BIOINFORMATICS

Vol. 17 no. 11 2001
Pages 1077–1083

Database-driven Multi Locus Sequence Typing (MLST) of bacterial pathogens

Man-Suen Chan[1,*,2], Martin C. J. Maiden[1] and Brian G. Spratt[1,3]

[1]The Wellcome Trust Centre for the Epidemiology of Infectious Disease, University of Oxford, South Parks Road, Oxford OX1 3FY, UK

Received on March 21, 2001; revised on April 25, 2001; accepted on May 17, 2001

ABSTRACT

Motivation: Multi Locus Sequence Typing (MLST) is a newly developed typing method for bacteria based on the sequence determination of internal fragments of seven house-keeping genes. It has proved useful in characterizing and monitoring disease-causing and antibiotic resistant lineages of bacteria. The strength of this approach is that unlike data obtained using most other typing methods, sequence data are unambiguous, can be held on a central database and be queried through a web server.

Results: A database-driven software system (mlstdb) has been developed, which is used by public health laboratories and researchers globally to query their nucleotide sequence data against centrally held databases over the internet. The mlstdb system consists of a set of perl scripts for defining the database tables and generating the database management interface and dynamic web pages for querying the databases.

Availability: http://www.mlst.net.

Contact: mchan@molbiol.ox.ac.uk

INTRODUCTION

Global epidemiology of pathogenic bacteria requires the direct comparison of isolates obtained in laboratories in different areas of the world in order to track the development of potential outbreaks. Unfortunately, many existing bacterial typing methods are ill-suited for these types of comparisons as they need to be standardized in each laboratory and depend on specialized reagents and techniques (Achtman, 1996). In this regard, nucleotide sequence based methods are more suitable, as they enable direct unambiguous comparison between isolates typed in different locations (Maiden *et al.*, 1998).

*To whom correspondence should be addressed.

[2]Present address: Department of Paediatrics, The Institute of Molecular Medicine, University of Oxford, John Radcliffe Hospital, Headington, Oxford OX3 9DS, UK.

[3]Present address: Department of Infectious Disease Epidemiology, Imperial College School of Medicine, University of London, St Mary's Campus, Norfolk Place, London W2 1PG, UK.

Multi Locus Sequence Typing (MLST) is a sequence-based typing system which has been developed with a global epidemiology perspective. This method involves the sequencing of seven internal ~500 bp fragments from house-keeping genes and has been developed for several pathogenic bacterial species including *Neisseria meningitidis*, *Streptococcus pneumoniae*, *Staphylococcus aureus*, *Campylobacter jejuni* and *Streptococcus pyogenes*. Each unique sequence is given a unique and arbitrary allele number and the combination of allele numbers at the seven loci is known as the allelic profile. Each unique allelic profile is similarly assigned an arbitrary number, in the order of discovery, which is known as the sequence type. Validation studies have shown that these sequence types correspond to the lineages determined from other methods (Maiden *et al.*, 1998). Databases including representative isolates from different geographical areas, diverse lineages and both pathogenic and carriage isolates have been compiled (Maiden *et al.*, 1998; Enright and Spratt, 1998; Enright *et al.*, 1999, 2000, 2001; Jolley *et al.*, 2000; Zhou *et al.*, 2000; Dingle *et al.*, 2001).

Current internet technology is fully exploited in the design and practice of MLST. A database-driven web server acts as a focal point, allowing participants to directly query their data against the centrally held database as well as submitting their own strains for inclusion in these databases. The use of a www-interface is ideal, as it is easy to learn and enables many users to connect from anywhere in the world. We have developed a software system (mlstdb), which can be used to generate MLST databases, manage them through a web interface and provide dynamic web pages for querying the database. The software is written in perl with specifications for individual MLST systems (species) in XML format. The software is fully portable and available free of charge and can therefore be used to set up databases for other species as more MLST systems become available.

SYSTEM AND METHODS

The design of the MLST databases is determined by the nature of the MLST sequence data. Unlike other

sequence databases, which, in general, contain many unique sequences, MLST databases typically contain many replicates of the same sequence as the same loci are being sequenced for every isolate. Therefore, the database can be conveniently divided into tables for the overall allelic profile and tables for the sequences for each of the loci. This is also reflected in the way that the databases are queried: by locus or by profile.

