

GeneGram Developer Documentation

Martin Goodfellow, martin.goodfellow@cis.strath.ac.uk,
<http://martin.g.1986.googlepages.com/genegram>

Database Connectivity

All communication with the Oracle 10 or 11g database is done via Java Database Connectivity (JDBC). The database driver used is the Oracle driver which can be found in the Java package `oracle.jdbc.driver`. This is contained in a JAR file which can be downloaded from the Oracle website. The required JAR file depends on the version of the database and the version of Java being used. The supported Java versions are 1.5 and 6.

All communication with the database is performed via the `DatabaseFunctions` class. The methods within the class are as follows:

```
public static Connection connect(String url, String dbName, String username, String password);
public static ResultSet query(Connection conn, String query);
public static Boolean doesTableExist(Connection conn, String tableName);
public static Boolean isTableEmpty(Connection conn, String tableName);
public static Boolean doesTablespaceExist(Connection conn, String tablespaceName);
public static Boolean doesIndexExist(Connection conn, String tableName);
```

Attention should also be paid to the class `DatabaseQueries`. This class contains a Java function to produce a set of DDL statements for creating tablespaces, providing a genome name and tablespace part number. For example, based on:

```
CREATE TABLESPACE GENOMEPARTNUMBER.TS DATAFILE
'~DBFFiles/GENOMEPARTNUMBER.TS.DBF' SIZE 500M AUTOEXTEND ON NEXT 50M
MAXSIZE UNLIMITED LOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE
MANAGEMENT AUTO
```

we generate several statements like:

```
CREATE TABLESPACE RATPART1.TS DATAFILE '~DBFFiles/RATPART1.TS.DBF' SIZE 500M
AUTOEXTEND ON NEXT 50M MAXSIZE UNLIMITED LOGGING EXTENT MANAGEMENT
LOCAL SEGMENT SPACE MANAGEMENT AUTO
```

for creating the tablespaces for the rat genome.

