Published in final edited form as:

Curr Protoc Mol Biol. 2009 April ; 0 19: Unit–19.7. doi:10.1002/0471142727.mb1907s86.

Using PATIMDB to Create Bacterial Transposon Insertion Mutant Libraries

Jonathan M. Urbach¹, Tao Wei², Nicole Liberati¹, Daniel Grenfell-Lee³, Jacinto Villanueva⁴, Gang Wu¹, and Frederick M. Ausubel¹

¹Massachusetts General Hospital, Boston, Massachusetts

²Eli Lilly and Company, Indianapolis, Indiana

³Microbia Precision Engineering, Lexington, Massachusetts

⁴Mariposa Bioscience Inc., Burlingame, California

Abstract

PATIMDB is a software package for facilitating the generation of transposon mutant insertion libraries. The software has two main functions: process tracking and automated sequence analysis. The process tracking function specifically includes recording the status and fates of multiwell plates and samples in various stages of library construction. Automated sequence analysis refers specifically to the pipeline of sequence analysis starting with ABI files from a sequencing facility and ending with insertion location identifications. The protocols in this unit describe installation and use of PATIMDB software.

Keywords

transposon; mutant; library; PATIMDB; software; process tracking; pipeline

INTRODUCTION

Defined bacterial transposon insertion mutant libraries have demonstrated utility in reverse genetic screening strategies (Jacobs et al., 2003; Garsin et al., 2004; Liberati et al., 2005; Cameron et al., 2008). Development of such libraries typically proceeds by way of a massively high-throughput process of transposon mutant generation and automated colony picking into microtiter plate cultures, followed by semi-arbitrary PCR, nested PCR, and sequencing of the products (Caetano-Anollés and Bassam, 1993). At the end of such workflows, there are typically tens of thousands of ABI sequence files that require automated sequence analysis for transposon insertion location identification. One such workflow (Liberati et al., 2008) has already been used to generate transposon insertion mutant libraries in *Pseudomonas aeruginosa*, *Enterococcus faecalis*, and *Vibrio cholerae* (Garsin et al., 2004; Liberati et al., 2005; Cameron et al., 2008). To simplify management of plates and automation of mutant sequence analysis, these projects made use of the PA14 Transposon Insertion Mutation Database (PATIMDB) Application. The PATIMDB Application fulfills the need for a pipeline that both tracks the processing of plates and samples and performs automated sequence analysis, yielding information about what

particular genes were interrupted in each mutant. The process tracking features of the PATIMDB are designed for use with the workflow developed by Liberati et al. (2008), illustrated schematically in Figure 19.7.1. The automated analysis functions of the PATIMDB Application are more general, and should be applicable to most projects involving the sequencing of transposon insertions.

The PATIMDB Application makes use of client-server architecture. The PATIMDB Application itself, which is intended to be installed on a user's PC or workstation, is a client application for a MySQL database server application hosted on a database server (Fig. 19.7.2). The database server hosting the PATIMDB database can be a dedicated database server or can be the workstation itself. Since the PATIMDB Application uses the open-source MySQL relational database management system, the data generated by the pipeline can be made available to other applications, including Web applications designed to query the PATIMDB database. Because of this architecture, the PATIMDB database can be extended with additional applications, and the client application can be easily updated without requiring reinstallation of the database itself.

This unit describes the installation and usage of the PATIMDB Application. Basic Protocol 1 and the support protocols describe installation of the PATIMDB Application, and are recommended for IT professionals and those researchers already comfortable with installing software and using a command line. Basic Protocol 2 describes the creation of essential database entities (including organisms, transposons, mutant libraries, genomes, and genes) and the creation of a BLAST database for the automatic data analysis pipeline. Most of Basic Protocol 2 can be performed by users without any advanced computer skills, although some researchers may require assistance from someone familiar with bioinformatics applications to create a table of genes for import into the database. Basic Protocols 3, 4, and 5 describe usage of the PATIMDB Application. These do not require any advanced computer skills and are recommended for all researchers.

This protocol is written from the perspective of researchers using the published workflow (Liberati et al., 2008) to create and sequence a transposon insertion mutant library. The portion of the workflow handled by the PATIMDB Application's process tracking features includes two PCR steps and a sequencing step, as shown in Figure 19.7.1. Researchers who use a different workflow may still benefit from the process tracking features if their workflow shares these three steps, although they may not be able to use the barcodes. In any case, most users should be able to use the automated sequence analysis features of the PATIMDB Application, no matter what workflow they use. For those researchers using different plate formats, it is possible to modify the format of each plate type in PATIMDB, for example, from 96-well format to 384-well format. For more information, see the PATIMDB documentation.

BASIC PROTOCOL 1: INSTALLING AND CONFIGURING THE PATIMDB APPLICATION AND HELPER APPLICATIONS

Materials

Hardware

Windows, Mac OS X, or Linux Computer with J2SE installed (See Support Protocol 1, 2 or 3)

Computer running the MySQL server and client applications (See Support Protocol 4, 5 or 6)

Software

PATIMDB Application, available from the Ausubel Group at Massachusetts General Hospital, http://ausubellab.mgh.harvard.edu/patimdb_application

MySQL Connector/J, Version 3.0.15 or later, available from MySQL AB, <http://dev.mysql.com/downloads/connector/j/>

Phred, version 0.000925.c or later, available from the University of Washington (<http://www.phrap.org>) for academic licenses and from the Codon Code Corporation (<http://www.phrap.com>) for commercial licenses

NCBI BLAST, version 2.2.14 or later, available from the NCBI, <ftp://ftp.ncbi.nih.gov/blast/executables/LATEST> (NCBI BLAST packages include the formatdb application)

Obtain and install the PATIMDB application

For convenience, the PATIMDB Application is distributed as a .zip archive (for Windows) or a .tgz archive (for Mac OS X, Linux, and other Unix-like operating systems). These archives are named with the format patimdb_YYYYMMDD.zip and patimdb_YYYYMMDD.tgz, respectively, where YYYYMMDD is the date of the release. In actuality, the contents of the .zip and .tgz packages are identical, and both formats can be used on any of the supported platforms if appropriate archive-extraction utilities are available. On Windows, use any of the ubiquitous Zip compression utilities to extract the .zip file. To extract the archives on Mac OS X, Linux, or other Unix-like operating systems, use the following commands for .tgz or .zip archives respectively, substituting the actual name of the archive:

```
tar zxvf patimdb_YYYYMMDD.tgz
or
unzip patimdb_YYYYMMDD.zip
```

The archives will extract to a distribution directory with the name patimdb_YYYYMMDD (where YYYYMMDD is the release date). Within the archives are the start scripts (patimdb.sh and patimdb.bat) and several subdirectories:

patimdb_YYYYMMDD/doc: contains PATIMDB documentation
patimdb_YYYYMMDD/MySQL: contains the PATIMDB database creation script
patimdb_YYYYMMDD/lib: where the MySQL Connector/J jar file is to be installed
patimdb_YYYYMMDD/apps: recommended location of the helper applications phred, blastall, and formatdb
patimdb_YYYYMMDD/blastdb: recommended location of BLAST databases
patimdb_YYYYMMDD/test: location of testing/example files
patimdb_YYYYMMDD/scripts: location of helper and utility scripts

NOTE: The PATIMDB Application documentation is provided in HTML format, and can be accessed within a Web browser through the file patimdb_YYYYMMDD/doc/index.html, or through the help features of the PATIMDB Application itself.

Obtain Phred

The PATIMDB Application uses Phred to perform base-calling on ABI-format sequence chromatograms. Phred executables for Windows, Mac OS X, and Linux are available from Codon Code Corporation, <http://www.phrap.com>. Free limited licenses for Windows and Mac OS X are available to academic users (see <http://www.phrap.com/priceinfo.htm>), but commercial institutions must purchase a license to use Phred and Phrap (Phrap is a sequence assembly program designed to be used in tandem with Phred in genome sequencing projects). Academic Linux users should obtain the Phred source code from the authors, Phil Green and Brent Ewing, at the University of Washington (<http://www.phrap.org/>). It can take a few days or more for the license agreement to be processed by the University of Washington. In any case, procuring Phred should be the first order of business when installing the PATIMDB Application and its helper applications.

Install Phred executable on Windows and Mac OS X

Follow the directions to install Phred under Windows or Mac OS X. The Phred executable may be installed as recommended by the distributor or, for convenience (or if you do not have root or administrator privilege to install software), the Phred executable (phred.exe on Windows) and associated phredpar.dat file can be installed within the apps subdirectory of your PATIMDB distribution. You may optionally add the installation directory to your PATH environment variable, and set the PHRED_PARAMETER_FILE environment variable as directed in the associated Phred documentation, but these PATH variables can also be set when you configure the PATIMDB Application (see Configure the Paths of Helper Applications).

