



## BIONUMERICICS Tutorial:

# Setup and analysis using a publicly available MLST scheme

## 1 Introduction

---

In this tutorial, we will illustrate the most common usage scenario of the *MLST online plugin*, i.e. when you want to perform MLST using an already published MLST schema, made available via <http://pubmlst.org> or <http://www.mlst.net> or any other online repository. For many clinically relevant organisms, an MLST schema is already available and using this schema ensures a consistent nomenclature.

## 2 Creating a new database

---

To illustrate the complete setup of the *MLST online plugin* for using a publicly available MLST schema, we will start by creating a new, empty database.

1. Double-click on the BIONUMERICICS icon () on the desktop.
2. In the *BIONUMERICICS Startup* window, press the  button to enter the *New database* wizard.
3. Enter a database name, e.g. "Neisseria database".
4. Click <**Next**>.

A new dialog box pops up, asking whether to create a new relational database for data storage or to use an existing one.

5. Leave the default option **Create new** enabled and press <**Next**>.

The next dialog asks which database engine should be used for storing data.

6. Select the default option and press <**Finish**>.

The *Main* window opens with an empty database.

### 3 Installing the MLST online plugin

In this section we will install the *MLST online plugin* in our database and we will set it up to use the publicly available *Neisseria* MLST schema.

1. The *Plugins* dialog box is called from the *Main* window by selecting **File > Install / remove plugins...** (I/P).
2. Select the *MLST online plugin* from the list in the *Applications* tab and press the **<Activate>** button.

The next dialog asks to confirm the installation of the *MLST online plugin*. Installation of the plugin requires administrator privileges on the relational database.

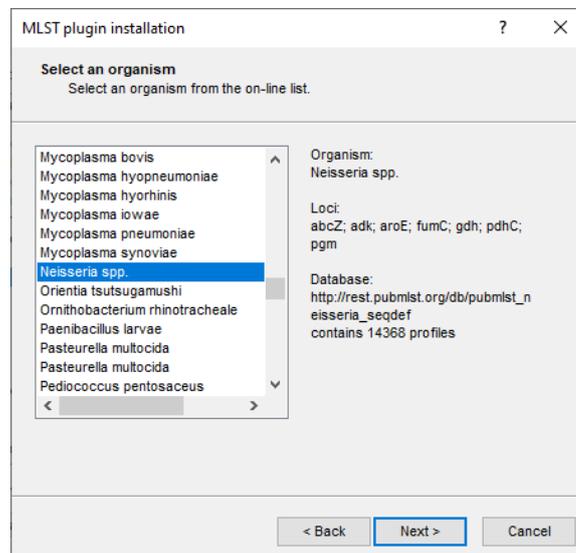
3. Confirm the installation of the *MLST online plugin*.

We are asked to select the organism source. In the next section we will import sequences from the organism *Neisseria* in our sample database. Since this organism has an MLST repository online we will use this online database to retrieve the allele, sequence type and clonal complex information for our sample data set.

4. Choose the option **Select organism from on-line list** and press **<Next>**.

Any organism for which an MLST repository is available online, is listed in the next step.

5. Select "Neisseria spp." from the list and press **<Next>** (see Figure 1).



**Figure 1:** Select organism from the list.

It is possible to specify a location for the profile and allele definition files, which can be located on your own computer, local area network or on the internet. However, the default location should point to the correct files already. The option **Update profiles and alleles at database startup** can also be checked, to avoid having to do a manual update.

6. Press **<Next>** to continue.
7. In the next step, leave **Calculate trimming patterns automatically** checked and press **<Next>**.

In the final step, the program prompts for database information fields to store the **Sequence types** and **Clonal complexes** information.

8. For this exercise, use the default "MLST ST" and "MLST CC" fields and press <Next>.

9. Pressing <Finish> starts with the installation of the *MLST online plugin*.



All remotely stored MLST information (allele numbering, sequence types and clonal complexes) for the selected organism will be downloaded in the BIONUMERICS database during the installation of the plugin. This might take several minutes. When querying for allele numbers, sequence types and clonal complexes in BIONUMERICS, this locally stored information will be used.

When the *MLST online plugin* is successfully installed, a confirmation message is displayed.

