@mattloose
matt.loose@nottingham.ac.uk

minoTour Installation and Set Up Guide

minoTour Web Version 0.47
minUp version 0.38

26 February 2015

Overview

minoTour consists of three key parts - 1) a mySQL database backend, 2) a php based website with support scripts and 3) a system for data upload - ‘minUp’. Initial setup of minoTour will take some time and you may require help from a system administrator or IT expert from your institution. We make certain assumptions about firewalls - in particular that your server is open on the mySQL port 3306. Please seek advice from your local IT support services as required. Once the system is up and running, the majority of features can be completed via the web interface or upload scripts (*some of these processes are still being developed*). The database should take care of itself. Note that minoTour is disk hungry - it will use a lot of drive space in storing data although runs can be archived after analysis to save space. We have made some improvements to this in this current version. Please see the new features of minUp for information. Excitingly, as nanopore technology improves our system becomes more stressed - we’ll keep trying to optimise and improve.

Please note that the PHP version should be >= 5.4. We are working on backwards compatibility…

There are three ways by which we recommend using minoTour. 1) Contact us (matt.loose@nottingham.ac.uk) and request an account on minoTour.nottingham.ac.uk - this requires very little work on your part! 2) Configure your own local server to run your version of minoTour. 3) Take a copy of our amazon machine instance for minoTour and run your own version in the cloud.

Upgrade Notes: If you are upgrading from a previous version of minoTour you should read these instructions still. There are changes to the new version which will require attention. These will be highlighted in the notes below as 'Upgrade Notes:'.

Set up a local installation of minoTour.

We will walk you through configuring each component of the system. We assume some basic system knowledge on your part. minoTour can be installed in a variety of different configurations - here we describe a scenario where the website and mySQL database are on the same computer and provide two means by which data can be uploaded -

either from the windows machine connected to the minION device or from a secondary computer.

Included Files

This minoTour installation set has been downloaded from GIThub. You should have a number of folders. "mT_web" contains the website itself. "mT_server" includes a number of scripts and programs that facilitate creating the necessary databases for minoTour to function. These also enable a number of important background processes that run on the server and will be developed in the future.

---

1) mySQL Database - Gru and friends…

The database works as follows. Each new run is added to its own unique database. This allows for relatively simple management of run data and shifting of databases to different locations in the future. To manage these databases, a master database is created (‘Gru’ - it controls the minIONs…). Gru allocates databases to users, controls access to the website and is absolutely essential to the operation of minoTour.

Upgrade Notes: *You already have the accounts configured for mySQL so you do not need to follow the steps below. However you must pay close attention to configuring the db.php file and others with your existing credentials.*

In order for this system to work, you create mySQL users that have permission to create and delete databases. This creates risks for your mySQL system and it is unlikely that any centrally managed facility will allow you to do this - you therefore need access to your own mySQL installation. To negate these risks, we create users that can only create and destroy databases which belong to them, defined by their own user names - this means that users cannot delete other databases in your mySQL installation - PHEW! To simplify the creation of these accounts, we have generated a number of scripts to automate this process (found in the folder 'mT_server/db_control/'). These scripts assume that you have access to the mySQL server and can connect to mySQL as root. You should ensure that the scripts have the necessary permissions to run on your server.

Firstly you need to create the basic database Gru and establish an account on mySQL that can be used by the website to show all the data (the ‘webuser'). You need to use the initialiseDB script and have the Gru.sql file (found in the 'mT_server/db_control/setup' folder).

This account has full access to all the databases created by the upload scripts. The website archive features require that the webuser account can drop tables and can access all users data. The datasets that can be seen by individual users is managed by the Gru database itself.

Assuming that you have root access to your mySQL install go to the command line and execute the initialiseDB script. You need to pass the script three variables. The username you wish to use for your webuser, the password and the ip address of the web server which will be connecting to the database. If your web server and mySQL database are on the same machine, use ‘localhost’ for this last option. The script will request your mySQL root password twice whilst it is running.

>./initialiseDB <webusername> <webpassword> <ip>

If this process has been successful you should now be able to log in to mySQL with the following command:

>mysql -u <webusername> -p

and provide the <webpassword> on request. Typing

mysql> show databases;

should list the presence of the Gru database.

The username and password created here will be used later to connect the minoTour website to your mySQL database.

Next you need to create accounts for users to actually upload data to the database. How you manage this process is up to you. We allow users to log in to the website and create an account for accessing the website choosing there own username. We then use this username to enable connection to the mySQL database. However these account names do not have to be the same.

IMPORTANT: *Do not use _ in the user name. This will break functionality elsewhere in the package.*

To create an account use the ‘createuser’ script (located in 'mT_server/db_control/admin'). This requires 3 arguments. The <username>, <userpassword> and the <ip> address from which they are connecting to your mySQL server. This is an important security feature as it limits the possibility of malicious connection to your database. To add additional IP addresses in the future we provide the script addusersource (see below). The simplest option is to use the wildcard symbol %. This will allow the user to connect from any computer anywhere. It will also resolve an issue where the machine connecting to the database does not have a fixed IP address. To create a user account execute the ‘createuser’ script. The username you use here must be associated with an account on the web site. e.g If Joe Bloggs creates a user account on the website called JB123, his account for uploading data in mySQL should also be called JB123.

>./createuser <username> <userpassword> <ip>

You will need to communicate the <userpassword> to the user directly. There is no way that they can find this information out otherwise. The option to change the user password on the website will not allow alter the mySQL password set here.

To add additional connection points for this user run the addusersource script providing the username and the additional IP address.

>./addusersource <username> <new_ip>

If you need to delete a user then run the dropuser script. This requires only the username to prevent any access to the database for that user. This does not remove any data OR prevent the user from seeing the data in the minoTour website.

>./dropuser <username>

Having done this, we shall now configure the website.

Important: *we will return to mySQL to configure our admin account shortly.*

---

2. minoTour WebSite installation.