Install Phred executable on Linux

Compiling the Phred executable on Linux or other Unix-like operating systems requires a C compiler, preferably GCC, but is otherwise straightforward (see operating system documentation for more information on the GCC compiler). The source code will be e-mailed to you as a compressed tar archive of files without an enclosing directory. This tar

archive will be named with the format phred-dist-VERSION.tar.Z. For tidiness, create a compilation directory for the source files, substituting the appropriate version:

```
mkdir phred-dist-VERSION
```

Change the default directory to your compilation directory and uncompress the source archive, substituting the path to your phred distribution tar archive for /PATH/TO/phred-dist-VERSION.tar.Z:

```
cd phred-dist-VERSION
tar zxvf /PATH/TO/phred-dist-VERSION.tar.Z
```

The INSTALL file contains directions for compilation. You can, if necessary, edit the Makefile, but on most Linux distributions, building requires only a single command:

```
make
```

If compilation is successful, you will end up with an executable named phred in the compilation directory. You may optionally copy the phred executable and the phredpar.dat file to the patimdb_YYYYMMDD/apps subdirectory. Otherwise you can copy the phred executable to a directory in your path, such as /usr/local/bin, and set the PHRED_PARAMETER_FILE environment variable as directed in the associated phred INSTALL file. The PATH to the phred executable and the phredpar.dat file can also be set when you configure the PATIMDB Application.

Install BLAST

Download the BLAST package from NCBI appropriate for your platform and operating system from <ftp://ftp.ncbi.nih.gov/blast/executables/LATEST>. For Mac OS X and Linux, BLAST is distributed as a compressed tar archive with a name in the format blast-VERSION-ARCHITECTURE-OS.tar.gz, and can be extracted with the following command, substituting the actual name of your downloaded BLAST distribution:

```
tar zxvf blast-VERSION-ARCHITECTURE-OS.tar.gz
```

This command will extract the BLAST distribution as a directory named blast-VERSION. Within this directory is a bin subdirectory containing the blastall and formatdb executables. For convenience, the blastall and formatdb executables may be copied into the patimdb_YYYYMMDD/apps subdirectory.

The Windows distribution of BLAST is a self-extracting archive with a name in the format blast-VERSION-ARCHITECTURE-OS.exe. Prior to extraction, it should be placed in its own folder (which you can call blast-VERSION). To extract, either double-click the self-

extracting archive, or run it on the command line. It will extract the BLAST executables, documentation, and data files into the blast-VERSION directory. The blastall.exe and formatdb.exe applications can then optionally be copied to the patimdb_YYYYMMDD/apps subdirectory.

Install MySQL Connector/J

Download MySQL Connector/J from MySQL's Web site <http://dev.mysql.com/downloads/connector/j/>, either as a .zip file with the name format mysql-connector-java-VERSION.zip (best for Windows) or as a tar.gz file with a name of the format mysql-connector-java-VERSION.tar.gz (best for Mac OS X, and Linux). Extract these archives with your preferred extraction utility. Within the extracted directory, there is a .jar file with the name format mysql-connector-java-VERSION-bin.jar. Place this .jar file in the patimdb_YYYYMMDD/lib subdirectory.

Create the empty PATIMDB database

This part of the installation process must be performed by someone with administrative (root) privileges on the MySQL database server. When in doubt, consult an IT professional.

The creation script for the PATIMDB database is located in the MySQL directory, and named create_PATIMDB.sql. If you are running the MySQL server program on a computer other than the local workstation where you are installing the PATIMDB Application, and you have no locally installed MySQL client program with root access privileges, you will have to transfer the create_PATIMDB.sql script to the server using ftp, sftp, or another file transfer program.

For the next steps, you need to choose a name for your database, a database user name, and a database password. For the purposes of this protocol, we will simply use DBNAME, DBUSER, and DBPASSWORD respectively for these. You will also need to know the IP address or hostname of the database server, and of the client workstation where the PATIMDB program is installed. For this protocol, we will use DBSERVER and DBCLIENT, respectively. Note that if DBSERVER and DBCLIENT are the same computer, you should use either localhost as the hostname or 127.0.0.1 as the address for that computer.

On the MySQL server computer, launch the command-line MySQL client as the root user:

```
mysql -h DBSERVER -u root -p
```

You will be prompted for the root password. Type it in it and press Enter. Create a user with full privileges for the PATIMDB database who can access that database from the client computer with the following command:

```
GRANT ALL ON DBNAME.* TO DBUSER@DBCLIENT IDENTIFIED BY  
'DBPASSWORD' ;
```

If this command worked, you should see a message such as Query OK, 0 rows affected (0.00 sec). To prompt the MySQL server to reread the user tables, and activate the changes to database privileges, issue the following command:

```
FLUSH PRIVILEGES;
```

At this point, you may create the DBNAME database and tables by issuing the following commands, substituting the actual database name for DBNAME and the actual path the create_PATIMDB.sql script:

```
CREATE DATABASE DBNAME;  
USE DBNAME;  
SOURCE /PATH/TO/create_PATIMDB.sql
```

If table creation worked, this command will produce a series of Query OK messages and no error messages. For a list of tables that have been created, issue the following command:

```
SHOW TABLES;
```

This should produce a list of 25 table names:

```
+-----+  
| Tables_in_DBNAME |  
+-----+  
| BlastDB           |  
| Gene              |  
| GeneAnnotation    |  
| GenomeSeq         |  
| InsertGenome      |  
| Location          |  
| Mutant            |  
| MutantLibrary     |  
| Plate             |  
| PlateParentChildLink |  
| PlateSample       |  
| PlateType         |  
| PositionGenomeGeneLink |  
| ProcessExecution  |  
| ProcessPlateLink  |  
| ProcessSeq        |  
| Protein           |  
| Protocol          |
```

```

| RawSeq          |
| RawSeqData      |
| Result          |
| SampleLineage   |
| Scientist        |
| Species          |
| Transposon       |
+-----+
25 rows in set (0.00 sec)

```

NOTE: On Windows and Mac OS X, these table names may be in all lower-case.

At this point you are ready to attempt to launch the PATIMDB Application.

Run and configure the PATIMDB application

The PATIMDB Application can be launched by invoking its start script on the command line from within its distribution directory:

```
cd patimdb_YYYYMMDD
```

On Linux and Mac OS X, the script to use is `patimdb.sh`. Make sure this file is executable by executing the following command:

```
chmod +x patimdb.sh
```

To run the script, execute:

```
./patimdb.sh
```

On Linux, this script can also be run by simply double-clicking it in your file manager, and then selecting Run In Terminal. In Windows, the script is called `patimdb.bat` which can be executed by typing its name in the Command Prompt window obtained via the Start menu, or by double clicking. If you get one of the following messages, you need to make sure that the Java executable is located in your PATH:

```

./patimdb.sh: line 2: java: command not found
or
'java' is not recognized as an internal or external
command, operable program or batch file.

```

The first time that the PATIMDB Application is launched, a dialog box will be displayed with the message “Alert: No pa.properties configuration file was detected. Do you want to

configure?” (Fig. 19.7.3). Click the OK button, and the PATIMDB Application Configuration window will be displayed (Fig. 19.7.4). In this window, you will enter five parameters for setting up the connection to the PATIMDB MySQL database. The first parameter is the Java class name of the database driver. Since we are using MySQL Connector/J, the default value, `com.mysql.jdbc.Driver`, is correct. In the URL box, enter the URL `jdbc:mysql://DBSERVER/DBNAME`, substituting your actual database server hostname or IP address, and database name for *DBSERVER* and *DBNAME*, respectively. In the UserID box, enter your database user ID (we are using *DBUSER* from above), and, under password, enter your database password (we are using *DBPASSWORD*). In the Connection Options box, we recommend `?autoReconnect=true` if you are using MySQL Connector/J versions earlier than version 5. (The preappended `?` is required. The `autoReconnect=true` option is specific to the MySQL Connector/J driver, and is being discontinued, so in later versions it will not be necessary. Check the documentation for your version of MySQL Connector/J for more information.) To save these values, click the Save Options button and then the Close button. If there is a problem with the connection to the database, you will have the opportunity to re-enter the configuration interface.

If the database connection is successful, the PATIMDB Application will display a dialog box with the message No Users in Database. Be sure to add some. Click OK to close the dialog box, and the PATIMDB Main Window will open (Fig. 19.7.5), along with the User Manager (Fig. 19.7.6). In order to enable password protection of the PATIMDB Application, you will need to create one or more users with Owner privileges. Until such a user is created, the application will launch as “owner,” meaning that anyone who runs the program will have full access to alter the database. Note that only rudimentary security is provided by password protection; there are back doors to the database! The first user you create should probably be yourself. To do this, enter a user name, a login name (which you will type at the login screen), and a password, and set the privilege of this first user to Owner. If you like, you may also enter your address and personal contact information. Click the Save button to write your user data to the database. You can then use the New button to create more users, or just close the window. For more information about adding users, click the Help button or see the PATIMDB Application documentation. The next time you start the PATIMDB Application, you will be prompted for your login (not your user name) and password (Fig. 19.7.7).

Configure paths of helper applications

In order to enable automated sequence analysis, the installed helper applications and the `PHRED_PARAMETER_FILE` must be made available to the PATIMDB Application. The three applications, `blastall`, `formatdb`, and `phred`, may be made available to the PATIMDB Application by placing them in one of the locations defined in the operating system’s `PATH` environment variable, or by explicitly specifying their locations for the PATIMDB Application. Likewise, the location of the `PHRED_PARAMETER_FILE` can be determined by defining the `PHRED_PARAMETER_FILE` environment variable, or else must be explicitly configured. To explicitly configure the paths of the helper applications and the `PHRED_PARAMETER_FILE`, select Options > Analysis Settings to open the Analysis Settings window (Fig. 19.7.8).