10. Press <OK> twice.

11. Press <Exit> to close the *Plugins* dialog box.

12. Close and reopen the database to activate the features of the *MLST online plugin*.

The *MLST online plugin* installs menu items in the main menu of the software under **MLST** (see Figure 2). In the *Main* window, the *MLST online plugin* has installed following items:

- Extra information fields in the *Database entries* panel (default names: **MLST ST**; **MLST CC**).
- One character type called **MLST**, one composite dataset called **MLST\_CMP**, and seven sequence types, each named after a housekeeping gene.

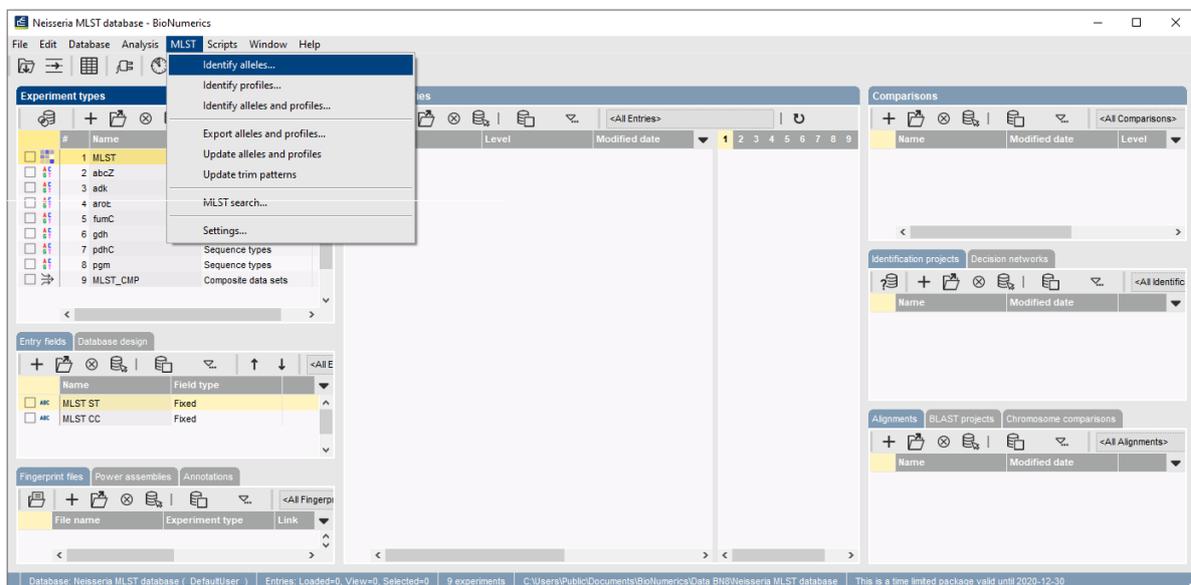
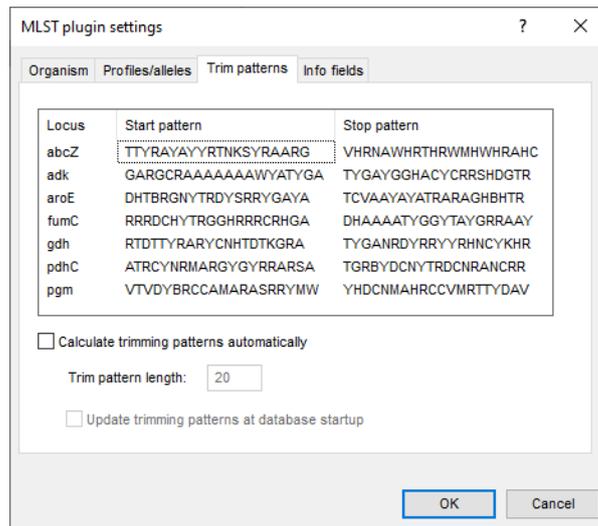


Figure 2: The *Main* window after installation of the MLST online plugin.

Upon installation of the plugin, the trimming patterns of *Neisseria* were automatically calculated. Since a lot of sequences exist for each housekeeping gene in the online repository, the trimming patterns of the *Neisseria* housekeeping genes contain a lot of degenerated positions. The presence of these degenerated positions might result in the assignment of the wrong trimming positions on our sample sequences. Less degenerated patterns can be found in the MLST trimming

patterns.txt file that can be downloaded from the Applied Maths website (<https://www.applied-maths.com/download/sample-data>, click on "MLST sample SCF trace files").

