Supporting Search with Lucene

out of date content - available for reference purposes only

- Introduction
- How to create and deploy Lucene indexes for IGB
    - Build Lucene.zip
    - Get felix.jar
    - Designate fields to be indexed by file type
    - Specify index target using shell variable (optional)
    - Index the target file(s)
    - Deploy the indexes
- To test
- Troubleshooting

Introduction

IGB supports search using Lucene indexes starting with IGB release 7.0.

When a user loads a data set from QuickLoad, IGB queries the server to determine if a Lucene index for that data set is available. If a Lucene index is available, then IGB enables Lucene searching for that data set. At that point, if the user runs a search, then IGB will zoom and scroll to whatever region was found to contain an item that matched the search. To actually view the data, the user then has to click **Loa d Data**. Also, search results are show in the **Advanced Search** tab.

Lucene indexes are most useful for enabling search of data files such as EST or probe set alignment files that users don't normally load into memory.

How to create and deploy Lucene indexes for IGB

The code you need to build a Lucene index in IGB resides in tools/LuceneIndexing under the Genoviz project. As with nearly everything else in IGB, the Lucene indexing code is developed as an OSGi bundle.

Build Lucene.zip

1. Define an environment variable IGB_WORKSPACE, which should point to your copy of the checked-out genoviz trunk or branch.
2. Run `ant` to compile the code. A new directory named `full_dist` will appear that contains `LuceneIndexing.zip`.
3. Change into the `full_dist` directory and unzip `LuceneIndexing.zip`.

Example)

```
export IGB_WORKSPACE=$HOME/src/genoviz/trunk
ant # build the code
cd full_dist
unzip LuceneIndexing.zip
```

Get felix.jar

Currently, to run the indexing code, you'll need a copy of felix.jar that contains a Main class and enables you to launch the OSGi framework. We'll probably change this at some point, but for now, download the latest felix.

Example)

```
wget
http://mirrors.gigenet.com/apache//felix/org.apache.felix.main.distribution-4.2.1.t
ar.gz
```

Unpack the felix distribution and copy `bin/felix.jar` into `full_dist/bin`. Note that there is already a copy of `felix.jar` at that location. It was copied by the build process from elsewhere in the genoviz project and lacks a Main method. So you can't use it.

Designate fields to be indexed by file type

Edit resources/index.properties. Use this file to specify which fields in which file types will be indexed.

Example)

```
_default=id,name,gene name,description
_ignore=method,source,type,seq
psl=id
```

In this example, the "id" field of files of type "psl" (blat output format) will be indexed and nothing else.

Specify index target using shell variable (optional)

Set shell variable lucene_index_dir to specific index source, which can be a single file or a directory. If the source is a directory, then all the files in that directory will be indexed.

Index the target file(s)

Run ant using build.xml that was packaged within LuceneIndexing.zip. Change into the `full_dist` directory and run ant.

Examples)

Make indexes for Arabidopsis ATH1 and AG array:

```
ant
-Dext.lucene_index_dir=$HOME/src/genomes/pub/quickload/A_thaliana_Jun_2009/Affymetr
ix
```

Make indexes for Arabidopsis EST alignments:

```
ant -Dext.lucene_index_dir=$HOME/src/genomes/pub/quickload/A_thaliana_Jun_2009/EST
```

Note: when you run `ant` from inside the `full_dist` directory, ant will invoke `java -jar bin/felix.jar` using whatever target file or directory you provide using the `lucene_index_dir` parameter.

Deploy the indexes

The indexes will appear in the same directory as the source files. They do not need to be moved unless they need to be transferred onto a server.

To deploy indexes onto a QL site, move them into the same directory as their target files.

To test

Open the target data files in IGB and run a search. IGB should zoom and scroll to the region containing a feature that matched your query. Click the **Load Data** button to view the item that was found.

Troubleshooting

You may have trouble building or running the indexing if the top-level build.xml file in tools/LuceneIndexing has not been kept up to date with the rest of the genoviz project. Note that the indexing code is using SymLoaders and other code from the genometry and related bundles, which means that if a new bundle is added to the larger project, the Lucene indexing code may fail unless you include those new bundles in its `build.xml` script. So if you can't build or run Lucene indexing, check dependencies on other bundles.