TR 5421 Sequence Deleter

Author: sc

Created: December 30, 2004

Last Modified: March 3, 2005 09:44

1 Purpose of Document

To list the requirements and design for deleting sequences from MGI that are no longer represented by the provider.

2 Introduction

A separate process is needed to delete sequences that are incrementally loaded into MGI. Currently, this includes the GenBank and RefSeq loads. Initially GenBank and RefSeq were loaded from a release. Since that time, May 2004, new and changed sequences have been loaded from non-cumulative update files from each provider, but no deletions have been done.

Upon each GenBank and RefSeq release a file is provided that contains sequences that have been deleted since the last release. This document describes how we will use these files to 'delete' sequences from MGI

3 Definitions

- delete - set SEQ_Sequence._SequenceStatus_key to 'deleted'

4 Other Documents

1. /mgi/all /wts_projects/3400/3404/requirements.pdf
2. /mgi/all /wts_projects/5300.5325/SequenceLoad/sequenceloaddesign.pdf

5 Requirements

5.1 Sequence Deleter Requirements

Upon each release:

1. mirror RefSeq and GenBank delete files
2. status mouse sequences in each delete file as deleted in MGI
    1. If a mouse sequence in the delete file is not in MGI ...QC report ?
    2. If a mouse sequence in the delete file is statused other than 'ACTIVE'
       e.g.

- ‘DELETED’ (key = 316343)
- ‘SPLIT’ (key = 316344, 2/16/2004 63 sequences)
- ‘Not Loaded’ (key = 316345)

what do we do... ?

5.2 Incremental Sequence Load Requirements

1. If a sequence to be updated is statused as deleted update as usual and status as ACTIVE?

   Note that refseqs found in the refseq release*.removed-records file have a ‘removed status’ column - see Section 6.2.2, “Incremental Delete Input” on page 4 column 8.. Sequences with removed status of ‘temporarily suppressed’ and even ‘permanently supressed’ could be restored at a later date.

5.3 WI Requirements

The JSAM requirements doc - see Section 4, “Other Documents” on page 1 - in section 4.4.3 states ‘Deleted Sequences will continue to appear in query results. The sequence detail page for a deleted sequence states “This sequence has been deleted from the provider database.”

5.4 EI Requirements

1. Add a restriction that you cannot annotate to a deleted sequence (there is a stored procedure that can handle this, ACC_verifySequenceAnnotation)?

6 Input

6.1 GenBank

In order to determine all SEQ_Sequences in MGI which have been deleted from GenBank we must initially do a what is in MGI that is not in Genbank.. GenBank creates a file of all the accession ids in GenBank every Sunday night and puts it on its ftp site. After the initial diff is processed using this file, the delete file in each successive release may be processed to keep MGI current with GenBank. See also genbank.ncbi.mail.fm in this project directory.

6.1.1 Initial Diff Input

ftp://ftp.ncbi.nih.gov/genbank/gbacc.idx.gz

This file contains all nucleotide accession numbers for sequences in GenBank. It is in the format:

“accid.version tab locus-name tab division tab accid”

e.g. A00001.1	A00001	PAT	A00001

6.1.2 Incremental Delete Input

ftp://ftp.ncbi.nih.gov/genbank/gbdel.txt.gz. (Update mirror_ftp/ftp.ncbi.nih.gov to fetch this file.)

This file contains all "accession numbers of entries deleted since the previous release" (quote from release notes). It is in the format (pipe '|' delimited):

"division file# | accession id"

e.g. ROD1|V06122.

6.2 RefSeq

In order to determine all SEQ_Sequences in MGI which have been deleted from RefSeq we must initially what is in MGI that is not in RefSeq. RefSeq creates a file of all the accession ids in a given RefSeq release. After the initial diff is processed using this file, the delete file in each successive release may be processed to keep MGI current with RefSeq. See also refseq.ncbi.mail.fm in this project directory.

6.2.1 Initial Diff Input

ftp://ftp.ncbi.nih.gov/refseq/release/release-catalog/RefSeq-release*.catalog where * = release number (9 is the current release 2/16/2004).

This file contains information about all RefSeqs in the current release and is in the following format (this was cut and paste from ftp://ftp.ncbi.nih.gov/refseq/release/release-catalog/README):

Content: Tab-delimited listing of all accessions included in the current

RefSeq release.