13. Select **MLST** > **Settings** to call the *MLST plugin settings* dialog box.
14. Click on the *Trim patterns* tab.
15. Uncheck **Calculate trimming patterns automatically** and copy the patterns from the MLST trimming patterns.txt file to the correct genes in the grid (see Figure 3).



**Figure 3:** Trimming patterns after correction.

16. Close the *MLST plugin settings* dialog box.

The plugin is now set up and we are ready to start assembling allele sequences.

## 4 Importing and assembling trace files in batch

A set of *Neisseria* trace files can be downloaded from the Applied Maths website (<https://www.applied-maths.com/download/sample-data>, click on "MLST sample SCF trace files") and are used in this guide to explain the work flow of the *MLST online plugin*.

1. Select **File** > **Import...** (, **Ctrl+I**) to call the *Import* dialog box.
2. Select **Import and assemble trace files** under **Sequence type data** and press <**Import**>.
3. Select the <**Browse**> button, navigate to the correct path, select all the sequence trace files and press <**Open**>.

The *Import sequence traces* wizard page is updated (see Figure 4).

4. Press <**Next**> to go the next step.

The way the information should be imported in the database can be specified with an import template. In the example data set, the **Key** and **Sequence experiment name** are provided in the trace file name.

5. Select the predefined template **Example import 2** and press the <**Preview**> button to check the rules applied on the selected files (see Figure 5).

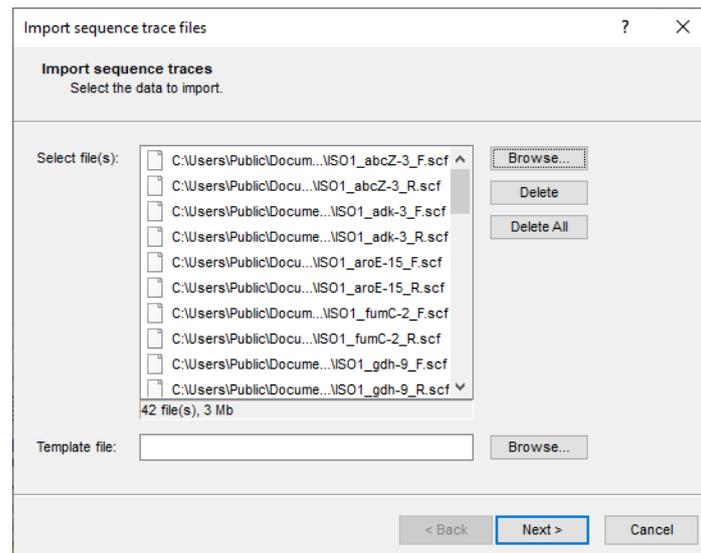


Figure 4: Select trace files.

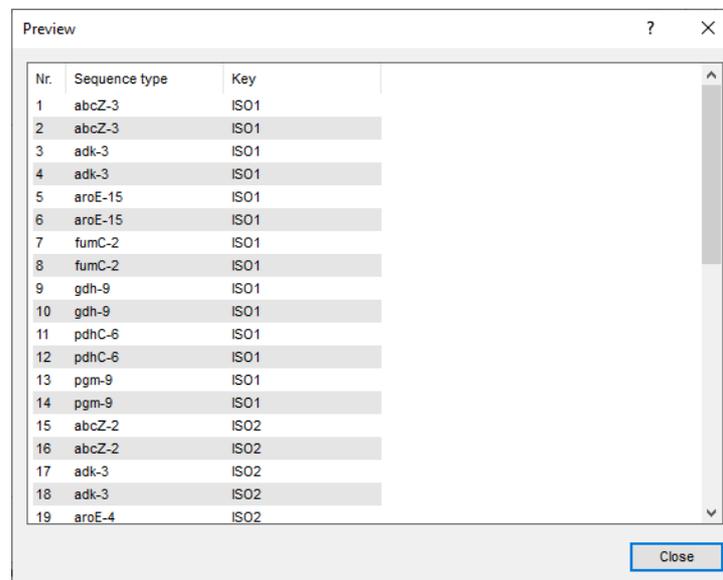


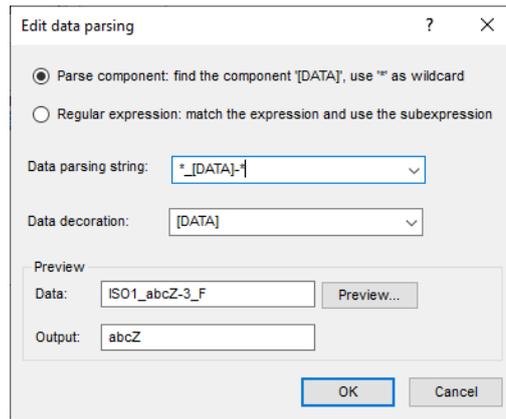
Figure 5: Preview.