The design also requires a system flexible enough to be applied to any bacterial species without the need for additional programming for each new species. To do this, the software must be able to handle different loci and number of loci being sequenced for each bacterial species, and different additional information being stored for each species. This is achieved by using XML to define each MLST system and then using the XML file to generate the database and web interface.

Database design

The MLST databases comprise one table for the allelic profile and additional strain information, and one table for each of the loci (all the current systems use seven loci). Also, there is one table of users. These are described below and the database design is shown in Figure 1.

(A) *Users table.* The users table holds two types of users, 'curators' who have privileges to update/insert/delete data and 'users' who do not. 'Users' are individuals/organizations who contribute data to the database, which is entered by the curator(s). A unique number and user name is given to each user and first name and surname, institution and email are also stored; all these fields are required. Individuals simply querying the database are not included in the user table. All the biological data entered in the database is linked back to the user table in two ways. The 'sender' in these tables is the user who contributed the data, the curator is the one who actually entered the data into the database.

(B) *Locus tables.* Each of the locus tables contains the sequences of all the alleles, the allele number being the primary key of each table. Administrative information such as sender, curator and date information is also stored. If the sequence is published, the GenBank/EMBL/DDBJ accession number is included in the table. The sequence on these tables cannot be updated as this may cause integrity problems with the profiles table. If a sequencing error is found, updating the sequence would also update the sequence for other strains with the same allele which may not be correct. Instead, the data can then be amended by editing the allele number in the profiles table, entering a new allele, or deleting the allele and re-entering it with the new sequence as appropriate.

(C) *Profiles table.* The profiles table includes the allelic profile, sequence type and epidemiological, microbiological and medical information about the strains submitted. No information that could identify the patient is included. The values of the allelic profile are linked to the locus tables so it is not possible to enter values for which no sequence has been entered. The database is designed to allow the monitoring of the distribution of bacterial strains geographically. Therefore, it is possible to enter several examples of the same genotype recovered from different patients.

Specification of database design for individual species using XML

The general structure of databases for different MLST systems is the same. However, a flexible method for configuring the databases for each species is required. This is achieved by specifying species-specific design using XML. This is a similar design to the software system PISE (Letondal, 2001), which uses XML files to specify different programs, these files are then used to generate dynamic web pages. The Document Type Declaration (DTD) for the MLST systems includes the following elements.

(1) Name of the species.

(2) System parameters consisting of a long and short code and a short description (one set per species).

(3) Authors (one or more).

(4) References (one or more).

(5) Doclinks (one or more).

(6) Curators (one or more).

(7) Loci (one or more, usually seven). Loci are specified by name, full name of gene and length. Optionally, sequencing primers can be added. Normally, for MLST, all the alleles are of the same length but this is not essential in the software system.

(8) Profiles fields: these contain additional fields in the profiles table (i.e. the profiles table will comprise all the profiles fields plus a field for each locus). Attributes of each field include its type, length (in characters), whether it is required, whether it is shown on the main web page and whether it has an option list. The option list contains allowable values for an option such as types of disease associated with a strain. The fields are kept as uniform as practicable between different species but the scheme allows flexibility including alternative typing schemes, etc. The XML file is used to generate the database and web pages automatically.

