Application Note

"External Applications": adding functionalities to the CLC Genomics Server

Francesco Lescai, Bela Tiwari, Jacob de Meza, Leif Schauser, Jonathan Jacobs

QIAGEN® Bioinformatics – Aarhus, Denmark

Introduction

QIAGEN's CLC bioinformatics software portfolio provides user-friendly and intuitive solutions that run on any platform. This helps scientists to focus on the biology of their research without requiring them to write code, or compile and run software from the command line. At QIAGEN Bioinformatics we understand, however, that no single piece of software can meet the needs of every bioinformatics challenge. Sometimes, you need to supplement standard pipelines with your own scripts, open-source tools or third party applications from the command-line. This application note describes how to use CLC software together with external applications to create an integrated bioinformatics analysis environment.

The CLC Genomics Server's[a] "External Applications" functionality enables users to integrate non-interactive applications or scripts, that would typically be run via the command-line, into the CLC environment. Once tools are configured as external applications, they can be made readily available to anyone connecting their CLC Workbench[b] software to the CLC Genomics Server, using our free External Applications Client Plugin[c]. Integrated tools can be launched directly using the graphical CLC Workbench menu system, or can be added to analysis pipelines within a workflow. Using the CLC workflow system, researchers can quickly configure and run complex genomics analyses reproducibly, using CLC and external applications, in one unified user-friendly environment.

The "External Applications" functionality is highly configurable and allows you to control what software is made available from the CLC Genomics Server, which parameters can be set by users at run-time, and which users and groups should have access to each tool. Parameters, files, and intermediate result files can be passed to an external application. In return, results produced by these applications can be imported by the CLC software, making them available to users from within the graphical CLC Workbench environment.

[a] https://www.qiagenbioinformatics.com/products/clc-genomics-server/

[b] CLC Workbench software includes our Biomedical Workbench, Genomics Workbench, Main Workbench, CLC Command-Line Tools, and our CLC Genomics Server software.

[c] https://www.qiagenbioinformatics.com/plugins/external-applications-client-plugin/

In this application note, we describe three example cases to illustrate the broad potential use of our External Applications functionality:

Use-Case Examples	
1. Using R for Data Visualization	Leveraging an external R script, using data produced by CLC analysis tools, to generate a data analysis view currently not available using CLC software.
2. Using Open-Source tools for analysis	Integration of an open-source statistical analysis tool for phylogenetic tree model testing, wrapped within a bash script. Multiple alignments are exported from the CLC software, and then selected best trees are seamlessly reimported for further analysis.
3. Integrating an External Tool into a CLC Workflow	Integrating a statistical tool into a workflow, alongside existing CLC alignment and tree building tools, enabling the comparison of trees generated by different tools under different models.

Step by step guides to installing the external applications described in this document, and links to the example files and scripts used, are provided in the appendix.

In this application note we give some examples of how external applications can be configured and used to provide extensive, customized functionality to CLC software users.

Configuring External Applications is described in the CLC Genomics Server Administator Manual.

Creating an External Application

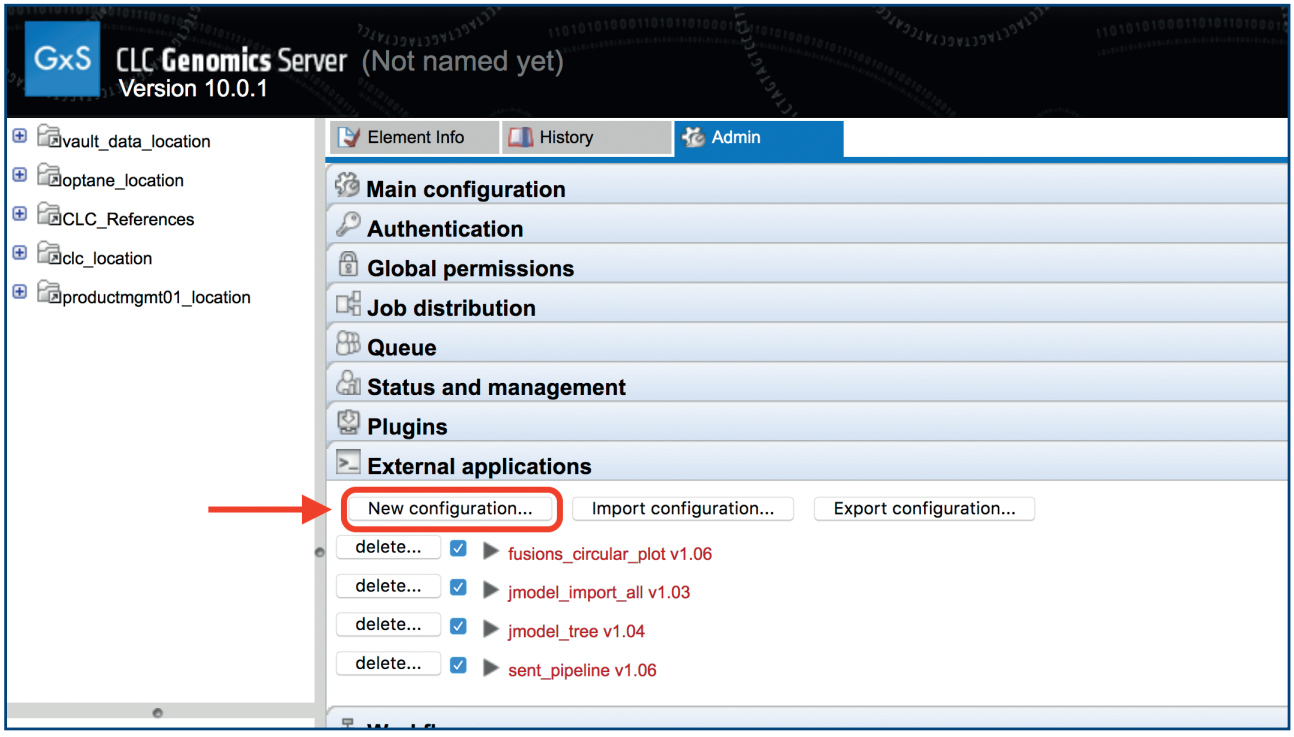

Figure 1. Clicking on the New configuration button under the External Applications tab opens up a form for configuring a tool or script as an external application. After configuration is complete, it will be available for use by CLC Genomics Workbench users logged into the CLC Genomics Server and by CLC Server Command Line Tools users.

To offer access to a command line tool or script via the CLC environment, a CLC Genomics Server administrator uses a web-form via the the CLC Genomics Server web administration interface. Required details about the external application include specifying the command to run, the type of inputs and outputs, and any required parameters.

The configuration form for a new external application is opened by clicking the "New configuration" button, found under the External Applications tab of the CLC Genomics Server web administrative interface (Figure 1). Under this tab, existing external applications configurations can also be viewed, edited, imported and exported.