In the preview we see that the sequence type name is not correctly parsed from the file names. In a next step, we will modify the rule for the **Sequence type**.

6. Make sure the predefined template **Example import 2** is still selected and press the **<Edit>** button to edit the rules.
7. Check the option **Show advanced options**, make sure the first row is selected in the grid panel and press the **<Edit parsing>** button.
8. In the *Data parsing* dialog box, update the data parsing string: **"\*\_[DATA]-\*"**.

This parsing string will only take into account the text occurring between the first underscore (.) and the hyphen (-). The asterisk (\*) serves as a wildcard, meaning that all characters before the first underscore and after the hyphen will be ignored.

9. Press the **<Preview>** button and press **<OK>** when the parsing is correct (see Figure 6).



**Figure 6:** Preview parsing.

10. Press **<Next>** and **<Finish>**.

11. Make sure the updated template **Example import 2** is selected and press the **<Preview>** button.

The preview should now look like [Figure 7](#).

Nr.	Sequence type	Key
1	abcZ	ISO1
2	abcZ	ISO1
3	adk	ISO1
4	adk	ISO1
5	aroE	ISO1
6	aroE	ISO1
7	fumC	ISO1
8	fumC	ISO1
9	gdh	ISO1
10	gdh	ISO1
11	pdhC	ISO1
12	pdhC	ISO1
13	pgm	ISO1
14	pgm	ISO1
15	abcZ	ISO2
16	abcZ	ISO2
17	adk	ISO2
18	adk	ISO2
19	aroE	ISO2

**Figure 7:** Preview of the parsing.

12. Close the preview.

13. Make sure the updated template **Example import 2** is selected and press **<Next>**.

14. Press **<Next>** to confirm the creation of 3 new entries (see [Figure 8](#)).

The *Processing* wizard page opens (see [Figure 9](#)).

In the *Reports* panel, the **Maximum# of unresolved bases reported** can be specified (default value 20). Likewise, the **Maximum # of align inconsistencies reported** can be entered (default value 20). Align inconsistencies are positions where the consensus is resolved, but where one or more sequences are different from the consensus.

15. Press **<Trimming settings>** to pop up the *Assembly trimming settings* dialog box.

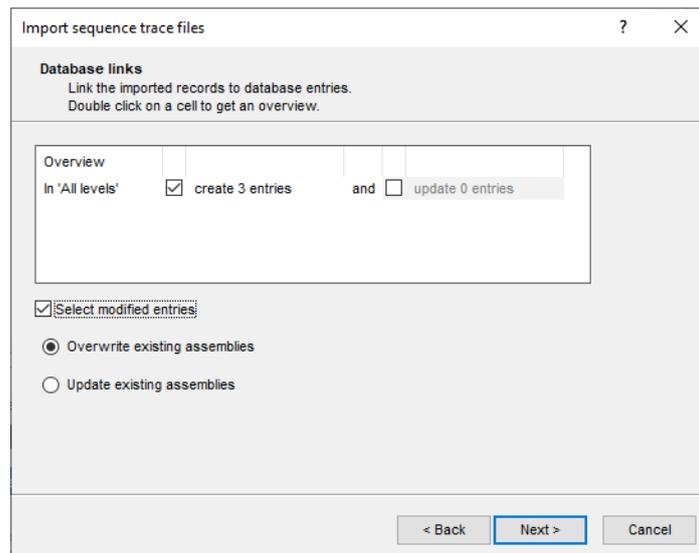
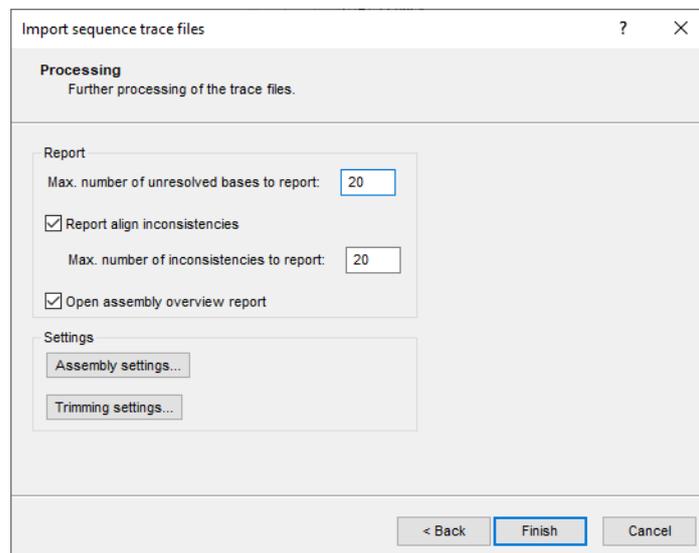