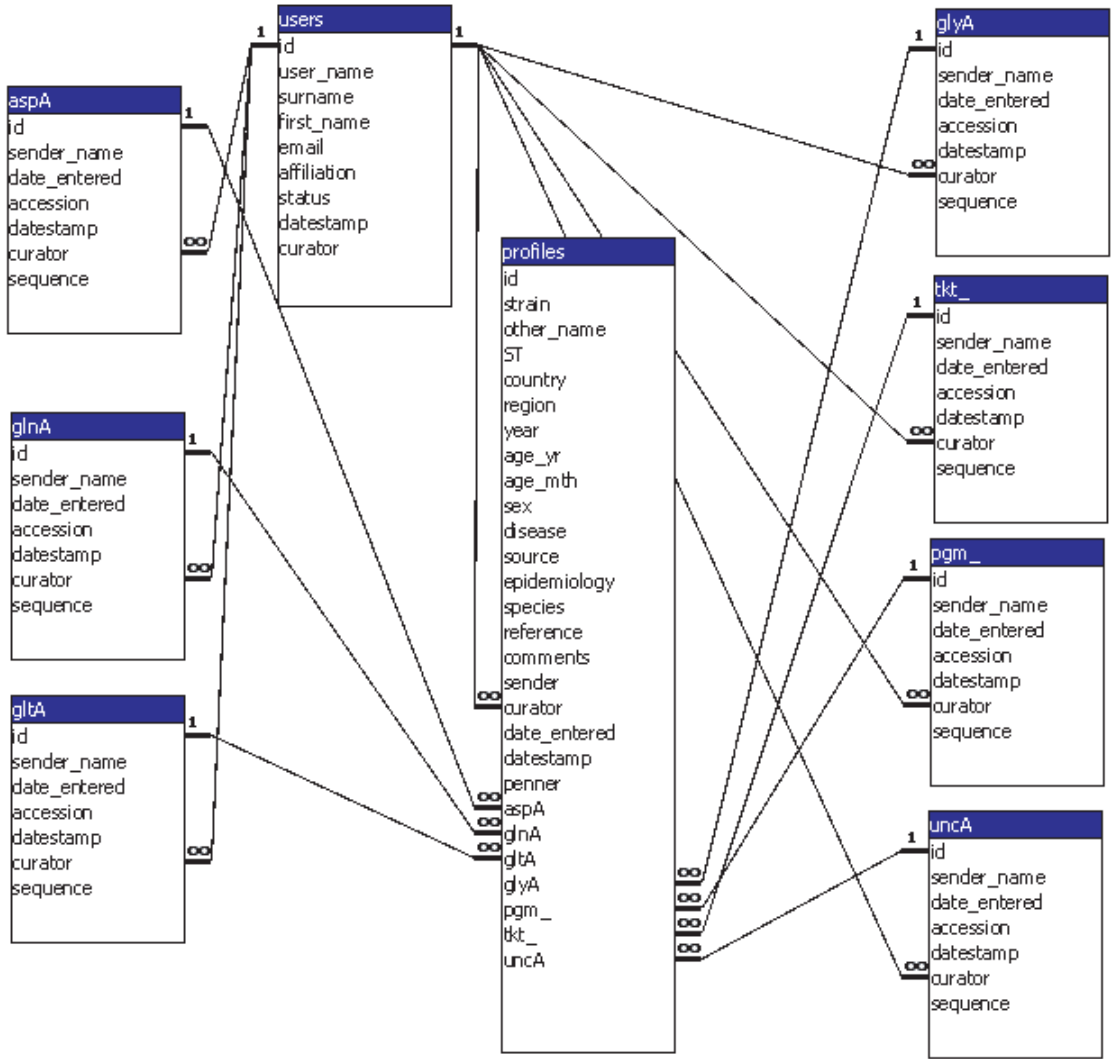

Fig. 1. The database design.

IMPLEMENTATION

The mlstdb system is implemented as a set of perl scripts for generating, managing and querying the MLST databases. The XML database files are parsed using the perl module XML::Parser::perlSAX (MacLoed, 2000). The different functions of the scripts are described below.

Generation of databases and informational web pages using sql and perl scripts

The XML files specifying the database design for each species is used as a basis for generating the database itself and files associated with it. After generating the XML file, its syntax is first tested with the program **nsgmls** (Clark, 2001) to ensure it conforms to the document type declaration.

The script **mlstreport.pl** produces an html web page which describes the MLST system, lists the loci to be sequenced and gives a table of the fields in the profiles with information about them. It also prints option lists for those fields that have them. This is displayed on the web server to inform users about the system. The report for the database specified in Figure 1 can be viewed at http://campylobacter.mlst.net/dbqry/cj-test.htm.

To generate the database itself, the script **mlstdbdef.pl** is used to generate the SQL commands to create the tables of the database. This is done automatically from the XML definition and does not require additional information. The definition includes all the foreign key constraints as well as CHECK constraints for the option lists. The output file of the script can then be run from your database management system to create all the tables.

