** |  | | 589 | ( exon 1 ) | ( mRNA-A ) | (parte de) | 0 |  | | 590 | ( exon 2 ) | ( mRNA-A ) | (parte de) | 0 |  | | 591 | ( exon 3 ) | ( mRNA-A ) | (parte de) | 0 |  | | 592 | ( exon 4 ) | ( mRNA-A ) | (parte de) | 0 |  | | 593 | ( exon 5 ) | ( mRNA-A ) | (parte de) | 0 |  | | 594 | ( exon 2 ) | ( mRNA-B ) | (parte de) | 0 |  | | 595 | ( exon 4 ) | ( mRNA-B ) | (parte de) | 0 |  | | 596 | ( exon 6 ) | ( mRNA-B ) | (parte de) | 0 |  | | 597 | ( exon 7 ) | ( mRNA-B ) | (parte de) | 0 |  | | 598 | ( proteína-p2 ) | ( mRNA-A ) | (deriva de) | 0 |  | | 599 | ( proteína-p1 ) | ( mRNA-B ) | (deriva de) | 0 |  | | 600 | ( mRNA-A ) | ( gene X ) | (parte de) | 0 |  | | 601 | ( mRNA-B ) | ( gene X ) | (parte de) | 0 |  | |

**Exemplo: Análise Computacional**

**Nota:** Este é um exemplo baseado em um exemplo de Scott Cain de um [workshop](http://gmod.org/wiki/index.php/Image:ChadoWorkshopArthopod2008.pdf) anterior do [Chado](http://gmod.org/wiki/index.php/Image:ChadoWorkshopArthopod2008.pdf) .

Você pode armazenar os resultados da análise computacional, como execuções do BLAST ou BLAT no módulo de sequência. Aqui está um exemplo de como você poderia armazenar um resultado de par de pontuação mais alto (HSP) do BLAST.

Com os resultados do HSP, existem duas sequências de referência, o que significa duas entradas na tabela de featureloc. (As tabelas de recursos de análise e análise, no módulo Análise da empresa, são usadas para armazenar informações sobre como a análise foi feita e quais pontuações resultaram.)

[](http://gmod.org/wiki/File:CompAnalysisFeature.png)

Cada linha horizontal torna-se um registro na tabela de recursos e cada linha vertical torna-se um registro na tabela feature\_relationship.

**Outras anotações de recurso**

Link para qualquer recurso via [feature\_id](http://gmod.org/wiki/index.php/Chado_Sequence_Module" \l "Table:_feature) :

* Termos GO em [feature\_cvterm](http://gmod.org/wiki/index.php/Chado_Sequence_Module" \l "Table:_feature_cvterm)
* Links de banco de dados em [feature\_dbxref](http://gmod.org/wiki/index.php/Chado_Sequence_Module" \l "Table:_feature_dbxref)
* Recursos diversos em [featureprop](http://gmod.org/wiki/index.php/Chado_Sequence_Module" \l "Table:_featureprop)
* Atribuição em [feature\_pub](http://gmod.org/wiki/index.php/Chado_Sequence_Module" \l "Table:_feature_pub)

**Estendendo Chado: tabelas de propriedades e novos módulos**

Chado foi construído para ser facilmente extensível. Novas funcionalidades podem ser adicionadas adicionando novos módulos. Os módulos Mage e Stock foram adicionados depois que Chado ficou fora por um tempo. Ambos trataram de necessidades que não foram atendidas ou foram tratadas de forma inadequada na versão original.

Chado pode ser adaptado às necessidades de uma organização individual usando *tabelas de propriedades* . As tabelas de propriedades são um meio de adicionar virtualmente novas colunas sem ter que modificar o esquema. As tabelas de propriedades estão incluídas em muitos módulos.

As tabelas de propriedades são incrivelmente flexíveis e tornam o Chado extremamente extensível, mas o fazem com um custo.

* O SQL para obter atributos de tabelas de propriedades é muito mais complexo e
* É mais trabalhoso impor restrições aos dados em tabelas de propriedades do que em colunas regulares.

Prática

**Pré-requisitos**

Usando dados do [MAKER](http://gmod.org/wiki/MAKER" \o "CRIADOR) .

**PostgreSQL**

PostgreSQL 9.1 já instalado via apt-get.

