MGX User Guide

Sebastian Jaenicke
sebastian.jaenicke@Computational.Bio.Uni-Giessen.DE
sjaenick@CeBiTec.Uni-Bielefeld.DE

July 25, 2017

Contents

1 Installation

1.1 Prerequisites

MGX is a client/server application based on the Netbeans Platform and implemented in Java. To run the client application, the following dependencies have to be met:

- Java Runtime Environment, 1.8 or later
- Supported Operating Systems: Windows, Linux, Mac OS X
- Memory requirements: >= 512MB RAM
- (preferably broadband) Internet connection

1.1.1 Java

MGX requires a working Java Runtime Environment (JRE) to be installed on the computer. Typically, Java is already installed on most systems or can be obtained free of charge from

`http://www.java.com/`.

The installed Java version can be checked e.g. at

`http://www.java.com/de/download/installed.jsp`.

1.1.2 Supported Operating Systems

The MGX application is developed and regularly used on Unix-based systems, but has already successfully been used on computers running Windows and Mac OS X.

1.1.3 Memory and disk requirements

512MB of available main memory are sufficient to run MGX. Installation of the software requires about 60MB of disk space.

1.1.4 Internet connection

The network communication protocol used by the MGX framework has been heavily optimized to allow usage even with low-throughput connections. Thus, typical usage of the application like visualization of analysis results does not require a lot of bandwidth, although overall performance may suffer with high-latency or low-bandwidth connections. However, as sequence datasets obtained by metagenome sequencing tend to be quite large, a broadband connection is recommended at least for initial data upload to the MGX server or when exporting sequence data from a MGX project.

1.1.5 Obtaining MGX

We regularly publish new releases of the MGX client application, which are available for download at

`ftp://ftp.cebitec.uni-bielefeld.de/pub/software/mgx/`.

An installation isn't necessary, just unzip the file and start the software from the `bin/` subdirectory. Please check whether an updated version is available before reporting bugs.

2 Using the MGX application

2.1 Obtaining a MGX project

Currently, there is no way to automatically create new MGX projects. If you would like to analyse your data using MGX, send a mail to the MGX team, which can be contacted at mgx@computational.bio.Uni-Giessen.DE. MGX projects and users are managed by the General Project Management System (GPMS) developed at CeBiTec, which provides Single Sign-On (SSO) for all applications provided by CeBiTec's Computational Genomics group.

To apply for a new project, please provide

- suggested short project name, e.g. MGX_AcidMine
- a one-line description of your project ("Acid Mine Drainage metagenome")
- a contact address of a single person responsible for the project (PI or group leader) and the corresponding GPMS login name.

If you don't have a GPMS account yet, please sign up at the GPMS web site `https://www.cebitec.uni-bielefeld.de/gpms/`.

2.1.1 MGX roles

MGX offers different access levels (roles), which are assigned individually for each project:

- Admins are equal to Users (see below), but can grant access to additional users.
- Users have full access to a MGX project and are able to define new or modify present datasets, import new sequences and execute analysis jobs. They are also able to delete all data associated with a project.
- Guests are provided read-only access to a project, i.e. they are able to access all information already present, view analysis results and export data; however, they are unable to perform new analysis or delete data from the project.

For all MGX projects, the person requesting the project is automatically added as an Admin. As new users can always be added to an existing MGX project, all registered users are required to carefully protect their login credentials and not to share them with any third party.

2.2 Basic concepts

2.2.1 Metadata

In addition to the sequence data, MGX requires a user to provide additional information about a dataset, e.g. further details about the investigated habitat as well as sampling and sequencing procedures. Metadata in the MGX platform is organized in a hierarchical manner describing

- the geographical location of a habitat,
- the sample taken from a habitat,
- the DNA extraction procedure,
- sequencing technology and protocol.

2.3 Connecting to a MGX server

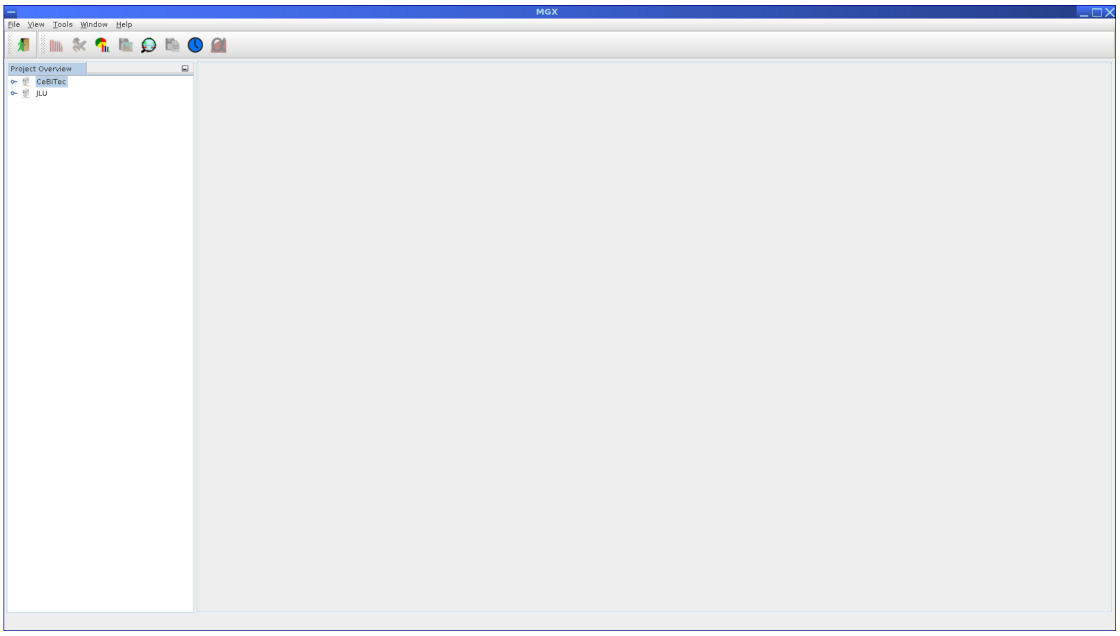
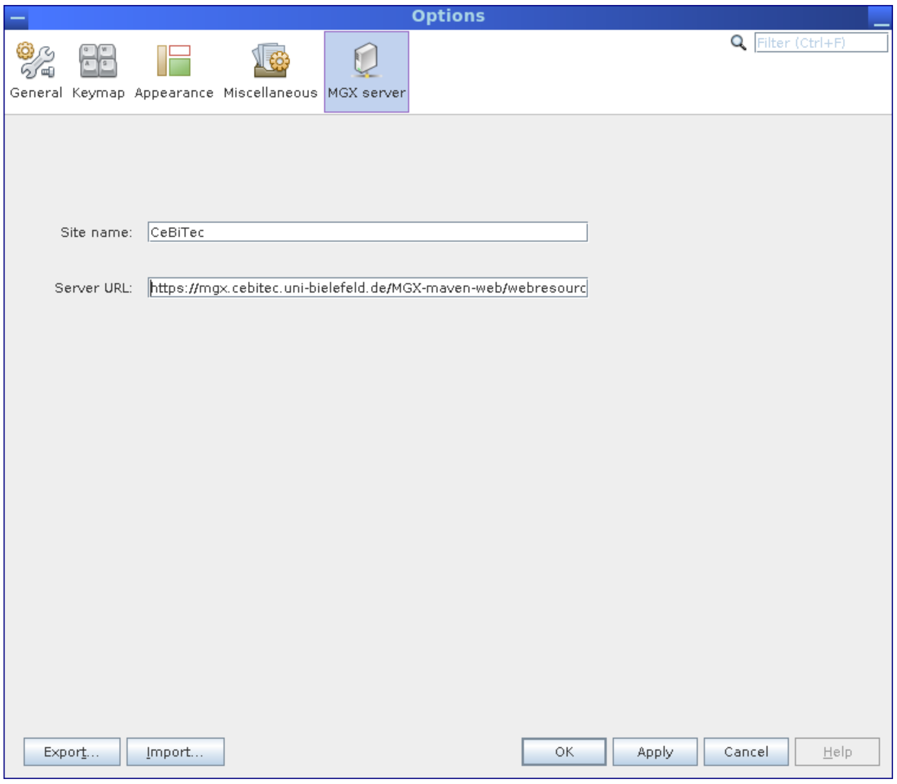


Figure 2.1: Screenshot of the client application right after startup.

After installation, the MGX application is already preconfigured to connect to the MGX server instance hosted at CeBiTec, Bielefeld University. In case a different MGX server should be used, the default server can be changed choosing Tools →Options from the menu and navigating to the MGX server tab (2.2). While the site name can be freely chosen by the user, the server URL has to be entered as provided by the site administrators.



Site name: CeBiTec

Server URL: https://mgx.cebitec.uni-bielefeld.de/MGX-maven-web/webresourc

Figure 2.2: A different default server instance can optionally be configured in the MGX server tab, which is available from the Tools →Options menu.

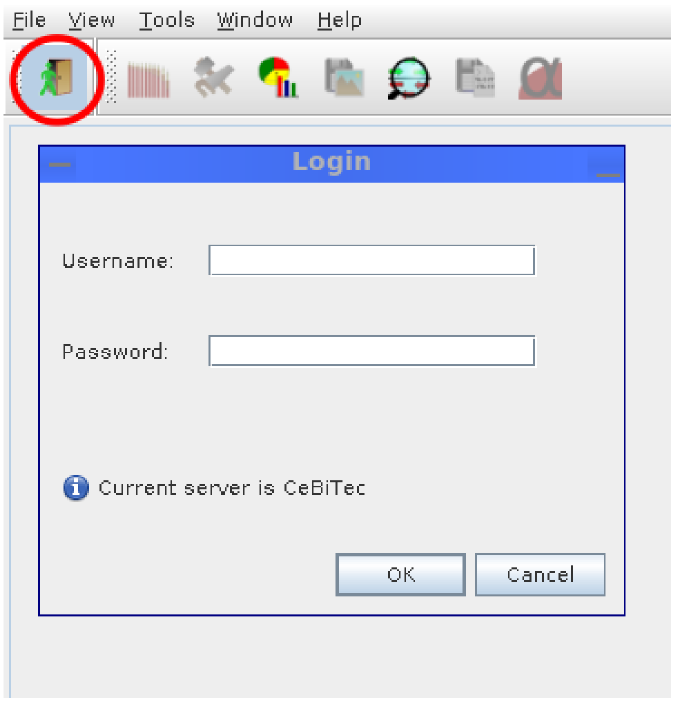


Figure 2.3: The first button in the menu toolbar will bring up the login dialog, allowing to connect to the configured server; the login screen also reflects the name of the current server.

All communication between the MGX user interface and the MGX server is encrypted using the standardized SSL (Secure Sockets Layer) protocol, ensuring confidentiality of unpublished data and protecting the integrity of login credentials. After successfully logging in, a new window is automatically opened. The MGX Explorer window lists all available projects a user is allowed to access, including both public as well as private projects. Projects are easily opened or closed by simply expanding the corresponding nodes in the Project Explorer window (2.4).