Configuring the website is straightforward. Copy the entire 'minoTour' folder located in the mT_web folder (which contains the web application) into the appropriate root folder for your webserver. In OSX this folder is usually called:

/Library/WebServer/Documents

In unix systems it may be

/var/www/html/

The name of the folder dictates the url for your website. You can change this if you wish.

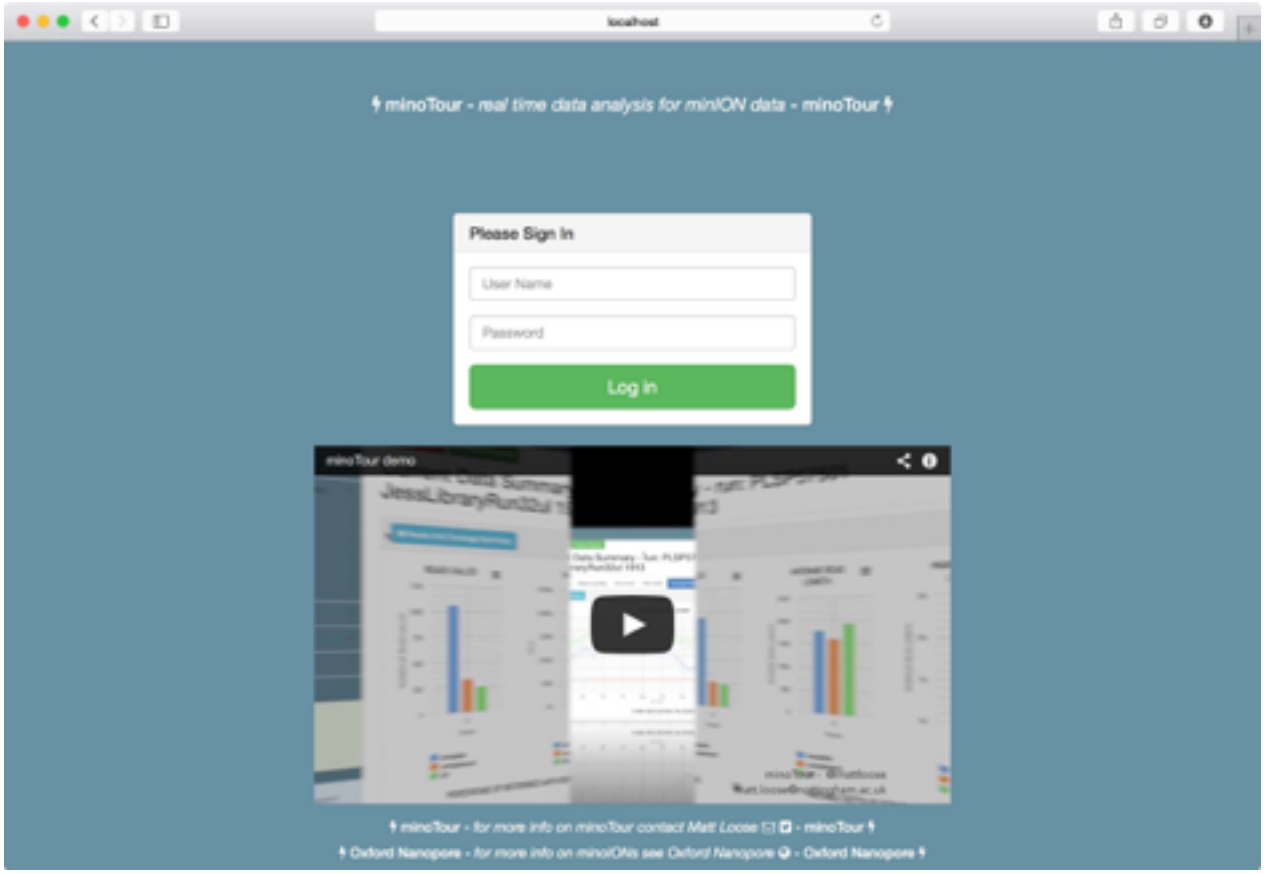
Upgrade Notes: *You may wish to install your new version of minoTour in place of the previous version. We would suggest first setting the site up in an alternative location and checking the configuration before writing over your previous installation. The new version of minoTour includes a 'previous versions' tab which will allow you to go back and use older versions of minoTour. Most importantly you need to keep note of the DB-HOST and DB_PASS values from the db.php file in the config folder and ensure these values are used in the new minoTour install.*

You may need to use 'sudo' privileges to copy this folder into the appropriate location.

Now - assuming you are on the same machine as your webServer and have not changed the folder name, open a browser window and navigate to:

http://localhost/minoTour

And you should see a nice welcoming page… - *if you don't - check that you have php and apache correctly configured on your machine. Google is your friend…*



If you do not see the video, then your webServer is unable to connect to the outside world. This isn't critical but will limit some functionality in our system. In particular it will prevent you submitting bug reports and feature requests via the website to us.

Before going any further, you need to configure the website to talk to your database.

To do this, navigate to the folder 'config' which is located within the minoTour folder. In this folder is a file called 'db_example.php'.