After the external application configuration is saved and made available, it can be launched by users logged into the CLC Genomics Server via a CLC Genomics Workbench, as illustrated for this example in Figure 2 or by using the CLC Server Command Line Tools.

In the following sections we describe three concrete use cases, two where the toolset available via CLC software is extended through external applications, expanding plotting capabilities with R, and the integration of additional statistical tools, and a third, where an external application is included alongside CLC tools within a workflow.

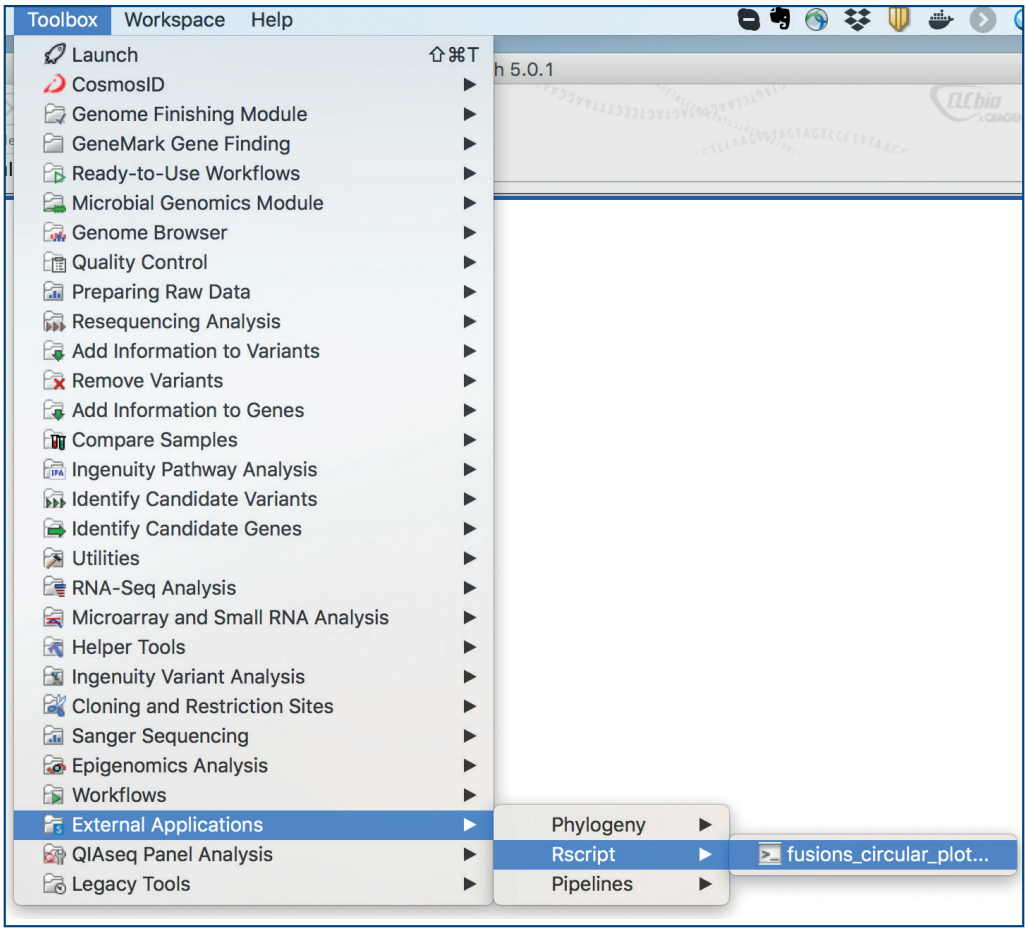

Figure 2. External applications can be launched using menu items in the External Applications folder of a CLC Workbench Toolbox. Here, the example external application was configured to be placed in a subfolder called Rscripts.

Case 1: Expanding Data Visualization with R

In this first use-case, we present an example designed to plot a circular view of the human genome to support the visualization of fusion genes. A step-by-step guide to installing this external application, as well as where the the files and scripts used in this example can be downloaded from, is included in Appendix 1.

In the External Applications configuration form, the command line that should be executed is entered in the "Command line" field (Figure 3). This includes the full path to the executable and associated parameters, which are specified within curly brackets. For each parameter entered in the Command line field, a corresponding configuration row appears in the General configuration area. Parameters values that users should set when launching the external application are configured by specifying the value type, and, where relevant, by providing a list of options a user will be able to choose from.

The "plotfusions.R" script (see Appendix 1) is used in this example. It expects two positional parameters. The value of the first parameter, "fusiongenes", is used to specify the