Management of the database using the WDBI system (optional)

The database can be managed through a web interface using the WDBI system (Rowe, 1999). The WDBI system is free software, which can be used to manage many types of databases. The user customizes the data management interface using fdf (format definition files). These are simply text files which determine how each field is displayed and types of input accepted, etc. Optionally perl subroutines can be added into the fdf file, which can be run when data is entered to check they are suitable values, etc. The **mlstdb** system includes fdf files/perl scripts, which generate fdf files (**mlstprofilefdf.pl, mlstloci.pl, users.fdf**) specifically for MLST. For the locus tables, this includes subroutines to check the allele sequences. New sequences are checked to ensure they are not already in the database, that they are of the right length and that they are sufficiently similar to existing alleles to be a valid new allele (>70% identity at DNA level). For fields with option lists, these appear as dropdown menus. The system will also check the person entering the data has curator status. The use of the WDBI system is not essential to run the MLST system as any database management system can be used. However, it is recommended as it has been customized for MLST, is easy to use and can be accessed by multiple users over a web interface.

Perl modules for MLST

Many of the MLST scripts contain common elements, such as subroutines to check the allele sequences, etc. For this reason, these elements have been incorporated into perl modules, which are separate files from the actual perl scripts for specific tasks. This modular approach makes programming more clear and the whole system easier to extend. Four perl modules are included with the package, one each comprising the subroutines for each type of database table and one general parsing module:

(1) **mlst::parser**: this is the perl module for parsing the XML file and generating the data structures (**$mlst::parser::species, %mlst::parser::system, @mlst::parser::loci, @mlst::parser::fields**, etc.) associated with the MLST system.

(2) **mlst::users**: this module contains functions for converting between user i.d. and name, checking the status of a user (to check write permission for a database), etc.

(3) **mlst::loci**: this module contains the functions for analyzing the sequences, including checking the allele number, finding the most similar sequence, importing and exporting sequences (from/to fasta or mega format) and determining the next free allele number. The threshold identity for inclusion as a new allele is 70% at the DNA level. Any sequence, showing greater differences is not permitted as a new allele. An alignment with the most similar sequence is performed using the EMBOSS program **stretcher** (Rice *et al.*, 2000) and the output of this alignment is displayed to the user.

(4) **mlst::profiles**: this module contains functions for manipulating allelic profiles. These include identifying the sequence types, finding all sequence types with at least a given number of matches to the query profile and finding the next free sequence type or i.d. number.

Wherever possible, the database queries are performed directly using an SQL SELECT query to the database. The perl module DBD::Pg is used for this. However, this is not always possible, such as with the sequence analysis. In this case, custom procedures, which require an initial SELECT, followed by some external manipulation of the data, are required.

Generation of dynamic web content for database queries

Dynamic content is generated using CGI scripts written in perl. The same set of scripts is used for each MLST system. The script parses the XML file and displays a page with parameters relevant to the MLST system queried. The following types of queries are available:

(1) *Locus queries.* This is a query with a DNA sequence. The sequence is pasted into a TEXTAREA and submitted for query against the database. The server first checks that the input is a DNA sequence of the correct length. It then sends an SQL SELECT query to the database. If the exact sequence is found, its allele number is reported. Otherwise, the percentage similarity of the closest allele is determined, and an alignment shown. If the similarity is over 70%, the user is informed that this is possibly a new allele, otherwise, they are informed that it is probably the wrong sequence. This is a reasonable threshold since our data shows different alleles in the same species to differ at <30% of sites. This is the level of similarity at which homologous recombination is thought to occur.
As well as the simple single locus query, two other locus queries are available: a multiple locus query can be used. This identifies all the loci and the sequence type for one strain on one page. In addition, it is possible to submit a batch of sequences to the single locus batch query. This is particularly useful during data processing in large sequencing projects as the entire contents of one sequencing plate can be queried at the same time.