Figure 8: Database links.

Figure 9: The *Processing* wizard page.

The trimming patterns entered in the *Trim patterns* tab (see Figure 3) are shown in the **Start pattern** and **Stop pattern** columns (see Figure 10).

16. Double-click on the **<Edit>** button for experiment **abcZ** to call the *Assembly trimming settings* dialog box (see Figure 11).

In the *Assembly trimming settings* dialog box, a number of additional settings can be specified for each individual sequence experiment:

- **Minimum # of sequences** specifies the minimum number of trace sequences that should contribute to the subsequence on the consensus that matches the trimming targets. For example, if “2” is entered, a trimming target will only be set if the matching region on the consensus is *fully* defined by at least 2 sequences.
- For both the **Start position** and **Stop position**, a **Trim pattern** is displayed. The use of IUPAC code for ambiguous positions is supported. The **Tolerance** defines the number of

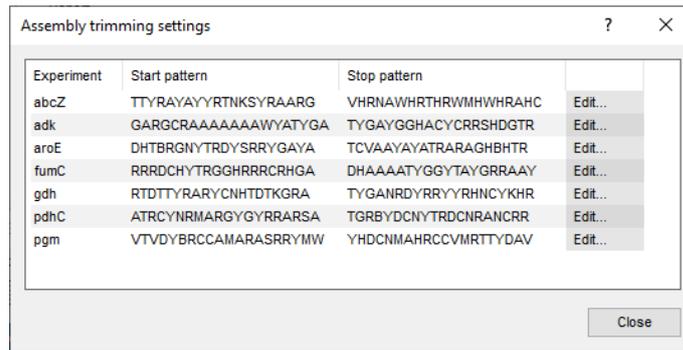


Figure 10: The *Assembly trimming settings* dialog box.

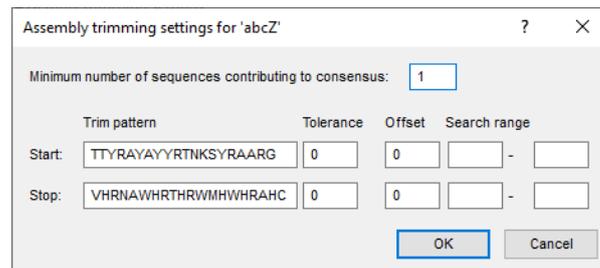


Figure 11: The *Assembly trimming settings* dialog box.

mismatches allowed for a sequence to be recognized as a trim pattern. With the **Offset**, one can specify that the consensus is trimmed at a certain offset from the start and end trimming target positions. If no offset is specified (zero), the trimming targets are included in the trimmed consensus. With the **Search range** one can restrict the search to certain regions on the consensus, e.g. to prevent incidental matches inside the targeted consensus sequence.

The entered trim patterns will be searched on the consensus sequence in both directions, i.e. on the consensus as it appears as well as on its complementary strand. In case the trim patterns match the complementary strand of the consensus, it will be automatically invert-complemented. If the **Trim pattern** text boxes are left empty, no preference sense is available.

17. Leave the predefined settings unaltered and press <OK> and <Close> to close dialog boxes.
18. Press the <Assembly settings> button to call the *Assembly settings* dialog box (see Figure 12).
19. Double-click on the <Edit> button for experiment **abcZ** to call the *Assembly settings* dialog box (see Figure 13).