By default, the BLAST, formatdb, and phred executables are set as the unqualified executable names, blastall, formatdb, and phred, without an associated path. The PATIMDB Application can execute unqualified executables if they are located in the default executable path specified by the PATH environment variable. To determine whether or not blastall, formatdb, and phred are available in your default path, execute these programs in a terminal window or command prompt without any arguments. Executing the phred command without any arguments should produce the message no input files specified, if phred is available in your path. Likewise, executing the command blastall on the command line should print more than 100 lines of an explanation of the arguments of the blastall command. Executing formatdb should give the message [formatdb] ERROR: No database name was specified. If these programs are not available in your executable path, you will most likely either get the message command not found or, on Windows, something along the lines of 'phred' is not recognized as an internal or external command, operable program or batch file. If these executables are found in your path, the default unqualified executable names in the Blast Executable, Formatdb Executable, and Phred Executable boxes, blastall, formatdb, and phrap should work just fine and not require modification. Otherwise, click the Browse boxes to the left of these settings, and browse to the correct executable file.

If you have defined the PHRED_PARAMETER_FILE environment variable, you should see the correct location of the phredpar.dat listed in the Phred Parameter File text box. To be sure, you should check to make sure that the phredpar.dat file is indeed in the indicated location. If there is no listed phredpar.dat file, click the Phred Parameter File Browse button to select the actual location.

Once the paths to blastall, formatdb, phred, and phredpar.dat have been properly set, clicking the Apply Changes button saves this path information. For more information on the various configurable parameters in the Analysis Settings window, click the Help button, or see the PATIMDB Application documentation. Click the Close button to close the window.

Selecting PATIMDB Application Main Window > File > Exit quits the PATIMDB Application.

BASIC PROTOCOL 2: MUTANT LIBRARY SETUP

Materials

Hardware

Windows, Mac OS X, or Linux Computer with the PATIMDB installed (Basic Protocol 1)

Computer hosting the PATIMDB Database

Data

Genome sequence of your transposon insertion library host organism in GenBank (preferred) or FASTA format

Gene descriptions in GenBank or tabular format (see below)

15- to 30-base segment of the transposon sequence that abuts the transposon junction and can be detected in the sequencing reads

Prepare your database for library production

Prior to creation of libraries, various database entities need to be defined including the host organism, the transposon, the mutant library, the genome sequence (or sequences), and genes. Furthermore, a BLAST database needs to be exported and formatted. Figure 19.7.9 illustrates the dependencies and relationships between the various entities in the PATIMDB database. For example, to define a mutant library, you must already have defined a host organism and a transposon, and to export a BLAST database, you must have already defined a genome sequence. Therefore, the following steps should be executed in the recommended order.

Define an organism

Organisms are specified in the PATIMDB with the attributes Genus, Species, and Strain. For the purposes of the PATIMDB, the Strain is used as the unique identifier of an organism, and is typically a short string of about four characters. To define an organism, select PATIMDB Application Main Window > Manage > Organisms. The Organism Manager window will be displayed (Fig. 19.7.10). Click the New button, and enter appropriate values for Genus, Species, and Strain. Click the Save button to store the organism in the database, then select Close.

NOTE: For the purpose of generating human-readable barcodes for plates, it is advantageous to use a strain identifier that is four characters or less. For more information on defining organisms, click the Help button.

Define a transposon

In addition to being a prerequisite for defining mutant libraries, the definition of a transposon provides the PATIMDB Application with a way to search for the edge of a transposon in raw sequence reads, thereby allowing the PATIMDB Application to determine precisely the genome locations of transposon insertions. To define a transposon, first open the Transposon Manager Window by selecting PATIMDB Application Main Window > Manage > Transposons (Fig. 19.7.11). Click the New button and enter the required Transposon ID, TnSearchSequence, MaxLevenshteinDistance, TnMaxPosition, and Default Offset in the appropriate text fields. The Transposon ID should be a short, preferably four-character identifier. Note that for the purpose of generating human-readable barcodes for plates, it is advantageous to use a strain identifier that is four characters or less. The TnSearchSequence is a string representing the 15 to 30 bases of sequence adjacent to the transposon insertion junction that can be identified in sequence reads. The PATIMDB Application searches the raw sequence read for the best possible match, or partial match to the TnSearchSequence, allowing a certain number of mismatches or gaps. Matches are scored by Levenshtein distance, the number of mismatches plus gap bases. Since the PATIMDB Application scores by Levenshtein distance, and lower scores are preferred, extra unmatchable bases only count against the match. Therefore, do not use a longer sequence than can generally be identified in the sequence under standard sequencing

conditions! The TnMaxLevenshteinDistance parameter is the maximum Levenshtein distance allowed when searching for the TnSearchSequence in the sequence read, and is optimally no more than 1/3 the length of the TnSearchSequence. This value determines the sensitivity of the transposon sequence search. If TnMaxLevenshteinDistance is too high, false transposon sequence matches may be detected in the raw sequence reads. If it is too low, then true matches may be missed. TnMaxPosition tells the PATIMDB Application how far into the sequence read to search for the (TnSearchSequence) transposon sequence. Appropriate values for these three parameters, TnSearchSequence, TnMaxLevenshteinDistance, and TnMaxPosition should be decided after inspection of actual sequence data from putative transposon insertion mutants. The PATIMDB Application documentation describes in detail how to pick parameters for transposon sequence matching. To commit additions and edits of transposons to the database, click the Save button. For more information on Transposons, click the Help button, or see the PATIMDB Application documentation.

Define a mutant library

To define mutant libraries, select PATIMDB Application Main Window > Manage > Libraries (Fig. 19.7.12). The Library Manager will be displayed. Click the New button and enter a Library ID, a Creation Date, and, from the drop-down menus, select a Library Creator, Transposon, and Host Organism. Optionally, you can select a “preparation protocol.” To save additions and changes to the Mutant Library, click the Save button. For more information, click the Help button or see the PATIMDB Application documentation.

Import a genome sequence

The PATIMDB Application is capable of importing Genome sequences from GenBank and FASTA format files. Suitable GenBank files for many bacterial genomes are available from the NCBI FTP site at <ftp://ftp.ncbi.nih.gov/genomes/Bacteria>. Note that GenBank Web pages containing embedded HTML tags are not supported. GenBank files can be downloaded from NCBI GenBank Web pages, though this may not work in all cases. See Question 6 in the FAQ section of the PATIMDB application for more information on how to do this. If no GenBank file is available for your genome, you can also import genome sequences in FASTA format. GenBank files are preferable to FASTA, because a single GenBank file can be used for both Genome Sequence and Gene import. Otherwise, a FASTA genome sequence file is needed, and a separate GFF, GTF, or PATIMDB TSV file is required for Gene import (see Importing Gene Records, below).

To import Genome Sequences, open the Genome Sequence Manager by selecting PATIMDB Application Main Window > Manage > Genome Sequences (Fig. 19.7.13). To create a new genome sequence, click the New button. In the Strain drop-down menu, select the appropriate strain, then, next to the Sequence File box, click the Browse button. A file browser will open. Use this to browse to the genome sequence GenBank or FASTA file. After clicking Select Genome Sequence File to close the browser, make sure the correct file type is selected in the Sequence File Type window, either GenBank or FASTA. To commit changes and additions, click the Save button. The genome sequence will be imported into the database by the PATIMDB Application, and a progress bar will be shown, indicating its

progress. If you are importing a genome from a GenBank file, you will then be prompted with a dialog Do you wish to import genes from the GenBank file? (see below). It is recommended that you click YES. Otherwise, when export is completed, click the Close button to close the progress bar. Note that if you get an error relating to packet size, you need to increase your maximum allowed packet size (see Support Protocol 4, 5, or 6). For more information on defining genome sequences, click the Help button or see the PATIMDB Application documentation.

Import gene records

The Gene Manager supports import of gene records from standard, publicly available file formats, including GenBank, GTF, and GFF files, as well as the PATIMDB Application's native TSV (tab-separated values) format, described below. Suitable GenBank and GFF files for many bacterial genomes can be obtained from the NCBI FTP site at <ftp://ftp.ncbi.nih.gov/genomes/Bacteria>, and GTF files are available from the Web sites of numerous other bacterial genome sequencing projects. If these file formats are not available for the genome being used, the Gene Manager can import a table of genes in the PATIMDB TSV format into the PATIMDB database.

The PATIMDB TSV format

The PATIMDB database contains a Gene table listing the start and stop coordinates of specific genes in a particular genome sequence. Importing gene records into the database to populate the Gene table requires that the corresponding genome sequence must already have been imported in the database. The gene table allows the PATIMDB to identify genes that have been interrupted by a transposon insertion. The PATIMDB TSV format contains this same information in what is essentially a spreadsheet format, something like a simplified Excel file. In fact, TSV format can generally be exported from spreadsheet applications, and many genome sequencing projects have made their gene coordinates available as TSV files. The various fields in the TSV file are identified by their field names, which are on the top line of the TSV file. The following fields are required:

GeneID: The unique primary key identifier of the gene.

