Documentos

ISSN 1677-9274
Dezembro, 2010
106

Eutils-search – Manual do Usuário

Embrapa

ISSN 1677-9274
Dezembro, 2010

Empresa Brasileira de Pesquisa Agropecuária
Embrapa Informática Agropecuária
Ministério da Agricultura, Pecuária e Abastecimento

Documentos 106

Eutils-search - Manual do usuário

Roberto Hiroshi Higa
Rodrigo Shizuo Yasuda

Embrapa Informática Agropecuária
Campinas, SP
2010

Embrapa Informática Agropecuária
Av. André Tosello, 209 - Barão Geraldo
Caixa Postal 6041 - 13083-886 - Campinas, SP
Fone: (19) 3211-5700 - Fax: (19) 3211-5754
www.cnptia.embrapa.br
sac@cnptia.embrapa.br

1ª edição on-line 2010

Dados Internacionais de Catalogação na Publicação (CIP)
Embrapa Informática Agropecuária

Higa, Roberto Hiroshi.
Eutils-search – manual do usuário / Roberto Hiroshi Higa, Rodrigo Shizuo Yasuda. - Campinas : Embrapa Informática Agropecuária, 2010.
17 p. : il. – (Documentos / Embrapa Informática Agropecuária ; ISSN 1677-9274, 106).

1. Software eutils-search. 2. Biotecnologia. 3. Biologia molecular. 4. Mineração de texto. I. Yasuda, R. S. II. Título. III. Série.
005.15 CDD (21. ed.)

Autor

Roberto Hiroshi Higa
Doutor em Engenharia Elétrica
Pesquisador da Embrapa Informática Agropecuária
Av. André Tosello, 209, Barão Geraldo
Caixa Postal 6041 - 13083-970 - Campinas, SP
Telefone: (19) 3211-5862
e-mail: roberto@cnptia.embrapa.br

Rodrigo Shizuo Yasuda
Estagiário da Embrapa Informática Agropecuária
e-mail: rodrigoyasuda@gmail.com

Apresentação

O software Eutils-search tem por objetivo trazer do banco de dados PubMed informações sobre artigos relacionados a genes de um organismo específico, de acordo com as regras referentes à taxa de acesso impostas pelo site. As informações trazidas são, então, armazenadas localmente em um banco de dados para acesso rápido. Além disso, o software também gera documentos XML correspondentes às informações do organismo requisitado.

O eutils-search é uma ferramenta de apoio ao desenvolvimento de aplicações de mineração de textos voltadas para os domínios de biotecnologia e biologia molecular, baseada em informações textuais obtidas do banco de dados PubMed.

Este documento apresenta os pré-requisitos e a descrição dos parâmetros necessários para utilização do software, bem como uma descrição de alguns aspectos internos do software, para melhor entendimento do processo que ele automatiza, além de alguns exemplos e uso.

Kleber Xavier Sampaio de Souza
Chefe Geral
Embrapa Informática Agropecuária

Sumário

Eutils-search – Manual do Usuário

Roberto Hiroshi Higa
Rodrigo Shizuo Yasuda

Pré-requisitos

Para executar o eutils-search, é necessário um servidor de banco de dados PostgreSQL e a correspondente biblioteca de acesso JDBC.

Em termos de uso de memória em disco, é necessário ter disponível uma quantidade mínima de 3 GB, embora esse parâmetro varie de acordo com o organismo requisitado.

Parâmetros de chamada

O eutils-search é executado por meio da seguinte linha de comando:

```
java –jar eutils-search.jar [parâmetros] tax_id
```

onde tax_id é o ID do organismo requisitado e pode ser encontrado no site do NCBI por meio do link[1]. Atualmente, o eutils-search considera apenas

[1] *Disponível em: <http://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?mode=Root>.*

tax_ids que se referem a organismos, species na terminologia utilizada pelo NCBI. Por isso, ao longo deste documento, os termos organismo e tax serão usados indistintamente, em geral, para referenciar um tax_id específico.

Também é necessário que os arquivos **gene2pubmed, gene_info e generifs_basic** e **nodes.dmp** estejam no mesmo diretório do arquivo eutils-search.jar. Esses arquivos podem ser obtidos, via ftp, dos sites[2].