The Assembly settings are grouped in tabs per settings dialog box in *Assembler: Quality* assignment, **Assembly** and **Consensus** determination. For a detailed description of the Assembler program settings, see the BIONUMERICS manual. In the last tab the Assembly settings can be copied from or to another sequence type experiment.

20. For this exercise, do not change the settings and press <OK> and <Close>.
21. Make sure the option **Open assembly overview report** is checked and press <Finish> to assemble the selected trace files from the example dataset into separate contig projects.

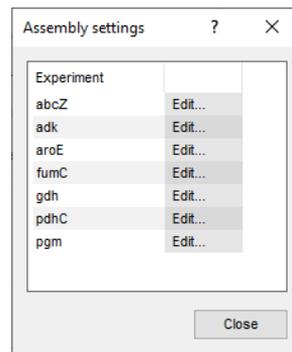


Figure 12: The *Assembly settings* dialog box.

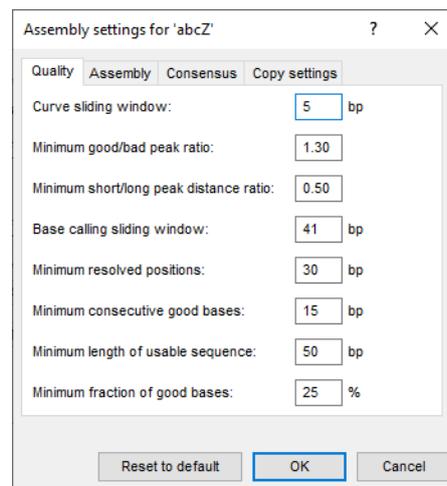


Figure 13: The *Assembly settings* dialog box.

## 5 Checking the sequence assemblies

When the assemblies are processed, an interactive report window appears (see Figure 14). This window can also be displayed from the *Main* window with **Analysis** > **Sequence types** > **Batch assembly reports...**

Key	abcZ	adk	aroE	fumC	gdh	pdhC	pgm
ISO1	warning	OK	OK	warning	warning	warning	error
ISO2	OK	OK	error	error	warning	error	OK
ISO3	OK	OK	error	error	error	OK	error

Figure 14: The *Batch sequence assembly report* window.

The *Overview* panel displays the entries (keys) as rows and the experiments as columns. Each cell in the grid, corresponding to a key/experiment pair, provides information about the current

status of the contig project. Only for those entries that have a green (= **OK** or **Solved**) or orange (= **Warning**) status, the allele IDs can be assigned.

1. Click a cell, e.g. **ISO1/pgm** to update the *Details* panel on the right-hand side.
2. In the *Details* panel double-click on the first message.

This will open the sequence in the *Contig assembly* window, with the corresponding position in focus. The position can now be examined and - if needed - the base calling can be changed manually. In the Assembler project of the pgm of ISO1, three unresolved bases are present in the consensus sequence. For each of these unresolved bases, an error is listed in the *Details* panel.

3. In the *Contig assembly* window click on the *Aligned traces* panel.
4. To obtain an optimal view of the curves, use the zoom sliders in the *Traces* panel or use the zoom buttons.



**Figure 15:** The *Aligned traces* panel.

The three unresolved bases in the consensus sequences are displayed in pink. The consensus sequence can be screened for (nearest) allele matches:

5. In the *Assembly view*, select **MLST > Identify allele**.

The consensus sequence is screened against the downloaded allele information of the selected organism.

The *MLST identification plot window* pops up. The sequence type of the sequence that is shown in Assembler is displayed in white. The other sequence types are shown in gray (see Figure 16). The identification result is shown below the sequence type boxes.