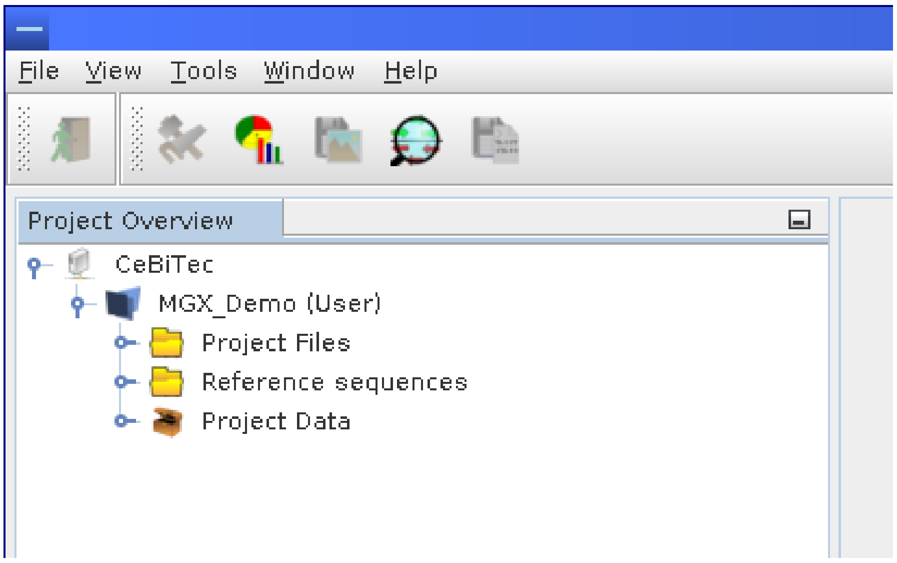


Figure 2.4: After successful login, the Project Explorer component will automatically open, showing a list of MGX projects available to the current user. Shown is just one project, MGX_Demo, with “User” access level indicated behind the project name.

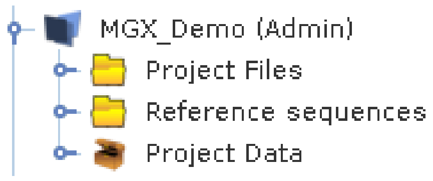


Figure 2.5: Divided into three different parts, a MGX projects offers (from top to bottom) dedicated storage for files to be used by analysis pipelines, managed reference sequences (including annotation data, if available) and general project data containing metadata as well as sequence datasets.

Each project contains metagenome datasets as well as structured storage (2.5), where user-provided databases can be uploaded to be used in custom analysis pipelines (Chapter 3).

2.4 Creating a new habitat

New habitats are defined choosing ”Add habitat” (2.6) from the context menu of the ”Project Data” node. This will bring up a wizard allowing to select the corresponding geographical location as well as specifying a habitats name and biome type (2.7). In a second step, an optional description for this habitat may be entered, as well.

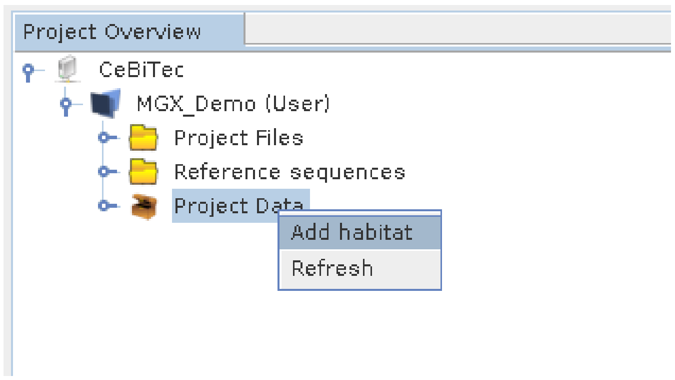


Figure 2.6: A new habitat can be added selecting the appropriate entry from the context menu of the "Project Data" node.

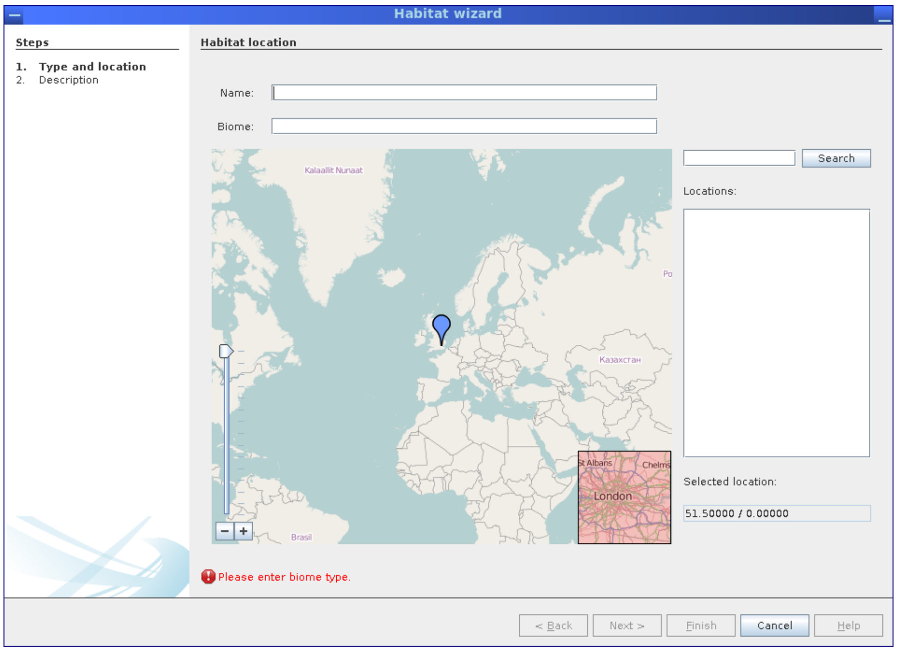
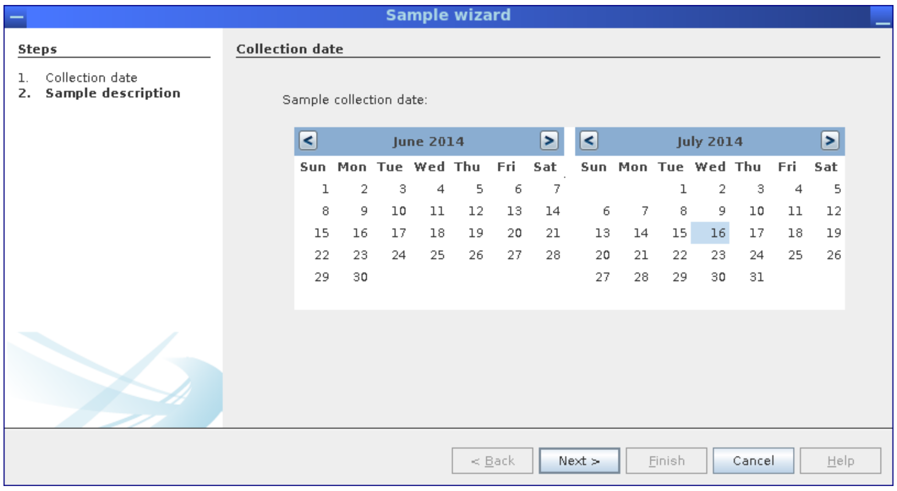


Figure 2.7: The habitat wizard allows to define new habitats specifying their location, name and biome type.

2.5 Creating new samples and DNA extracts

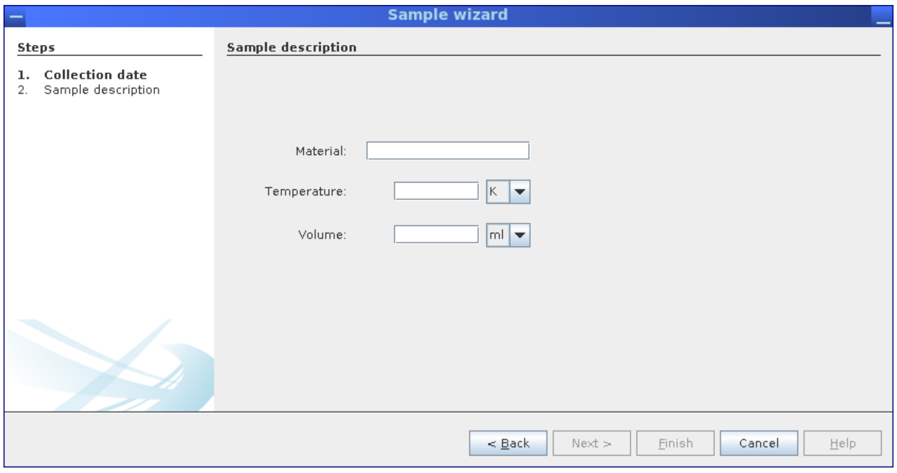
Samples and DNA extracts are created in the same way as habitats, except that samples are defined for habitats and DNA extracts are defined based on samples. Thus, the corresponding wizards are available from the context menu of the "Habitat" and "Sample" nodes, respectively.



June 2014						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

July 2014						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Figure 2.8: In the first step of the sample wizard, the sampling date is selected.

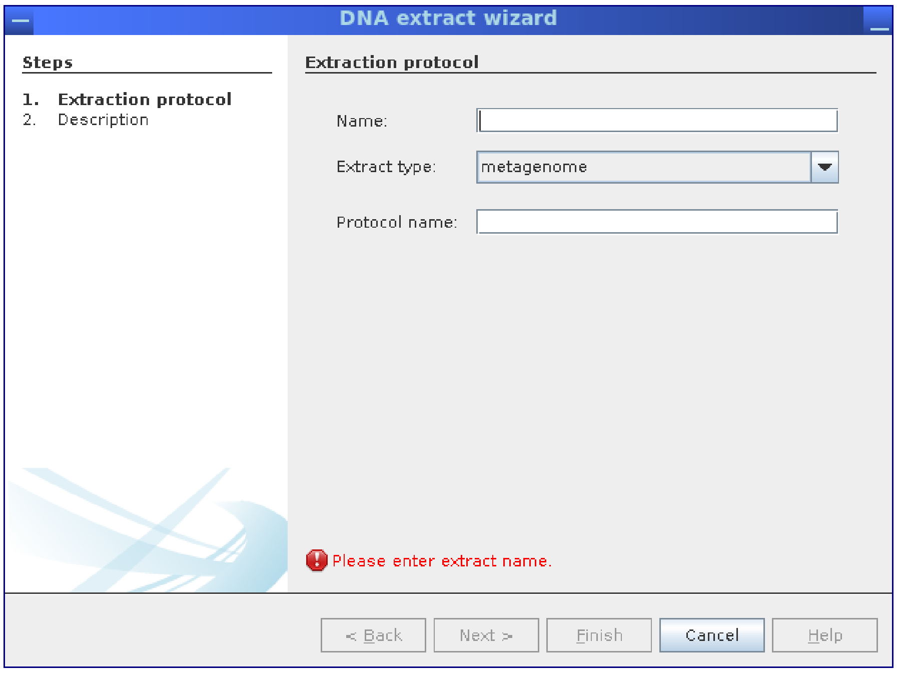


Material:

Temperature: K

Volume: ml

Figure 2.9: In the second step, the sampled material as well as temperature and sampled volume/weight need to be entered.