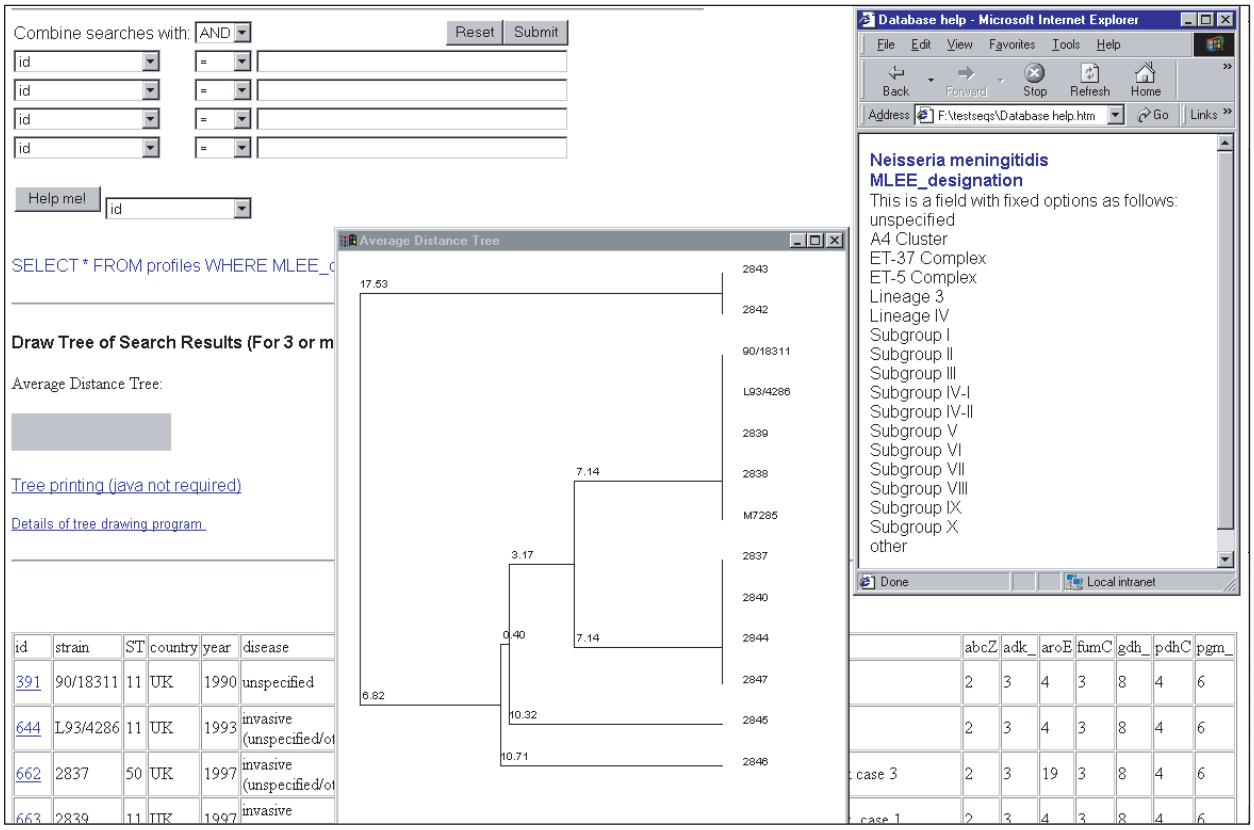

Fig. 2. Advanced-query interface.

(2) *Allelic profile query.* Once the allele numbers at each locus for a given isolate are known, it is possible to use the allelic profile query to determine the sequence type of that isolate and tabulate other strains with that sequence type. Optionally, it is also possible to list the strains, which have 3, 4, 5 or 6 or more matches with the query strain. This is useful for finding a group of closely related strains.

(3) *Database searching.* The databases can also be queried using other information fields such as country, serotype, etc. and combinations of these fields. The 'database browsing' simple query interface is intended as a coarse filter only and allows query with one field. The 'advanced query' interface (Figure 2) allows querying in up to four fields simultaneously and allows combining searches with AND or OR. For each field specified, queries can be made using =, NOT and for numerical fields, < (less than) and > (more than). There is also a help button that will list available options for each field. In both interfaces, the query results are tabulated with hyperlinks to the complete information on each strain.

As well as database searching, there are also additional data visualization and analysis tools in the web software. Results of the allelic profile and database searches can be viewed as UPGMA trees (Sokal and Michener, 1958). This is provided by an applet that can be activated by pressing on the button marked 'Tree' after a query has been made.