Open this file with a text editor and you will see lines including:

```
define("DB_HOST", "127.0.0.1");
define("DB_NAME", "Gru");
define("DB_USER", "");
define("DB_PASS", "");
define("DB_PORT", 3306);
```

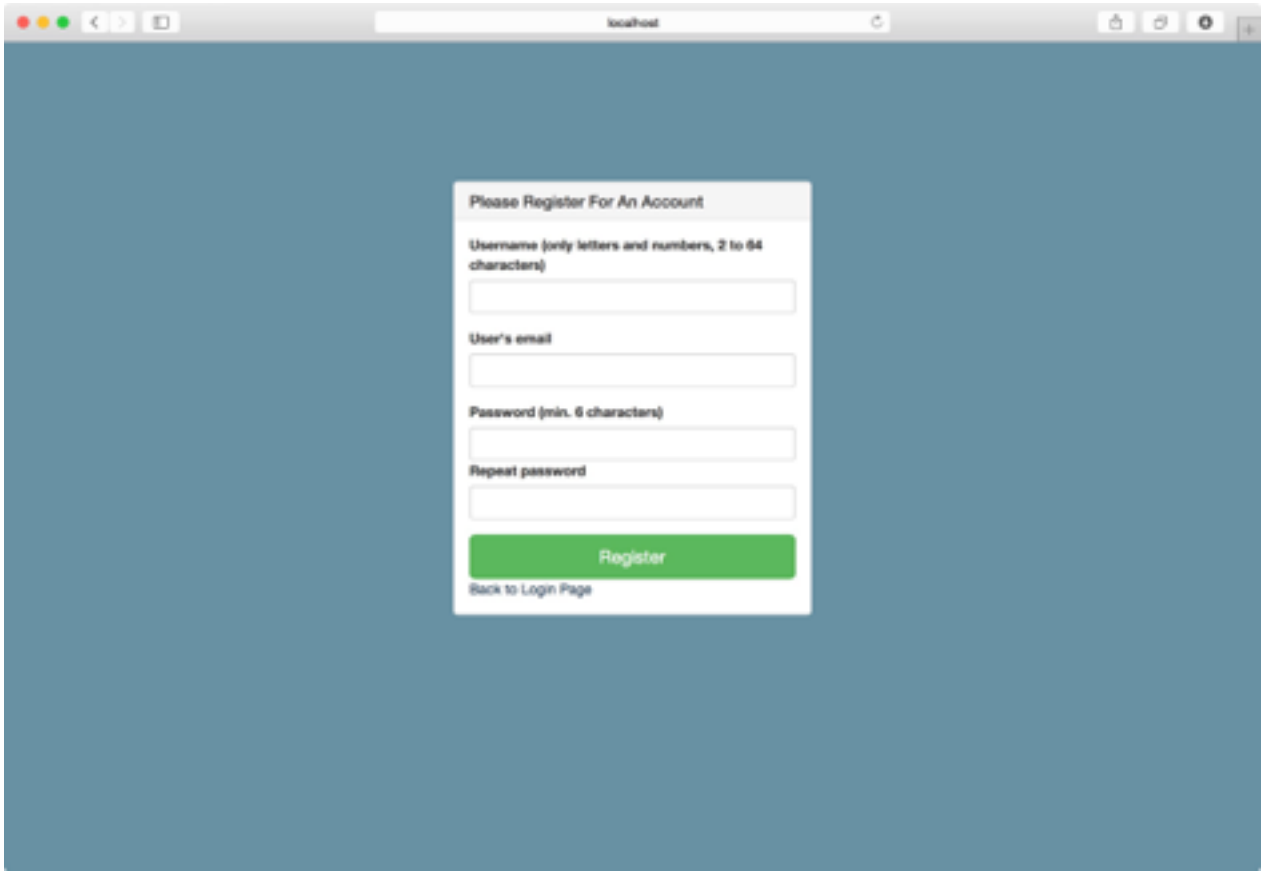
If your database is on the same machine as your webserver, leave DB_HOST set to "127.0.01". If this doesn't work, switch it to "localhost". If your mySQL database is on a remote machine, then enter the IP address of that machine here. Then set DB_USER

and DB_PASS to the user name and password that you used when setting up the database above. If you have mySQL configured on a non standard port you can set that here. **When you save this file, save it as db.php in the config folder**.

Important: *Each time a new release of minoTour is made available you should check the db_example.php file to see if new global parameters have been established.*

Now return to your web browser - we will make an account for you to access the site - to do so go to register_new.php .

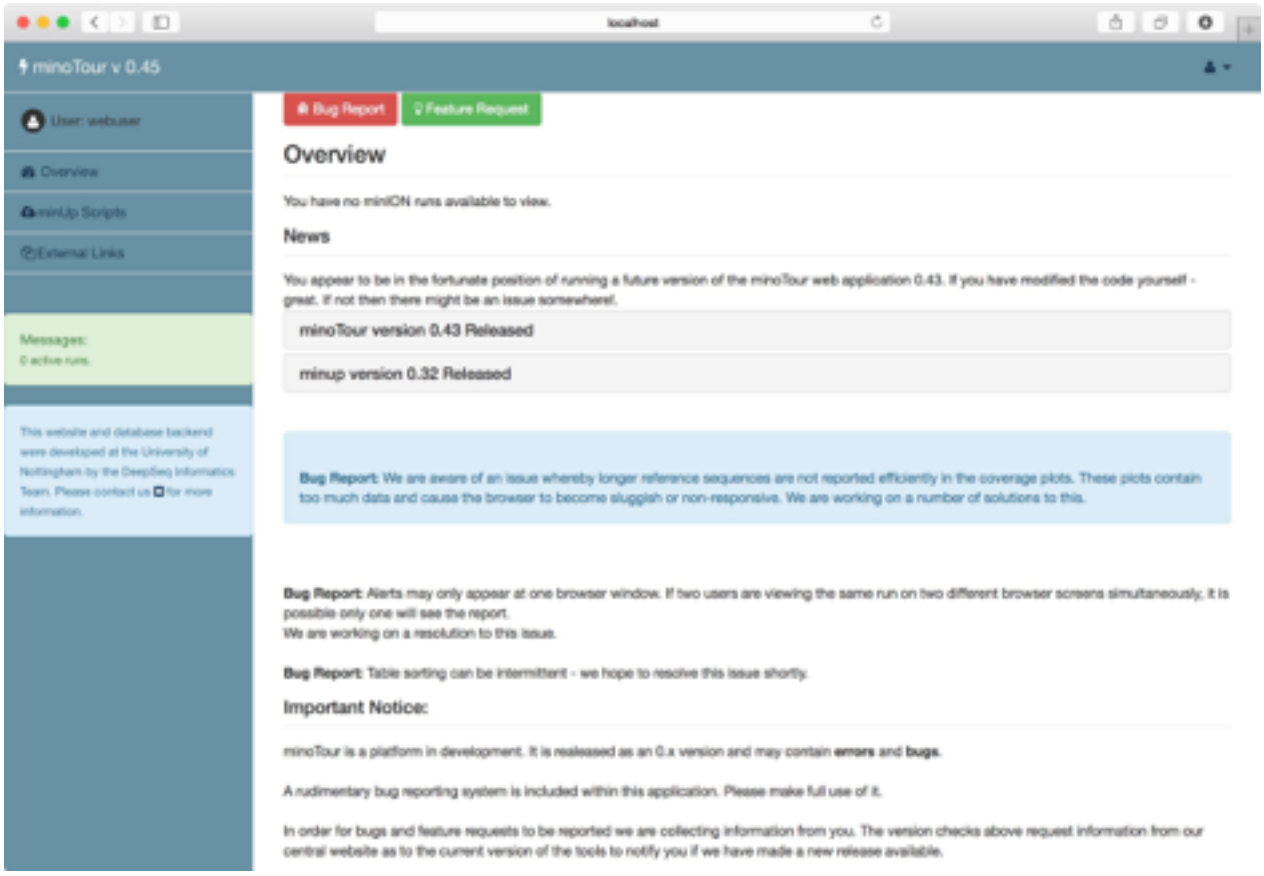
Important: *For minoTour versions 0.43 and beyond we have removed the publicly accessible registration page. This prevents malicious account creation. To find the page to register a new user go to minoTour/register_new.php rather than index.php .*