Columns:

1. taxonomy ID
2. species name
3. accession.version
4. gi
5. refseq release directory accession is included in complete + other directories '|' delimited
6. refseq status

   na - not available; status codes are not applied to most genomic records

   INFERRED

   PREDICTED

   PROVISIONAL

   VALIDATED

   REVIEWED

   MODEL

   UNKNOWN - status code not provided; however usually is provided for this type of record

7. length

Example:

```
10090   Mus musculus    NM_008155.1     6680066 complete|vertebrate_mammalian   PROVISIONAL     1985
```

6.2.2 Incremental Delete Input

ftp://ftp.ncbi.nih.gov/refseq/release/release-catalog/release3.removed-records. (Update mirror_ftp/ftp.ncbi.nih.gov to fetch this file).

This file contains information about refseqs that were included in the previous release but are not included in the current release. It is in the same format as Section 6.2.1, “Initial Diff Input” on page 3 with an 8th column:

8. removed status
   - dead protein: protein was removed when genomic record was reloaded and protein was not found on the nucleotide update. This is an implied permanent suppress.
   - temporarily suppressed: record was temporarily removed and may be restored at a later date.
   - permanently suppressed: record was permanently removed. It is possible to restore this type of record however at the time of removal that action is not anticipated.
   - replaced by accession: the accession in column 3 has become a secondary accession that cited in column 8.

Example:

```
10090   Mus musculus    NG_004108.3     53749217        complete|vertebrate_mammalian   na      885     temporarily suppressed
```

7 Design

7.1 Initial Diff Processing

Quick and dirty.

1. grep accid from input file > Provider_accid.txt
2. query MGI for accids of _MGIType_key = 19, _LogicalDB_key = 9 or 27, preferred = 1 > MGI_accid.txt
3. comm -13 Provider_accid.txt MGI_accid.txt > MGI_todelete.txt to get accids unique to MGI
4. create an java interpreter and process MGI_todelete.txt with the incremental deleter

7.2 Incremental Processing

The seqdeleter is a DLALoader. It has:

7.2.1 Configuration file for each provider e.g. refseqdeleter.config, gbseqdeleter.config

Of note are the following:

- SEQ_INTERPRETER full package path to the interpreter class e.g.
  - org.jax.mgi.app.seqdeleter.GBSeqDeleterInterpreter
  - org.jax.mgi.app.seqdeleter.RefSeqDeleterInterpreter
- INPUTDIR - full path to the input file directory
- APP_INFILES - full path to the input file(s)
- FILEDIR - full path to the directory under /data/loads where logs, output, archive and report directories reside. FILEDIR is referenced by seqdeleter_common.config
- APP_CAT_METHOD - how this providers files are piped to stdin (See INFILE_NAME in Section 7.2.2, “Common seqdeleter configuration - seqdeleter_common.config” on page 5).
- JOBSTREAM - jobstream name - MGI_User.login
- SEQ_LOGICALDB - logical db for the provider - ACC_LogicalDB.name
- MAIL_LOADNAME - load email notification subject string

7.2.2 Common seqdeleter configuration - seqdeleter_common.config

Of note are the following variables:

- DLA_START=org.jax.mgi.shr.dla.loader.DLAStart - DLA frameworks main class which creates an instance of the configured DLA_LOADER
- DLA_LOADER=org.jax.mgi.app.seqdeleter.SeqDeleter - The loader class created by DLA_Start
- INFILE_NAME=STDIN - we pipe file(s) to stdin so that we may configure the cat method. e.g. refseq delete file is mirrored uncompressed thus uses /usr/bin/cat, genbank delete file is mirror compressed, thus uses /usr/local/bin/gunzip -c
- SEQ_QUERY_BATCHSIZE=400 - each sequence must be looked up in MGI, this is the number of sequences to batch in one query.

7.2.3 jobstream script - Very similar to other DLALoader jobstream scripts, it:

- archives output files
- sources configurations files and DLA Jobstream functions
- Invokes the seqdeleter, piping in the input

7.2.4 The following classes reside in lib_java_dla. A brief description is provided here.

1. SeqDeleter.java

   This class extends the DLALoader and implements its initialize, preprocess, run, and postprocess methods.

- initialize() creates the iterator over the input file which uses the configured Interpreter. It creates a SeqDeleterProcessor to handle deleting of sequences
- preprocess() is an empty implementation
- run() iterates thru the sequences in the input passing them to the SeqDeleterProcessor.processDelete(String seqidToDelete).
- postprocess does final tasks, closing streams and reporting load statistics.

2. SeqDeleterProcessor.java

   This class provides the processDelete(String seqIdToDelete method which looks up the sequence in MGI and if it is found, statuses it as deleted.

3. RefSeqDeleterInterpreter.java

   This class simply pulls the seqid out of the RefSeq delete file. The interpret() method returns the seqid String.

4. GBSeqDeleterInterpreter.java

   This class simply pulls the seqid out of the GenBank delete file. The interpret() method returns the seqid String.