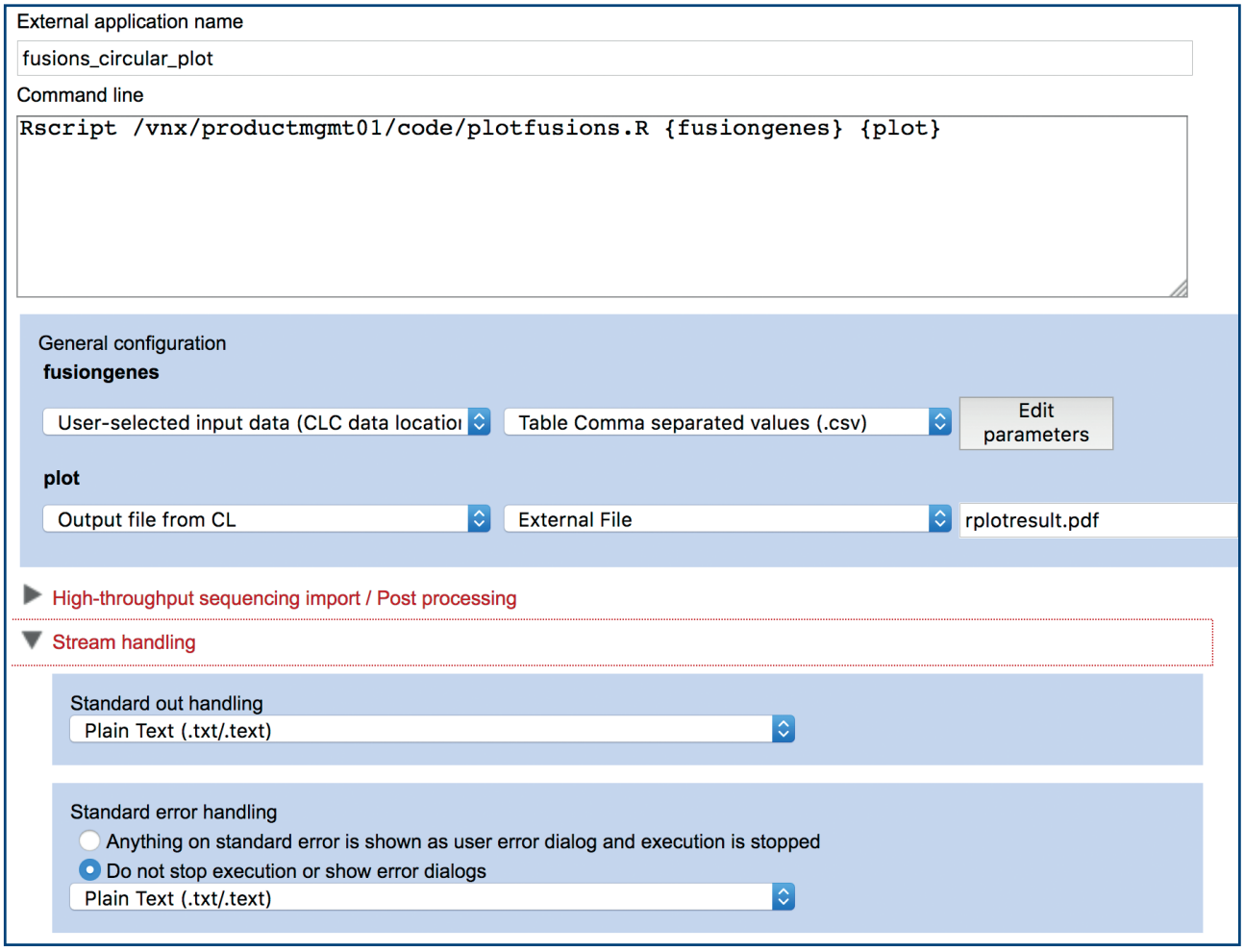

Figure 3. Configuration of an Rscript as an external application.

location and format of the input. We configure these details in the "General configuration" area as shown in Figure 3. Here, users will select the input from their CLC data location, and the data they select will be exported from the CLC Genomics Server as a csv file, which can be used by the plotfusions.R script. The value of the second parameter, "plot", relates to the output from the R script. Here, we indicated that the output should be imported into the CLC environment as an "External file", meaning it will not be converted into a CLC native format, and we provided the name of the file to be produced.

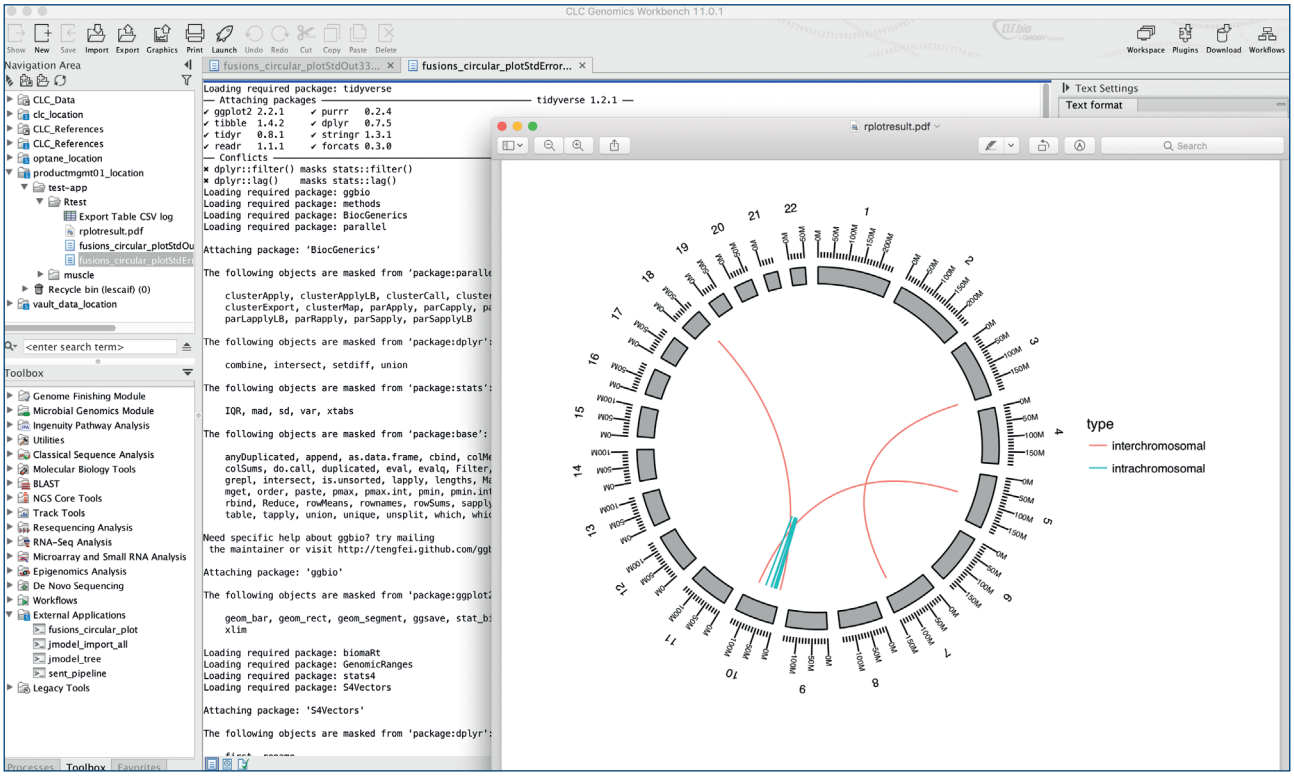

Figure 4. The results of the example R script run as an external application can be viewed via the CLC Workbench graphical interface. Here, the textual information written by R is seen on the left, and the pdf plot is open on the right.

Standard output (stdout) and standard error (stderr) can be collected from the underlying application, as illustrated for this example in Figure 3. Under the Stream Handling section, we have specified a format, "Plan Text", replacing the default, which is "No standard import". By selecting a format, we have indicated that information sent to stdout by the application will be in plain text and should be collected and imported into the CLC Genomics Server. There are two options for how stderr is handled. The default is the upper of the two options shown in Figure 3, which would result in users being shown the contents of stderr and the external application being halted. Here, we chose instead the second option, where any information written to stderr by the application will be collected and imported into the CLC Genomics Server, and the application will not be halted if information is written to derr.

Once the external application completes its run, the user will be able to access the results of the R script from the CLC location they specified when launching the tool. Here, the results consist of a PDF file containing the plot, and a text file containing the stdout generated by R. A view of the results via a CLC Genomics Workbench is shown in Figure 4.

Case 2: Integration of statistical tools

We can build upon the extensive offering of the CLC Genomics Workbench and CLC Genomics Server for multiple alignment and phylogenetic anaysis by integrating the external software jModelTest1,2. While similar CLC functionality exists for maximum likelihood phylogeny and model testing, jModelTest is useful for testing a larger number of combinations of construction methods and substitution models when building phylogeny trees, and it ranks them according to alternative information criteria. A step by step guide to installing this external application, as well as where the the files and scripts used in this example can be downloaded from, is included in Appendix 2.

jModelTest accepts a phylip alignment as input and generates a plain text report with the statistical test outcomes, as well as the model for the best tree.

To aid interpretation of the results when viewed via a CLC Workbench, we wrote a Perl script that converts the jModelTest output into a separate file for each of the best trees according to two different information criteria, AIC and BIC.