O eutils-search possui os seguintes parâmetros:

createdb

Essa opção indica que todas as tabelas necessárias para a execução do programa devem ser criadas, eliminando qualquer tabela anteriormente criada. Essa operação deve ser executada uma única vez com o intuito de criar as tabelas ou se o usuário realmente quiser remover as tabelas existentes.

populate

Essa opção indica que todos os dados do arquivo gene2pubmed, que contém as relações entre taxonomias, genes e artigos publicados, devem ser carregados para o banco de dados PostgreSQL. Essa operação é computacionalmente muito custosa e deve ser executada uma única vez. Para atualizar o banco de dados, deve-se utilizar o parâmetro --**updatepubmed**.

updatepubmed

Essa opção indica que o banco de dados deve ser atualizado com novas informações do arquivo gene2pubmed.

---

[2] *Disponível em: < ftp://ftp.ncbi.nih.gov/gene/> e <ftp://ftp.ncbi.nih.gov/pub/taxonomy/>*

updaterif

Essa opção indica que o banco de dados deve ser atualizado com informações do arquivo **generifs_basic**, que contém a descrição RIF dos artigos do PubMed.

updategeneinfo

Essa opção indica que o banco de dados deve ser atualizado com informações do arquivo **gene_info**, que contém informações sobre os genes dos organismos.

creategenexml

Essa opção indica que os arquivos XML referentes ao organismo requisitado devem ser criados. Cada arquivo XML criado agrupa as informações referentes a um gene do organismo requisitado.

createarticlexml

Essa opção indica que os arquivos XML do organismo requisitado devem ser criados. Cada arquivo XML criado contém informações referentes a um artigo relacionado a um ou mais genes do organismo requisitado.

fetchdata

Essa opção indica que o download dos títulos, datas de publicação e resumos dos artigos relacionados ao organismo requisitado devem ser exe-

cutados. Devido às restrições impostas para acesso do PubMed, essa operação faz o download de informações com 100 artigos a cada requisição. O intervalo entre cada requisição é de 1 segundo de segunda a sexta-feira. Não existe intervalo entre requisições das 23h as 5h ou durante sábados e domingos. Essa operação pode demorar muito tempo e deve, preferencialmente, ser feita durante os fins de semana. Além disso, ela não deve ser interrompida. Sua interrupção acarretará no seu reinício, ou seja, todo o download feito até o momento será feito novamente.

Estrutura interna

A utilização do eutils-search é seguida de um processo de mineração de textos, em que os dados obtidos do PubMed são tratados. Dependendo de como esse processo está estruturado, os dados obtidos por meio do eutils-search podem ser acessados de diferentes formas. Por isso, é interessante que o usuário conheça alguns detalhes do processo que o eutils-search automatiza e de sua estrutura interna.

O processo automatizado pelo eutils-search pode ser dividido em quatro etapas:

- Criar as tabelas no banco de dados local;
- Ler os arquivos nodes.dmp, gene2pubmed, gene_info e generi fs_basic, nessa ordem;
- Obter os dados do PubMed (dado o tax_id referente a um organis mo específico);
- Gerar os documentos XML;

Alguns passos não precisam ser executados, caso já o tenham sido anteriormente. Além disso, o passo 2 pode consistir apenas na atualização das informações já existentes. Se o usuário quiser atualizar as informações existentes, bastará utilizar um dos parâmetros update, apresentados na sessão anterior.

O eutils-search foi desenvolvido utilizando a linguagem java e, internamente, está organizado em sete classes: **Article, Gene, myXMLParser, Searcher, SQLConn, Tax** e **configHandler**.

Classes Article, Gene e Tax

As classes **Article, Gene** e **Tax** refletem a estrutura dos dados tratados. Cada uma dessas classes é responsável por armazenar um artigo, um gene e uma taxonomia, respectivamente, criando uma hierarquia na qual os Organismos estão na raíz da árvore, os **Genes** são filhos dos Organismos e os Artigos (**Article**) são filhos dos **Genes**. Resumidamente, um conjunto de artigos está relacionado a um gene que, por sua vez, está relacionado a um organismo.