6. In our example, four mismatches are reported with allele ID 9 (see Figure 16).

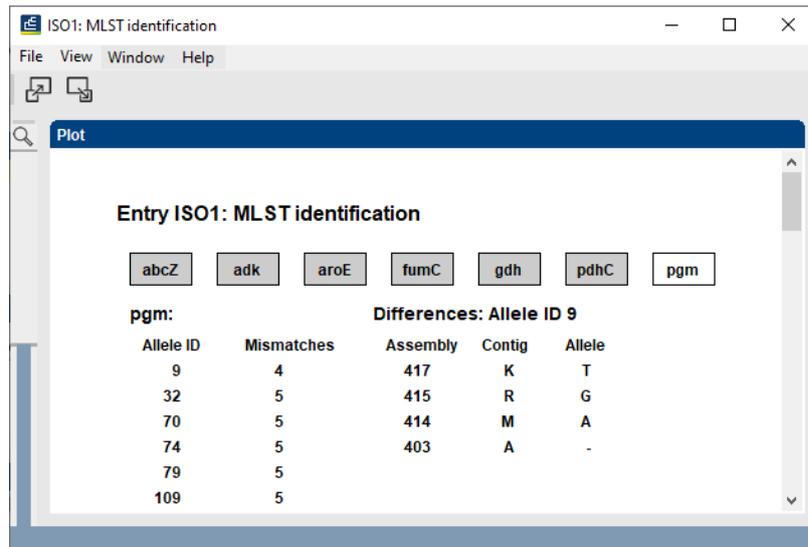


Figure 16: The *MLST identification plot window*: Best match with allele ID 9.

7. Select the first mismatch from the list.

The focus in *Assembler* is updated.

8. Change the G at position 820 in the reverse sequence to a T. The consensus sequence at position 820 is automatically updated.

9. Update the consensus sequence according to other editing suggestions: enter G at position 818, enter A at position 817, and remove A at position 806. Save the contig project with **File > Save** (📁, **Ctrl+S**).

10. Select **MLST > Identify allele**. The consensus should now have a perfect match with allele ID 9.

11. Select **Batch sequence assembly > Set report to solved, save and close** (**Ctrl+Shift+S**) in the *Contig assembly window*.

The corresponding key/experiment cell in the *Overview* panel is updated and displayed in green. The status "solved" is displayed in the cell and in the **Status** column of the *Details* panel.



Figure 17: Solved status.

12. Click on the **ISO1/abcZ** warning cell in the *Overview* panel to update the *Details* panel on the right-hand side.

13. In the *Details* panel double-click on the first warning message to open the *Contig assembly window*.

Six alignment inconsistencies are reported. False peaks in the trace files result in the insertion of six false bases in the consensus sequence. These bases can be deleted from the consensus sequence as follows:

14. Click on the first alignment inconsistency in the *Details* panel. The focus is automatically updated.
15. Press the **Del**-key on the keyboard to delete the base from the consensus sequence.
16. Repeat this step for the other alignment inconsistencies.
17. Save the contig project with **File > Save** (📁, **Ctrl+S**).
18. Select **MLST > Identify allele**. The consensus should now have a perfect match with allele ID 3.
19. Select **Batch sequence assembly > Set report to solved, save and close** (**Ctrl+Shift+S**) in the *Contig assembly* window.
20. Check all other warnings for ISO 1. In the *Assembly view*, select **MLST > Identify allele** to get an idea of the (closest) matches.
21. As an extra exercise, check the errors and warnings of the ISO 2 and ISO 3 samples.

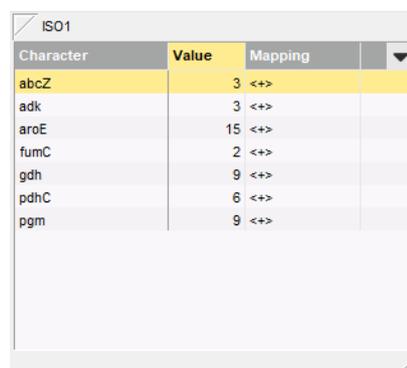
## 6 Identifying alleles and profiles

After all sequence assemblies are checked, we can use the *MLST online plugin* for the identification of our sequences.

1. Make sure all three entries in the database are selected.
2. In the *Main* window, select **MLST > Identify alleles**.

The matched allele IDs are stored as character values in the **MLST** character type, as can be seen in the *Experiment card* window:

3. Click on the green dot in the **MLST** column of the *Experiment presence* panel to open the character *Experiment card* window for an entry (see Figure 18).



Character	Value	Mapping
abcZ	3	<=>
adk	3	<=>
aroE	15	<=>
fumC	2	<=>
gdh	9	<=>
pdhC	6	<=>
pgm	9	<=>

**Figure 18:** Character card.

4. Close the *Experiment card* window by clicking in the small triangle-shaped button in the left upper corner.

Next we will screen the allelic profiles of the entries against the sequence type and clonal complex information of *Neisseria*.

5. In the *Main* window, select **MLST > Identify profiles**.

The matched sequences types and clonal complexes are displayed in the MLST information fields "MLST ST" and "MLST CC", respectively.