We wish to run jModelTest and pass its output to our Perl script, but need a single command to configure an external application. Thus we wrote a small bash wrapper script that allows us to launch the jModelTest and the Perl script via one command.

The bash wrapper script takes four positional parameters. These are entered in the Command line field of the external application configuration, as shown in Figure 5. Each parameter is then further described in the General configuration area. The input to the external application is specified by the "alignment" parameter. It is configured as "User selected input data (CLC data location)", meaning that users will select the data to use as input from among their CLC data files. The data they choose will be exported to phylip format, which is the format expected by jModelTest.

Three parameters, "modelresults", "aictree" and "bictree", are then configured, representing each of the three output files from jModelTest and our Perl script. All are configured as type "Output from CL", signifying that they are outputs from the command line application. The format of each of these output files is also specified, so they are imported appropriately into the CLC Genomics Server. Here the statistical results are specified as plain text, and the trees as Newick trees. The results will be saved to a location that the user specifies when launching the external application.

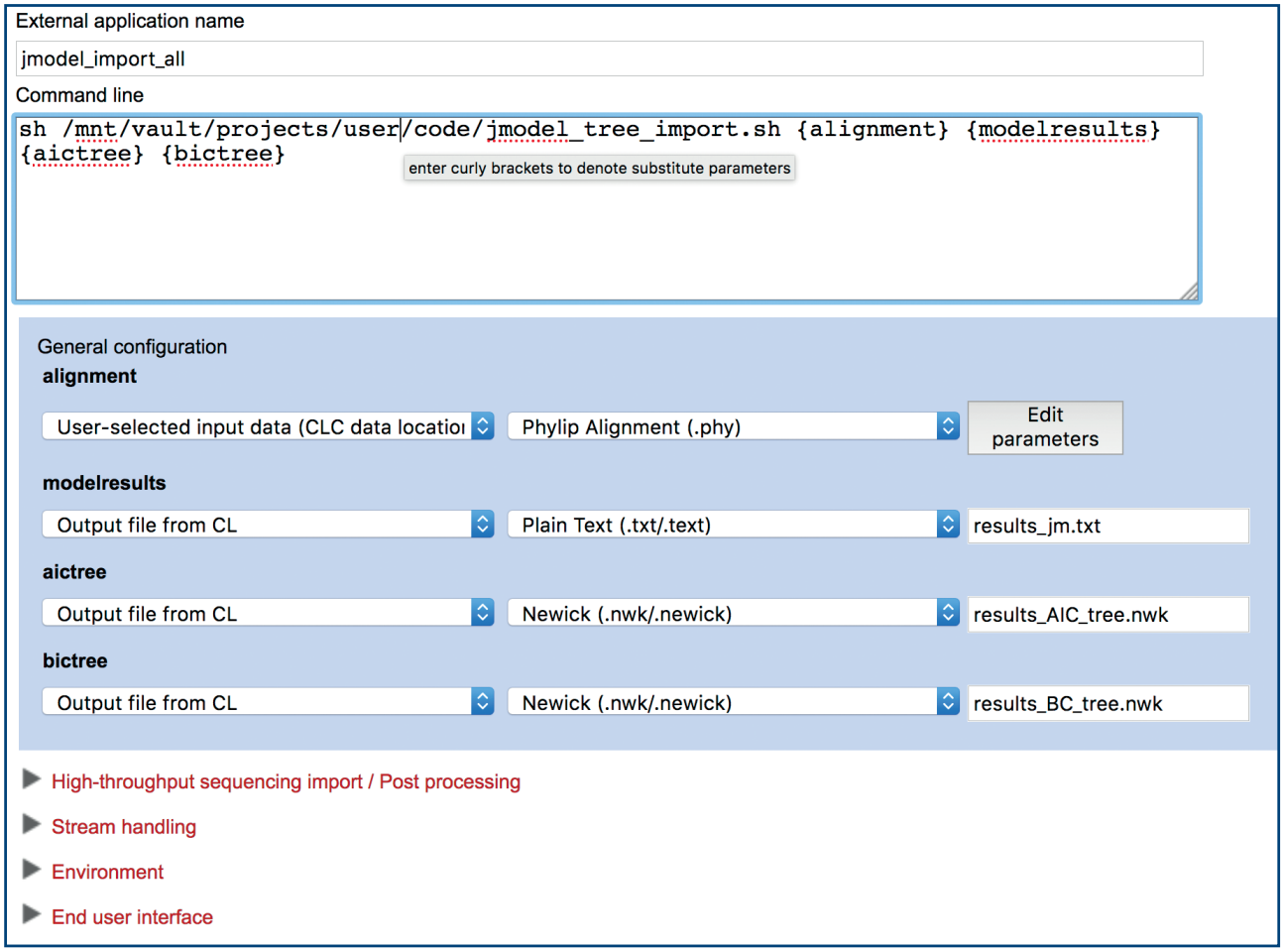

Figure 5. - Configuration of the jModelTest external application

Once imported, the results can be visualized using the CLC Genomics Workbench, as shown in in Figure 6.

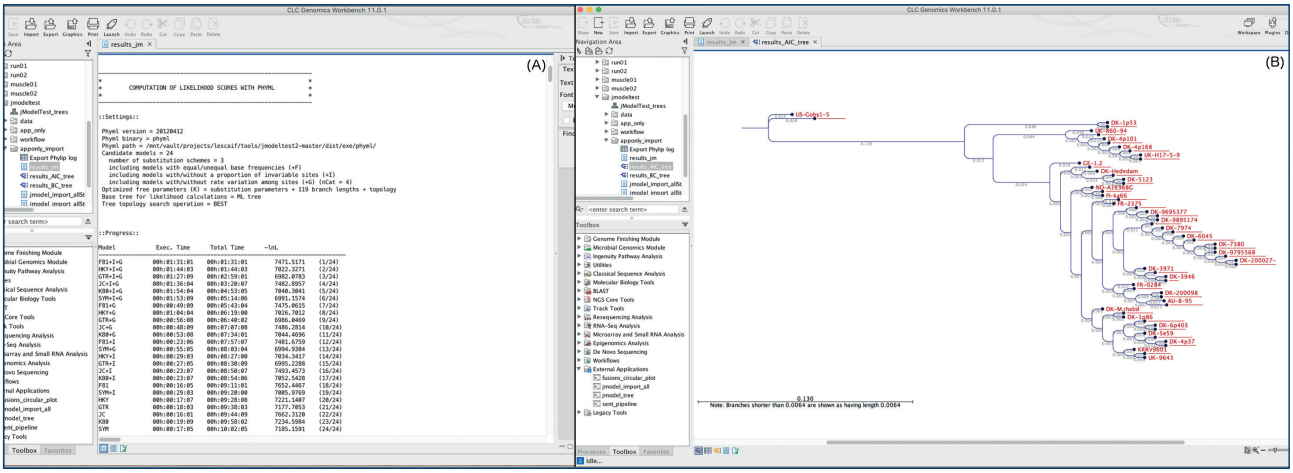

Figure 6. jModelTest results visualization. The statistical results imported as plain text can be seen on the left (A) and one of the trees imported is shown open on the right (B).