Essas três classes representam cada tipo específico de dado. Por exemplo, a classe **Article** armazena o PMID (ID do artigo no PubMed), o título do artigo, seu resumo e outras informações. A classe **Gene** armazena dados como a descrição do gene e seu símbolo, e uma coletânea de artigos referentes a esse gene. A classe **Tax** armazena apenas o tax_id (ID do organismo) e uma lista de genes que o compõe.

Além disso, a classe **Tax** é responsável por carregar os dados de 3 arquivos: generifs_basic, gene2pubmed e gene_info. Esses arquivos serão explicados a seguir, sendo que mais informações podem ser obtidos em[3].

Arquivos genrifs_basic, gene2pubmed e gene_info

O arquivo generifs_basic pode ser obtido em [4] e deve ser descompactado no mesmo diretório onde o arquivo executável eutils-search.jar está localizado. Ele possui anotações sobre alguns dos genes do PubMed, e está estruturado em cinco colunas: **TaxID, GeneID, PubmedID, Last Update, GeneRIF Text**, sendo que as colunas PubmedID e Last Update não são lidas pelo eutils-search. Cada linha desse arquivo contém a informação RIF para uma tripla TaxID, GeneID, PubmedID. A informação GeneRIF Text é armazenada no banco de dados local e é, posteriormente, utilizada para gerar os documentos XML. Note que, apesar da informação

[3] *Disponível em: <ftp://ftp.ncbi.nih.gov/gene/README>*

[4] *Disponível em: <ftp://ftp.ncbi.nih.gov/gene/GeneRIF/generifs_basic.gz>.*

PubmedID estar presente nesse arquivo, ela não indica que o GeneRIF é uma informação específica de um artigo, dado que o PubmedID presente nesse arquivo é o primeiro artigo encontrado que está relacionado com o GeneID em questão. Vale ainda notar que o RIF é uma informação entrada manualmente no banco de dados do PubMed por um processo que pode ser realizado pelo link [5]. No banco de dados utilizado pelo eutils-search, a informação sobre o RIF é armazenada em uma tabela específica, pois a quantidade de genes que possuem anotação é muito pequena, comparada com a quantidade de genes no PubMed.

arquivo gene2pubmed pode ser obtido em [6] e deve ser descompactado no mesmo diretório onde o executável eutils-search.jar está localizado. Ele contém a relação entre artigos, genes e organismos, sendo dividido em três colunas: TaxID, GeneID, PubmedID. Toda a hierarquia entre artigo, gene e taxonomia é definida nesse arquivo que, portanto, tem um papel central no processo automatizado pelo eutils-search.

O arquivo gene_info pode ser obtido em [7] e deve ser descompactado no mesmo diretório em que o executável eutils-search.jar está localizado. Ele contém as informações sobre genes, como descrição e símbolo. Nesse arquivo, que possui diversas colunas, apenas 5 colunas são relevantes para o eutils-search: TaxID, GeneID, Symbol, Description e Other_Designations.

As classes SQLConn, myXMLParser, configHandler Searcher

A classe **SQLConn** é responsável por realizar a comunicação entre o eutils-search e o servidor de banco de dados local (PostgreSQL). Ela é a responsável pela realização de consultas, leitura dos arquivos **generifs_basic, gene2pubmed** e , e inserção de dados no banco de dados, armazenamento dos dados obtidos do PubMed, bem como pela criação das tabelas do banco de dados. Os parâmetros de conexão com o servidor de banco de dados são obtidos da classe

---

[5] Disponível em: <http://www.ncbi.nlm.nih.gov/projects/GeneRIF/GeneRIFhelp.html>.

[6] Disponível em: <ftp://ftp.ncbi.nih.gov/gene/DATA/gene2pubmed.gz>.

[7] Disponível em: <ftp://ftp.ncbi.nih.gov/gene/DATA/gene_info.gz>.

O diagrama de tabelas do banco de dados utilizado por eutils-search é apresentado na Figura 1.