DNA extract wizard

Steps

1. Extraction protocol
2. Description

Extraction protocol

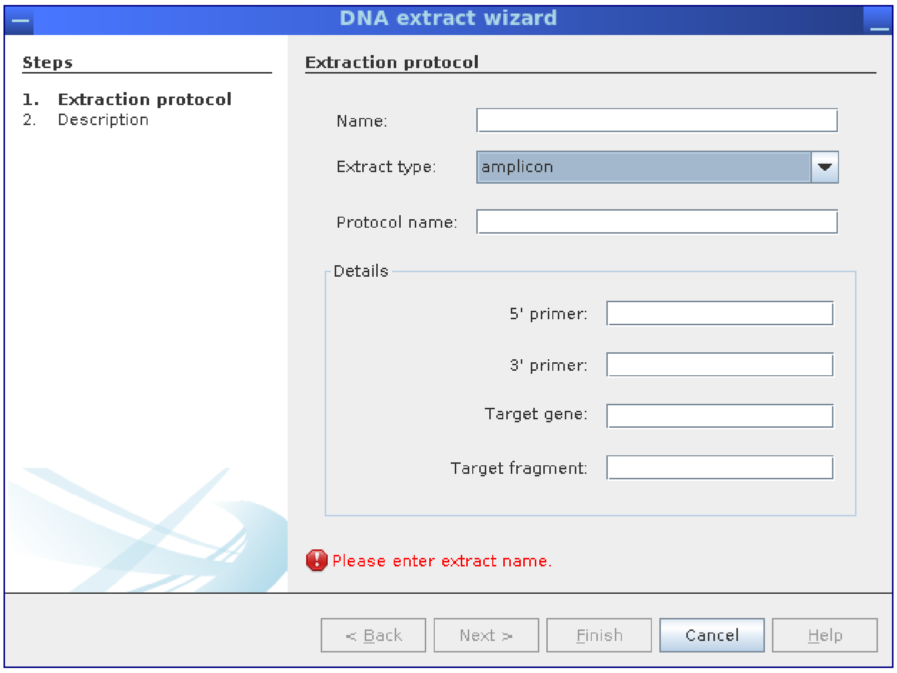
Name:

Extract type: metagenome

Protocol name:

Please enter extract name.

Figure 2.10: The DNA extract wizard allows to specify the type of DNA extract (metagenome, metatranscriptome, amplicon) and protocols used to extract the DNA.


DNA extract wizard

Steps
1. Extraction protocol
2. Description

Extraction protocol

Name:	
Extract type:	amplicon
Protocol name:	

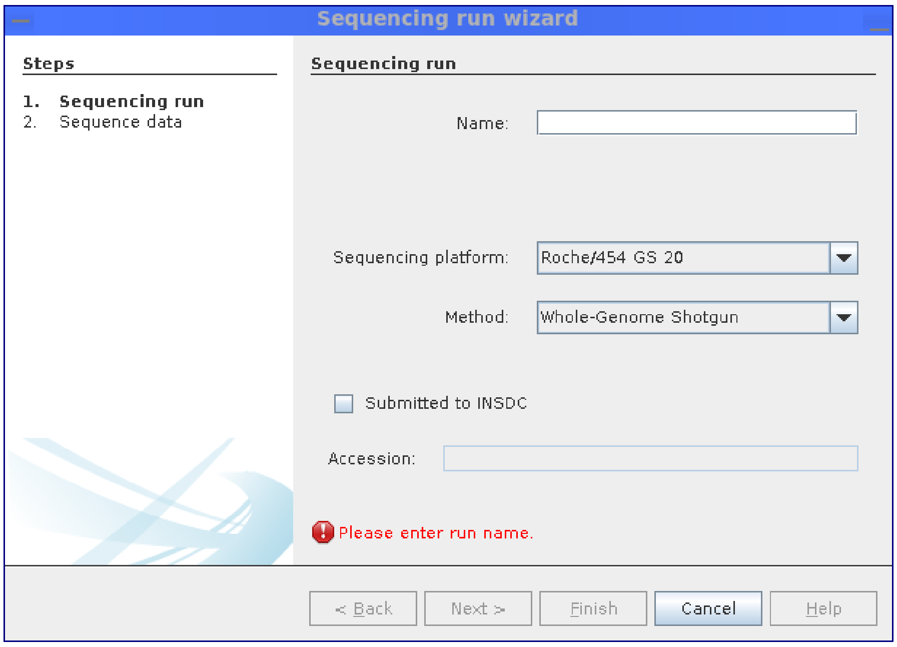
Details

5' primer:	
3' primer:	
Target gene:	
Target fragment:	

Please enter extract name.

Figure 2.11: Depending on DNA extract type, additional data can be provided; for amplicons, primer names and the corresponding target gene (fragment) can be entered, for metatranscriptomes, the type of RNA depletion methods can be selected.

2.6 Importing sequence data



Sequencing run

Name:

Sequencing platform: Roche/454 GS 20

Method: Whole-Genome Shotgun

☐ Submitted to INSDC

Accession:

Please enter run name.

Figure 2.12: Before sequence data can be uploaded, the employed sequencing platform and technology have to be specified; for data already submitted to or obtained from public INSDC repositories (e.g. NCBI, EBI, DDBJ), the corresponding accession number can be stored, as well.

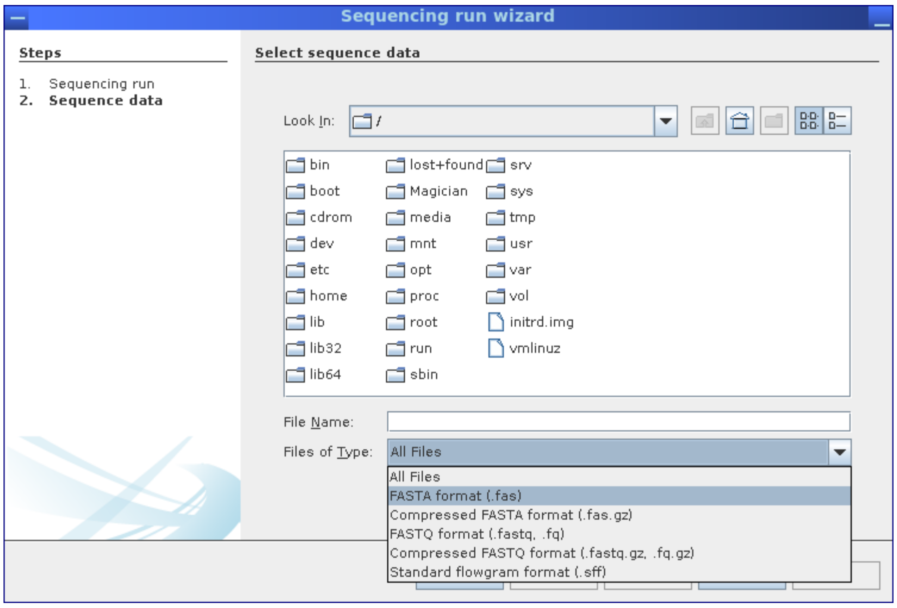


Figure 2.13: Finally, the file containing the sequence data is selected; MGX supports all commonly used file formats such as FASTA, FASTQ, or SFF.

2.7 Quality control

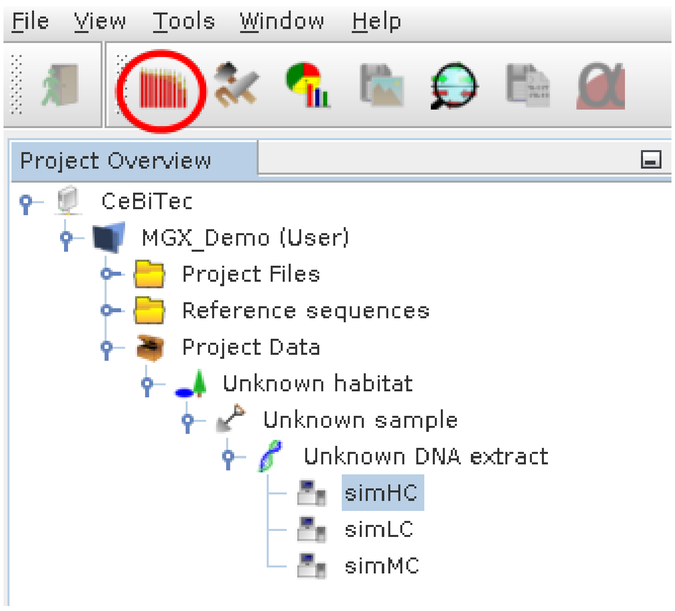


Figure 2.14: After selecting a sequencing run object, the Quality Control component can be opened from its menubar icon (red circle).

After sequence import, Quality control reports generated within MGX should be inspected (2.14) before proceding with data analysis. MGX currently offers three types of QC reports: Distribution of GC content, sequence length and nucleotide distribution within the DNA sequences. Those can be used to evaluate overall sequence data quality and check for possible signs of contamination. For demonstration purposes, data shown relates to the artificial simHC metagenome dataset created by the FAMeS[Mavromatis et al., 2007] project. The actual sequence data is publicly available and can be obtained from the FAMeS web site (`http://fames.jgi-psf.org/Retrieve_data.html`).

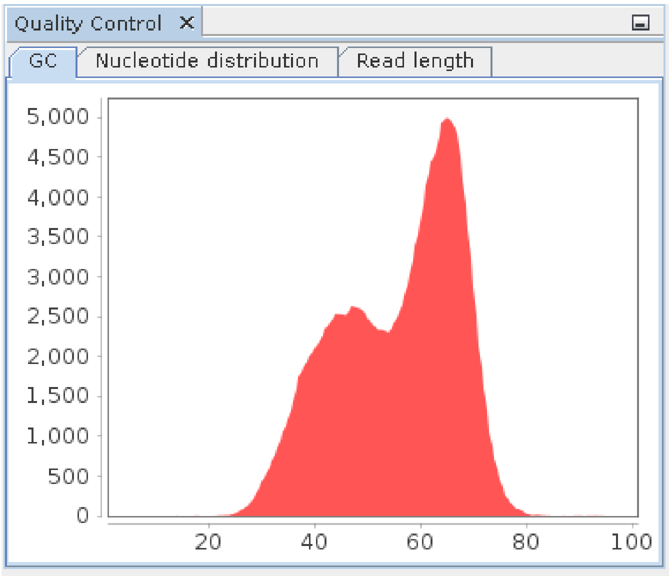


Figure 2.15: GC distribution of the simHC dataset.

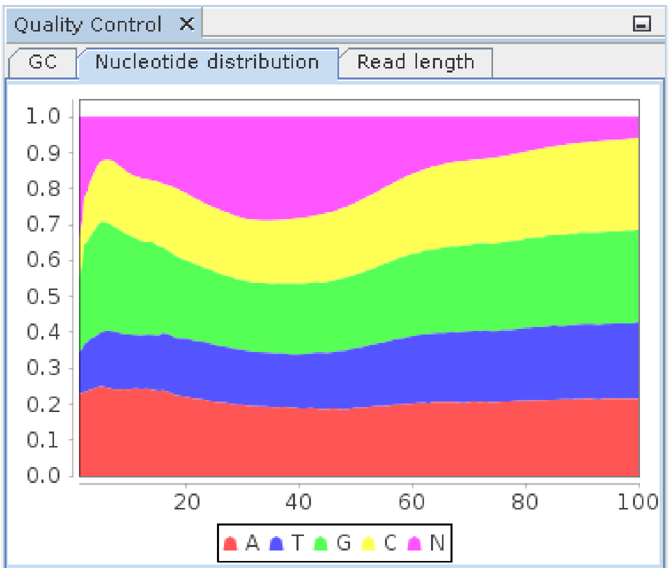


Figure 2.16: Nucleotide distribution of the simHC dataset. A high fraction of uncalled bases is apparent from the chart.