The user can then take advantage of functionality within the CLC Workbench to change the layout of the tree, annotate it with metadata, and customize the plot.

Case 3: Inclusion of external applications in CLC workflows

Tools configured as external applications can be directly incorporated into CLC workflows, making it simple to create and share complex, configurable, reproducible pipelines of analyses.

In Figure 7, we illustrate a workflow that allows a user to compare the results of three different tree reconstruction approaches. When this workflow is launched, the user selects input sequences to be aligned with the Workbench's ClustalO alignment tool. The alignment is then passed to three different tree reconstruction tools: jModelTest and two instances of the CLC Maximum Likelihood Phylogeny tool, each with different substitution models. All results generated are to be saved, as indicated by the blue output boxes connected to each of the elements in the workflow.

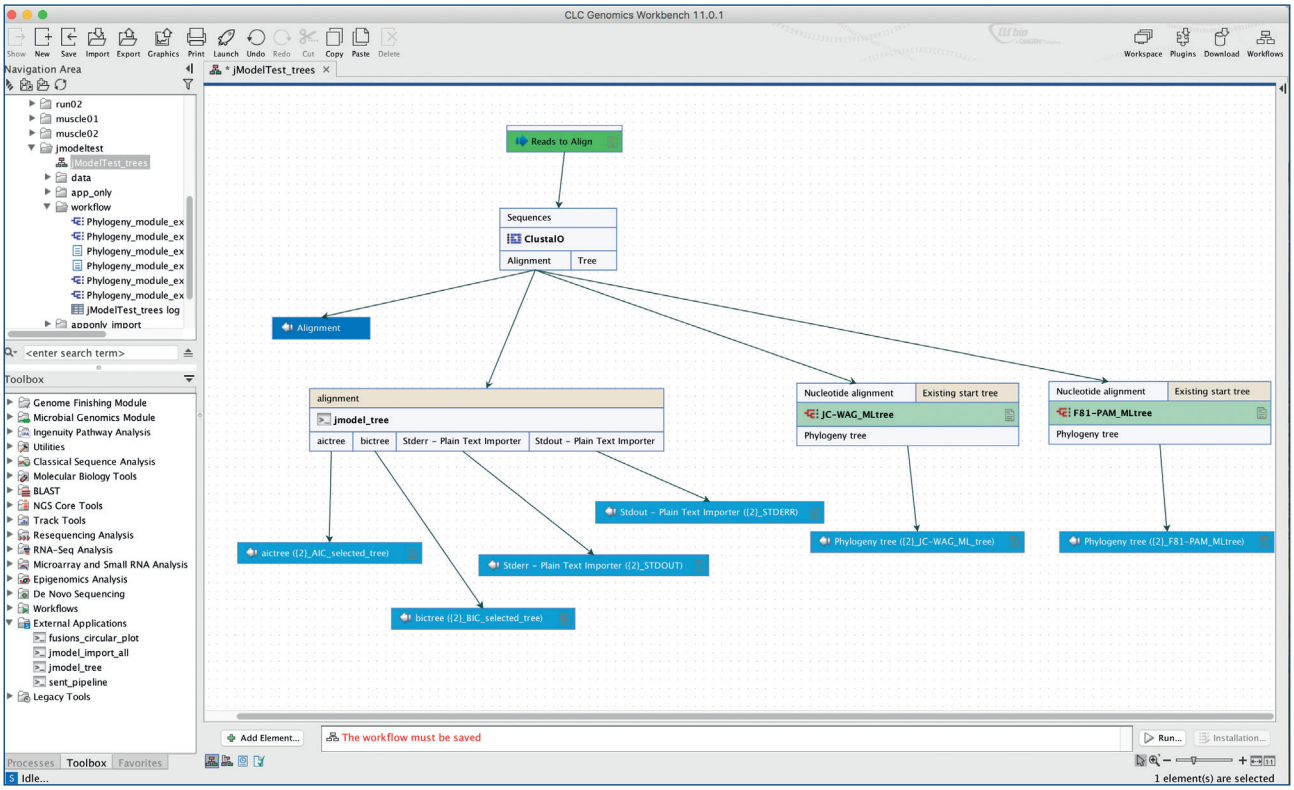

Figure 7. Integration of an external application in a workflow. Here, jModelTest, described in Case 1, is included in a workflow alongside CLC tools. The workflow elements have been renamed: jModelTest as jmodel_tree, and two instances of Maximum Likelihood Phylogeny have element names JC-WAG_MLtree and F81-PAM_MLtree.

For users, launching a workflow is a simple as launching a single tool, and they just need to specify any required information like the data to use as input, any necessary parameter values, and where to save the outputs. In the CLC Workbenches, this is all done via a graphical interface. After the workflow has finished running, the results can be viewed in a CLC Workbench.

Conclusions

The CLC Genomics Server's framework for External Applications allows the functionality available to users of CLC software to be extended to include tools or scripts that can be launched via a command line. Configuring an external application is simple, and allows complex analyses and integrated solutions to be easily and quickly made available to all users of CLC Workbench and Genomics Server.

The CLC Genomics Server External Applications framework allows expert bioinformaticians to focus on designing analysis pipelines and solutions, and provides a means to deploy those solutions to scientists seeking to take advantage of the powerful graphical interface and flexibility of the CLC Workbench platform.

Appendix 1 – How to install and run the example external application to plot circular genomes with R

The main focus of this appendix is to allow the user to test the functionality of an external application running R code, to plot circular genomes. In order to run this example, we have used the results of the analyses explained in the tutorial "Fusion Detection Using QIAseq RNAscan Panels" .

Prerequisites

The user should have R installed on their system, as well as the following packages:

- tidyverse (which includes ggplot[3] and dplyr[4]) for plotting capabilities and data manipulation;
- ggbio[5] for the specific functions to plot genomes and genomic features;
- biomaRt[6] to retrieve information from the ENSEMBL database;
- GenomicRanges[7] to handle genomic intervals;
- biovizBase for genome and caryogram information

CLC Genomics Server version 10.0.1 should be installed, and either Biomedical Genomics Workbench 5.0.1 or CLC Genomics Workbench 11.0.1 should be available to run the application from a desktop client.

Download and install code and configuration files

1) download the code package from our website http://resources.qiagenbioinformatics.com/external-applications/Rplot.zip, and unzip it to the location of choice on your computer.

2) login into the administrative interface of the CLC Genomics Server, select the "Admin" tab, and click on the External applications tab to open it.

3) Click on "Import configuration"as shown in Figure 8, browse and select the CLCServerConfiguration_Rcircularplot.xml file as shown in Figure 9, and choose import: you will now find "fusions_circularplot" application under the External applications.

4) Copy the Rscript "plotfusions.R", which you will find in the folder where you unzipped the code package in step 1, to the location of choice on your server.

