

KEEPING REFSEQ NON-REDUNDANT AND CONTINUOUSLY UPDATED IN-HOUSE

BY GETDB™

GETDB™ BUSINESS CASE

context

RefSeq is a database of reference sequences, made available by the National Centre for Biotechnology Information (NCBI) as a release split into several directories corresponding to major taxonomic groups. Each directory contains a number of files with protein, RNA or genomic sequences data in various formats. Daily updates, also made available by the NCBI, contain information which must either be added to, or replace the information present in a specific release.

The "dependency problem"

RefSeq has a two month release cycle with daily updates. The updates must be applied to the correct release, that is, the release of the same update cycle. Unfortunately the NCBI does not make this dependency explicit. It is implicit, in that when the release files are replaced on the remote site, the updates corresponding to the original release are also removed.

Downloading the release files typically takes some time (a few hours), whereas downloading the updates is very quick (a few minutes). It follows that each time a new release is produced, an erroneous updated release can be built by applying the new updates to the old release.

objective

The objective for GetDB™ is to make an up-to-date version of RefSeq available on a daily basis. This involves downloading the "release" resource, applying the appropriate "daily" updates as they appear and removing the redundancy created by the updates.

GetDB™ solution

- Three GetDB™ resources are created.

Two *Basic Resources* : one for the release and one for the daily updates

- RefSeq_Release (RR)
- RefSeq_Update (RU)

A *Composite Resource* for the updated, non-redundant data (daughter of the two Basic Resources)

- RefSeq_Updated_Release (RUR)

GetDB™ objects

Resource: The basic unit on which GetDB™ applies data processing methods (plugin sequences). Resources represent data sets which have internal identifiers, names, versions and states. There are two main types of resources, namely, basic resources and composite resources.

Basic Resource: A GetDB™ resource which is acquired from another server either on the internal information system, or accessible through the internet. The resource consists of a set of files whose names match a filter expressed as a regular expression, which are locally updated by acquiring only new or modified data. Once acquired, normal plugin execution occurs on the basic resource.

Composite Resource: A GetDB™ resource which is created based on one or more other GetDB™ resources (the parent resources). The actual "building" or "construction" of the composite resource is handed off to a construction method. Once created, normal plugin execution occurs on the composite resource.

Construction Method: An arbitrarily complex program used to build a composite resource. The construction method mimics the download process of basic resources.

Plugin: A program which makes up the basic unit of processing in GetDB™. Plugins range from simple wrapper scripts to full fledged data processing programs. Plugins are run by GetDB™ whenever new data is incorporated into a resource. Dependencies between plugins can be expressed, thereby, creating plugin sequences.

- Construction of the composite resource

Concerning the construction of the RUR resource, the main difficulty results in resolving the "dependency problem". This issue is linked to the fact that the "daily" resource files sometimes mustn't be applied to the current Release resource files.

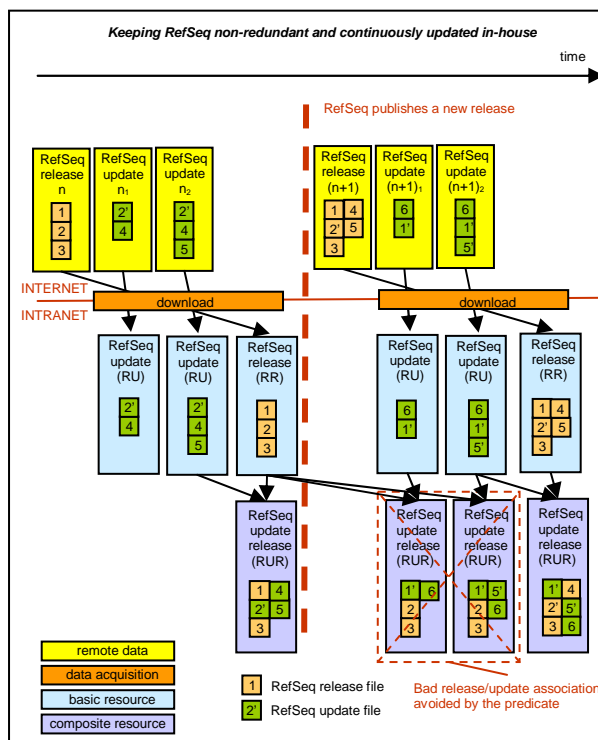
In order to decide whether a new composite resource should be built or not, a predicate is implemented. It applies the following rules:

| | <i>New daily cycle</i> | <i>Current daily cycle</i> |
|------------------------|------------------------|----------------------------|
| <i>New release</i> | Build | Not build |
| <i>Current release</i> | Not build | build |

The decision is based on the qualification of the current release (current/new) and of the current daily cycle (current/new).

Current Release and New Release resource: When GetDB™ downloads and installs the RefSeq release resource, it creates different versions. In this way, the predicate can detect whether a Release resource involved in the potential new composite resource is the same as the Release resource involved in the previous composite resource construction.

Current Daily Cycle and New Daily Cycle: The RefSeq Daily resource on the NCBI can be considered as a cumulative resource (files are added every day) for about two months, then purged (all the files are deleted). This purge event is the moment when a new Daily Cycle has started. The predicate can thus compare the files contained in the two versions of the Daily resource in order to know if a purge event has occurred (meaning the Daily resource is in a new cycle) or not.



The purge event should be detectable when all files on the remote site have disappeared*.

The construction method is responsible for the actual creation of the daughter resource. This involves removing any old or redundant entries based on their identifiers.

results

GetDB™ provides a means for maintaining up-to-date data, even when complex processing is required. Composite resources can be used to provide users with updated data from, among many others, RefSeq, Genbank and Pubmed. The use of predicates provides the ability to base the decision of whether to create a new daughter resource version on information from various sources, including the internal information system.

GetDB™ composite resources, with their predicates and construction methods, provide a powerful and flexible mechanism to combine and integrate data.

GENOMING IN SHORT

Genomining is a bioinformatics company, created in 2001, which provides real time solutions for the discovery and interpretation of data in biological research. It provides the life science industry with products ranging from data integration and manipulation frameworks, virtual screening management and optimization tools, to supercomputing solutions.

CONTACTS

Katja Schuerer
 Bioinformatics Development Manager
Katja.schuerer@genomining.com
 Tel: +33 (0)1 42 31 08 01

Helene Chao
 Business Development
helene.chao@genomining.com
 Tel: +33 (0)1 42 31 08 01

OUR WEBSITE : www.genomining.com

*In fact, files may reside on the remote site but mustn't be included in the detection of the purge event. An example of a file which may not disappear in a purge event is the "README" file. To avoid this issue, a filter (regular expression) can be provided to the predicate. Files matching the expression are not considered in this detection.