This first account is important as we are going to make it the system administrator. Use your choice of username and password. Providing a valid email address is important as it allows us to contact you in the event of bugs or other problems. Complete the form and register away.

If all has worked well you will be rewarded with a success message… Return to the login page and use your shiny new account to log in.

You should see a page looking a bit like this (with 0.47 showing, not 0.45!):



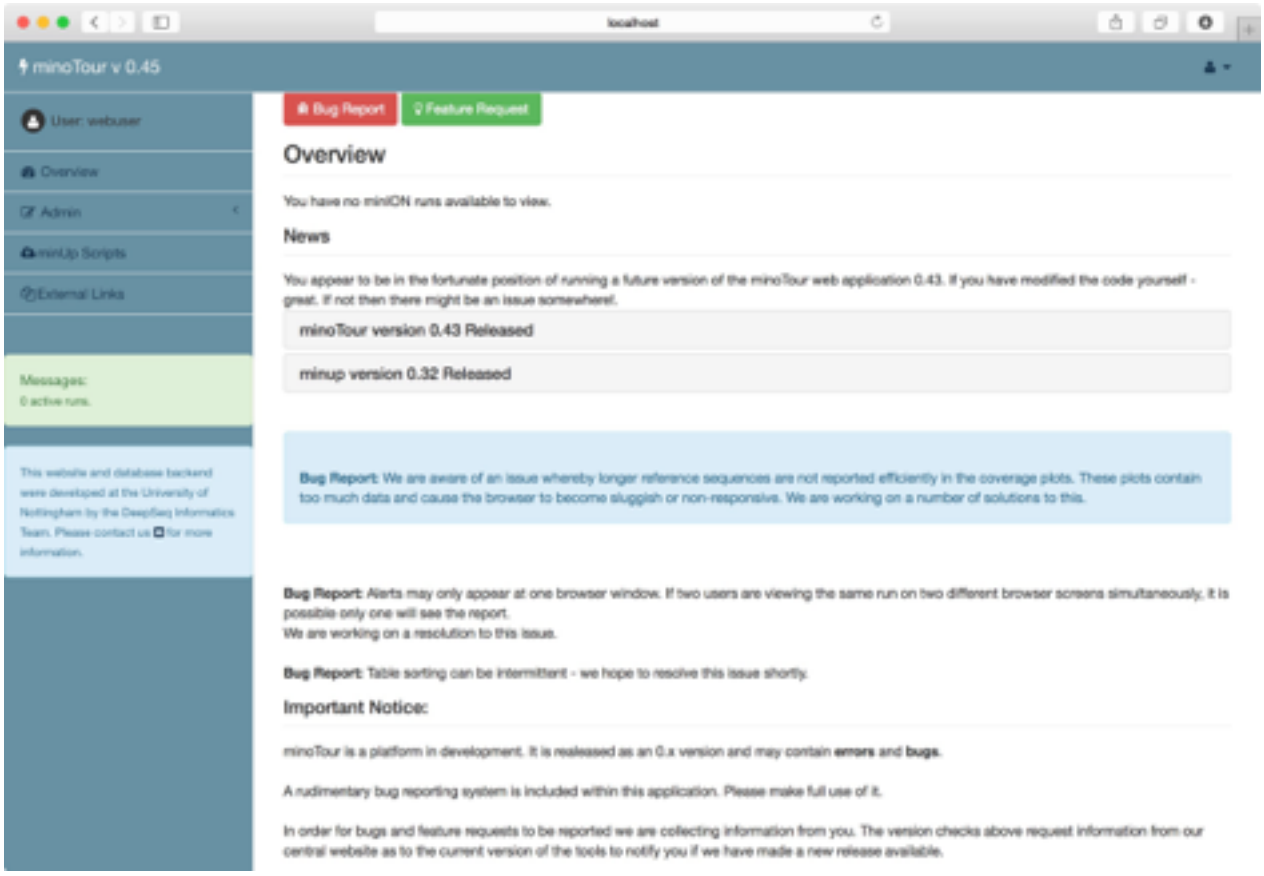
If your machine is on a network where it can see the outside world, you will have the news items listed. If you do, then the bug reporting and feature requests will work. If you do not, then these buttons won't function for you.