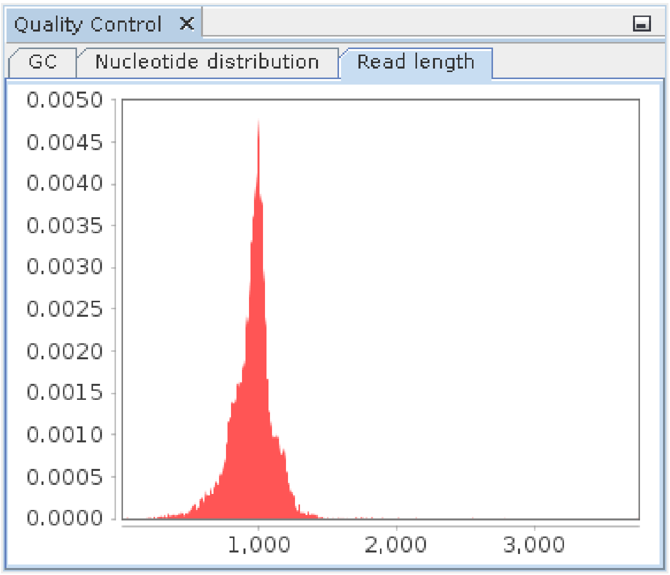
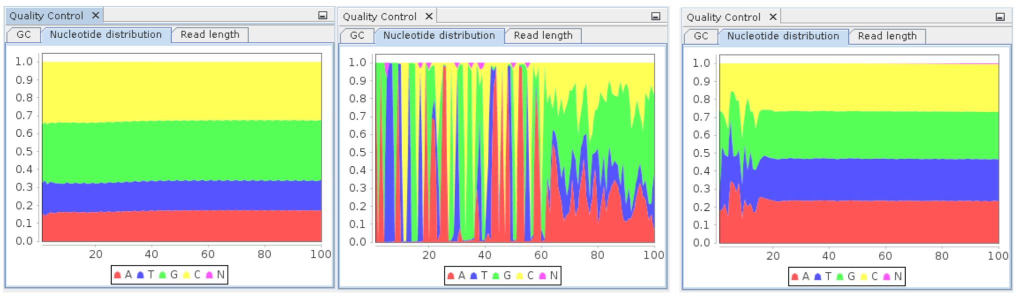


Figure 2.17: Read length distribution of the simHC dataset.



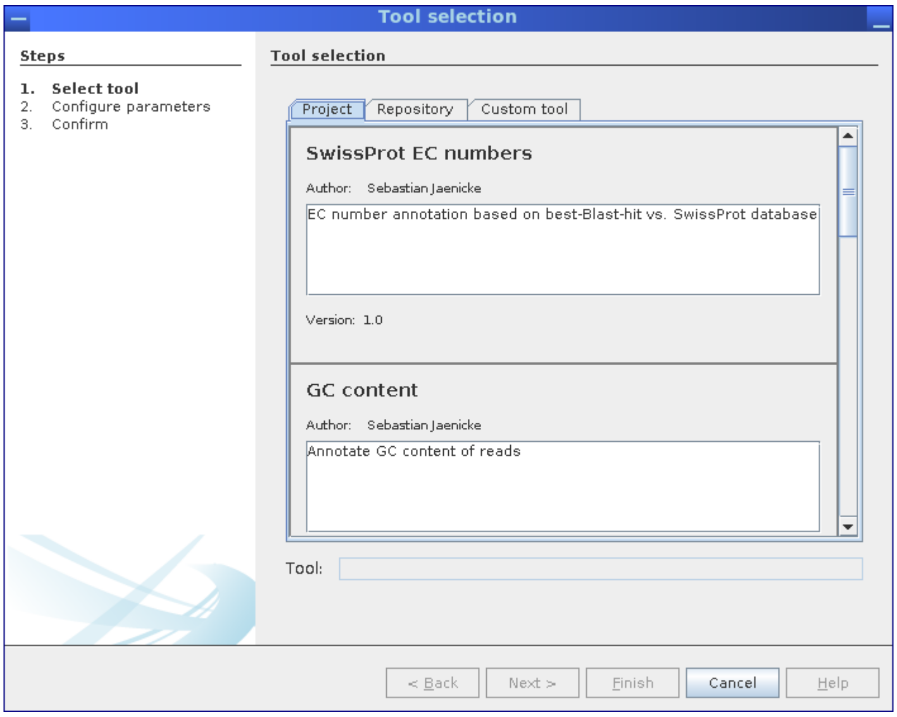
(a) High-GC (65%) data (b) Amplicon data (c) Adapter residue

Figure 2.18: Nucleotide distribution examples.

Depending on the kind of sequence data, different patterns might emerge (2.18), which might or might not warrant any further action. While small amounts of e.g. adapter residue are sometimes encountered and might be considered acceptable, it is up to the individual researcher to check back with their sequencing provider and ask for adapter sequences to execute additional trimming.

2.8 Defining and executing analysis jobs

2.8.1 Selecting an analysis pipeline



SwissProt EC numbers

Author: Sebastian Jaenicke

EC number annotation based on best-Blast-hit vs. SwissProt database

Version: 1.0

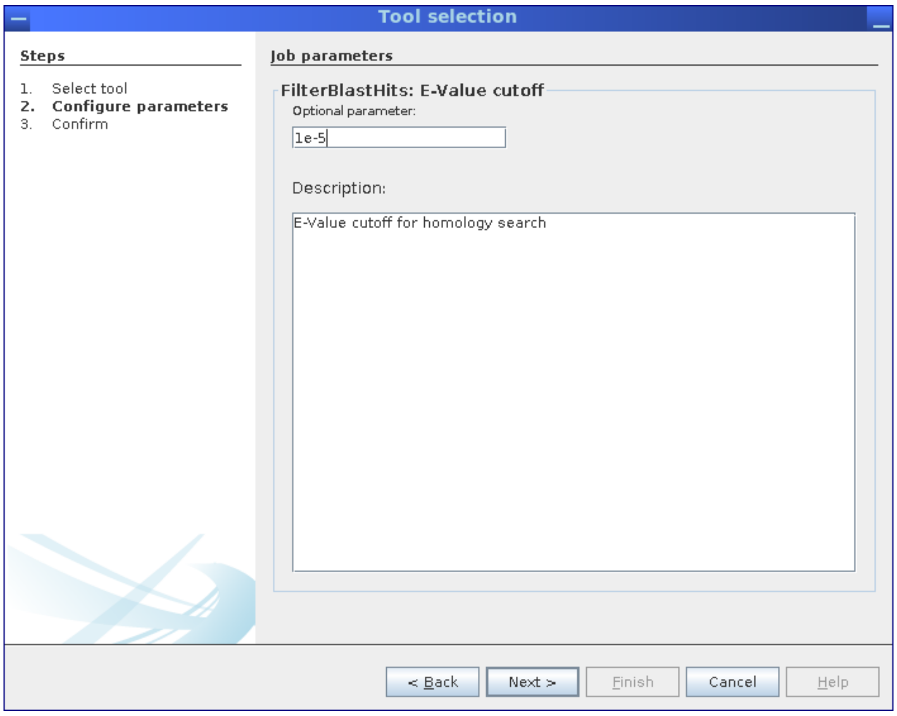
GC content

Author: Sebastian Jaenicke

Annotate GC content of reads

Figure 2.19: Analysis pipelines can be selected from the corresponding MGX project itself, from the repository of public pipelines provided by the server, or, a custom workflow can be uploaded and executed.

All analysis pipelines can be started from the context menu of the metagenome dataset to be analyzed, provided the user has been granted at least “User” level access. First, the user can choose the desired pipeline from the project itself, from the pipeline repository hosted on the MGX server, or upload an own pipeline implementation (2.19). Subsequently, analysis parameters can be reviewed and adapted (2.20) before submitting an analysis.



FilterBlastHits: E-Value cutoff

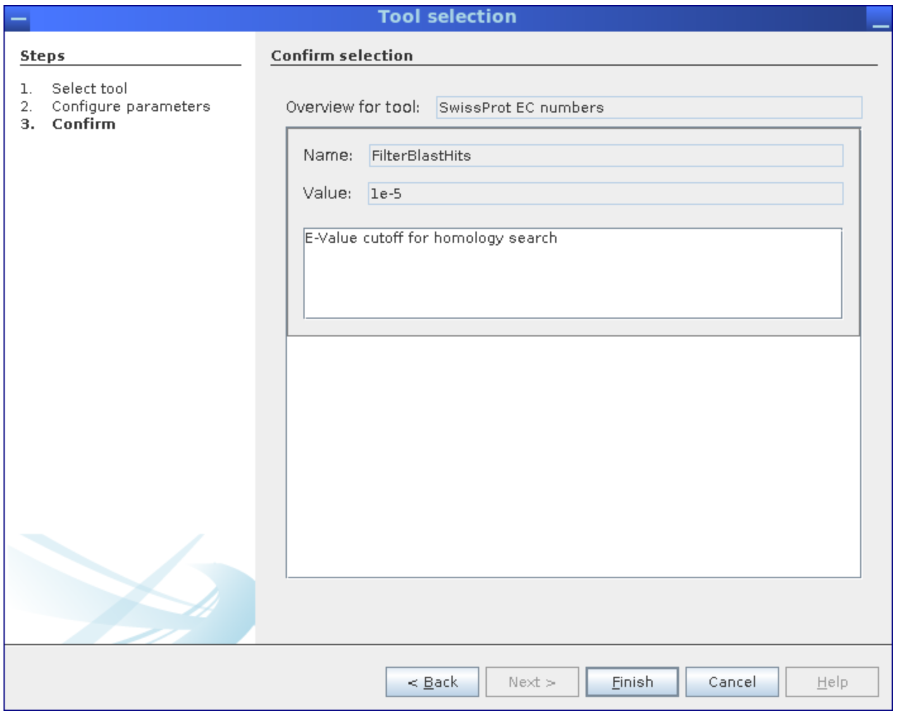
Optional parameter:

1e-5

Description:

E-Value cutoff for homology search

Figure 2.20: The wizard allows to inspect and adapt parameters for the selected pipeline. The actual number of steps depends on the number of parameters available for customization.