**Editar arquivos de configuração**

**sudo su**

**cd /etc/postgresql/9.1/main/**

**menos pg\_hba.conf**

Na parte inferior do arquivo, já alteramos o mesmo usuário de identidade para confiável . Isso significa que qualquer pessoa que esteja na máquina local pode se conectar a qualquer banco de dados. Se você quiser permitir que as pessoas se conectem de outras máquinas, seu endereço IP ou uma combinação de endereço IP e máscaras de rede pode ser usado para permitir o acesso remoto.

Se tivéssemos editado o arquivo pg\_hba.conf, teríamos que (re) iniciar o servidor de banco de dados:

**/etc/init.d/postgresql restart**

**saída # para sair do su**

**Crie um usuário gmod**

Normalmente, teríamos que mudar para o usuário postgres para criar um novo usuário de banco de dados, mas o usuário ubuntu já é um superusuário. Crie um novo usuário chamado "gmod":

createuser gmod

A nova função deve ser um superusuário? (s / n) s

**[BioPerl](http://gmod.org/wiki/BioPerl" \o "BioPerl)**

Versão 1.6.901 já instalada.

**Vamos lá!**

**variáveis ​​ambientais**

em ~ / .bashrc add:

**GMOD\_ROOT = '/ usr / local / gmod'**

**exportar GMOD\_ROOT**

e fonte do perfil:

**fonte ~ / .bashrc**

**Instalando Chado**

**cd ~ / sources / chado / chado /**

**perl Makefile.PL**

Usar valores em '/home/ubuntu/chado/chado/build.conf'? [S] **n**

Isso permite que você especifique o Pythium quando for solicitado um organismo padrão.

> "chado" para o nome do banco de dados

> "ubuntu" para nome de usuário do banco de dados

> "Pythium" para o organismo padrão

> "público" para o esquema em que o Chado residirá

Basta pressionar Enter quando for solicitada uma senha para o usuário gmod.

Este Makefile.PL faz várias coisas para preparar a instalação. Entre outras coisas:

* Configura uma variedade de parâmetros de usuário do banco de dados
* Prepara os arquivos SQL para instalação
* Ele pode reconstruir [Class :: DBI](http://search.cpan.org/perldoc?Class::DBI) api para [Chado](http://gmod.org/wiki/Chado" \o "Chado) , mas não é agora (eles são pré-construídos para o esquema padrão)
* Todas as coisas 'normais' que aconteceriam ao executar Makefile.PL , como copiar arquivos lib e bin para ficar pronto para instalação

**faço**

**sudo make install**

**make load\_schema**

**make prepdb**

Nesse ponto, o instalador verifica o banco de dados para ver se o organismo que colocamos como nosso organismo padrão está presente no banco de dados. Como não é, ele nos pedirá para adicioná-lo. Ele faz isso executando automaticamente um script que vem com o Chado, gmod\_add\_organism.pl. Você pode executar este script a qualquer momento que quiser [adicionar organismos ao seu banco de dados](http://gmod.org/wiki/Load_GFF_Into_Chado#Add_an_Entry_for_Your_Organism) :

Adicionando Pythium ao banco de dados ...

Gênero e espécie são necessários; por favor forneça-os abaixo

Gênero do organismo? Pythium

Espécies de organismo? ultimo

Abreviatura do organismo? [P.ultimum]

Comentário (pode estar vazio)?

Continuando ...

fazer ontologias

Ontologias disponíveis:

[1] Ontologia de relacionamento

[2] Ontologia de sequência

[3] Gene Ontology

[4] Propriedades do recurso Chado

[5] Ontologia de planta

Quais ontologias você gostaria de carregar (delimitado por vírgulas)? [0] **1,2,4**

Você pode escolher qualquer ontologia que desejar, mas o GO levará mais de uma hora para ser instalado e não importa muito o que você escolher, porque iremos explodir esse banco de dados em breve de qualquer maneira.