The last thing we need to do is set this account as an administrator. To do that, we go back to mySQL - log in using your mySQL account - in our case webaccess. Follow these instructions replacing the minoTouruser with the user name of your choice:

```
mysql> USE Gru;

mysql> UPDATE USERS SET admin=1 WHERE user_name =
'minoTouruser';
```

Now return to the website and refresh the page… Assuming all has gone to plan you will have an admin account and the page will look like this:



You should see a new ‘Admin’ menu has appeared. Currently the Admin processes are limited - you can allocate databases to different users. The second page 'cache administration' allows you to check if the new optimised features of minoTour are working correctly.

Update Notes: *This is a new feature of minoTour that dramatically speeds up displaying data. You should follow these instructions.*

If you click on 'cache administration' you will see a message that minoTour is currently not using memcached. Memcached is a unix tool that allows a website to store parameters in memory on the server. We use this to dramatically improve performance. You will need to have memcached installed on your host to use this service. We also provide background scripts that allow the server to process data.

To configure memcached in an Ubuntu environment type (you should modify as appropriate for your flavour of unix):

```
sudo apt-get install memcached php5-memcache
```

Depending on the size of your minoTour install you can dedicate more or less memory to memCached. We recommend at least 64Mb but larger installations with more users may benefit from more memory. To configure the memory allocated to memcached see the man pages for your particular flavour of unix/memcached.

After configuring memcached correctly you should see the message has changed on your 'cache administration' page to indicate that minoTour can see the memcache.

The perl scripts we are going to run in the background require the memcached-perl library - this can be installed on Ubuntu by typing:

```
sudo apt-get install libcache-memcached-perl
```

We now need to configure these scripts to work with your installation. As with the web site there is a central configuration file located in the 'nefario' folder within the 'mT_server' folder. The example configuration file is called mT_param.example. Open this in a text editor and complete each line as follows:

directory - is the path to the folder containing your minoTour website. Do not forget the trailing '/' e.g. /var/www/html/minoTour/

memcahce - is the memcache ip and port - this will work as 127.0.0.1:11211 unless you have changed your memcache install.

dbhost - the database host ip

dbuser - the database username you configured on page 2 of this document

dbpass - the database password you configured on page 2 of this document

phploc - the path to php to execute php from the command line

Now you can set the background processes running by executing the command:

```
perl mT_control.pl
```

In order to run this in the background you need to use nohup or an equivalent methodology - e.g.

```
nohup perl mT_control.pl &
```

If you return to the website and 'cache administration' you should now see a message that your memcache is working and the backend scripts are running.

Great. Nearly there… Now to upload some data…

---

3. a) minupv.0.38W.exe - Data Upload - Windows

We have a version of the minup script which has been compiled to operate on windows. This is available as the file minup.v0.38W.zip and can be downloaded from the website itself from the minUp Scripts menu option. This comes with a windows compatible version of the Last Aligner and is easier to use than the linux mounting system suggested in 3b) and 4 below. However it is unable to take advantage of GNU-PARALLEL for speeding up the last analysis but does seem to perform very well on the Windows machine as specified by Oxford Nanopore.

@mattloose
matt.loose@nottingham.ac.uk

To install this software, download the folder minup.v0.38W.zip from the website from under the 'minUp Scripts' menu and decompress into a folder in C:\ . Alongside our program you will also need to install CygWin available from https://www.cygwin.com . Once you have installed CygWin you must ensure that the CygWin bin folder containing the file cygwin1.dll is added to your windows path. The command for doing this in command.exe is:

```
set PATH=%PATH%;C:\cygwin64\user\bin\
```

This will allow our minup program to access the linux like commands it requires to function. You should then also add the folder you have downloaded from us to your path. Assuming you decompressed the folder and placed it at the root of the C: you would type the following:

```
set PATH=%PATH%;C:\minupv0.38W\
```

Within this folder are three key programs. minup.v0.38W.exe is the windows data uploader. lastal.exe is the last aligner compatible with windows, and lastdb.exe formats the last database.

To check your installation, try each of these programs.

```
C:\lastal
```

will generate the following output:

```
lastal: please give me a database name and sequence file(s)

Usage: lastal [options] lastdb-name fasta-sequence-file(s)
```

Then:

```
C:\lastdb
```

will generate the following output:

```
lastdb: please give me an output name and sequence file(s)

Usage: lastdb [options] output-name fasta-sequence-file(s)
Prepare sequences for subsequent alignment with lastal.

Main Options:
-h: show all options and their default settings
-p: interpret the sequences as proteins
-c: soft-mask lowercase letters
```

