Usage Instructions for kClust Phameration

Charles Bowman- cab106@pitt.edu – May 7, 2015

Introduction and Installation

This document describes how to install and use the newly updated kClust Phameration script package for creation of databases. The new script is available along with several other rewritten helper scripts from a bitbucket repository at:

https://bitbucket.org/chazbot/k_phamerate

It is recommended that you obtain the scripts using Git with the following command:

```
git clone https://chazbot@bitbucket.org/chazbot/chazbot.com.git
```

After initially cloning the depository, you should check for updates by running `git pull` within directory where you initially fetched the repository into.

Script Dependencies

The scripts require the following programs, as well as MySQL and Pyhon, to be installed to function properly:

- **Phamerator -** http://phamerator.webfactional.com/PhameratorNativeInstallation.pdf
  Install the Phamerator client (or at least its required dependencies) to examine databases after creation. Most of the dependencies used by Phamerator itself are used in these scripts.
- **Legacy BLAST Executables -** ftp://ftp.ncbi.nlm.nih.gov/blast/executables/release/LATEST/
  These are installed when you launch the Phamerator client, but if you elected not to install and launch it, you should download and install these somewhere on your system.
- **kClust -** ftp://toolkit.lmb.uni-muenchen.de/pub/kClust/
  Install kClust by downloading “kClust” from the above link. Simply place this executable in a folder and link it to a bin for global execution. It is highly recommended that you read the kClust paper to understand the new process.
- **kAlign2 -** http://msa.sbc.su.se/downloads/kalign/current.tar.gz
  Compile and install kAlign2 based on the README file in the distribution. Following these instructions creates the links necessary to execute the program globally.
- **HH-suite -** ftp://toolkit.genzentrum.lmu.de/pub/HH-suite/
  Download and extract the HH-suite tools to a directory, then link them into a bin for global execution. Alternatively you can add the “bin” directory in the distribution to your PATH. In order for global execution of hhconsensus and hhmake to work, you also must add the HHLIB variable to the PATH of the user in question. You can accomplish both of these requirements by appending the following to your user’s ~/.profile

  export HHLIB=/path/to/hhsuite/base/directory/
  PATH=”$PATH:/path/to/hhsuite/bin/folder/”

These programs should be installed and globally executable via links or PATH modifications. You should be able to type the commands to launch them – kClust, kAlign, hhmake and hhconsensus – globally and be greeted by usage instructions and/or version information.

Additionally, in order to perform conserved domain searches, you should download and extract NCBI's Conserved Domain Database (CDD). It is available from NCBI at:

ftp://ftp.ncbi.nih.gov/pub/mmdb/cdd/little_endian/Cdd_LE.tar.gz

Definition of Contents

This repository contains all the scripts necessary for database creation from start to finish. Arguments for each script will be highlighted later, but are available in headers of the scripts themselves.

- **import_phage.py** – This is a simplified script for importing genomes to a database from GenBank flat text formatted files. It outputs an import log to your temporary directory detailing any issues.
- **k_phamerate.py** – This script replaces a majority of the old Phamerator workflow. It is responsible for taking all of the genes in the database, comparing them using kClust, and alignments, and then assigning phamilies. This is accomplished in three major steps: an original clustering using kClust,

alignment and consensus extraction of first iteration groupings, and then a second kClust iteration. During this process, a few extra precautions are taken to conserve phamily names and color assignments.

- **cdd_pp.py** – This script searches NCBI's Conserved Domain Database for potential matches and stores them in the database using a parallel python framework, allowing for optimal resource usage and faster searching.
- **update_host.py** – This simple script updates the "HostStrain" column of the database from a csv file containing PhageID and HostStrain on each line.
- **update_cluster.py** – This simple script updates the "cluster" column of the database from a csv file containing PhageID and cluster on each line.
- **phix_circle.py** – This script restores functionality of the Phamily Circle feature at the client level. It performs pairwise BLAST and ClustalW2 searches within a phamily and stores the results in the database for the client to access. These calculations should not be calculated server-side.
- **schema/k_blank.sql** – This is the Database Definition Language (DDL) file containing the schema for an up-to-date and fully functional database. You can use this file to create a new empty database.
- **schema/k_schema_updates** – This file contains SQL statements necessary to bring an existing database up to standards for use with the k_phamerate tools.