The results of a single locus query can be compared to existing alleles in the database using an applet called Jalview developed by Michele Clamp at European Bioinformatics Institute (Clamp, 1998). The tree applet described above is also derived from the Jalview code. Jalview is a fully functional multiple alignment viewer with many analysis tools such as tree drawing, principal components analysis, pairwise comparisons, etc. A Jalview window with some MLST sequences loaded is shown in Figure 3.

WEB SERVER AND DATABASE MANAGEMENT

The system is implemented on a Linux server running the Apache web server (www.apache.org). The database software used is postgresql version 7.0 (Momjian, 2001),

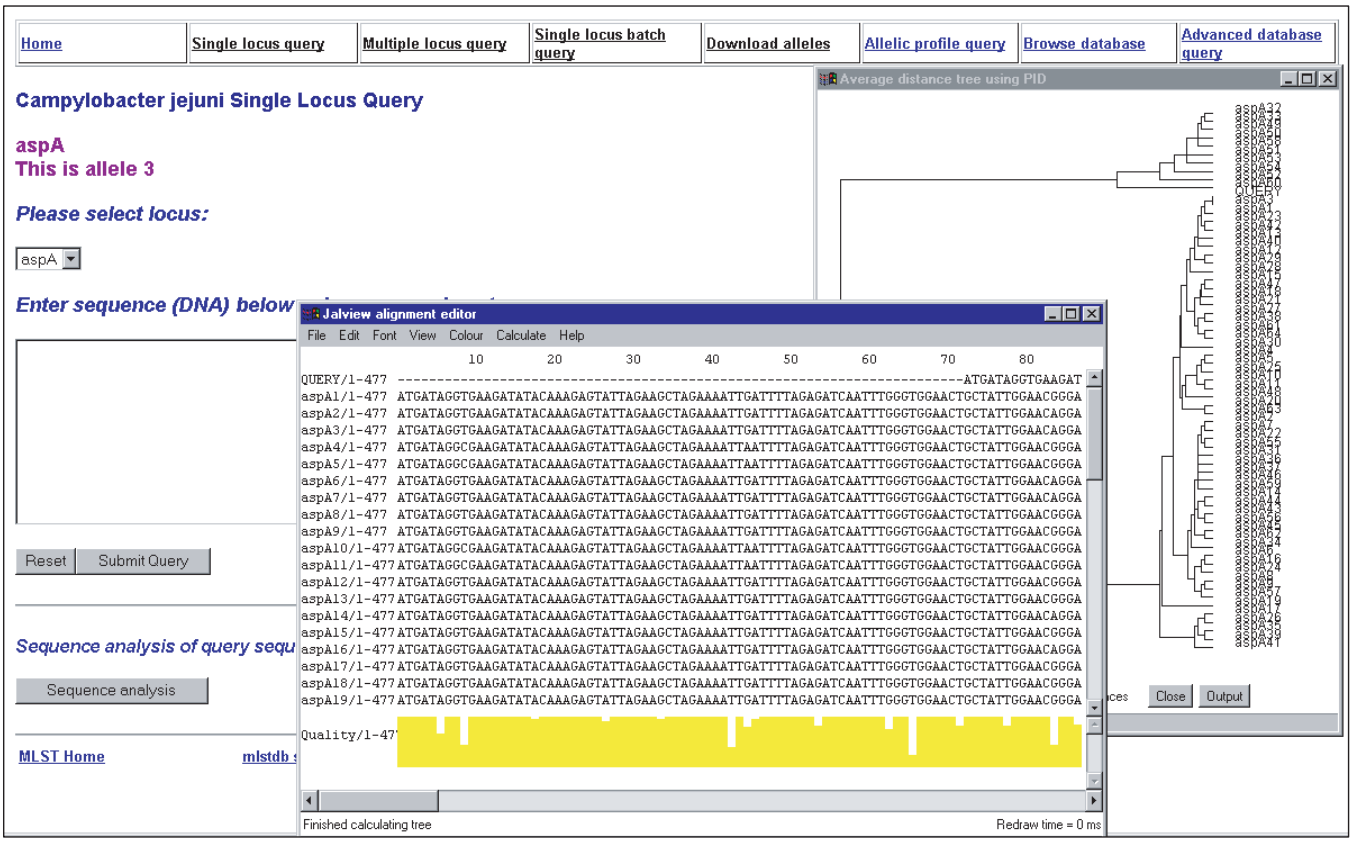

Fig. 3. Screenshot showing MLST sequence data in Jalview window.

available free from www.postgresql.org. Other database software could also be used with minimal modifications to the code. The WDBI interface, which is written in perl, is available free from http://www-stel.asu.cas.cz/wdbi/. All the standard perl modules can be downloaded from the perl website at www.perl.com.