Finally minup.v0.38W.exe:

```
C:\minup.v0.38W -h
```

Which will generate the following output:

```
usage: minup.v0.38.py [-h] [-dbh DBHOST] -dbu DBUSERNAME [-dbp DBPORT] -pw
                      DBPASS [-f REF_FASTA] -w WATCHDIR [-n THREADS] -u
                      MINOTOURUSERNAME [-s VIEW_USERS] [-o FLOWCELL_OWNER]
                      [-r RUN_NUM] [-c] [-m] [-a] [-t] [-d] [-v]

minup: A program to analyse minION fast5 files in real-time or post-run. Args
that start with '--' (eg. --mysql-host) can also be set in a config file (minup_posix
.config ) by using .ini or .yaml-style syntax (eg. mysql-host=value). If an
arg is specified in more than one place, then command-line values override
config file values which override defaults.

optional arguments:
  -h, --help            show this help message and exit
  -dbh DBHOST, --mysql-host DBHOST
                        The location of the MySQL database. default is
                        'localhost'.
  -dbu DBUSERNAME, --mysql-username DBUSERNAME
                        The MySQL username with create & write privileges on
                        MinoTour.
  -dbp DBPORT, --mysql-port DBPORT
                        The MySQL port number, else the default port '3306' is
                        used.
  -pw DBPASS, --mysql-password DBPASS
                        The password for the MySQL username with permission to
                        upload to MinoTour.
  -f REF_FASTA, --align-ref-fasta REF_FASTA
                        The reference fasta file to align reads against. Using
                        this option enables read alignment provided LastAl and
                        LastDB are in the path. Leaving this entry blank will
                        upload the data without any alignment. To use multiple
                        reference fasta files input them as one text string
                        seperated by commas (no white spaces)
  -w WATCHDIR, --watch-dir WATCHDIR
                        The path to the folder containing the downloads
                        directory with fast5 reads to analyse - e.g.
                        C:\data\minion\downloads (for windows).
  -n THREADS, --aligning-threads THREADS
                        The number of threads to use for aligning
  -u MINOTOURUSERNAME, --minotour-username MINOTOURUSERNAME
                        The MinoTour username with permissions to upload data.
  -s VIEW_USERS, --minotour-sharing-usernames VIEW_USERS
                        A comma seperated list (with no whitespaces) of other
                        MinoTour users who will also be able to view the data.
  -o FLOWCELL_OWNER, --flowcell-owner FLOWCELL_OWNER
                        The name of the minion owner. 'minionowner' is the
                        default
  -r RUN_NUM, --run-number RUN_NUM
                        The run number of the flowcell. The default value is
                        0.
  -c, --commment-true   Add a comment to the comments field for this run.
                        Follow the prompt once minup starts .
  -m, --upload-maf-true
                        Upload MAF format alignment data.
  -a, --upload-align-true
                        Upload long alignment data.
  -t, --insert-tel-true
                        Store all the telemetry data from the read files
                        online. This feature is currently in development.
  -d, --drop-db-true    Drop existing database if it already exists.
  -v, --verbose-true    Print detailed messages while processing files.
```

To reduce the complexity of running minUp, a customised config file can be generated by the website. Note that the config file for the Windows version of minup is slightly different to the linux version. The file should be placed in the same folder as the minup.v0.38W.exe file.

You are free to edit this file and change the parameters or add common parameters (such as you're usual shared users as an example).

If you are running minup on the same machine as your mySQL database you may need to modify the config file to set the host name to 'localhost' or equivalent.

To test your installation we have created a small sample of data from the recently released Loman Lab dataset (http://dx.doi.org/10.5524/100102). This sample set consists of just the first 100 or so reads from the run. It is also available from the web site itself. Decompress the demo_data_set.zip folder to a location on your machine. The folder structure is important here. metrichor returns files to a folder called ‘downloads’ and minup looks for this folder in any location you point it at - so keep the folder structure. We also include a copy of the reference genome in this dataset.

To upload this data type the command minup.v0.38W at the command prompt and complete the options as appropriate - for example:

C:\minup.v0.38W -dbh <your_hostname_or_ip> -dbu <minoTouruser> -pw <PASSWORD1> -u <minoTouruser> -f /<PATH>/<TO>/<YOUR>/<COPY>/<OF>/demo_data_set/reference/EcoliMG1655.fasta -w /<PATH>/<TO>/<YOUR>/<COPY>/<OF>/demo_data_set/ -o Joe_Bloggs -r 1 -m -c -t

You will be asked to enter a comment (due to the -c option set above) describing your run. This can be a brief note on the run as you set it up and will be associated with the run record in minoTour.

If this works you will see the following messages in command.exe…

```
processing the reference fasta.
finished processing reference fasta.
monitor started.
2014-09-23 10:48:19  CACHED:  101 PROCESSED:  0
...
2014-09-23 10:49:26  CACHED:  0 PROCESSED:  101
2014-09-23 10:49:36  CACHED:  0 PROCESSED:  101
```

Now jump over to the website, refresh the page, click on ‘Current Sequencing Run’ and have a look at the data. To end the processing step, just hit CTRL-C…

If this hasn’t worked, then give us a shout… @mattloose or matt.loose@nottingham.ac.uk!

You are now up and running and will be able to upload old run data (just point the —watch-dir to wherever you store your old reads) or perform live read uploads. You can ignore the rest of this manual!

3. b) minup python script - Data Upload - Linux/OsX

We also provide a unix version of minUp. To load your first data set into minoTour we have created a small sample of data from the recently released Loman Lab dataset (http://dx.doi.org/10.5524/100102). This sample set consists of just the first 100 or so reads from the run. Decompress the demo_data_set.zip folder to a location on your machine. The folder structure is important here. metrichor returns files to a folder called 'downloads' and minup looks for this folder in any location you point it at - so keep the folder structure. We also include a copy of the reference genome in this dataset.

minUP is a python script (the current version is minup.v0.38.py) and it has a number of prerequisites. Firstly decompress the file - then to determine what you need to install type:

```
>python minup.v0.38.py
```

If you have missing prerequisites you can install them using 'pip install' (if you don't have pip-install then hit sudo apt-get install python-pip - you might also need sudo apt-get install python-dev - oh and you will probably need sudo apt-get install python-dev libmysqlclient-dev). A complete list of the prerequisites is provided here:

watchdog
h5py
cython
biopython
simplemysql
configargparse

Note: for simplemysql on MAC OSX you may need to create a symbolic link from /usr/local/mysql/lib/libmysqlclient.18.dylib to /usr/lib/. See http://stackoverflow.com/questions/6383310/python-mysqldb-library-not-loaded-libmysqlclient-18-dylib for more information.

In addition, minUP uses the LAST aligner and assumes that it is in the path. To install the LAST ALIGNER please go to http://last.cbrc.jp and follow the instructions. You also need to install GNU-Parallel - see http://www.gnu.org/software/parallel/ for info.