Confirm selection

Overview for tool: SwissProt EC numbers

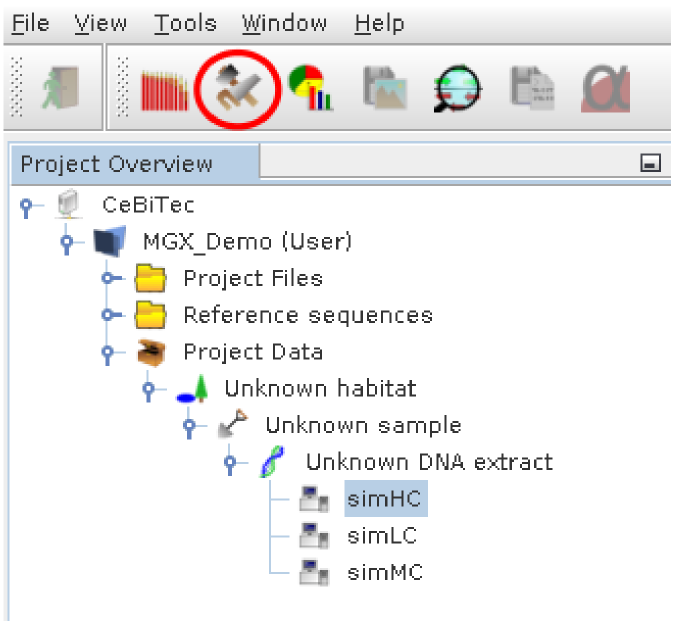
Name: FilterBlastHits

Value: 1e-5

E-Value cutoff for homology search

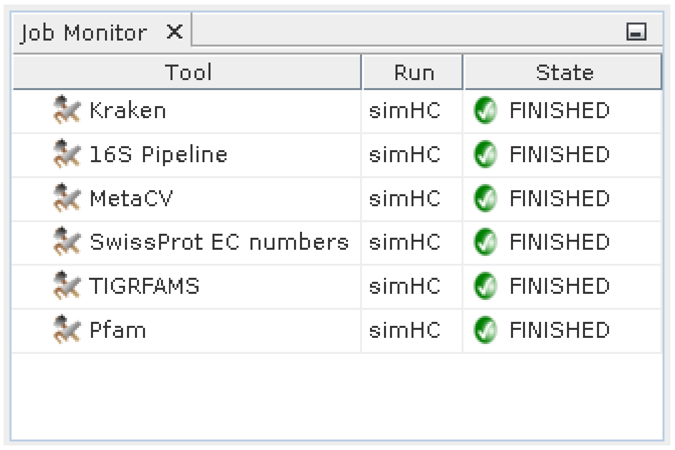
Figure 2.21: Before executing an analysis pipeline, a final overview of all parameters is shown. Once confirmed, the pipeline is submitted and scheduled for execution on the MGX server. Here, the selected pipeline has only one single parameter.

2.8.2 Monitoring job progress



- CeBiTec
  - MGX_Demo (User)
    - Project Files
    - Reference sequences
    - Project Data
      - Unknown habitat
        - Unknown sample
          - Unknown DNA extract
            - simHC
            - simLC
            - simMC

Figure 2.22: The Job Monitor component can be opened using its icon in the toolbar.



Tool	Run	State
Kraken	simHC	FINISHED
16S Pipeline	simHC	FINISHED
MetaCV	simHC	FINISHED
SwissProt EC numbers	simHC	FINISHED
TIGRFAMS	simHC	FINISHED
Pfam	simHC	FINISHED

Figure 2.23: The Job Monitor provides an overview of job states. Depending on context, all jobs within a MGX project or only jobs for a single dataset will be displayed.

The Job Monitor component is provided to inspect the state of jobs present within a MGX project. It can be opened from the toolbar (2.22)and will display the state of jobs scheduled for execution, currently running or already finished (2.23). In addition, it can also be used to delete jobs and corresponding analysis results when no longer needed.

Depending on the selected item in the Project Explorer, the Job Monitor will by default display all jobs within a project; if a single sequencing run is selected, only jobs for this dataset will be shown.

2.9 Visualization of results

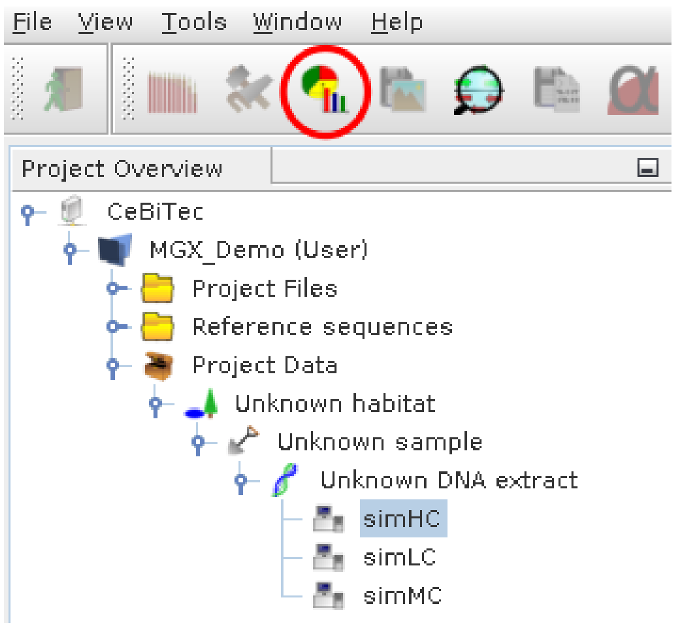


Figure 2.24: Icon for the visualization module.

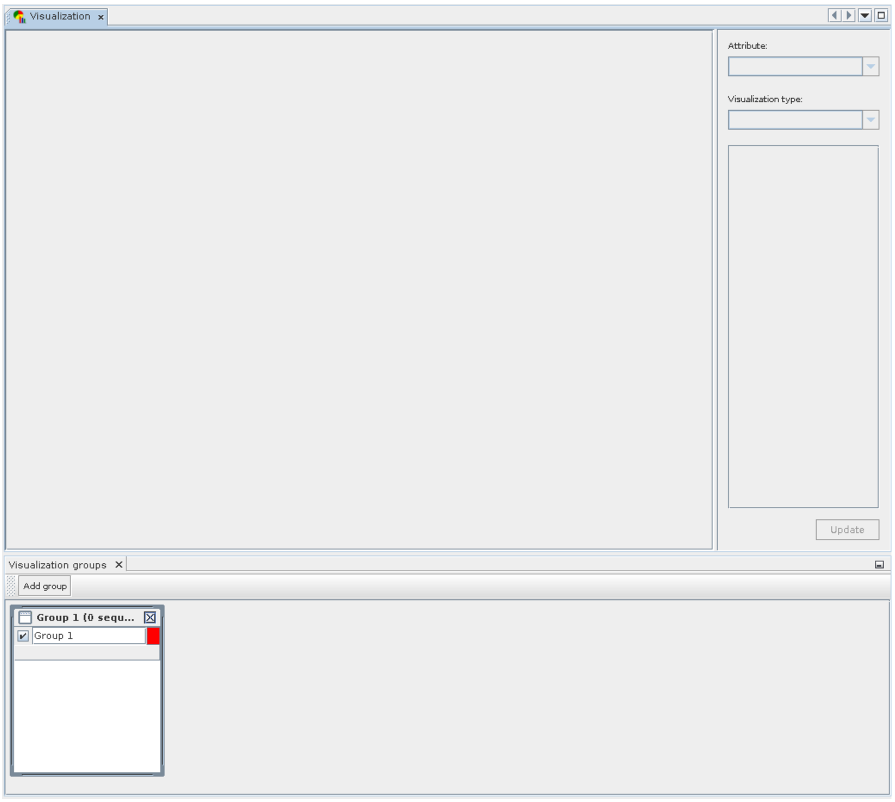
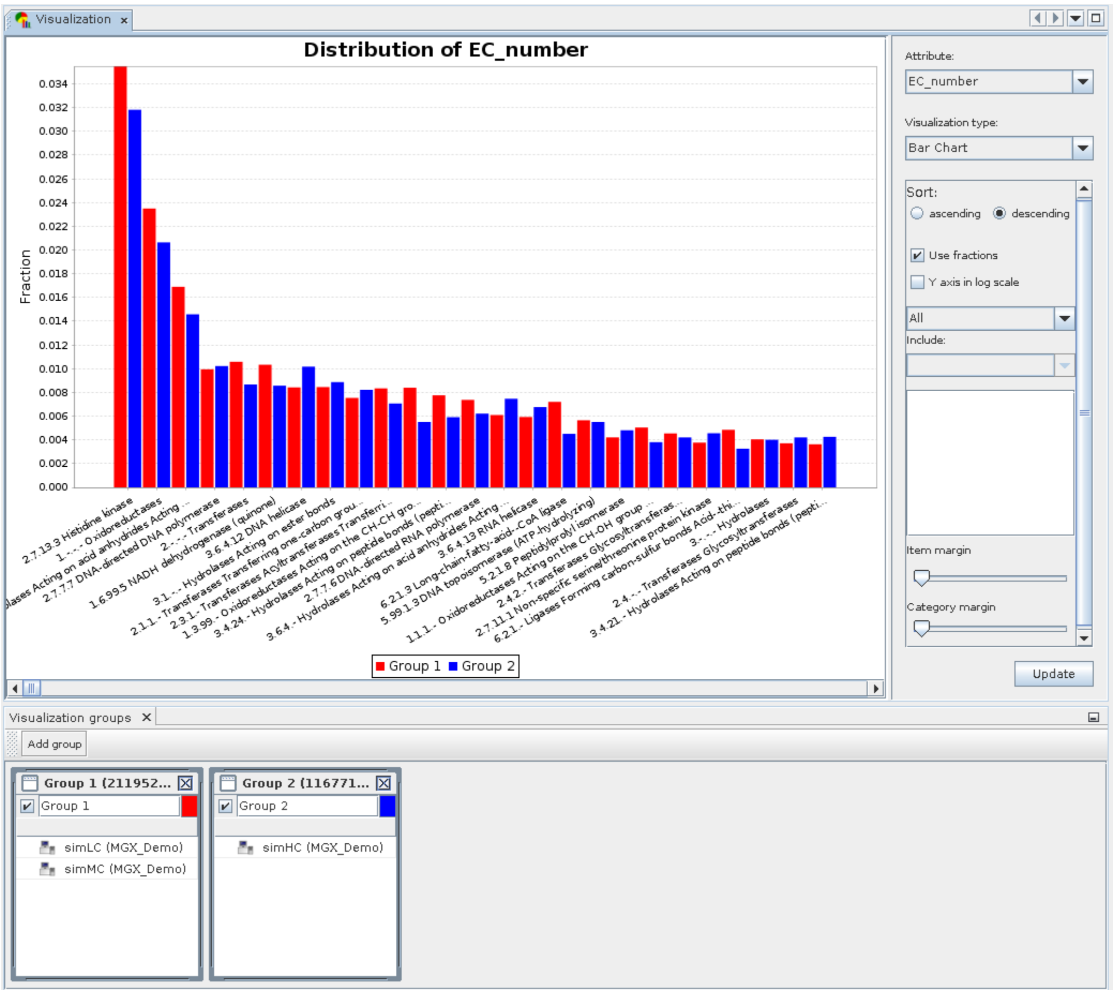


Figure 2.25: Components of the visualization module. The bottom window is used to create and define groups, while the top window allows to select result and visualization type, customize options and display the final chart.

The visualization component actually consists of two separate windows (2.25). The Group Window is located at the bottom and used to create, name and define groups used for data display. Sequencing runs can be added to individual groups using Drag and Drop. While any sequencing run can be present in several groups at the same time, it is not possible to add it to a group more than once. Groups may be assigned a name, their display color can be chosen by the user, and they may be temporarily excluded from display.
The main visualization window is shown at the top; once sequencing runs are added or changed, the type of analysis result to be shown can be chosen on the top right. Afterwards, the desired visualization type is selected and may be customized, as well.