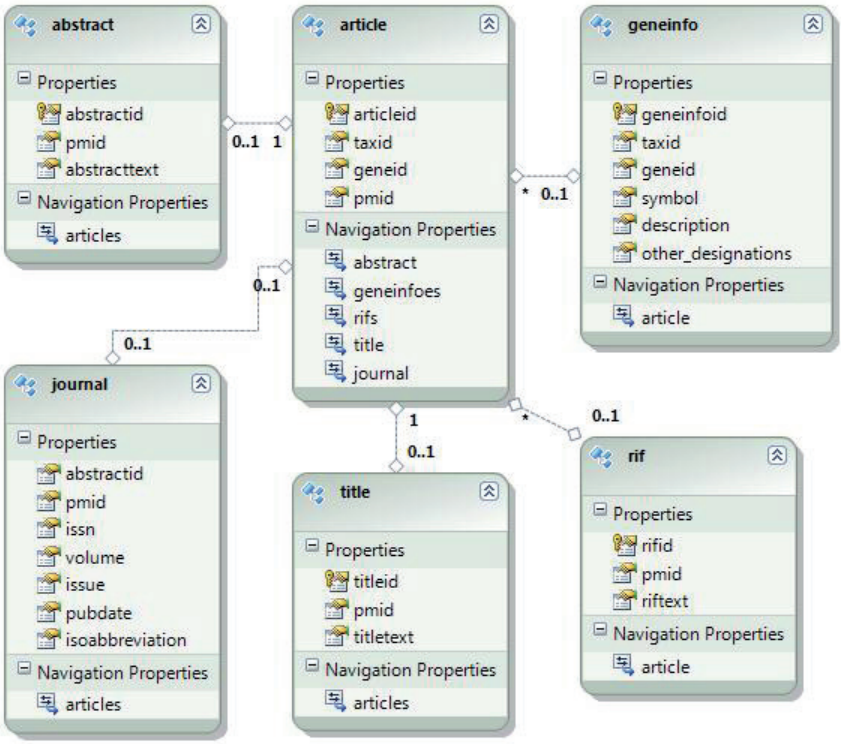

Figura 1. Diagrama de tabelas do banco de dados.

A classe **myXMLParser** é responsável por tratar os arquivos obtidos do PubMed no formato XML e criar os correspondentes arquivos XML. Para a leitura de arquivos XML, essa classe faz uso do JavaX e do módulo Xpath, e para criação dos arquivos XML, ela faz uso da classe **SQLConn** para obtenção de dados do banco de dados.

A classe **configHandler** tem como único propósito ler o arquivo de configurações eutils.conf que armazena informações de conexão com o banco de dados. O arquivo de configuração armazena os parâmetros *database, username e password*, utilizados para conexão com o banco de dados

local. Esse arquivo deve estar no mesmo diretório do programa executável **eutils-search.jar.** No arquivo **eutils.conf**, esses parâmetros são definidos utilizando a sintaxe <parâmetro>=<valor>. Por exemplo:

- `database=mydb`
- `username=myuser`
- `password=mypass`

A classe **Searcher** contém o procedimento principal do software, onde os parâmetros de entrada são lidos e as funções necessárias chamadas, de acordo com o que foi requisitado pelo usuário. Além disso, nessa classe é feito o controle de requisições ao PubMed, de acordo com os limites propostos pelo sistema. Entre o período de 21h e 5h, durante a semana, e aos sábados e domingos não há limite de requisições. Com exceção desses períodos, existe um limite de três requisições por segundo e 100 requisições por dia.

Exemplo de uso

Para realizar todo o processo, ou seja, criar o banco de dados, ler os **arquivos generifs_basic, gene2pubmed** e **gene_info**, fazer o download dos dados do PubMed e obter os documentos do organismo Bos taurus `(tax_id = 9913)`:

```
java –jar eutils-search.jar --createdb --populate
--updaterif --updategeneinfo --fetchdata --createarti-
clexml 9913
```

Supondo que o banco de dados já foi criado, para populá-lo com os arquivos generifs_basic, gene2pubmed e gene_info:

```
java –jar eutils-search.jar --populate --updaterif
--updategeneinfo
```

Supondo que o banco de dados já foi criado e populado com os arquivos generifs_basic, gene2pubmed e gene_info, para fazer o download dos dados adicionais do PubMed (títulos e resumos dos artigos):

```
java –jar eutils-search.jar --fetchdata 9913
```

Supondo que os dados já foram obtidos do pubmed, para criar os documentos XML:

```
java –jar eutils-search.jar --createarticlexml 9913
```

Informática Agropecuária

Ministério da
Agricultura, Pecuária
e Abastecimento

CGPE 9080