Once minUP is correctly configured you will see a list of all the parameters for minUP printed to the screen:

```
>python minup.v0.38.py - h

usage: minup.v0.38.py [-h] [-dbh DBHOST] -dbu DBUSERNAME [-dbp DBPORT] -pw
                      DBPASS [-f REF_FASTA] -w WATCHDIR [-n THREADS] -u
                      MINOTOURUSERNAME [-s VIEW_USERS] [-o FLOWCELL_OWNER]
                      [-r RUN_NUM] [-c] [-m] [-a] [-t] [-d] [-v]

minup: A program to analyse minION fast5 files in real-time or post-run. Args
that start with '--' (eg. --mysql-host) can also be set in a config file (minup_posix
.config ) by using .ini or .yaml-style syntax (eg. mysql-host=value). If an
arg is specified in more than one place, then command-line values override
```

```
config file values which override defaults.

optional arguments:
  -h, --help            show this help message and exit
  -dbh DBHOST, --mysql-host DBHOST
                        The location of the MySQL database. default is
                        'localhost'.
  -dbu DBUSERNAME, --mysql-username DBUSERNAME
                        The MySQL username with create & write privileges on
                        MinoTour.
  -dbp DBPORT, --mysql-port DBPORT
                        The MySQL port number, else the default port '3306' is
                        used.
  -pw DBPASS, --mysql-password DBPASS
                        The password for the MySQL username with permission to
                        upload to MinoTour.
  -f REF_FASTA, --align-ref-fasta REF_FASTA
                        The reference fasta file to align reads against. Using
                        this option enables read alignment provided LastAl and
                        LastDB are in the path. Leaving this entry blank will
                        upload the data without any alignment. To use multiple
                        reference fasta files input them as one text string
                        seperated by commas (no white spaces)
  -w WATCHDIR, --watch-dir WATCHDIR
                        The path to the folder containing the downloads
                        directory with fast5 reads to analyse - e.g.
                        C:\data\minion\downloads (for windows).
  -n THREADS, --aligning-threads THREADS
                        The number of threads to use for aligning
  -u MINOTOURUSERNAME, --minotour-username MINOTOURUSERNAME
                        The MinoTour username with permissions to upload data.
  -s VIEW_USERS, --minotour-sharing-usernames VIEW_USERS
                        A comma seperated list (with no whitespaces) of other
                        MinoTour users who will also be able to view the data.
  -o FLOWCELL_OWNER, --flowcell-owner FLOWCELL_OWNER
                        The name of the minion owner. 'minionowner' is the
                        default
  -r RUN_NUM, --run-number RUN_NUM
                        The run number of the flowcell. The default value is
                        0.
  -c, --commment-true   Add a comment to the comments field for this run.
                        Follow the prompt once minup starts .
  -m, --upload-maf-true
                        Upload MAF format alignment data.
  -a, --upload-align-true
                        Upload long alignment data.
  -t, --insert-tel-true
                        Store all the telemetry data from the read files
                        online. This feature is currently in development.
  -d, --drop-db-true    Drop existing database if it already exists.
  -v, --verbose-true    Print detailed messages while processing files.
```

We hope this is fairly self explanatory, but to be absolutely clear… Optional parameters are in square brackets. The minotour-username parameter is the database username created in step 1. The password is the mySQL password from that step, NOT the website password you created in step 2. An example command which will work with the test dataset is provided below.

As with the windows version, a customised config file can be generated from the website. It needs to be placed in the same folder as the minup.v0.38.py script.

If you are running minup on the same machine as your mySQL database you may need to modify the config file to set the host name to 'localhost' or equivalent.

```
>python minup.v0.38.py -dbh localhost -dbu minoTouruser -pw
PASSWORD1 -u minoTouruser -f /<PATH>/<TO>/<YOUR>/<COPY>/<OF>/
demo_data_set/reference/EcoliMG1655.fasta -w /<PATH>/<TO>/<YOUR>/
<COPY>/<OF>/demo_data_set/ -o Joe_Bloggs -r 1 -c -m -t
```

Note: --run-number refers to libraries running on the same flow cell - if you are loading a second library after washing you may wish to know this fact. To prevent minoTour aligning to a reference simply omit the -f option.

Note 2: If something goes wrong and you wish to re-upload the same data, you have to give minup permission to delete the old database. If you don't then you will receive an error about duplicate entries. To delete the old database, add the option --drop-db-true - or -d to the minup command.

You will be asked to enter a comment (due to the -c option set above) describing your run. This can be a brief note on the run as you set it up and will be associated with the run record in minoTour.

If this works you will see the following messages in the terminal…

```
processing the reference fasta.
finished processing reference fasta.
monitor started.
2014-09-23 10:48:19  CACHED:  101 PROCESSED:  0

…
2014-09-23 10:49:26  CACHED:  0 PROCESSED:  101
2014-09-23 10:49:36  CACHED:  0 PROCESSED:  101
```

Now jump over to the website, refresh the page, click on 'Current Sequencing Run' and have a look at the data. To end the processing step, just hit CTRL-C…

If this hasn't worked, then give us a shout… @mattloose or matt.loose@nottingham.ac.uk!

General comments on minUp

A number of new options are available in minUp. We have simplified the upload of alignment data. If you are aligning your reads to a reference you should specify the -m option to upload the alignment data. This sends a compressed maf alignment to the database which is subsequently processed by the mT_control scripts described above. The -a upload option for alignments remains from previous versions of minoTour but will be removed in future.

The -c comment field enables you to add a notes on a run. These can be added to and amended on the new 'run reports' option. Comments added via the browser are persistent even if you delete and re-upload a run so you can use them to track the history of a data set.

The -t field enables the upload of telemetry on the run to the database. This includes the ability to look at squiggles from the underlying reads. It will also enable kmer analysis and other features on the minoTour development roadmap. We strongly encourage you to use this option as standard.

I apologise for the length of these instructions. If you have read this far - congratulations and thanks for using minoTour. Please don't hesitate to get in touch with questions - we'll try our best to respond as quickly as possible.