**Salvando seu progresso até este ponto**

Geralmente, é uma boa ideia salvar seu progresso quando terminar de carregar as ontologias, antes de tentar carregar quaisquer outros dados. Dessa forma, se algo der errado, é muito fácil restaurar até esse ponto. Para fazer um db dump, faça o seguinte:

**pg\_dump chado | bzip2 -c --best> db\_w\_ontologies.bz2**

O -c diz ao bzip2 para aceitar a entrada padrão e despejá-la na saída padrão . Para restaurar de um dump, elimine e recrie o banco de dados e, em seguida, descompacte o dump nele, como este:

dropdb chado

createdb chado

bzip2 -dc db\_w\_ontologies.bz2 | psql chado

**Uma nota sobre Redos**

Se em algum momento você sentir que deseja reconstruir seu banco de dados do zero, você precisa se livrar do diretório temporário onde os arquivos de ontologia estão armazenados. Você pode fazer isso com rm -rf ./tmp **(tenha muito cuidado com isso ./ na frente de tmp )** ou com um alvo make que foi projetado para isso: make rm\_locks , que apenas se livra dos arquivos de bloqueio, mas deixa o arquivos de ontologia no local.

**Preparando dados**[**GFF**](http://gmod.org/wiki/GFF)**para carregamento**

Problemas:

* separando 'anotações' de 'análise computacional'
* arquivos muito grandes
* fasta → gff
* scripts utilitários para várias atividades de preparação

**Trabalhar com arquivos**[**GFF**](http://gmod.org/wiki/GFF)**grandes**

Arquivos grandes (mais de 3-500.000 linhas) podem causar dores de cabeça para o carregador em massa [GFF3](http://gmod.org/wiki/GFF3) , mas os arquivos GFF3 de anotação de genoma podem frequentemente ter milhões de linhas. O que fazer? O script gmod\_gff3\_preprocessor.pl ajudará com isso, dividindo os arquivos em pedaços de tamanho razoável e classificando a saída para que faça sentido.

Observe que se seus arquivos já estão classificados (e neste caso, isso significa que todos os recursos pai vêm antes de seus recursos filhos e as linhas que compartilham IDs (como os CDSes às vezes fazem) estão juntos), então tudo que você precisa fazer é dividir seus arquivos . Freqüentemente, os arquivos já estão classificados e evitar a classificação é bom por dois motivos:

1. Leva muito tempo para fazer a classificação em comparação com a divisão (analisa cada linha, carrega em tabelas temporárias e extrai linhas por meio de consulta para reconstruir o arquivo GFF)
2. O processo de classificação torna mais difícil ler o GFF resultante, uma vez que o recurso pai não estará mais perto dos recursos filho no arquivo GFF. Chado não se preocupa com isso, mas você pode.

**Carregando**[**GFF3**](http://gmod.org/wiki/GFF3)

Trabalhar com o grande número de arquivos que sai do pré-processador pode ser um pouco chato, então "desenvolvi" alguns truques. Basicamente, eu crio um script bash que executará todas as cargas de uma vez. A maneira mais fácil (para mim) de fazer isso é:

**ls \* .gff3> load.sh**

**vi load.sh**

e usar o vim regex goodness para escrever os comandos do carregador em cada linha do arquivo:

**:% s / ^ / gmod\_bulk\_load\_gff3.pl --analysis -g /**

que coloca o comando no início de cada linha. O --analysis diz ao carregador que ele está trabalhando com os resultados da análise, portanto, as pontuações precisam ser armazenadas no módulo de análise da empresa. Embora os arquivos de genes não sejam realmente resultados de análises, eu os falsifiquei para se parecerem com os resultados de FgenesH para que a Apollo tivesse algo com que trabalhar. Em seguida, edito manualmente o arquivo para adicionar e remover as poucas modificações de que preciso:

* Adicione --noexon ao arquivo genes (uma vez que contém recursos de exon e CDS, não quero que o carregador crie recursos de exon a partir de recursos de CDS).

Em seguida, adiciono #! / Bin / bash na parte superior do arquivo (o que não é realmente necessário) e executo

**bash load.sh**

e carrega todos os arquivos sequencialmente. Eu fiz isso com 60-70 arquivos por vez, deixando o carregador rodar por alguns dias (!)

**Capturar a saída para verificar se há problemas**

Se você vai deixar um carregamento rodar por muito tempo, provavelmente deve capturar a saída para verificar se há problemas. Existem duas maneiras de fazer isso:

* execute dentro do comando de tela :

**tela -S loader**

que cria uma nova 'tela' separada do seu login. Em seguida, execute o comando de carregamento na tela. Para sair da tela, mas permitir que o comando de carregamento continue em execução nela, digite ctrl-a seguido por a d (para desconectar) e você obterá o terminal original de volta. Para se reconectar à tela do carregador, digite

**tela -R loader**

* capturar stdout e stderr para um arquivo

Ao executar o comando load, você pode usar o redirecionamento para coletar o stdout e o stderr para um arquivo:

**bash load.sh> & load.output**

**Realmente carregando dados**

OK, vamos colocar os dados no chado:

**gmod\_bulk\_load\_gff3.pl -g /home/ubuntu/sources/pyu\_data/scf1117875582023.gff**

Isso não deve funcionar (ainda!)