UML Diagrams

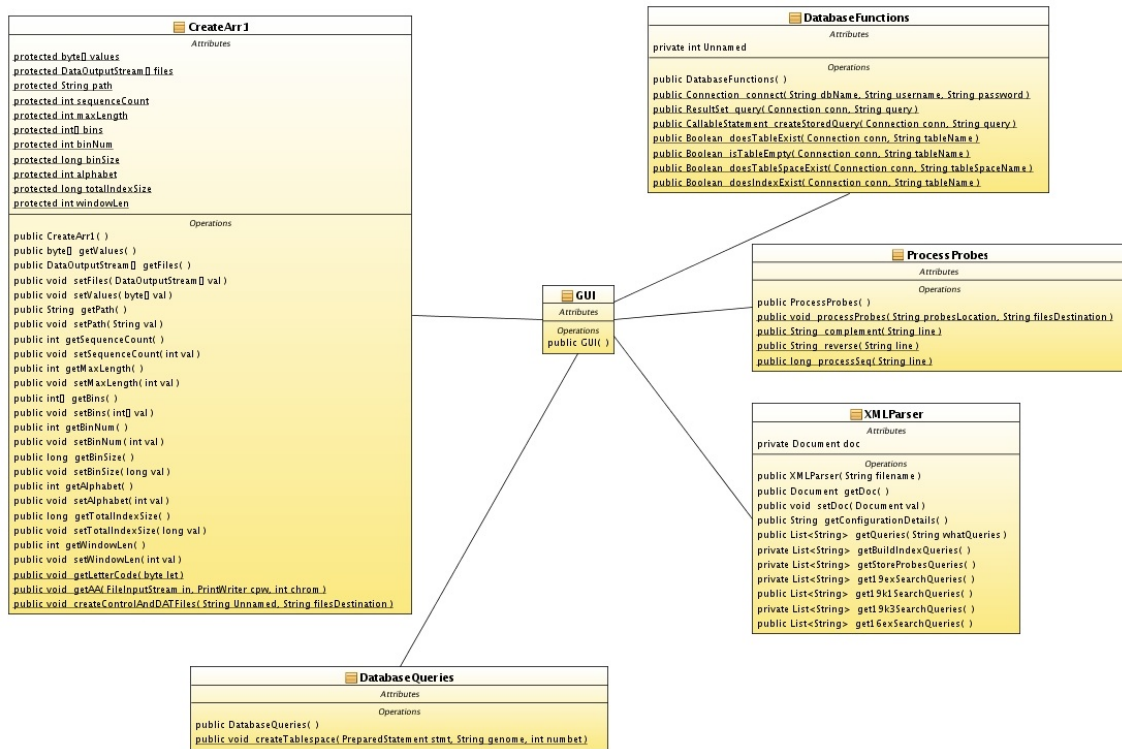


Figure 1: Class Diagram

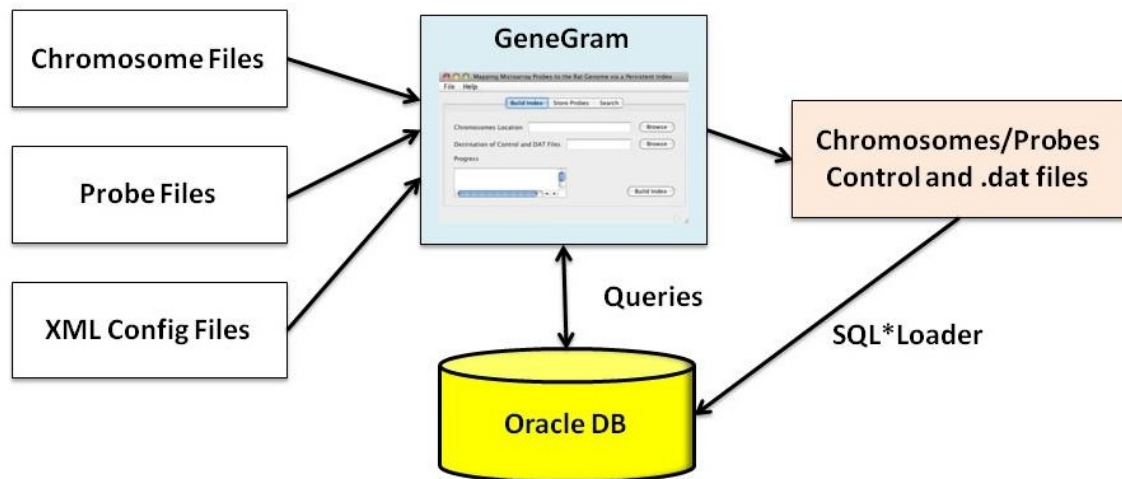


Figure 2: Software Architecture Diagram

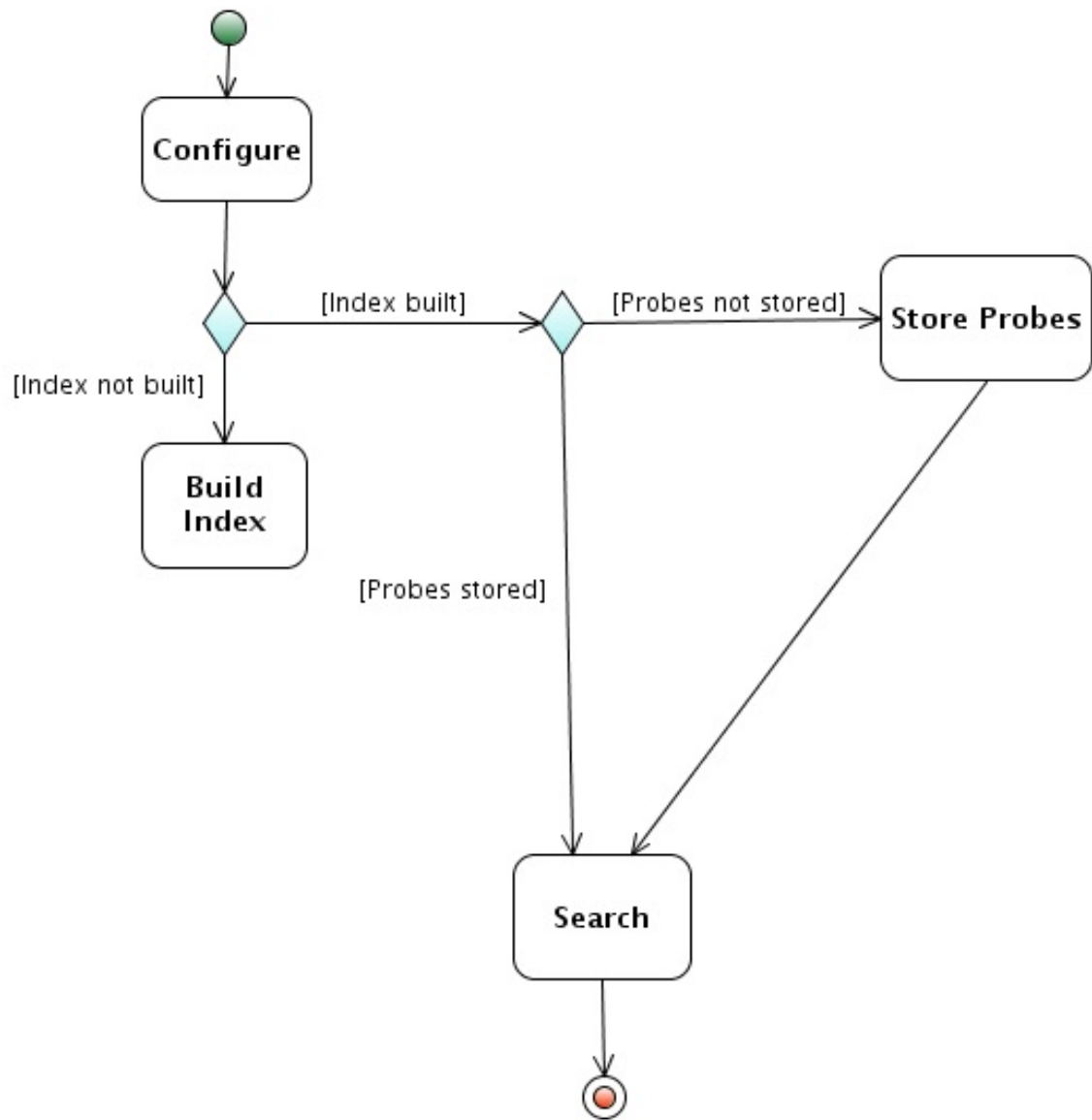


Figure 3: Activity Diagram