5) Under the External applications tab in the CLC Genomics Server web interface, click on the arrow on the left of the "fusions_circularplot" and change the path to the "plotfusions.R" in the command line form as shown in Figure 10, to indicate the folder you chose in the previous step 4

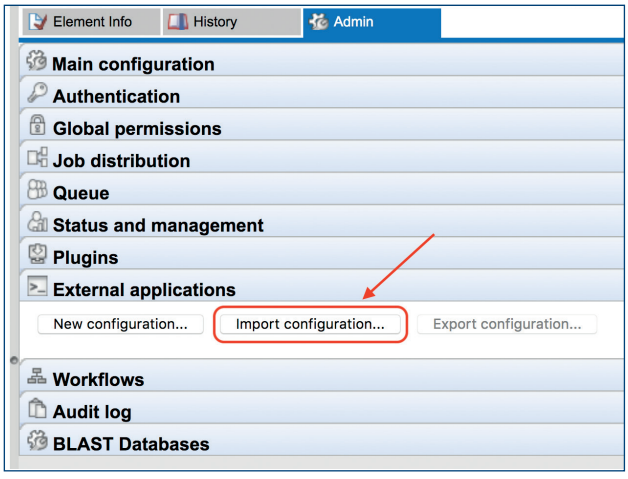

Figure 8. Import External application configuration file

The script plotfusions.R was developed for use with R version 3.4.4 and the following Bioconductor packages:

GenomicRanges_1.30.3, GenomeInfoDb_1.14.0, IRanges_2.12.0, S4Vectors_0.16.0, biomaRt_2.34.2, ggbio_1.26.1, BiocGenerics_0.24.0, forcats_0.3.0, stringr_1.3.1, dplyr_0.7.5, purrr_0.2.4, readr_1.1.1, tidyr_0.8.1, tibble_1.4.2, ggplot2_2.2.1, tidyverse_1.2.1