Visualization
Distribution of EC_number
Fraction
0.034
0.032
0.030
0.028
0.026
0.024
0.022
0.020
0.018
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
2.7.13.3 Histidine kinase
1.-.-.- Oxidoreductases
lases Acting on acid anhydrides Acting ...
2.7.7.7 DNA-directed DNA polymerase
2.-.-.- Transferases
1.6.99.5 NADH dehydrogenase (quinone)
3.6.4.12 DNA helicase
3.1.-.- Hydrolases Acting on ester bonds
2.1.1.- Transferases Transferring one-carbon grou...
2.3.1.- Transferases Acyltransferases Transferri...
1.3.99.- Oxidoreductases Acting on the CH-CH gro...
3.4.24.- Hydrolases Acting on peptide bonds (pepti...
2.7.7.6 DNA-directed RNA polymerase
3.6.4.- Hydrolases Acting on acid anhydrides Acting ...
3.6.4.13 RNA helicase
6.2.1.3 Long-chain-fatty-acid--CoA ligase
5.99.1.3 DNA topoisomerase (ATP-hydrolyzing)
5.2.1.8 Peptidylprolyl isomerase
1.1.1.- Oxidoreductases Acting on the CH-OH group ...
2.4.2.- Transferases Glycosyltransferas...
2.7.11.1 Non-specific serine/threonine protein kinase
6.2.1.- Ligases Forming carbon-sulfur bonds Acid--thi...
3.-.-.- Hydrolases
2.4.-.- Transferases Glycosyltransferases
3.4.21.- Hydrolases Acting on peptide bonds (pepti...
🟥 Group 1 🟦 Group 2
Update
Visualization groups
Add group
Group 1 (211952...
Group 2 (116771...

Attribute: EC_number

Visualization type: Bar Chart

Sort: ascending [✓ descending]

[✓ Use fractions]

Y axis in log scale

All

Include:

Item margin

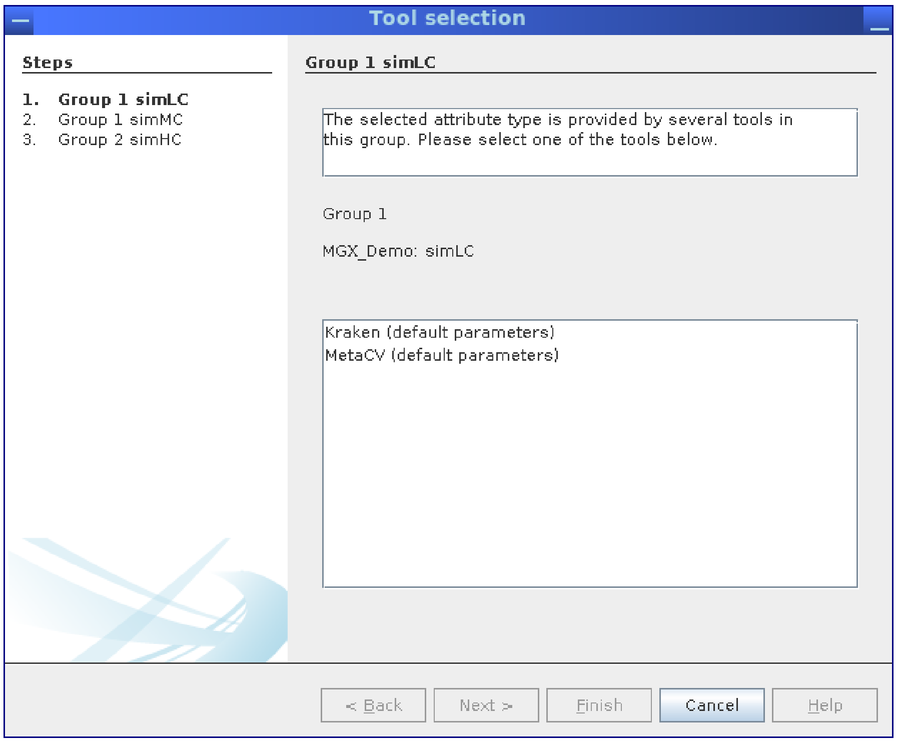
Category margin

Group 1 (211952...
[✓] Group 1 🟥
simLC (MGX_Demo)
simMC (MGX_Demo)

Group 2 (116771...
[✓] Group 2 🟦
simHC (MGX_Demo)

Figure 2.26: Visualization example displaying a bar chart for two groups.

Figure 2.26 shows a simple visualization example. Two groups were defined containing the simLC and simMC metagenomes in the first and the simHC metagenome in the second group. The main visualization window shows a bar chart generated for these groups based on assigned EC numbers. To account for differences in dataset size, both groups are normalized to fractions.



The selected attribute type is provided by several tools in this group. Please select one of the tools below.

Group 1

MGX_Demo: simLC

Kraken (default parameters)
MetaCV (default parameters)

Figure 2.27: In case a result is provided by more than one job, e.g. different taxonomic assignment methods, a dialog allows to select between jobs. In this example, taxonomic assignments are provided by MGX pipelines employing Kraken[Wood and Salzberg, 2014] as well as MetaCV[Liu et al., 2012].

Whenever a selected result type is provided by more than one analysis job, an interactive dialog allows the user to review possible jobs and select one (2.27). This scenario will occur when a tool is executed several times with different parameters, or when different taxonomic classifiers were used to analyse a dataset.

2.9.1 Exporting sequences

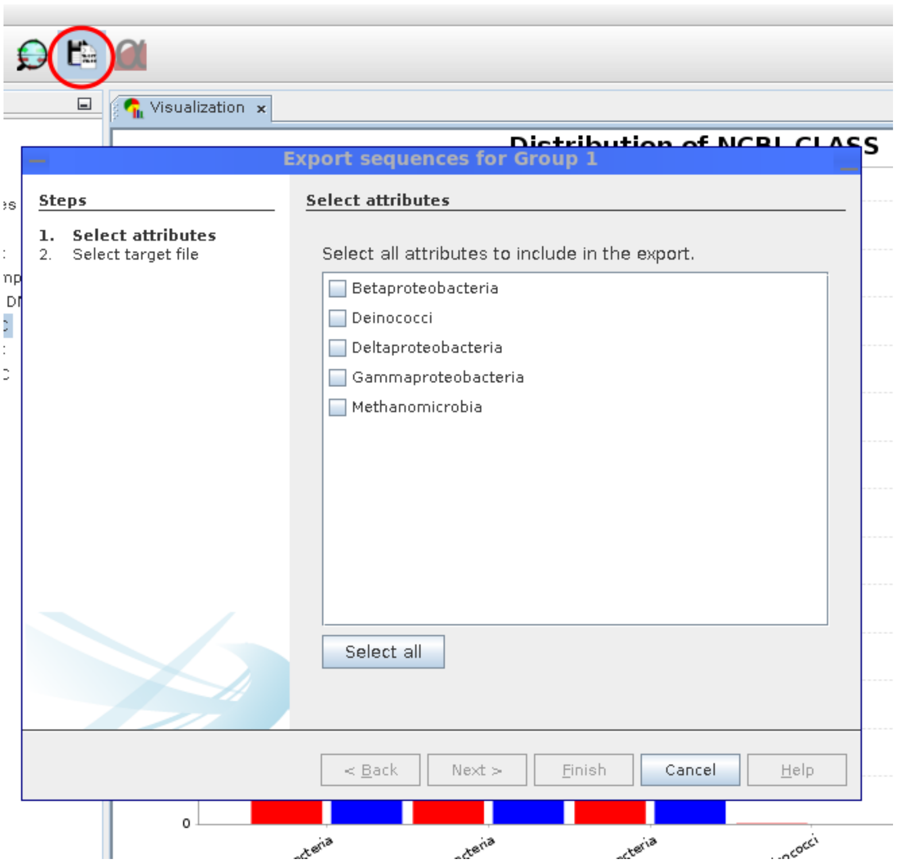
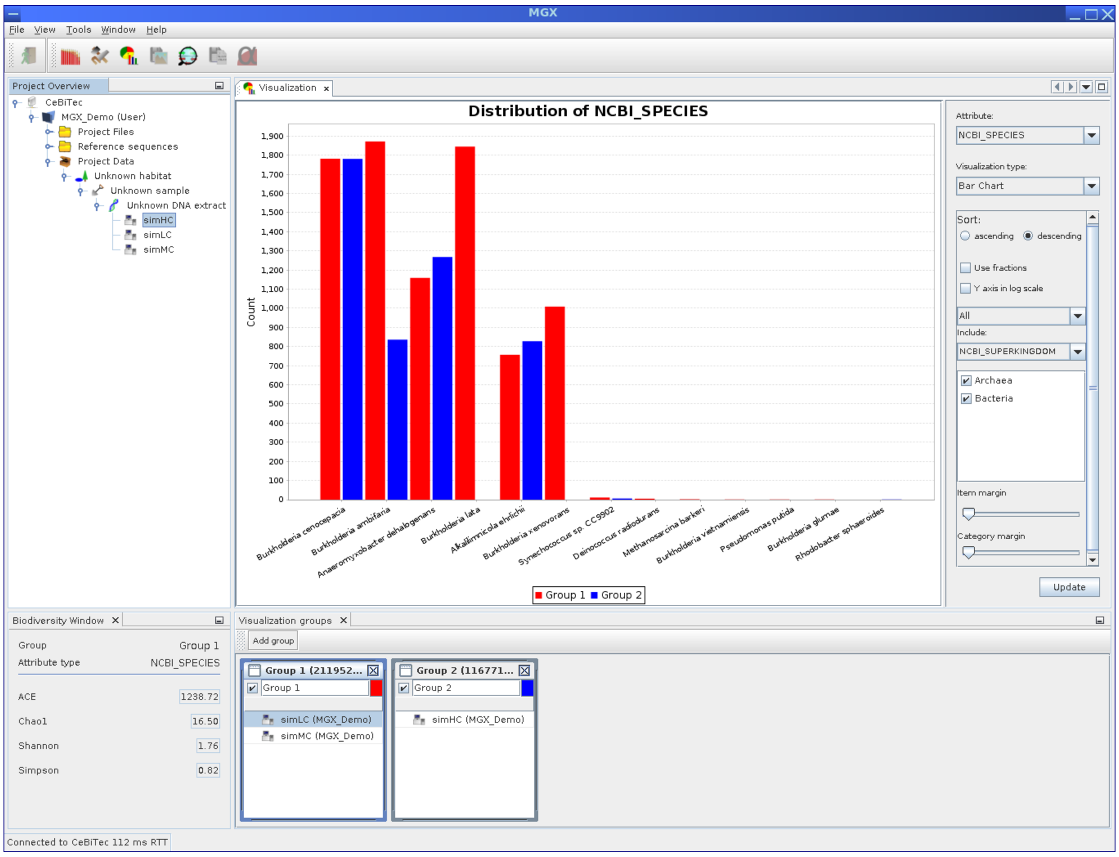


Figure 2.28: Sequences can be exported for each group individually. The user can freely choose which attributes should be included.

The sequence export wizard allows to export sequences conditionally based on analysis results. For each visualization group, it is possible to choose the attributes for which sequences should be obtained (2.28). Sequences are subsequently downloaded from the MGX server and saved in FASTA format files for each group.

2.9.2 Biodiversity indices



Attribute: NCBI_SPECIES
Visualization type: Bar Chart
Sort: ascending [✓ descending]
☐ Use fractions
☐ Y axis in log scale
All
Include: NCBI_SUPERKINGDOM
☑ Archaea
☑ Bacteria
Item margin
Category margin

Biodiversity Window

Group	Group 1
Attribute type	NCBI_SPECIES
ACE	1238.72
Chao1	16.50
Shannon	1.76
Simpson	0.82

Figure 2.29: Biodiversity indices (bottom left) such as ACE or Shannon are computed for the currently selected visualization group.