**Tente novamente**

Doh! O carregador está tentando nos dizer que isso se parece com dados de análise (ou seja, dados produzidos por computador em vez de humanos).

Precisamos dizer ao carregador que na verdade são os resultados da análise:

**gmod\_bulk\_load\_gff3.pl --analysis -g /home/ubuntu/sources/pyu\_data/scf1117875582023.gff**

Mate, *mate* , **mate!**(ctrl-c) o carregamento assim que você ver esta mensagem:

Existem recursos de CDS e exon neste arquivo, mas

você não definiu a opção --noexon, que provavelmente deseja.

Por favor, veja `perldoc gmod\_bulk\_load\_gff3.pl para mais informações.

Argh! Agora o carregador está apontando que esse arquivo [GFF](http://gmod.org/wiki/GFF) tem recursos de exons e CDS e o Chado prefere algo um pouco diferente. Embora o carregador carregue esses dados conforme escritos, eles não serão "padrão". Em vez disso, adicionaremos a opção --noexon (que diz ao carregador para não criar recursos de exon a partir dos recursos do CDS, uma vez que já os temos). Mais uma (pelo menos) vez:

**gmod\_bulk\_load\_gff3.pl --noexon --analysis -g /home/ubuntu/sources/pyu\_data/scf1117875582023.gff**

Sucesso! (Provavelmente.) Se isso falhar, quando for executado novamente, podemos precisar da opção --recreate\_cache , que recria uma tabela "temporária" que o carregador usa para controlar os IDs.

**Carregando outros dados**

[Chado XML](http://gmod.org/wiki/Chado_XML" \o "Chado XML) e [XORT](http://gmod.org/wiki/XORT" \o "XORT) , mas realmente, carregador de arquivos em massa [Tripal](http://gmod.org/wiki/Tripal" \o "Tripal) .

Chado para expressão, genótipo, fenótipo e diversidade natural

Esta seção é sobre alguns dos módulos Chado menos usados.

**Expressão**

Do ponto de vista do banco de dados do organismo, a expressão significa transformar isso:

|  |  |
| --- | --- |
| [ZfinGSC.jpg](http://gmod.org/wiki/File:ZfinGSC.jpg) | Fig. S5 Efeito sinérgico de zdia2 sMO e profilin I tMO. Os embriões foram injetados com 8 ng de zdia2 sMO (A e B) ou co-injetados com 4 ng de zdia2 sMO e 4 ng de profilin I tMO (C e D). Os embriões foram incubados até o estágio de botão de cauda (10 hpf), fixados e corados com ribossonda ntl e gsc. As fotografias foram tiradas para vista lateral (A e C) e vista dorsal (B e D) após WISH.  De [Lai](http://dx.doi.org/10.1371/journal.pone.0003439)*[et al.](http://dx.doi.org/10.1371/journal.pone.0003439)*[, 2008 - A formina 2 relacionada ao diáfano e a profilina I são necessárias para os movimentos das células de gastrulação. PLoS One 3 (10): e3439](http://dx.doi.org/10.1371/journal.pone.0003439) |

Nisso:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Gene** | **Peixe** | **Etapa** | **Anatomia** | **Ensaio** |
| gsc | tipo selvagem (não especificado), MO: diaph2, pfn1 | Bud | placa precordal | ISH |
| gsc | tipo selvagem (não especificado), MO: diaph2 | Bud | placa precordal | ISH |
| ntla | tipo selvagem (não especificado), MO: diaph2, pfn1 | Bud | notocorda | ISH |
| ntla | tipo selvagem (não especificado), MO: diaph2, pfn1 | Bud | botão da cauda | ISH |
| ntla | tipo selvagem (não especificado), MO: diaph2 | Bud | notocorda | ISH |
| ntla | tipo selvagem (não especificado), MO: diaph2 | Bud | botão da cauda | ISH |
| Do [ZFIN: Figura: *Lai et al.*, 2008, Fig. S5](http://zfin.org/cgi-bin/webdriver?MIval=aa-fxfigureview.apg&OID=ZDB-FIG-081103-25) | | | | |

**O que define um padrão de expressão?**

A expressão pode incluir muitas coisas diferentes, e o que ela inclui depende da comunidade:

**Gene, ou transcrição, ou proteína, ou ...**

O que estamos medindo?