MLST internet-based collaborations require maintenance, organization and curation. Each database (i.e. each species) has an appointed curator who checks all the submissions. In order to assign a new allele, the sender must submit forward and reverse sequence traces (electropherograms) to the curator who checks that there are no ambiguities in the sequence. Also, each user must register although registration is free. Submissions are only accepted from registered users with a valid email address in case there are queries regarding the submission. Since some of the data in the databases may not yet be published in a journal or submitted to a public database, registered users are required to agree to a policy document regarding the use of the data. An email list of registered users is maintained and informed of new updates and matters for discussion.

DISCUSSION

The MLST project is a successful and cost effective model for international co-operation in the monitoring of infectious diseases. The most important factors in this success have been the portable, sequence-based typing method which has proved popular in both public health laboratories and research institutions, the availability of the data through a user-friendly interface on the world wide web and a web site providing a forum for sharing information and techniques. Due to these factors, users of the service have been happy to deposit the data in the centralized databases, which are growing on a daily basis. The larger databases each have approximately 50 individuals/organizations contributing data. In addition, the number of hits on the web-site has been increasing steadily since its establishment.

As the MLST project progressed the challenge was to develop a software design that is sufficiently flexible to allow new, fully functional MLST databases systems to be set up without additional programming. The database, its management system and the dynamic web pages can all be created using only the XML specification of the

MLST system. The software is based on a set of scripts which could be used individually. This allows flexibility in the type of database software, web server, etc., allowing MLST to be integrated into existing bioinformatic services, epidemiological databases and software for analyzing the population biology and evolution of bacteria (Feil *et al.*, 1999, 2000; Holmes *et al.*, 1999). The incorporation of most of the functions of MLST into perl modules also allows the web pages for a particular system to be further customized. The software is fully portable and can be downloaded free of charge from the MLST web site.

In future, MLST is likely to become a more popular method for bacterial typing. A major factor will be decreasing costs of automatic sequencing. This will lead to new challenges in several areas. Firstly, future MLST projects will be bigger, with larger and an increasing number of databases and increasing frequency of hits on the web servers. This would require scaling up the capacity of the system and an increasing need to organize the curation of the databases. Secondly, the projects will become more diverse. For example, MLST systems are being developed for some eukaryotic organisms and this will require further development of the software to incorporate these systems. Thirdly, as new systems are developed, software that will allow projects to be distributed at different sites and with multiple participants will need to be developed. Therefore as the MLST technology develops, the use of MLST as a typing method and the development of the internet services and software will need to be closely interlinked.

ACKNOWLEDGEMENTS

This project would not have been possible without the curators of the databases (Angela Brueggemann, Kate Dingle, Mark Enright, Bill Hanage, Keith Jolley, Rachel Urwin) and all who have contributed data to the MLST databases. We are grateful for the continued enthusiasm and support of all the MLST database users. We also wish to thank Steven Dunstan for advice in setting up the web site, Michele Clamp for permission to use the Jalview code and Bela Tiwari for comments on the manuscript. This work was funded by the Wellcome Trust.

REFERENCES

Achtman,M. (1996) A surfeit of YATMs? *J. Clin. Microbiol.*, **34**, 1870.

Clamp,M. (1998) Jalview: a java multiple alignment editor: http://www.ebi.ac.uk/~michele/jalview/.

Clark,J. (2001) NSGMLS: an SGML system conforming to international standard ISO 8879—Standard Generalized Markup Language http://www.jclark.com/sp/nsgmls.htm.