The visualization module features another component used to display commonly used biodiversity indices, such as the ACE, Shannon[Spellerberg and Fedor, 2003], Chao1[Hughes et al., 2002] and Simpson[Simpson, 1949] indices. The component will automatically show the index values for the currently selected visualization group based on the chosen attribute type (2.29).

2.10 Uploading own files

For each project, MGX provides dedicated storage (2.30) in order to allow users to provide custom data, subsequently to be used with analysis pipelines. Thus, own sequence collections or HMM model files for genes of interest can easily be uploaded and later on be included into analysis pipelines.

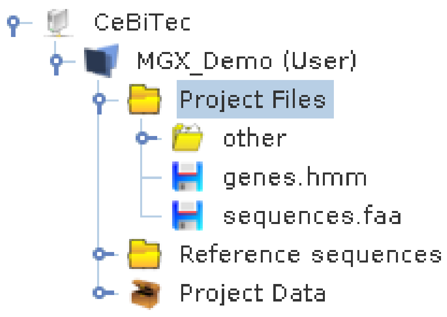


Figure 2.30: Each project includes flat storage where custom data can be stored, e.g. own FASTA files to be used as reference databases for metagenome analysis.

While it may be necessary to implement an own pipeline depending on the desired kind of analysis, the MGX repository hosts predefined pipeline templates addressing the most common cases: The "BestHit-Blast" template can be used to annotate metagenome sequences with the description of a Blast hit after the user has uploaded a FASTA file containing amino acid sequences, and the "BestHit-HMM" template provides the same functionality for hidden Markov models (HMMs).

2.11 Reference mapping

2.11.1 Importing reference genomes

MGX provides several analysis pipelines to align metagenome reads to reference genomes; before these pipelines can be used, the corresponding reference genome has to be added to the project. There are two possible ways to achieve this: The MGX repository hosts published and annotated reference genomes obtained from the NCBI; in addition, users may choose to upload a custom reference sequence in FASTA, GenBank or EMBL format, e.g. a finished but unpublished genome not available from official sources.

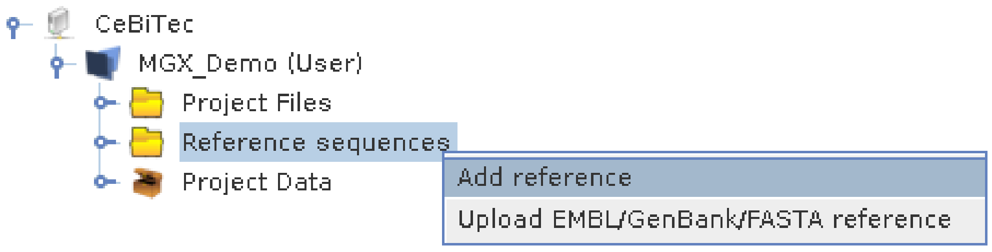


Figure 2.31: Reference sequences to be used as mapping targets may be imported from the global repository or uploaded in EMBL/GenBank/FASTA format.

To add a reference genome, right-click on the "Reference sequences" node within the project view and select either “Add reference” to access the MGX repository or “Upload EMBL/GenBank/FASTA reference” to provide an own sequence (2.31).

Once the import is complete, the reference genome is available for analysis and can be selected for the corresponding analysis pipelines which provide reference mapping, e.g. BowTie or FR-HIT.

2.11.2 Displaying reference mappings

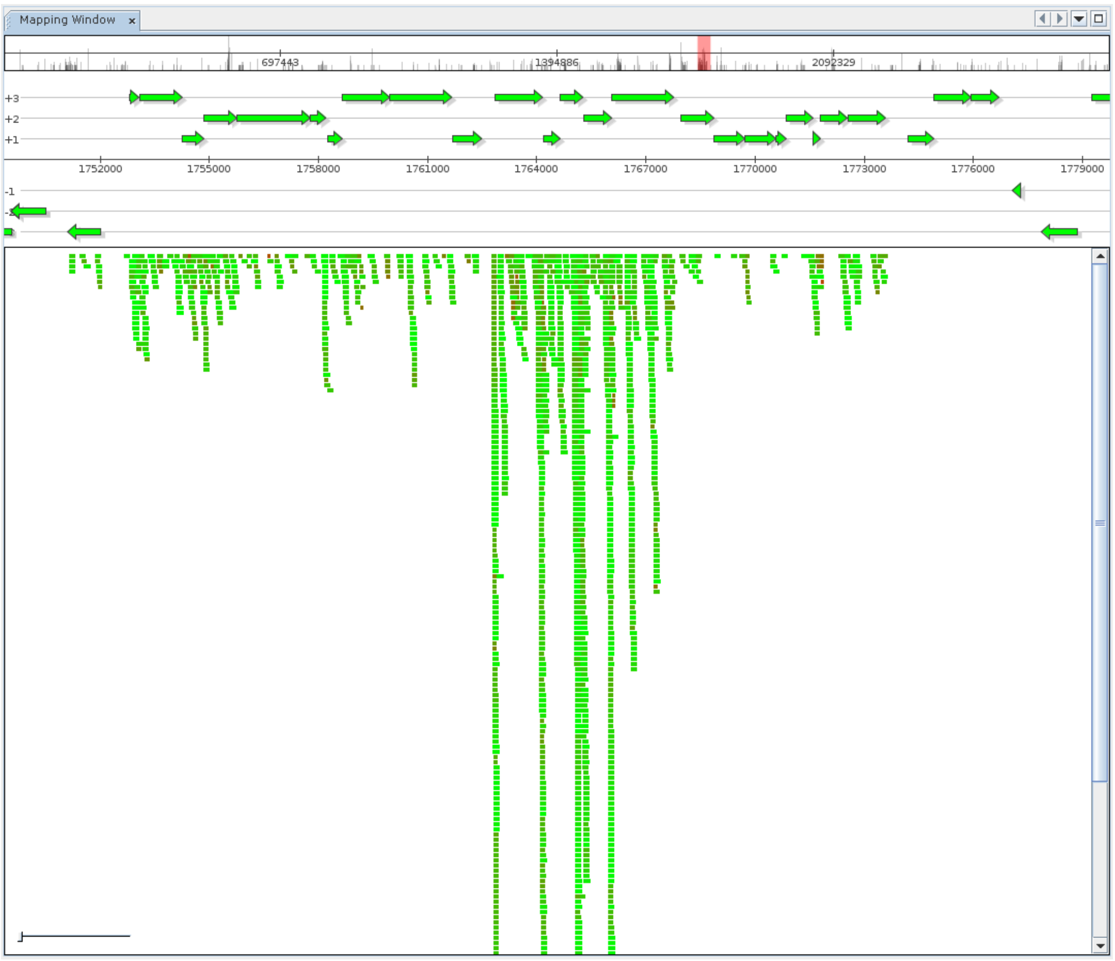


Figure 2.32: The reference mapping component showing alignment results for a metatranscriptome dataset mapped to the reference genome of one of the dominant organisms. From top to bottom, the component displays a) navigation and coverage histogram, b) currently selected interval and c) aligned DNA sequences for the interval. Color coding refers to relative sequence identity.

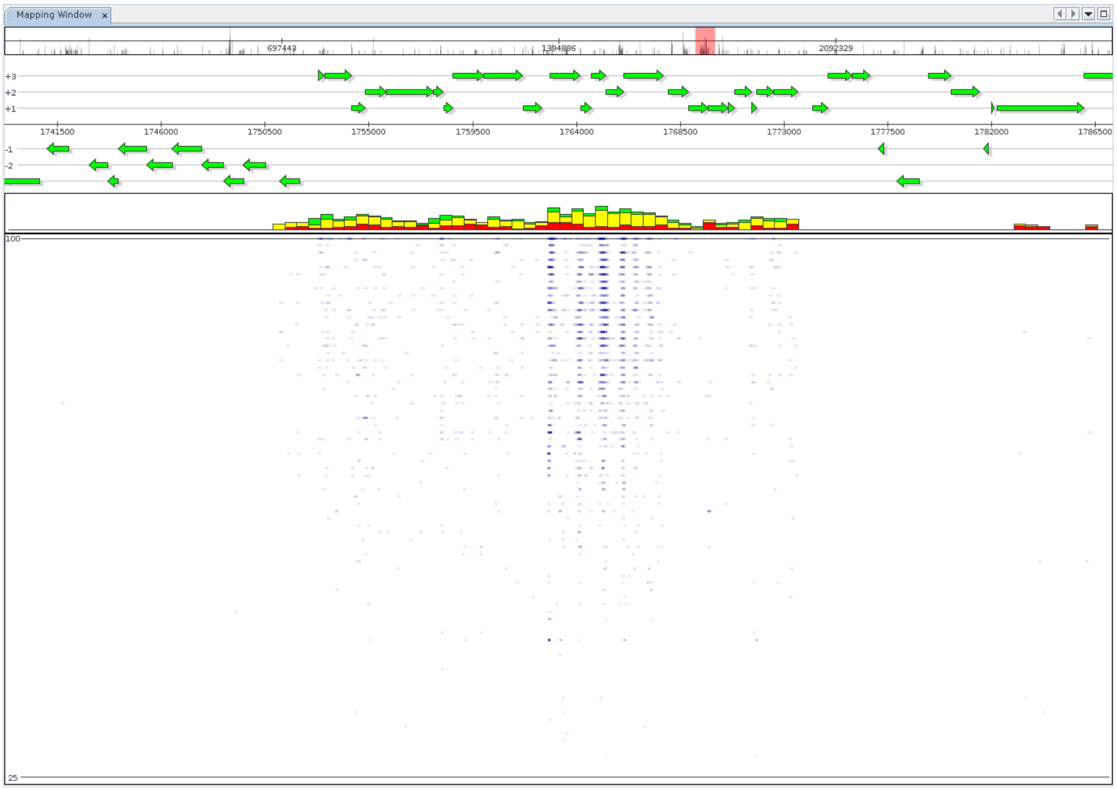


Figure 2.33: An alternate visualization mode is the generation of fragment recruitment plots, here showing the same data as described previously (2.32). The view mode features the fragment recruitment plot itself and additionally provides stacked bars summarizing mapping identity within reference intervals, grouped into low (red), medium (yellow, ≥ 75%) and high (green, ≥ 97%) quality mappings.

Alignment of metagenome or metatranscriptome data to reference sequences of known origin allows researchers to evaluate relative identity between metagenome sequences and the actual strain or to obtain an overview of gene expression within a meta-transcriptome. MGX currently provides predefined pipelines employing BLAST[Altschul et al., 1990], FR-HIT[Niu et al., 2011] and Bowtie 2[Langmead and Salzberg, 2012]. The reference mapping component is provided to inspect and browse alignment results, offering both a generic alignment view where each mapped sequence is colored according to mapping identity, as well as a fragment recruitment view. Switching between view modes is possible from the context menu of the mapping component.