Appendix

Configuring Ubuntu

If you wish to install minoTour to Ubuntu linux here are the instructions for configuring minoTour on a bare bones Ubuntu 14.04 install.

Firstly make sure you have ports open for http, https, ssh and mysql on your server. Then…

(Note - you really should set a password for the mySQL root user when prompted to at this point…you will use this later on.)

Step 1 - LAMP Installation and memcached

:-$ sudo apt-get update
:-$ sudo apt-get install lamp-server^
:-$ sudo apt-get install memcached php5-memcache

Configure the perl environment to run background scripts

You need to install the perl memcached module

```
sudo apt-get install libcache-memcached-perl
```

```
You should also install the following perl module (#idiot check - you may need
to install make if it isn't already on your system… sudo apt-get install
make )

Parallel-Loops
```

You should now be able to see a default ubuntu webpage if you browse to your server IP or URL. If not - reboot your system and try again…

Step 2 - Configure mySQL for minoTour.

If you are using `ubuntu`, you have to use the following steps to enable mySQL to communicate with the outside world:

1. run the command `vim /etc/mysql/my.cnf`
2. comment `bind-address = 127.0.0.1` using the # symbol
3. restart your mysql server once.

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

At this point you should be good to go following the instructions outlined above.

Setting up your own Amazon instance.

minoTour will run on the free tier of amazons EC2 hosting service. We therefore provide an Amazon Machine Instance which is partially configured. By following the instructions below you should be able to set up your own working minoTour in the cloud.

Please note that you have to be able to connect out from your institution on port 3306 for this to work.

The AMI ID is ami-058b0972 and you can search for it as “minoTour” - this will return two instances - an old and current version - you need the new one (minoTour_v0.47).

We have been testing this on a single processor with 1 gig ram and 30 gig SSD (the free tier) - obviously go for whatever budget and common sense seems reasonable to you!

Once you have the AMI up and running you will need to follow these steps to configure the databases. We have not specified any default accounts on the minoTour version on this AMI so it can be linked securely to your own passwords. There is no back door built in for us. However the mySQL root password has been left blank. We recommend that you set a root password.

1. Go to

/home/ubuntu/minoTour_install/mysqlscripts

In this folder are a number of scripts. The first to execute is initialiseDB - this will setup a blank database to control minotour and will also create your username and password account for connecting the website to the database. It requires three arguments in this order - username password source - where the username and password are the account that will be used to connect the website to the database. Source will be localhost unless you are setting up a system with the web server and mySQL databases on different machine.

Therefore type something like

./initialiseDB webuser webpass localhost

Note that the default root password for mysql has been left blank on this install - when you are asked for a password by this script type nothing and just hit return. You will be asked for the password twice - this is expected behaviour.

2. Configure the website - go to

/var/www/html/minoTour/config/

You need to edit the file 'db.php'. There are a number of parameters in here, but the key two are DBUSER and DBPASS. These should be set to the username and password you specific at initialiseDB above.

You also need to configure the new background processes - these are found in

/home/ubuntu/minoTour_install/nefario

The first to edit is mt_Control.pl - in this file is a line beginning "my $dbh = " - you need to replace the values #DB_USER and #DB_PASS with the username and password from the initialiseDB command above (webuser and webpass) in this example.

A similar line is found in mT_align.pl - here it begins with my $dbh2 - you need to again replace the values of #DB_USER and #DB_PASS with the username and password from the initialiseDB command above.

Once this is done you should test that the website is working - go to http:/your.server.address/minoTour/ and you should see the index page. (note the capital T in the middle of minoTour).

You should see the login page.

To create a new account to use minoTour go to http:/your.server.address/minoTour/register_new.php and create an account. This can be whatever username you wish - just don't forget it!

This should work if you database connection is set up correctly. Now you can log in to minoTour using this account.

Now we want to turn this user into an admin user.

To do so, log in to mySQL on the server using the root account (remember it has no password).

Then:

```
mysql> use Gru;
mysql> update users set admin =1 where user_name = "username";
```

Username here will be the user you just created on the minoTour website.

Almost there now!

Final step is to create a mySQL account to upload data from.

To do this go back to the mySQL scripts directory:

/home/ubuntu/minoTour_install/mysqlscripts

Now run the createuser script. Again this takes three arguments, username password source - the username should be the same username as the minoTour account name that will upload the data. The password can be the same if you wish but does not have to be. The most secure source address is the IP address of the computer minion data will be uploaded from, but otherwise you can just use the % wildcard to upload from anywhere.

Now - to test if everything is working…

First start the processing script on the server - go to

/home/ubuntu/minoTour_install/nefario

Execute mT_control.pl remembering to use nohup:

nohup perl mT_control.pl &

You can check if this is running by going to the website, clicking on the admin option in the left hand menu, and selecting Cache Administration. If everything is working as it should you see a message like:

This minotour installation is using memcache.
Congratulations, you have memcache up and running and the perl script is communicating well with your web backend!

If so you should now try uploading data. There is a new version of minup which can be downloaded now directly from the website. See the minup tab for details.

Note that typing

python minup.v0.38.py -h

will give you a detailed help file for this new version. There are a number of new options. The most important are:

-t, --insert-tel-true
Store all the telemetry data from the read files
online. This feature is currently in development.
-m, --upload-maf-true
Upload MAF format alignment data.
-c, --commment-true Add a comment to the comments field for this run.
Follow the prompt once minup starts .

The -t option allows you to view raw trace data in minoTour.

@mattloose
matt.loose@nottingham.ac.uk

The -m option is our preferred method for uploading alignment data. It is a lot faster than the previous -a option (which still remains in this version but is likely to be removed in future).
The -c option allows you to enter a comment to annotate your run. Please avoid long comments and/or punctuation that may represent script commands. If a run fails to upload, try removing the -c option. You can now add comments to a run through the web browser as well under the run report menu item for a single run. Comments added via the browser are persistent even if you delete and re-upload a run so you can use them to track the history of a data set.