Database Creation Workflow:

The following steps will walk you through creation of a new database containing four bacteriophage genomes from NCBI GenBank. All steps involving MySQL can be alternatively accomplished through a GUI interface such as MySQLWorkbench or MyPHPAdmin. This example with use the empty schema in the "schema" folder and example files in the "examples directory". You will need your MySQL credentials throughout this process.

1. **Database creation**

   a) Navigate to the k_phamerate script directory using the terminal

   b) Enter the MySQL prompt at a level where you can create or modify databases:
   ```
   mysql -u root -p
   ```
   c) Create an empty database:
   ```
   create database test;
   ```
   d) Exit the MySQL prompt and load our blank template:
   ```
   exit;
   mysql test < schema/k_blank.sql
   ```

2. **Adding genomes to the database**

   a) Use import_phage.py to add the example genomes:
   ```
   python import_phage.py test Examples/MFPDB/gb/
   ```
   b) Check `/tmp/phage_import_log.txt` to ensure your phages were added properly.

   c) Use update_cluster.py to set phage cluster designation:
   ```
   ./update_cluster.py test Examples/MFPDB/test_clusters.csv
   ```
   d) Use update_host.py to set phage host designation:
   ```
   ./update_host.py test Examples/MFPDB/test_hosts.csv
   ```

3. **Perform comparative operations on the database**

   a) Use k_phamerate.py to compute phamily groupings:
   ```
   ./k_phamerate.py test
   ```

b) Change paths to rpsBLAST and CDD in cdd_pp.py to match your system (L49/50)

c) Use cdd_pp.py to add domain information to genes:

```
./cdd_pp.py test
```

4. Polish and export the database for use in clients

a) Set the version number of the database:

```
mysql -u root -p
use test;
insert into version values(1);
exit
```

b) Export the database and version tracking file:

```
mysqldump --skip-comments test > test.sql
echo 1 > test.version
```

c) Upload test.sql and test.version to your database repository or access them using your client locally.

After completing these steps, you will have created a viable Phamerator database using the new kClust oriented workflow!

Alternative Workflows:

All of the functionalities mentioned above apply not only to creating a database from scratch, but also modifying existing databases.

- **Addition of Genomes**

  To add genomes to the database, simply start from step two of the above instructions.

- **Removal of Genomes**

  To remove genomes from the database, you can either use a GUI MySQL administration program to remove genomes from the phage table, or through the command line:

  ```
  mysql -u root -p
  use DATABASE;
  REMOVE FROM `phage` WHERE `phageid` = ‘string’;
  ```

  Then continue the workflow from step three to complete the removal.

- **Replacement of Genomes**

  To replace a genome in a database, first remove it, then add its replacement and continue the workflow from step 2C.

FAQ and Tips for Database Management

- You need to perform some changes to the SQL schema of a legacy database for use with the new script, as phage deletion is now a smoother process. Refer to the file **k_schema_updates** for the relevant queries. Upon execution, you will be able to remove phages from the database using direct SQL query or an SQL editor, such as MySQLWorkbench:

  REMOVE FROM `phage` WHERE `phageid` = ‘string’;

- As the Phameration no longer requires alignment information, you can shrink your databases substantially by truncating the scores_summary table. (see **k_schema_updates** for details)
- **k_phamerate.py** runs an iteration of kClust, does alignments and extractions, then inserts new phamily designations after a second kClust iterations. Phamily names will be conserved if they exist in the database beforehand. **DO NOT INTERRUPT THE PROCESS**. If a Phameration is interrupted, restore from backup and try again. Contact us if you experience any issues!