**Etapa**

Armazenar estágios únicos ou janelas de estágio?

O que significam as janelas do palco ou estágios adjacentes: em toda a janela ou não temos certeza do palco (alto rendimento)

**Cepa / Genótipo**

Em que geneótipo vimos isso.

**Anatomia**

"Expresso no cérebro" significa expresso em todos, na maioria ou em algum lugar do cérebro?

**Ensaio**

ISN, anticorpo, sonda, ...

**Fonte**

Publicação, tela de alto rendimento, laboratório, projeto, ...

Genótipo

Ambiente

**Padrão**

Você quer acompanhar homogêneo, graduado, irregular

**Não expresso**

Você controla as ausências? Se sim, o que isso significa? (não *detectado* )

**Imagem**

Uma imagem é obrigatória ou opcional?

**Força**

O que significam os pontos fortes em diferentes experimentos?

**Como Chado lida com essa variedade?**

*Pós-composição* , que é uma forma muito chadoísta de fazer as coisas.

* Abrace uma definição mínima do que é um padrão de expressão. No Chado, tudo o que é necessário é um *nome* , por exemplo, *SLC21A no rim do participante do curso GMOD após 4 dias no NESCent* , ou apenas "GMOD0002347".
* Você também pode fornecer uma descrição. Se o seu nome é "GMOD0002347", pode ser uma boa ideia
* Os detalhes são então pendurados nisso.

**Exemplo FlyBase**

Um exemplo específico de FlyBase:

Aqui está um exemplo de um caso simples do tipo de dados que o [FlyBase seleciona](http://flybase.org/) .

O transcrito *dpp* é expresso no estágio embrionário 13-15 no segmento cefálico, conforme relatado em um [artigo de Blackman *et al.*em 1991](http://flybase.bio.indiana.edu/reports/FBrf0053806.html) .

Isso seria implementado no módulo de expressão vinculando o recurso de transcrição *dpp* à expressão por meio de [feature\_expression](http://gmod.org/wiki/Chado_Tables" \l "Table:_feature_expression" \o "Mesas Chado) . Em seguida, vincularíamos os seguintes cvterms à expressão usando [expression\_cvterm](http://gmod.org/wiki/Chado_Tutorial" \l "Table:_expression_cvterm) :

* *estágio embrionário 13* onde o cvterm\_type seria o estágio e a classificação = 0
* *estágio embrionário 14,* onde o cvterm\_type seria o estágio e a classificação = 1
* *estágio embrionário 15* onde o cvterm\_type seria o estágio e a classificação = 1
* *segmento cefálico* onde o cvterm\_type seria anatomia e a classificação = 0
* *hibridização in situ* onde o cvterm\_type seria o ensaio e a classificação = 0

**Tradução**

No FlyBase, este seria um único registro de expressão, com 5 termos de Ontologia / CV anexados a ele.

* 1 dizendo para que anatomia se refere a expressão - *segmento cefálico*
* 1 dizendo que o tipo de ensaio era *hibridização in situ*
* 1 para cada um dos 3 estágios - *estágios embrionários 13, 14, 15*

E

* 1 registro dizendo que este padrão de expressão é para *dpp* .
* 1 registro dizendo que esse padrão de expressão é de [Blackman](http://flybase.bio.indiana.edu/reports/FBrf0053806.html)*[et al.](http://flybase.bio.indiana.edu/reports/FBrf0053806.html)*[em 1991](http://flybase.bio.indiana.edu/reports/FBrf0053806.html) .

**Chado permite**

O design da tabela do módulo Expressão permite que cada padrão de expressão tenha

* 1 nome
* 0, 1 ou mais
  + Publicações / Fontes
  + Características
  + Imagens
  + Termos de anatomia
  + Estágios
  + Tipos de ensaio
  + Qualquer outro termo CV / Ontologia (por exemplo, detectado, não detectado)

**Dois pontos-chave**

1. **Chado pode apoiar qualquer coisa que sua comunidade decida sua definição de um padrão de expressão.**
2. **No entanto, Chado não irá impor essa definição para você.**

Se você precisar de um padrão de expressão para ter

* 1 nome
* 1 publicação / fonte
* 1 característica
* 1 termo de anatomia
* 1 estágio
* 1 tipo de ensaio
* 1 sinalizador detectado / não detectado
* 0, 1 ou mais imagens

então você terá que escrever um script para verificar isso.