Strand: Either + or – corresponding to the top and bottom strand of the genome, respectively.

Start: The numerical start coordinate of the gene, the lesser of either the first or last base of the gene.

Stop: The numerical stop coordinate of the gene, the greater of either the first or last base of the gene.

Seq: The sequence of the gene.

Fields must be named precisely as shown. Other optional fields may also be included. For more information, click the Help button in Gene Manager, or check the PATIMDB Application documentation.

The Gene Manager

If you have just imported a genome sequence from a GenBank file, by answering Yes when prompted to import genes as well, then the Gene Manager will open with the appropriate Genome Sequence ID and source GenBank file in the Genes File text field. Otherwise, to open the Gene Manager, select PATIMDB Application Main Window > Manage > Genes (Fig. 19.7.14). To import gene records, make sure the appropriate Genome Sequence ID (GenomeSeqID) is selected in the corresponding drop-down menu. To specify a genes file, click the Browse button and use the file browser to select the appropriate genes file. Make sure that the appropriate Import File Type is selected. To proceed with the import, click the Import Genes button. A progress bar will be displayed as the file is parsed and the genes are imported. If you are importing a GenBank, GFF, or GTF file, an import dialog will appear prompting you to select what type of features you wish to import, and what unique identifier to use (Fig. 19.7.15). We recommend selecting feature type “gene” and identifier type “locus_tag”. Note that we do not recommend importing additional feature types, since the corresponding data is generally redundant. For more information on this, see the PATIMDB Application documentation. When gene import is completed, click the Close button. For more information on importing genes, click the Help button or see the PATIMDB documentation.

Prepare a BLAST database for automated analysis

To start the BLAST Database Manager, select PATIMDB Application Main Window > Manage > BLAST Database (Fig. 19.7.16). In the BLAST Database Manager window, you will see a small table of genome sequences present in the database. In the Export box, check all of the genome sequences that you want to use for BLAST analysis of the sequence reads. Next, enter the name of your BLAST database into the Genome Blast Database box. The default value of patim blastdb.tfa is fine, or you can browse to the patimdb YYYYMMDD/blastdb directory and enter a more descriptive name. Note that, due to an issue with formatdb and blastall, the path of the BLAST database may not have a space in it. This creates problems especially for Windows users whose default directories are located within C:\Documents and Settings\. The best solution is to simply put the BLAST database in a location that will have no spaces in its pathname, e.g., C:\blastdb\patimdb blastdb.tfa. Another work-around for this problem is to specify the path to the BLAST database with a path relative to the installation directory. Click the Export button to proceed with the BLAST database export. A progress bar will appear, and when the export is completed, click the Close button. For more information about setting up the BLAST database, click the Help button or see the PATIMDB Application documentation.

BASIC PROTOCOL 3: CREATING PLATE SETS AND PERFORMING PROCESSES WITH THE PATIMDB APPLICATION

The Plate Set Creator allows the user to create virtual plates, mutants, and samples in the database. In doing so, the user specifies that a given library has a given set of associated plates, mutants, and samples. By default, each plate set contains a 96-well working plate (WOR), a 96-well supernatant plate (SUP), a 96-well PCR1 plate (AR1), a 96-well PCR2 plate (AR2), and a 96-well sequencing (SEQ) plate, as well as a 384-well master plate

(MAS) and a 384-well duplicate (DUP) plate. Plate sets are identifiable by their associated LabelIDs that are part of a plate's human-readable label (see below). Because up to four 96-well format plates may exist for each 384-well format plate, 384-well plate wells are mapped to up to four 96-well plates, and therefore can be shared by up to four plate sets. The LabelID numbering of the 384-well format plates will only use every fourth number, or more precisely, the first of each set of four numbers. Table 19.7.1 shows an example of virtual plate sets created using strain PA14 and transposon PhoA. Because of the 4:1 relationship of 96-well plates to 384-well format plates, we recommend that the total number of plate sets created be divisible by 4, or else some wells in the 384-well plates will be unused. It is possible to change the default formats for each of the plate types. If the plate types are set to be all 96-well format, or all 384-well format, the plate LabelID number-skipping behavior will no longer occur. See the Frequently Asked Questions section of the PATIMDB Application documentation for more information on changing the format of a given plate type.

To create a plate set, select PATIMDB Application Main Window > Execute > Create Plate Set (Fig. 19.7.17) to open the Plate Set Creator window. In the Number of Plate Sets box, enter the number of plate sets you wish to create, and then in the Mutant Library pull-down, enter the Mutant Library for which you are creating the plates. Optionally, you may specify a storage location with the drop-down menu. In the Scientist drop-down menu, select the plate set's creator, probably yourself, and, lastly, select a destination for the barcode files that will be automatically generated, and which will provide a way of labeling your plates. Lastly, click the Create Plate Set button to create your plates. When the Plate Creator window reports Status: DONE, you can close the window. Barcode files will be created in the specified location with the following name format:

[SCIENTIST NAME] [YYYY-MM-DD] [PLATE TYPE].barcode

The human-readable names of the plates, stored in the barcode files will be in the format:

[STRAIN] [TRANSPOSON] [LABELID] [TYPE]

In these barcode files, these labels will alternate with numerical PlateIDs. For more information about plate creation, click the Help button, or see the PATIMDB Application documentation.

Process tracking is performed through the Process Execution window. This interface allows the user to record that a particular process has been performed for a particular plate. The current supported processes are:

PCR1 protocol: SUP plate → AR1 plate

PCR2 protocol: AR1 plate → AR2 plate

Sequencing: AR2 plate → SEQ plate

NOTE: These processes are indicated with bold arrows in Figure 19.7.1.

To execute a process, open the Process Execution Window by selecting PATIMDB Application Main Window > Execute > Execute A Process (Fig. 19.7.18). Select a Process Type from the Process Type drop-down, either PCR1, PCR2, or Sequencing. A list of one or more Protocol IDs will appear in the Protocol ID drop-down menu. You can use the default, or if you have defined more than one protocol for a given process type, pick the one you want. Select the investigator performing the process in the Scientist drop-down, and select the input plate from the list (input plates must be available for this to happen). The corresponding output plate will appear in the Output Plate window. To execute the process, click the Execute Process button. For more information about process execution, click the Help button or see the PATIMDB documentation.

It is possible to repeat process execution, if necessary. For more information, see the PATIMDB Application documentation.

BASIC PROTOCOL 4: PERFORMING AUTOMATED SEQUENCE ANALYSIS WITH THE PATIMDB

When you have received a set of ABI files from a sequencing facility corresponding to a particular sequencing plate, you are ready to perform automated sequence analysis. Select Application Main Window > Execute > Analyze Sequences to open the Sequence Analysis window (Fig. 19.7.19). In order to maintain the correct association of sequence files to ABI files, files must be stored with a particular arrangement of subdirectories and named according to a particular set of conventions.

Sequences must be stored with the following directory structure:

```
[STRAIN]/[TRANSPOSON]/[LABELID]v[VERSION]
```

For example, PA14/PhoA/1v1 is a valid directory structure given STRAIN PA14, TRANSPOSON PhoA, LABELID 1, and VERSION 1. Sequencing files should be named with the convention:

```
[LABELID]v[VERSION] [WELL] [ARBITRARY].ab1
```

In this case, wells are named in the format LETTERNUMBER (e.g., A01, H12), and ARBITRARY is an arbitrary sequence at the discretion of the user or sequencing facility. For example, the following are valid sequence file names:

```
1v1 A01 a1 001.ab1
```



```
7v1 G09 jonathan.ab1
31v2 D12 1223.ab1
```

If your sequencing facility does not support this file-naming convention, you will have to rename the files returned by the facility. This can easily be accomplished using a Perl script. See the Frequently Asked Questions section of the PATIMDB Application documentation for an example of a Perl script to perform this task.

To perform automatic sequence analysis, click the Browse button to select the desired Sequence Source Directory. From the pull-down menu, select as the source type ABI Files. [If you have already performed Phred and your sequence file directory contains properly named sequence and Phred files, you can select DNA Sequence Files (FASTA + QUAL) as the source type. Make sure you have both FASTA and QUAL files in the directory, or you will get an error.] To begin automated sequence analysis, click the Start Analysis button. The PATIMDB Application will then execute its automated sequence analysis pipeline (Fig. 19.7.20), performing base-calling, quality assessment, sequence trimming, transposon sequence matching, BLAST against the genome, and insertion site determination. When automated analysis is completed, the Sequence Analysis window will report Done. For more information, click the Help button in the Sequence Analysis window, or see the PATIMDB Application documentation.

It is possible to repeat automated sequence analysis. This feature is particularly useful when the BLAST database or gene set has been updated. For more information about this, select Application Main Window > Execute > Analyze Sequences and click the Help button, or see the PATIMDB Application documentation (Fig. 19.7.21).