Dingle,K.E., Colles,F.M., Wareing,D., Ure,R., Fox,A.J., Bolton,F.E., Bootsma,H.J., Willems,R.J., Urwin,R. and Maiden,M.C. (2001) Multilocus sequence typing system for *Campylobacter jejuni*. *J. Clin. Microbiol.*, **39**, 14–23.

Enright,M.C. and Spratt,B.G. (1998) A multilocus sequence typing scheme for *Streptococcus pneumoniae*: identification of clones associated with serious invasive disease. *Microbiology*, **144**, 3049–3060.

Enright,M.C., Fenoll,A., Griffiths,D. and Spratt,B.G. (1999) The three major Spanish clones of penicillin-resistant *Streptococcus pneumoniae* are the most common clones recovered in recent cases of meningitis in Spain. *J. Clin. Microbiol.*, **37**, 3210–3216.

Enright,M.C., Day,N.P.J., Davies,C.E., Peacock,S.J. and Spratt,B.G. (2000) Multilocus sequence typing for characterisation of methicillin resistant and methicillin susceptible clones of *Staphylococcus aureus*. *J. Clin. Microbiol.*, **38**, 1008–1015.

Enright,M.C., Spratt,B.G., Kalia,A., Cross,J.H. and Bessen,D.E. (2001) Multilocus sequence typing of *Streptococcus pyogenes* and the relationship between emm-type and clone. *Infect. Immun.*, in press.

Feil,E.J., Maiden,M.C.J., Achtman,M. and Spratt,B.G. (1999) The relative contributions of recombination and mutation to the divergence of clones of *Neisseria meningitidis*. *Mol. Biol. Evol.*, **16**, 1496–1502.

Feil,E.J., Maynard Smith,J., Enright,M.C. and Spratt,B.G. (2000) Estimating recombinational parameters in *Streptococcus pneumoniae* from multilocus sequence typing data. *Genetics*, **154**, 1439–1450.

Holmes,E.C., Urwin,R. and Maiden,M.C.J. (1999) The influence of recombination on the population structure and evolution of the human pathogen *Neisseria meningitidis*. *Mol. Biol. Evol.*, **16**, 741–749.

Jolley,K.A., Kalmusova,J., Feil,E.J., Gupta,S., Musilek,M., Kriz,P. and Maiden,M.C. (2000) Carried meningococci in the Czech Republic: a diverse recombining population. *J. Clin. Microbiol.*, **38**, 4492–4498.

Letondal,C. (2001) A web interface generator for molecular biology programs in unix. *Bioinformatics*, **17**, 73–82.

MacLoed,K. (2000) XML::Parser::PerlSAX—Perl SAX parser using XML::Parser. http://bitsko.slc.ut.us/libxml-perl/XML%3A%3AParser%3A%3APerlSAX.html.

Maiden,M.C.J., Bygraves,J.A., Feil,E., Morelli,G., Russell,J.E., Urwin,R., Zhang,Q., Zhou,J., Zurth,K., Caugant,D.A., Feavers,I.M., Achtman,M. and Spratt,B.G. (1998) Multilocus sequence typing: a portable approach to the identification of clones within populations of pathogenic microorganisms. *Proc. Natl Acad. Sci. USA*, **95**, 3140–3145.

Momjian,B. (2001) *PostgreSQL: Introduction and Concepts*. Addison-Wesley, Reading, MA.

Rice,P., Longden,I. and Bleasby,A. (2000) EMBOSS: the European Molecular Biology Open Software Suite. *TIG*, **16**, 276–277.

Rowe,J. (1999) WDBI—Web DataBase Interface. (http://www-stel.asu.cas.cz/wdbi/).

Sokal,R.R. and Michener,C.D. (1958) A statistical method for evaluating systematic relationships. *Univ. Kansas Sci. Bull.*, **28**, 1409–1438.

W3C (2000) Extensible Markup Language (XML) 1.0 (2nd edn). W3C Recommendation 6 October 2000. http://www.w3.org/TR/REC-xml.

Zhou,J., Enright,M.C. and Spratt,B.G. (2000) Identification of the major Spanish clones of penicillin resistant pneumoncocci via the internet using Multi Locus Sequence Typing. *J. Clin. Microbiol.*, **38**, 977–986.