**Tabela: expressão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Expressão Estrutura** | | | |
| **Tecla F** | **Nome** | **Modelo** | **Descrição** |
|  | expression\_id | serial | *CHAVE PRIMÁRIA* |
|  | Nome único | texto | *ÚNICO NÃO NULO* |
|  | md5checksum | personagem (32) |  |
|  | Descrição | texto |  |

Tabelas que fazem referência a esta por meio de restrições de chave estrangeira:

* [expression\_cvterm](http://gmod.org/wiki/Chado_Tutorial#Table:_expression_cvterm) , [expression\_image](http://gmod.org/wiki/Chado_Tables#Table:_expression_image) , [expression\_pub](http://gmod.org/wiki/Chado_Tables#Table:_expression_pub) , [expressionprop](http://gmod.org/wiki/Chado_Tables#Table:_expressionprop) , [feature\_expression](http://gmod.org/wiki/Chado_Tables#Table:_feature_expression) , [wwwuser\_expression](http://gmod.org/wiki/Chado_Tables#Table:_wwwuser_expression)

**Tabela: expression\_cvterm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Estrutura expression\_cvterm** | | | |
| **Tecla F** | **Nome** | **Modelo** | **Descrição** |
|  | expression\_cvterm\_id | serial | *CHAVE PRIMÁRIA* |
| [expressão](http://gmod.org/wiki/Chado_Tutorial#Table:_expression) | expression\_id | inteiro | *UNIQUE # 1 NOT NULL* |
| [cvterm](http://gmod.org/wiki/Chado_Tables#Table:_cvterm) | cvterm\_id | inteiro | *UNIQUE # 1 NOT NULL* |
|  | classificação | inteiro | *NÃO NULO* |
| [cvterm](http://gmod.org/wiki/Chado_Tables#Table:_cvterm) | cvterm\_type\_id | inteiro | *UNIQUE # 1 NOT NULL* |

Tabelas que fazem referência a esta por meio de restrições de chave estrangeira:

* [expression\_cvtermprop](http://gmod.org/wiki/Chado_Tables#Table:_expression_cvtermprop)

**Genótipo**

Definido no [Módulo Genético Chado](http://gmod.org/wiki/Chado_Genetic_Module) .

Um *genótipo* no Chado é basicamente um *nome* , com uma pilha de *características* associadas a ele.

Isso costumava significar um conjunto de alelos.

Não tenho certeza de como as cepas foram tratadas

Na era do sequenciamento de alto rendimento, isso pode ser muito mais detalhado.

**Tabela: genótipo**

Contexto genético. Um genótipo é definido por uma coleção de recursos, mutações, balanceadores, deficiências, blocos de haplótipos ou construções de engenharia.

|  |  |  |  |
| --- | --- | --- | --- |
| **Estrutura do genótipo** | | | |
| **Tecla F** | **Nome** | **Modelo** | **Descrição** |
|  | genotype\_id | serial | *CHAVE PRIMÁRIA* |
|  | nome | texto | Nome alternativo opcional para um genótipo, para fins de exibição. |
|  | Nome único | texto | *UNIQUE NOT NULL*  O nome exclusivo de um genótipo; tipicamente derivado das características que constituem o genótipo. |
|  | Descrição | personagem variando (255) |  |

Tabelas que fazem referência a esta por meio de restrições de chave estrangeira:

* [feature\_genotype](http://gmod.org/wiki/Chado_Tables#Table:_feature_genotype) , [phendesc](http://gmod.org/wiki/Chado_Tables#Table:_phendesc) , [phenotype\_comparison](http://gmod.org/wiki/Chado_Tables#Table:_phenotype_comparison) , [phenstatement](http://gmod.org/wiki/Chado_Tables#Table:_phenstatement) , [stock\_genotype](http://gmod.org/wiki/Chado_Tables#Table:_stock_genotype) , [wwwuser\_genotype](http://gmod.org/wiki/Chado_Tables#Table:_wwwuser_genotype)