http://resources.qiagenbioinformatics.com/tutorials/Fusion_Detection.pdf

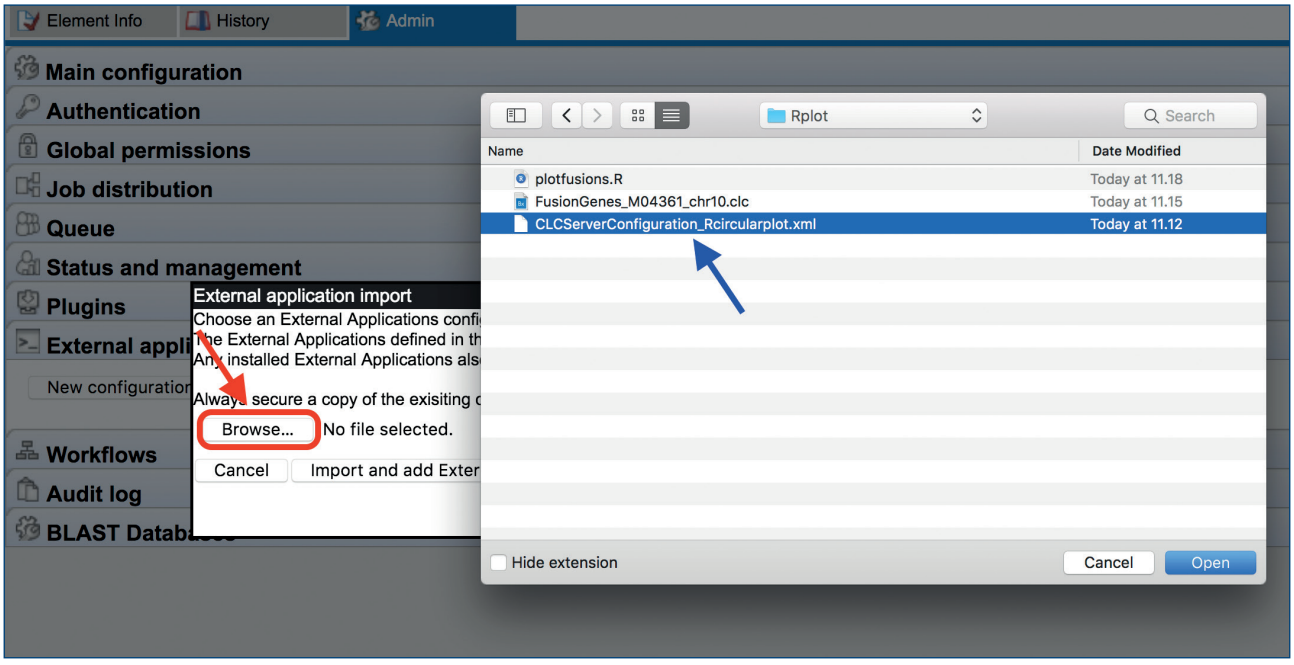

Figure 9. Select configuration file

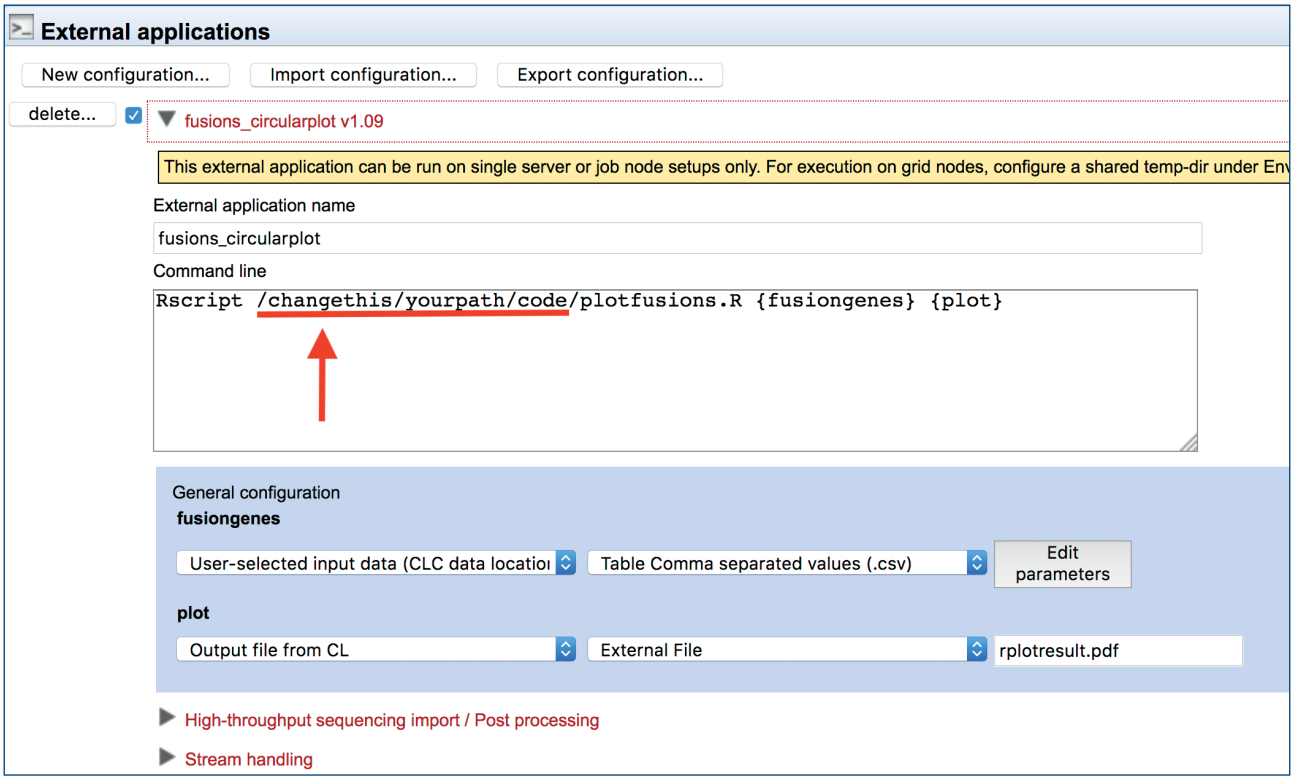

Figure 10. Provide the correct path to the script in the Command line field.

Import Data and run the application

Now the external application called "fusion_circularplot" is ready to be used, and available in the Biomedical or CLC Genomics Workbench, when logged to the CLC Server it has been imported on.

In order to run it, from the Workbench:

1) Choose from the menu, "import" and then "standard import": a window will open as in Figure 11; find the location where you have unzipped the previously downloaded package and select "FusionGenes_M04361_chr10.clc" to import it in a desired CLC location.
2) From the menu, choose "Toolbox" | "External Applications" | "Rscript" | "fusion_circularplot".
3) In the dialogue box choose to run from the server (where you have installed R and the dependent packages), and in the "fusiongenes" selector locate the file you have imported in step 1.
4) Choose a location for the results, and click "Finish" to run the application: once the execution has completed, you will find the R standard output as well as the PDF plot in the chose location.

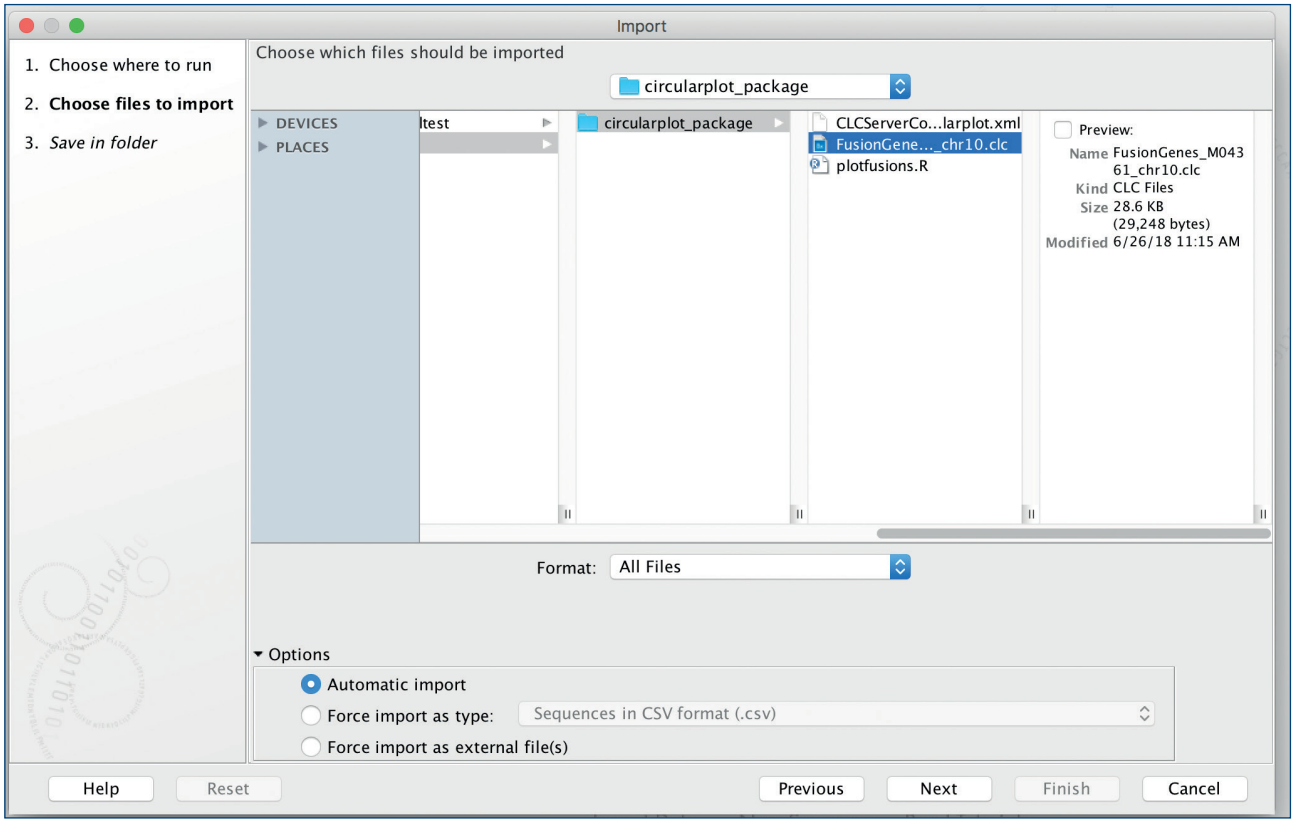

Figure 11. Import the example data

Appendix 2 – How to install and launch the example external application running jModelTest

The main focus of this appendix is to allow the user to test the functionality of an external application running jModelTest, to choose the most optimal tree construction method, starting from a multiple alignment. In order to run this example, we have used the multiple alignment that can be produced by running the tutorial "Phylogenic Trees and Metadata" .

Prerequisites

The user should have jModelTest and its dependencies installed: for download and installation instructions, please refer to their Github page.

CLC Genomics Server version 10.0.1 should be installed, and CLC Genomics Workbench 11.0.1 should be available to run the application from a desktop client.