BASIC PROTOCOL 5: RETRIEVING PLATE AND MUTANT REPORTS

Select Application Main Window > Retrieve > Plate Records to open the Plate Selector (Fig. 19.7.22). Use the Plate Type pull-down to select the plate types you want to review. When you select a plate type, the PlateID/Label pull-down will refresh. Select the plate you wish to examine, and click the Retrieve button to retrieve the corresponding plate report.

The PATIMDB Application will display a Report for Plate (Fig. 19.7.23) clickable plate map. Wells are labeled with their respective MutantIDs. Red wells are those that have an associated processed sequence, meaning that their reads had some good-quality sequence after trimming, whereas white wells had no or insufficient good quality sequence. Clicking on a red well opens a Mutant Report (or if there is more than one associated read for a mutant, clicking on a well will open a selection box). Below the plate map, there is a text area containing information about the selected plate including the Label, PlateID, Library, Creator, and Creation Date.

A mutant report (Fig. 19.7.24) is divided into two main sections. The upper section is a graphical sequence read map of a sequence read, the associated transposon sequence, and the stored BLAST hits (InsertGenome records). The trimmed sequence is represented as a gray bar, and the edge of the transposon is represented as a pointed violet region on the left side. Below the read is drawn a scale (calibrated in bases), and below that are the significant

BLAST alignments, represented as a series of flat-headed red lines. These BLAST alignments correspond to portions of the genome sequence that align with the read. Each BLAST alignment has a corresponding InsertGenomeID that corresponds to InsertGenome records in the text report section (see below).

The text report section is broken down into four subsections. The top subsection contains mutant information. The first bit of data is ABI file from which the sequence is derived. This is followed by the ProcessSeqID, the RawSeqID, and the MutantID, unique identifiers of the trimmed sequence, the raw sequence read, and the mutant, respectively. The LabelID and Library also included in this section identify the particular plate set and library to which the mutant belongs. The Background and the Transposon identify the background strain and transposon, respectively, used for mutant creation. Lastly, the TnLastBase, also known as the transposon last base, identifies the position of the last transposon base in the raw sequence read.

The second subsection of the text report contains various quality assessment metrics of the raw sequence, and, by extension, the insertion location determination. Transposon Found is a boolean indicating whether the PATIMDB Application was able to identify the edge of the transposon in the raw sequence read (optimally, this value is TRUE). Possible Contamination is a boolean indicating that BLAST results suggest multiple PCR products that appear to map to more than one location in the genome. Optimally, this value is FALSE. Ambiguous Location indicates that a sequence may map to more than one location because of identical BLAST scores rather than (necessarily) multiple PCR products. Optimally, this value is FALSE. Average Quality refers to the average Phred score of the raw sequence read. Higher values are preferred, although even sequences with low average quality can have segments of good sequence. Bases Above Quality Threshold is the number of bases in the Raw Sequence exceeding the QA Quality Threshold parameter (see the PATIMDB Application documentation on Analysis Settings for more information). This is a measure of the total amount of good quality sequence in the read. Total Hits is the number of significant recorded BLAST alignments.

The third section of the text report contains the processed (trimmed) sequence in FASTA format.

The fourth section of the text report contains a table listing significant BLAST alignments. InsertGenomeIDs are the unique identifiers of each BLAST hit, and are the same numbers represented in the sequence read map, above. The best hit for a given bacterial strain is given a star to indicate that its chromosomal position mapping is most likely to be correct. Strain Hit Rank refers to the ranking of the BLAST hit for a given strain (this is useful when the BLAST database contains multiple strains for mapping). Subject Genome/Strain refers to the GenomeSeqID and strain. While most strains will only have a single genome sequence, others may have more than one if they have more than one chromosome (GenomeSeqID might more properly be called ChromosomeSeqID). The BLAST Bit Score parameter is self-explanatory. Strain Score Separation is the alignment's bit score minus the bit score of the next best alignment, and is therefore a measure of hit uniqueness and ambiguity, especially for the top ranked score. Query Start and Query End are the

coordinates of the first and last bases respectively of the query sequence that align with the subject sequence. Subject Start and Subject End are the coordinates of those bases in the subject sequence that aligned with the Query Start and the Query End, respectively. Optimal Insertion Location is the insertion location as determined by the PATIMDB Application, and Disrupted Genes (Relative Tn Insertion Orientation) indicates those genes disrupted by transposon insertion, with the relative orientation of the read and the transposon insertion in parentheses (+ for collinear, – otherwise). For more detailed information on the Mutant Text Reports, click the Help button, or see the PATIMDB Application documentation.

Select Application Main Window > Retrieve > Mutant Records to open the Mutant Selector. The mutant selector allows users to directly retrieve mutants by MutantID (Fig. 19.7.6). To retrieve a report for a given mutant, enter the MutantID in the indicated box and hit the Retrieve button. If there is data for that mutant, a Mutant Report will be displayed (or if there is more than one associated read for a mutant, a selection box will be displayed). For more information, click the Help button, or see the PATIMDB Application documentation.

SUPPORT PROTOCOL 1: INSTALLING JAVA 2 PLATFORM STANDARD EDITION ON WINDOWS COMPUTERS

Materials

Hardware

A computer running Windows 2000, XP, or Vista

Software

Java 2 Platform Standard Edition (J2SE), version 5.0 or later, available from <http://java.sun.com/javase>

To install the J2SE on Windows computers, you will need to be logged onto your Windows workstation as Administrator. Download the appropriate version of the J2SE for your architecture, i586 or x64. Save the file, which will have a name such as jre-VERSION-windows-ARCHITECTURE.exe, on your hard drive. The Java runtime is available as part of other packages on Sun's Web site, including the Java Development Kit (JDK). Although these packages also fulfill the requirement for Java, most users will only need the J2SE package. Double-click this file to begin the install process. You will be presented with a Welcome screen, and you will be asked to decline or accept Sun's license agreement. Before you click Accept, check the "Change destination folder" box, so that you will have the option of selecting a destination folder in the next screen. Next, click the Accept button. In the Destination Folder screen, you will have the opportunity to be able to browse for a new destination folder. In any case, note the destination folder, most likely something like C:\Program Files\Java\jre6\). Once you have done so, click Next. Installation will proceed. Once installation is completed, click Finish. Java is now installed.

The destination directory of the J2SE will contain a subdirectory named bin where the java.exe application is located. To add this directory to your PATH environment variable, first go into the Windows Control Panel and double-click System. In the System Properties

item under Control Panel, click the Advanced tab, then Environment Variables. Under System Variables, select PATH and click Edit. In the Edit System Variable window, in the “Variable value” box, append the path to this J2SE bin subdirectory. Individual items in the Windows PATH environment variable must be separated by semicolons, so the string that should be appended to the end of the PATH environment variable should be something like; C:\Program Files\Java\jre6\bin. Click “OK” to close the Edit System Variable window. Click “OK” in the Environment Variables window, and again to close the System Properties window.

SUPPORT PROTOCOL 2: INSTALLING JAVA 2 PLATFORM STANDARD EDITION ON LINUX COMPUTERS

Materials

Hardware

A computer running Linux kernel version 2.4 or later

Software

Java 2 Platform Standard Edition (J2SE), version 5.0 or later, available from <http://java.sun.com/javase>

There are several options for installing the J2SE on Linux computers. For RPM-based Linux distributions, Sun provides RPM-based installation packages that install via the Red Hat Package Manager system. This is probably the best option for Red Hat-based distributions. Sun also provides generic packages that do not use the RPM system, which should work on all recent Linux distributions.

To install J2SE, you must have root privileges, either by logging in as root, or by issuing the su - command. Download the appropriate Java package from the Sun Web site, which will be named something like jre-VERSION-linux-ARCHITECTURE.bin or jre-VERSION-linux-ARCHITECTURE-rpm.bin. To make those files executable, execute the following command, substituting the actual package name:

```
chmod +x jre-VERSION-linux-ARCHITECTURE.bin
```

If you have chosen to install the RPM-based J2SE, start the installation by executing the following command, substituting your actual package name:

```
/jre-VERSION-linux-ARCHITECTURE-rpm.bin
```

If you chose to download the non-RPM-based package, you may want to create such a directory, e.g., /usr/local/java, with the command:

```
mkdir /usr/local/java
```

Change to the desired destination directory for installation:

```
cd /usr/local/java
```

To start the installation process, execute the following command on the command line, substituting the actual path and jre package name:

```
/PATH/TO/jre-VERSION-linux-ARCHITECTURE.bin
```

Having started the installer, you will be presented with the license agreement. Use the space bar to advance to the end of the license file, where you will be asked to accept or decline the license terms. Type yes to accept terms, and installation will proceed.