**Tabela: feature\_genotype**

|  |  |  |  |
| --- | --- | --- | --- |
| **feature\_genotype Structure** | | | |
| **Tecla F** | **Nome** | **Modelo** | **Descrição** |
|  | feature\_genotype\_id | serial | *CHAVE PRIMÁRIA* |
| [recurso](http://gmod.org/wiki/Chado_Tables#Table:_feature) | feature\_id | inteiro | *UNIQUE # 1 NOT NULL* |
| [genótipo](http://gmod.org/wiki/Chado_Tutorial#Table:_genotype) | genotype\_id | inteiro | *UNIQUE # 1 NOT NULL* |
| [recurso](http://gmod.org/wiki/Chado_Tables#Table:_feature) | chromosome\_id | inteiro | *ÚNICO # 1*  Uma característica do "cromossomo" do tipo SO. |
|  | classificação | inteiro | *A classificação UNIQUE # 1 NOT NULL*  pode ser usada para organismos n-plóides ou para preservar a ordem. |
|  | cgroup | inteiro | *UNIQUE # 1 NOT NULL*  Grupo espacialmente distinguível. grupo pode ser usado para distinguir os grupos cromossômicos, por exemplo (produtos de RNAi e assim por diante podem ser tratados como grupos diferentes, uma vez que não caem em um cromossomo particular). |
| [cvterm](http://gmod.org/wiki/Chado_Tables#Table:_cvterm) | cvterm\_id | inteiro | *UNIQUE # 1 NOT NULL* |

**Ambiente**

Também definido no [Módulo Genético Chado](http://gmod.org/wiki/Chado_Genetic_Module) .

**Tabela: ambiente**

O componente ambiental de uma descrição de fenótipo.

|  |  |  |  |
| --- | --- | --- | --- |
| **estrutura do ambiente** | | | |
| **Tecla F** | **Nome** | **Modelo** | **Descrição** |
|  | ambiente\_id | serial | *CHAVE PRIMÁRIA* |
|  | Nome único | texto | *ÚNICO NÃO NULO* |
|  | Descrição | texto |  |

Tabelas que fazem referência a esta por meio de restrições de chave estrangeira:

* [ambiente\_cvterm](http://gmod.org/wiki/Chado_Tutorial#Table:_environment_cvterm) , [phendesc](http://gmod.org/wiki/Chado_Tables#Table:_phendesc) , [phenotype\_comparison](http://gmod.org/wiki/Chado_Tables#Table:_phenotype_comparison) , [phenstatement](http://gmod.org/wiki/Chado_Tables#Table:_phenstatement)

**Tabela: environment\_cvterm**

|  |  |  |  |
| --- | --- | --- | --- |
| **estrutura environment\_cvterm** | | | |
| **Tecla F** | **Nome** | **Modelo** | **Descrição** |
|  | environment\_cvterm\_id | serial | *CHAVE PRIMÁRIA* |
| [ambiente](http://gmod.org/wiki/Chado_Tutorial#Table:_environment) | ambiente\_id | inteiro | *UNIQUE # 1 NOT NULL* |
| [cvterm](http://gmod.org/wiki/Chado_Tables#Table:_cvterm) | cvterm\_id | inteiro | *UNIQUE # 1 NOT NULL* |

**Fenótipo, Diversidade Natural e Suporte Atlas**

Fenótipos, diversidade natural e suporte de atlas são áreas de trabalho futuro no Chado. O Chado tem um módulo de fenótipo, mas não envelheceu tão bem quanto outros módulos. Seu suporte para a diversidade natural é limitado ao que é implementado nos módulos de genótipo, ambiente e fenótipo. Isso não é robusto o suficiente para lidar com estudos que são frequentemente feitos na comunidade de plantas e cada vez mais feitos em comunidades animais também.

Para resolver isso, há vários esforços em andamento. O [projeto Aniseed](http://aniseed-ibdm.univ-mrs.fr/) inclui atlas gráficos de anatomia, expressão e destino de células em 4 dimensões. Aniseed está atualmente em processo de reimplementação para usar o Chado. Este trabalho provavelmente levará a contribuições de volta ao GMOD (tanto no Chado quanto em uma interface da web) para melhor apoiar esses tipos de atlas.

Um melhor suporte à diversidade natural será adicionado no próximo ano. [NESCent](http://nescent.org/) desenvolveu um módulo de Chado de diversidade natural protótipo baseado no [PIBDM](http://www.maizegenetics.net/gdpdm/) , que irá adicionar suporte robusto para dados de diversidade natural.