Download and install code and configuration files

1) download the code package from our website http://resources.qiagenbioinformatics.com/external-applications/jmodeltest.zip, and unzip it to the location of choice on your computer.
2) login into the administrative interface of the CLC Genomics Server, move to the tab "Admin", and click on the row "External applications".
3) Click on "Import configuration"as shown in Figure 8, browse and select the CLCServerConfiguration_jmodeltest.xml file in the location chosen in step 1, and import it. You will now find "jmodeltest" application under the External applications.
4) Copy the scripts "extract_jmodel.pl" and "jmodel_tree_import.sh", which you find in the folder where you unzipped the code package in step 1, to the location of choice on your server.
5) Under the External applications tab in the CLC Genomics Server web interface, click on the arrow on the left of the "jmodeltest" and change the path to the "jmodel_tree_import.sh" in the command line form as shown in Figure 12, to indicate the folder you chose in in step 4 and save the configuration.
6) Edit the script "jmodel_tree_import.sh", in order to change the locations indicated in Figure 13 to the desired locations where you have installed jModelTest and where you copied the script in step 4.

http://resources.qiagenbioinformatics.com/tutorials/Phylogeny-module-visualization-of-Trees-and-Metadata.pdf
https://github.com/ddarriba/jmodeltest2

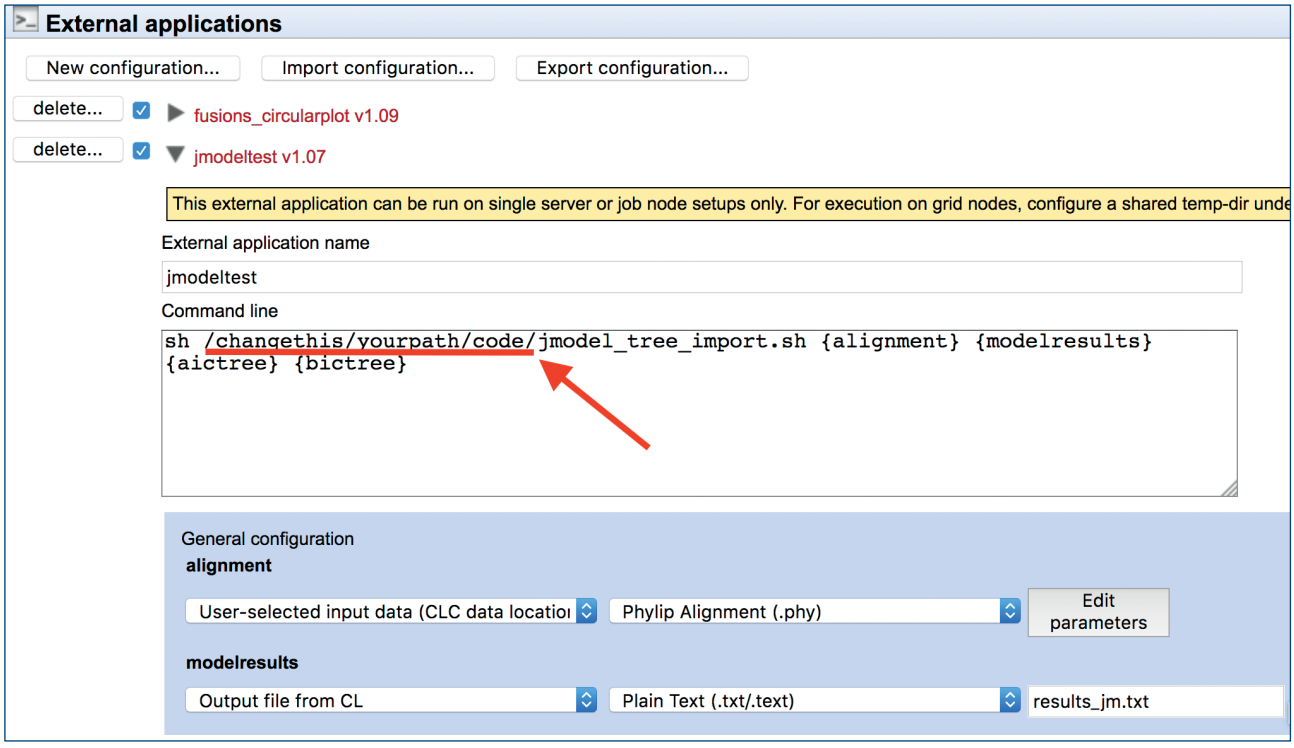

Figure 12. Customise the command line form for jModelTest

```
#!/bin/sh

alignment=$1
modelresults=$2
aictree=$3
bictree=$4

JM_DIR=/choose/your/path/jmodeltest2-master/dist

echo "starting jModelTest"

java -jar ${JM_DIR}/jModelTest.jar -d ${alignment} \
-g 4 \
-i \
-f \
-AIC \
-BIC \
-a \
-o ${modelresults} \
--set-property log-dir=`pwd` \
-w

echo "extracting best trees for AIC and BIC information selection"

perl /choose/your/path/extract_jmodel.pl -jmodel ${modelresults} -aictree ${aictree} -bictree ${bictree}
```

Figure 13. Customise file locations inside the jModelTest bash script

Import the data and run the application

Now the external application called "jmodeltest" is ready to be used. To launch it from the CLC Genomics Workbench:

1) "Under the "File" menu, choose "Import" | "Standard Import". A window will open as show in Figure 11 for the R application; find the location where you unzipped the previously downloaded package and select "phylogeny_alignment_example.clc" to import the data to a desired CLC location.
2) From the menu, choose "Toolbox" | "External Applications" | "Phylogeny" | "jmodeltest"
3) In the dialogue box choose to run the job on the Server (where you have installed jModelTest and its dependencies), and in the "alignment" selector locate the file you have imported in step 1
4) Choose a location for the results, and click "Finish" to run the application. Once the end of the execution the output of jModelTest, two different trees, standard error and standard output files will be found in the chosen location.

References

1. Darriba D, Taboada GL, Doallo R, Posada D. 2012. jModelTest 2: more models, new heuristics and parallel computing. Nature Methods 9(8), 772. https://github.com/ddarriba/jmodeltest2
2. Guindon S and Gascuel O (2003). A simple, fast and accurate method to estimate large phylogenies by maximum-likelihood". Systematic Biology 52: 696-704.
3. H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2009
4. Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2018). dplyr: A Grammar of Data Manipulation
5. Yin T, Cook D, Lawrence M (2012). "ggbio: an R package for extending the grammar of graphics for genomic data." Genome Biology, 13(8), R77.
6. Durinck S, Spellman P, Birney E, Huber W (2009). "Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt." Nature Protocols, 4, 1184–1191.
7. Lawrence M, Huber W, Pagès H, Aboyoun P, Carlson M, Gentleman R, Morgan M, Carey V (2013). "Software for Computing and Annotating Genomic Ranges." PLoS Computational Biology, 9

Discover more at **www.qiagenbioinformatics.com**

Other resources

External Applications documentation

http://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=External_applications.html

General information

CLC Genomics Server

https://www.qiagenbioinformatics.com/products/clc-genomics-server/

CLC Genomics Workbench

www.qiagenbioinformatics.com/products/clc-genomics-workbench/

CLC Server Command Line Tools

https://www.qiagenbioinformatics.com/products/clc-server-command-line-tools/

Tutorials

www.qiagenbioinformatics.com/support/tutorials/

For up-to-date licensing information and product-specific disclaimers, see the respective QIAGEN kit handbook or user manual. QIAGEN kit handbooks and user manuals are available at www.qiagen.com or can be requested from QIAGEN Technical Services or your local distributor.

Trademarks: QIAGEN®, Sample to Insight® (QIAGEN Group).
© 2018 QIAGEN, all rights reserved. PROM-12739-001