The installation directory will contain a subdirectory named bin. The path to this subdirectory will be something like /usr/local/java/jreVERSION/bin. Add this path to your executable path by editing your /etc/profile file. On Red Hat Linux, this can be accomplished by editing the /etc/profile file and adding the line:

```
pathmunge /usr/local/java/jreVERSION/bin after
```

More generally (on other distributions), the following line can be used to add the jrVERSION/bin path to your PATH environment variable:

```
export PATH=$PATH:/usr/local/java/jreVERSION/bin
```

SUPPORT PROTOCOL 3: UPDATING TO J2SE 5.0 ON MAC OS X COMPUTERS

Mac OS X ships with Apple's own version of the Java 2 Platform Standard Edition, and as of 2008, Macintosh computers include J2SE version 5.0. Some versions of Mac OS X may require an update of the J2SE. To determine if your Macintosh system requires an update, you can check by entering the following command in a terminal window:

```
java -version
```

Versions reported as 1.5.0 or higher fulfill the requirement for J2SE version 5.0. Note that Sun changed its numbering convention for Java after Java version 1.4. Version 5.0 still reports as 1.5.0.

Materials

Hardware

A computer running Mac OS X 10.4 or later (J2SE version 5.0 is not available for earlier versions of Mac OS X)

Software

J2SE 5.0, available through the Mac OS X Software Update utility, or from <http://developer.apple.com/java/download/>

To update the Mac OS X J2SE to version 5, in the Apple Menu, select About This Mac. The About this Mac window will open. Click the “Software Update...” button. The Software Update utility will check Apple’s Web site for package updates, and you will be presented with a list of available updates. A package named Java For Mac OS X will most likely be in that list of packages to update. Packages for installation may be selected by checking the Install box to the left of each package name. Make sure the box next to the Java package is checked and click the “Install...” button. After installation, Mac OS X may request a restart. It is advisable to do this.

If for some reason, Java For Mac OS X is not in the list of available packages for update, you can download the appropriate package from Apple’s Web site, <http://developer.apple.com/java/download/>, and install it manually.

SUPPORT PROTOCOL 4: INSTALLING MySQL SERVER AND CLIENT ON PACKAGE-BASED LINUX SYSTEMS

Materials

Hardware

A computer running Linux kernel version 2.4 or later

Software

MySQL Server and Command Line Client Programs, available from MySQL AB <http://dev.mysql.com/downloads/mysql/5.0.html>

The easiest way to install MySQL on a Linux system is to use one of the automated installation utilities that comes with most modern Linux distributions. The biggest advantages of using these utilities are ease of use, and the fact that the packages you will be installing are tailored for your precise Linux distribution, version, and architecture.

To install the MySQL server and command-line client programs, first become root, either by logging in as root, or with the su - command. On RPM-based distributions, such as Red Hat and Fedora Linux, that use the yum utility, the following command will probably install MySQLmysql:

```
yum install mysql-server mysql
```

Once yum ascertains which packages it needs to install, it will ask you to confirm that you want to download and install the required MySQL packages. Type Y to confirm.

The yum utility will then download and install the mysql-server and MySQL packages, as well as any dependencies. Note that the specific package names may vary on your OS distribution and version. When in doubt, use the following command to obtain a list of the available mysql packages for your version of the OS:

```
yum list | grep -i mysql
```

On Debian-based distributions, including Ubuntu, the standard automated installation utility is apt-get:

```
apt-get install mysql-server mysql-client
```

See your OS documentation for more information about the apt-get utility.

MySQL can be installed on older Red Hat Linux systems by simply downloading the appropriate RPM files from the MySQL Web site (see <http://dev.mysql.com/downloads/mysql/5.0.html>). Alternatively, both RPM files and DEB package files can often be downloaded from the distributor of your Linux distribution. Refer to your system documentation for information on how to install these packages on your system.

Once MySQL is installed, make sure the MySQL (or mysql.server) startup script is present in the /etc/rc.d/init.d directory and that there are appropriate symbolic links to this script in the default runlevel directory, for example, /etc/rc.d/rc5.d/. This can be accomplished in Red Hat and Fedora Linux by checking the MySQL box in the Services control panel (select Desktop > System Settings > Server Settings > Services).

The MySQL Server has a configuration file called my.cnf, usually located in /etc (see the MySQL documentation for your distribution for more information; note some versions of MySQL have used my.conf or my.ini, alternatively). Most of the default settings of my.cnf work well with the PATIMDB Application, with the exception of maximum allowed packet size. Add the following line to the [mysqld] section of the my.cnf file:

```
max allowed packet=32M
```

For this change to take effect, you will need to restart the MySQL server by executing (as root) the startup script with the argument restart:

```
/etc/rc.d/init.d/mysql restart
```

If your database server is not the same computer as your workstation, you may have to configure your firewall and security settings. In Red Hat and Fedora Linux, this can be accomplished in the Security Level control panel (select Desktop > System Settings > Security Level). Specifically, you will need to open up MySQL's port 3306. In the Firewall Options tab add 3306:tcp to the "Other ports" box. Also, if you have SELinux enabled, you may find that access to MySQL is blocked. If this is the case, under the SELinux tab, in the Modify SELinux Policy box, under SELinux Service Protection, check "Disable SELinux protection for mysqld daemon".

To set the root password for the MySQLserver, execute the following command, substituting the actual desired password for PASSWORD:

```
mysqladmin -u root password 'PASSWORD'
```

SUPPORT PROTOCOL 5: INSTALLING MySQL SERVER AND CLIENT ON GENERIC LINUX-, MAC OS X-, AND UNIX-BASED COMPUTERS

For a protocol to install MySQL on generic Linux, Mac OS X, and Unix-based computers, see Vastrik (2008).

To enable transfer of large packets, add the following line to the [mysqld] section of the my.cnf file:

```
max_allowed_packet=32M
```

SUPPORT PROTOCOL 6: INSTALLING MySQL SERVER AND CLIENT ON WINDOWS

The MySQL server and client programs are available from the MySQL AB web-site at <http://dev.mysql.com/downloads/mysql/5.0.html#downloads>. Windows binaries are available for 32-bit Windows x86, and 64-bit Windows x64. Three types of packages are available: a standard Windows installer, a "Windows Essentials" minimal installer, and a "without installer" package. We recommend using the standard Windows installer, since it is easy to use for installation, and allows the user to install the full set of MySQL features if desired. Execute the following steps:

1. Download the appropriate MySQL package, for example, mysql-5.0.51b-win32.zip.
2. Double-click the compressed zip archive to open. On older versions of Windows, you may need a zip archiving program to open, but versions of Windows since Windows XP treat zip archives as compressed folders: the archive will open as a folder, and you can simply click on the contents. Click Setup.exe.

3. A window will open with the message: Welcome to the Setup Wizard for MySQL Server 5.0. Click Next.
4. In the Setup Type window, select Typical (or, if you want all the features, select Complete) and click Next.
5. In the Ready to Install the Program window, click Install.
6. After the Installation is complete, you may see one or two windows advertising MySQL Enterprise. Click Next to proceed. If you see a second MySQL Enterprise advertisement window, Click Next again.
7. In the window displaying the message Wizard Completed, make sure the “Configure the MySQL Server now” box is checked, and click “Finish”.
8. In the MySQL Server Instance Configuration Wizard, when the message Welcome to the MySQL Server Instance Configuration Wizard 1.0.12.0 appears, click Next.
9. When prompted with the message Please select a configuration type., select Standard Configuration, and click Next. You may opt to perform Detailed Configuration, but this is not necessary.
10. When prompted with the message Please set the Windows Options., make sure Install as Windows Service is checked. Also, for convenience, we recommend you check Include Bin Directory in Windows PATH (this is not the installer’s default). Click Next.
11. When prompted with the message Please set the security options., check Modify Security Settings, and enter and confirm your desired MySQL root password in both text boxes.
12. When prompted with the message Ready to execute, click Execute.
13. A window labeled Processing Configuration will display, showing the progress of configuration. Click Finish when the process is complete to close the wizard.
14. By default, the installer installs MySQL in the location c:\Program Files\MySQL\MySQL Server 5.0\.. To enable transfer of large packets, use Notepad, Wordpad, or even Microsoft Word to add the following line to the [mysqld] section of the my.ini:

```
max_allowed_packet=32M
```

Be sure to save my.ini as a plain text file. It may be necessary to rename the resulting file to remove the .txt extension (file must always be named my .ini).

15. To restart the MySQL server, in the start menu, select Control Panel > Performance and Maintenance > Administrative Tools > Services. In the services list, scroll down to MySQL. Right-click on MySQL and, in the context menu, select Restart.

COMMENTARY

Background Information

The PATIMDB Application is a useful tool for assisting in the creation of bacterial transposon insertion mutant libraries. The application's process tracking features will be most useful for researchers using a workflow similar to that of Liberati et al. (2008), but the automated sequence analysis features are expected to be useful for any projects employing sequencing strategies similar to the widely used method of Caetano-Anollés and Bassam (1993), who used semi-arbitrary PCR followed by nested PCR to generate a sequencing template. In addition to its potential application in a wide variety of projects involving the generation of transposon insertion mutants, the PATIMDB Application has several convenient features. It is relatively lightweight, with only modest system requirements for running (its initial deployment was on a 300 MHz Pentium-based computer). Being Java-based, it has the ability to run on a wide variety of computer platforms. Furthermore, it implements an automated sequence analysis pipeline using several robust, widely-used tools, including NCBI BLAST and Phred. In addition, the PATIMDB Application stores its data using the freely available and widely used MySQL database, so that the data can potentially be made accessible to other applications, including Web-based applications. Such a Web-based application has already been used to publish the data from the first project using the PATIMDB, the *Pseudomonas aeruginosa* Transposon Insertion Mutant Database (<http://ausubellab.mgh.harvard.edu/cgi-bin/pa14/home.cgi>). Whereas it is possible to deploy relatively customized, sophisticated process tracking and analysis pipelines using other freely available tools, such as BioConductor (<http://www.bioconductor.org>), the setup and development time for these tools can be significant. By contrast, the PATIMDB Application provides a ready-to-use pipeline and can be deployed in as little as a few hours.

The PATIMDB Application is still being actively developed, and improvements to documentation, bug fixes, and new features are planned. Among the planned features are ways to search for mutants by disrupted gene or chromosomal location, easier ways to export reports and data, and an interface providing gene- and chromosome-based views, representing putative genome insertion locations graphically. The public is invited to suggest new features and report user experiences (contact Jonathan Urbach at urbach@molbio.mgh.harvard.edu). Since it is the plan to eventually make the PATIMDB Application open-source, developers are invited to contribute enhancements to the PATIMDB Application.

Troubleshooting

The PATIMDB Application is still in active development as of June 2008. While we do intend to add more features, much of the work on the application focuses on making it more user-friendly and making error-handling more robust. As these improvements occur, we anticipate that the ease of installation, configuration, and usage of the PATIMDB Application will improve. We suggest here certain strategies to make problems you encounter easier to solve.

Firstly, when you launch the application on the command line, pay attention to any text messages that may be printed to the command line interface itself (also known as “standard output”). Error message dialogs generally reflect the fact that a particular type of error was encountered and that an appropriate message to communicate the problem to the user was incorporated into the program. Unfortunately, there are usually more types of errors that can occur than developers can anticipate. Nevertheless, errors that bubble up from the guts of the application often produce error messages on the command line that can elucidate the problem, even if appropriate error-handling routines have not yet been implemented.

While it is hard to anticipate all the kinds of problems that can occur on diverse platforms and operating system versions, most of them fall into a few classes. The most common types of problems that you are likely to encounter have been outlined in Table 19.7.2, along with strategies for solutions. In addition to these general suggestions and strategies offered here, the PATIMDB Application documentation contains a troubleshooting section addressing more specific problems that are commonly encountered.

If these problems become an annoyance, or if you suspect that they are due to a bug, e-mail the developer, Jonathan Urbach (urbach@molbio.mgh.harvard.edu), with a detailed bug report.

The one step in the installation that seems most subject to problems is the installation of the MySQL database server, but solutions can frequently be found by searching MySQL AB’s Web site (<http://www.mysql.com>), or by using an Internet search engine such as Google, Yahoo, etc.

Time Considerations

Once all the required software packages have been obtained, it is possible for an experienced IT person to install the PATIMDB Application, and all its dependencies, including Java, MySQL, MySQL Connector/J, and the helper applications, blastall, formatdb, and Phred, in less than an hour, as described in Basic Protocol 1 and the support protocols. In most cases, it will probably take longer, however. As noted above, the installation of MySQL can be problematic, but so can the installation of the other applications. We recommend that you budget 3 to 4 hr for the entire process, especially if you have no experience with the installation of MySQL or Java. We also recommend that you plan ahead in procuring Phred, since you need to obtain a license for it first, a process that can take anywhere from a day to a week or more.

Completing Basic Protocol 2, including setting up your BLAST database, importing a list of genes, and creating your first mutant library may take some time as well. The main rate-limiting steps are likely to be deciding on appropriate search parameters for your transposon, and generating a table of genes that is properly formatted for import into the PATIMDB. For the former, you will probably have to peruse some sequence files to make a decision about the parameters you wish to use. The latter simply involves obtaining and reformatting the relevant data from the relevant genome sequencing project. We recommend budgeting 4 hr of time for Basic Protocol 2. Fortunately, most users will only have to perform Basic Protocol 2 once.

Basic Protocols 3, 4, and 5 should take no more than 5 to 10 min each to perform. Over the course of the generation of a transposon insertion mutant library, they may be performed repeatedly.

Acknowledgments

This work was supported by NIH grants AI-064332, DK-040561, and AI-044220, and DOE grant DE-FG02-ER63445 (to F. Ausubel).

Literature Cited

- Cameron DE, Urbach JM, Mekalanos JJ. A defined transposon mutant library and its use in identifying motility genes in *Vibrio cholerae*. *Proc Natl Acad Sci USA*. 2008; 105:8736–8741. [PubMed: 18574146]
- Caetano-Anollés G, Bassam BJ. DNA amplification fingerprinting using arbitrary oligonucleotide primers. *Appl Biochem Biotechnol*. 1993; 42:189–200. [PubMed: 7902694]
- Garsin DA, Urbach J, Huguet-Tapia JC, Peters JE, Ausubel FM. Construction of an *Enterococcus faecalis* Tn917-mediated-gene disruption library offers insight into Tn917 insertion patterns. *J Bacteriol*. 2004; 186:7280–7289. [PubMed: 15489440]
- Jacobs MA, Alwood A, Thaipisuttikul I, Spencer D, Haugen E, Ernst S, Will O, Kaul R, Raymond C, Levy R, Chun-Rong L, Guenther D, Bovee D, Olson MV, Manoil C. Comprehensive transposon mutant library of *Pseudomonas aeruginosa*. *Proc Natl Acad Sci USA*. 2003; 100:14339–14344. [PubMed: 14617778]
- Liberati NT, Urbach JM, Miyata S, Lee DG, Drenkard E, Wu G, Villanueva J, Wei T, Ausubel FM. An ordered, nonredundant library of *Pseudomonas aeruginosa* strain PA14 transposon insertion mutants. *Proc Natl Acad Sci USA*. 2005; 103:2833–2838. [PubMed: 16477005]
- Liberati, NT.; Urbach, JM.; Thurber, TK.; Wu, G.; Ausubel, FM. Comparing insertion libraries in two *Pseudomonas aeruginosa* strains to assess gene essentiality. In: Gerdes, SY.; Kato, J-I.; Peyser, BD.; Irizarry, R.; Spencer, FA.; Baba, T.; Mori, H., editors. *Methods in Molecular Biology Vol 416: Microbial Gene Essentiality: Protocols and Bioinformatics*. Humana Press; Totowa, N.J: 2008. p. 153-169.
- Vastrik I. Installing a local copy of the reactome Web site and database. *Curr Protoc Bioinform*. 2008;9.10.1–9.10.9.

Figure 19.7.1.

Workflow to generate a transposon insertion mutant library. The PATIMDB Application is designed to work with the workflow developed by Liberati et al. (2008). Starting with putative transposon mutants on large-format plates, colonies are transferred to 96-well culture plates (WOR) and grown up in suitable media. Transfer of culture into 384-well plates containing growth medium with glycerol produces master (MAS) and duplicate (DUP) plates. PCR template plates (SUP) are prepared by transferring culture into 96-well plates followed by lysis at 95°C and centrifugation. Arbitrary PCR1 plates (AR1) are

generated by transferring supernatant from SUP plates and performing the first round of arbitrary PCR. This PCR product is transferred as template to the arbitrary PCR2 plates (AR2) and the nested PCR step is performed. Lastly, transfer and cleanup produces the sequencing plates (SEQ), which are submitted to a sequencing facility.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Figure 19.7.2.

PATIMDB Application Client Server Architecture. The PATIMDB Application, which is intended to be installed on a PC or workstation, is a client application for a MySQL database installed on a database server. Communication with the database server takes place over a network.

Figure 19.7.3.

The first time the PATIMDB Application is launched, this message alerts the user that no configuration file exists. Clicking the OK button opens the PATIMDB Application configuration window. Clicking Cancel exits the application.

Figure 19.7.4.

The PATIMDB Application configuration window allows configuration of the applications database connection.

Figure 19.7.5.

The PATIMDB Application Main Window is the main interface of the PATIMDB Application. Functions are accessed through the various menus.

Scientist Name* Jonathan Urbach

Department Molecular Biology

Address

Phone

Fax

Email

Login* urbach

Password*

Privilege* owner

Previous Next New Delete Save Help

Total number of libraries : 1

Figure 19.7.6.

The User Manager is the interface for defining users of the PATIMDB Application. Define users with Owner privilege to enable password protection.

Figure 19.7.7.

When password protection is enabled, users are prompted for their login and password when the PATIMDB Application starts.

Figure 19.7.8.

The Analysis Settings interface allows users to configure paths to helper applications used by the PATIMDB Application for automated sequence analysis. Other analysis parameters may also be configured. See the PATIMDB Application documentation or click Help for more information.

Figure 19.7.9.

The PATIMDB is a relational database with many entities that are dependent on one another. During Mutant Library Setup (Basic Protocol 2), Organisms, Transposons, Mutant Libraries, Genome Sequences, and Genes are defined, and BLAST databases are exported. During plate creation (Basic Protocol 3), Plates, Mutants, and Samples are created. During automated sequence analysis, raw sequences are imported, processed sequences are generated, and BLAST alignments are used to generate InsertGenome records. Although the BlastDB is not, strictly speaking, a PATIMDB database entity, it is included here to show its interrelationship with the various database tables.

The image shows a software window titled "Organism Manager". It contains three text input fields: "Strain" with the value "PA14", "Genus" with the value "Pseudomonas", and "Species" with the value "aeruginosa". Below these fields is a row of six buttons: "Previous", "Next", "Delete", "New", "Save", and "Help". The "Save" button is disabled (grayed out).

Figure 19.7.10.

Defining an organism is the recommended first step in defining a mutant library. The strain field is used as the unique primary key for organisms. Since the strain identifier is used in generating human-readable barcodes (which can often only encode a limited number of characters), the strain identifier should be short, optimally four characters.

The screenshot shows a window titled "Transposon Manager". It contains several input fields and buttons. The fields are: "Transposon ID*" with the value "PhoA", "Parent" with the value "TnPhoA", "Description" (empty), "Tn Search Sequence*" with the value "CTGTATAAGAGTCAG", "Tn Max Levenshtein Distance (default 5)*" with the value "5", "Tn Max Position (default 70)*" with the value "70", and "Default Offset*" with the value "20". At the bottom, there are six buttons: "Previous", "Next", "New", "Delete", "Save", and "Help".

Figure 19.7.11.

The Transposon Manager is the interface for defining transposons. The Transposon ID is used as the unique primary key for transposons. Since the Transposon ID is used in human-readable barcodes, it should be short, optimally four characters. The other required fields in this window are the transposon search sequence (TnSearchSequence), TnMaxLevenshteinDistance, TnMaxPosition, and DefaultOffset.

The screenshot shows a window titled "Mutant Library Manager". It contains several input fields and dropdown menus. The fields are: "Library ID*" with the value "PA14_PhoA", "Creation Date*" with the value "2008-05-29" and a date format hint "yyyy-mm-dd", "Library Owner" (empty), "Library Creator*" with a dropdown menu showing "Jonathan Urbach", "Transposon*" with a dropdown menu showing "PhoA", "Host Organism*" with a dropdown menu showing "PA14", "Preparation Protocol" with a dropdown menu showing "1", and "Library Description" with the value "None". At the bottom, there are buttons for "Previous", "Next", "New", "Delete", "Save", and "Help". A status bar at the very bottom indicates "Total number of libraries : 1".

Figure 19.7.12.

The Mutant Library Manager is the interface for defining mutant libraries. The Library ID is the unique primary key. Other required fields are Creation Data, Library Creator, Transposon (as defined in the Transposon Manager), and Host Organism (as defined in the Organism Manager).

Manage and Import Genome Sequences

Genome Seq ID: 1

Strain: PA14

GenBank Accession: CP000438

Description:

Length: 6537648

Sequence File: :h/Desktop/PA14/CP000438.gbk Browse

Sequence File Type: GenBank

Previous Next New Save Close Help

Figure 19.7.13.

The Genome Sequence Manager allows users to define genome sequences by importing a FASTA format sequence file and associating the contained sequence with a particular bacterial strain (as defined in the Organism Manager). These genome sequences will be used later for BLAST analysis.

Figure 19.7.14.

The Gene Manager allows users to import a file containing gene identifiers, starts, ends, strands, and sequences. This data allows the PATIMDB Application to determine which genes have been interrupted by a transposon insertion mutant insertion. See the PATIMDB Application documentation for more information on the required TSV file format.

Figure 19.7.15.

During import of gene records from GenBank, GFF, and GTF files, the Genes File Import Dialog window appears, allowing users to specify which type of feature to import into the database and which unique identifier to use as a database key. These file formats contain gene, CDS, rRNA, and other feature types that may share unique identifiers, such as locus tags. Since “gene” is the most general of these feature types, it is generally sufficient to select the defaults, “gene” and “locus_tag.”

Figure 19.7.16.

The BLAST Database Manager allows users to specify which genome sequences to export and format as a BLAST database. Only these exported genome sequences will be used during automated sequence analysis.

The screenshot shows a software window titled "Create Plate Sets and Barcode Labels". It contains several input fields and buttons. The "Number of Plate Sets" field is set to 48. The "Mutant Library" dropdown is set to PA14_PhoA. The "Strain" field is set to PA14. The "Transposon" field is set to PhoA. The "Storage Location" dropdown is empty. The "Scientist" dropdown is set to Jonathan Urbach. The "Barcode File Destination" field is set to DB/PATIMDB_app_20080528, with a "Browse..." button next to it. At the bottom, there is a "Status: Ready" indicator in red, and two buttons: "Create Plate Set" and "Help".

Figure 19.7.17.
The Plate Set Creator allows users to create virtual plates for a given mutant library.

Figure 19.7.18.

The Execute A Process window is the main process tracking interface of the PATIMDB Application.

Figure 19.7.19.

In the Sequence Analysis interface, users can perform automated sequence analysis on sets of genome sequences. Users must select the directory containing the source sequence files and the source file type.

Figure 19.7.20.

Automated Sequence Analysis Pipeline. The pipeline that the PATIMDB Application uses to perform automated sequence analysis begins with ABI files as input. Phred is used to perform base-calling and quality assessment, generating sequence (.SEQ) and quality (.QUAL) files as output. The PATIMDB Application then performs transposon matching, and the sequence, quality, and transposon data are stored in the database. The PATIMDB Application uses the Phred quality data to trim the sequence, removing low-quality sequence from the 5' end, and the resulting processed sequence is exported as .PSEQ files and used as the query for BLASTN against the genome sequence. The processed sequence and the resulting BLAST results files are imported into the database. The BLAST results, together with the transposon matching data, are used to determine the genomic location of the transposon insertion, as well as which genes are disrupted by that insertion. In this figure,

analysis pipeline processes are represented as solid, black arrows, whereas insertions of data into the database are represented as dashed, gray lines.

Figure 19.7.21.

The Reanalyze Sequences interface allows users to repeat automated sequence analysis on reads that are already in the database.

Figure 19.7.22.

The Plate Selector window allows users to select plates to view by plate type, plate ID, and label.

Figure 19.7.23.

In the Report for Plate window, users may view information about selected plates. Mutant reports may be retrieved by clicking wells in the plate map graphic.

Figure 19.7.24.
The Mutant Report interface contains information about individual mutants.

Table 19.7.1

Example of Plate Labels from Plate Creation

Plate LabelID					
Plate type	Format	1	2	3	4
WOR	96	PA14_PhOA_1wor	PA14_PhOA_2wor	PA14_PhOA_3wor	PA14_PhOA_4wor
SUP	96	PA14_PhOA_1sup	PA14_PhOA_2sup	PA14_PhOA_3sup	PA14_PhOA_4sup
AR1	96	PA14_PhOA_1ar1	PA14_PhOA_2ar1	PA14_PhOA3ar1	PA14_PhOA_4ar1
AR2	96	PA14_PhOA_1ar2	PA14_PhOA_2ar2	PA14_PhOA_3ar2	PA14_PhOA_4ar2
SEQ	96	PA14_PhOA_1seq	PA14_PhOA_2seq	PA14_PhOA_3seq	PA14_PhOA_4seq
MAS	384	PA14 PhOA 1mas			
DUP	384	PA14 PhOA 1dup			

Table 19.7.2**Solving Common Problems**

Class of problem	What to consider and strategies for solving
Problems starting the PATIMDB Application	<p>Make sure that:</p> <ol style="list-style-type: none"> 1 The Java interpreter is available and in your path. 2 The PATIMDB app.jar file is located in the expected place, the dist subdirectory. 3 The installed version of Java is compatible with the PATIMDB Application. Running java -version should report v. 1.5 or greater.
Problems connecting to the PATIMDB database	<p>Make sure that:</p> <ol style="list-style-type: none"> 1 The MySQL server is properly installed and running. 2 The computer hosting the MySQL database is accessible over the network, if it is not the local computer, and there is no firewall or security configuration preventing MySQL's port from being accessed by client programs. 3 The MySQL Connector/J application is properly installed. 4 Your MySQL passwords are correct. If all else fails, try installing the mysql command line client on the local machine (if you haven't installed it already) and trying to use it to connect.
Problems with helper applications	<ol style="list-style-type: none"> 1 Does the PATIMDB Application complain about not being able to find a particular helper application: phred, blastall, or formatdb? Check the configured location. Make sure the path is valid. Is application capable of being run from the command line using the path you have configured in Analysis Settings? 2 If the PATIMDB Application reports that a particular command exited with an error status, try running that command in a terminal or command prompt. Segmentation faults and any kind of protection error indicate a serious problem with the binary application being used, and indicate the need for reinstalling or obtaining a newly compiled binary.
GUI glitches and general bugs	<ol style="list-style-type: none"> 1 If a certain combination of mouse clicks causes the PATIMDB Application's GUI to behave strangely or become unresponsive, try quitting the application, then restart. This will fix many problems. 2 If execution of the automated sequence analysis pipeline exits prematurely, and no sequence results have been imported, make sure that your paths are correct, and that you are indeed importing ABI files. See "Problems with helper applications," above.
Miscellaneous problems	Observe whatever message dialogs and command line error messages pop up. They can often give clues to solving the problem.