2.12 Search

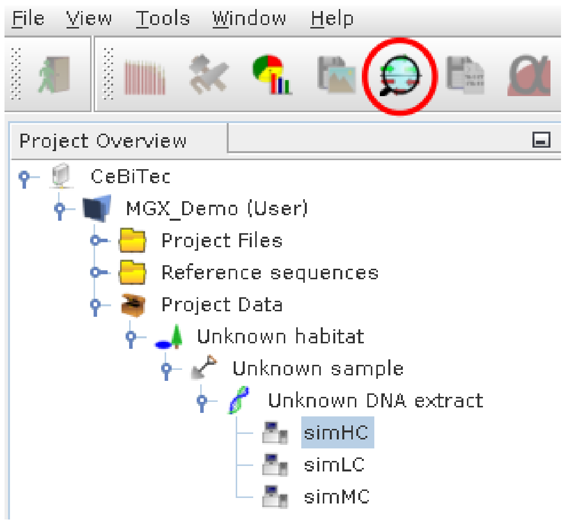
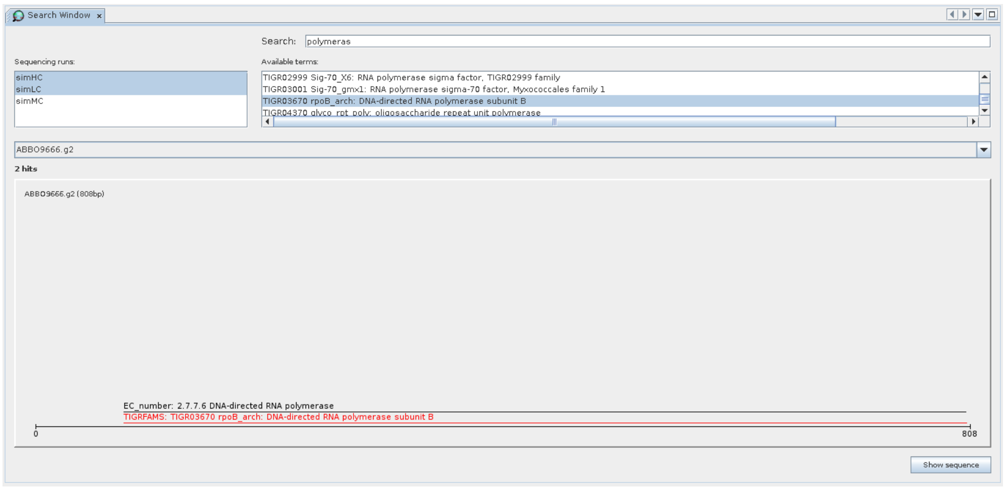


Figure 2.34: Icon for the metagenome search component.



Search: polymeras

Sequencing runs:
simHC
simLC
simMC

Available terms:
TIGR02999 Sig-70_X6: RNA polymerase sigma factor, TIGR02999 family
TIGR03001 Sig-70_gmx1: RNA polymerase sigma-70 factor, Myxococcales family 1
TIGR03670 rpoB_arch: DNA-directed RNA polymerase subunit B
TIGR04370 glyco_rpt_poly: oligosaccharide repeat unit polymerase

ABBO9666.g2

2 hits

ABBO9666.g2 (808bp)

EC_number: 2.7.7.6 DNA-directed RNA polymerase
TIGRFAMS: TIGR03670 rpoB_arch: DNA-directed RNA polymerase subunit B

Figure 2.35: Search component showing results for the term "polymeras". The search was performed within the select metagenomes simHC and simLC (top left); the bottom part shows an individual sequence identified by the search. Search results are displayed together with all other attributes available for a sequences, thus allowing to identify co-occurence of results.

3 Implementing custom analysis pipelines

All offered analysis tools provided by the MGX platform are implemented as workflows for the Conveyor[Linke et al., 2011] workflow engine developed by B. Linke. Within Conveyor, tools are provided as so-called "nodes", which resemble individual processing steps and which are used to implement novel analysis methods by simply arranging and connecting them into a larger workflow. Conveyor currently includes plugins providing typical bioinformatics tools like BLAST or HMMer, but has recently been extended with dedicated plugins aimed at metagenome analysis, like MetaCV, MetaPhyler or MetaPhlAn, which all perform taxonomic analysis. A dedicated Conveyor plugin provides access to MGX data structures, thereby enabling the analysis of metagenomes stored in the MGX system with processing tools provided by Conveyor itself. While workflow definitions are stored in a XML-based format, a graphical user interface, the Conveyor Designer (3.1), enables users to implement new analysis by simply placing and connecting nodes.

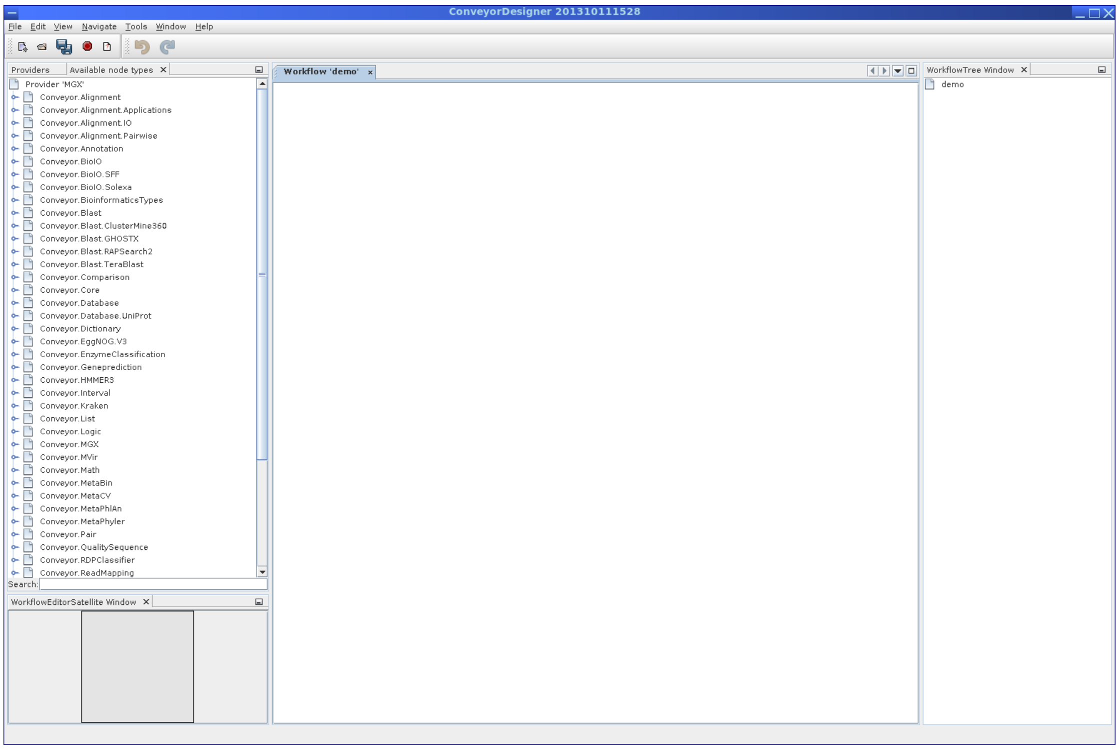


Figure 3.1: The Conveyor Designer application allows easy and user-friendly development of custom analysis algorithms in a graphical way.

As Conveyor is actively developed and new tools are continously integrated, giving a thorough introduction to Conveyor is beyond the scope of this document. The most up-to-date documentation describing Conveyor itself and the Conveyor Designer in particular can be found at the Conveyor web site `http://www.uni-giessen.de/fbz/fb08/bioinformatik/software/Conveyor`.

3.1 Getting started

In order to implement a custom workflow, the Conveyor Designer needs to be configured with a definition of available Conveyor plugins and node types. This is easily achieved by importing a plugin dump file, which contains a list of data types and nodes provided by a Conveyor installation.

To use the Designer to implement a workflow for the MGX framework, a corresponding plugin dump file can be obtained from within MGX by right-clicking on the project name (3.2).

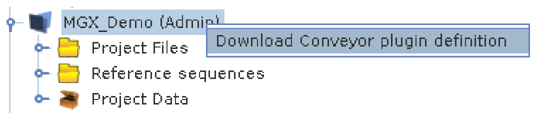


Figure 3.2: A plugin dump file for use with the Conveyor Designer can be obtained from within MGX by right-clicking on the project name.

Afterwards, start the Designer application and define a new provider (Right-click on “Available providers”). Make sure to specify “Plugin dump file” (3.3) as the type of plugin set and select the file generated by MGX. Once the plugin dump file has been imported, you are ready to implement new workflows. Initially starting with an empty sheet, nodes can be dragged from the list of all available nodes on the left and placed onto the sheet. Node connections are created by clicking on a node, keeping the mouse button pressed and releasing it over the connections target node, thus creating the link; in ambiguous cases, e.g. for nodes with several unconnected inputs/outputs, a dialog will allow to select the desired connection. Nodes may also require node-specific configuration, which can be edited from a nodes context menu. A red border around a node indicates missing configuration items or connections.

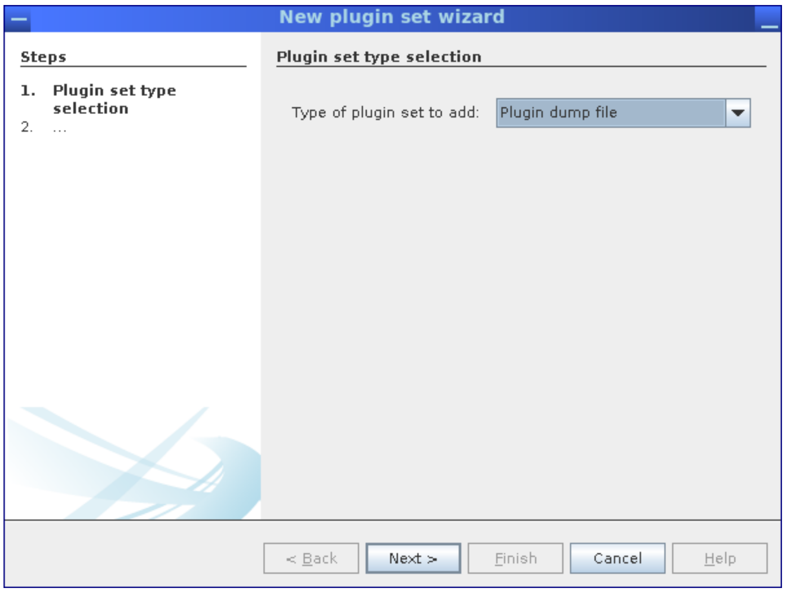

New plugin set wizard
Steps
1. Plugin set type selection
2. ...
Plugin set type selection
Type of plugin set to add: Plugin dump file
< Back
Next >
Finish
Cancel
Help

Figure 3.3: Importing a plugin dump file into the Conveyor Designer.

3.2 Workflow requirements

In order to design custom Conveyor workflows for later usage within the MGX platform, there are several constraints to be met which will be described in more detail.

First of all, a dedicated `GetMGXJob` node (Figure 3.4) has to be present within the workflow; in addition, this node has to be named "mgx". During execution of a pipeline within MGX, this node is configured via an external configuration file, providing required information about a jobs context, like e.g. access to a project database and associated storage.

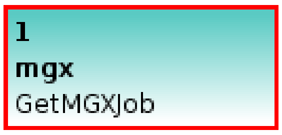


Figure 3.4: The `GetMGXJob` node provides necessary context for executing a workflow within MGX, such as database access. By convention, this node has to be named mgx.

Access to metagenome DNA sequences is provided via the `ReadCSF` node, which will provide all metagenome sequences for a sequencing run object within MGX, except those for which the "discard" flag has already been set. As pipelines are always executed for one single analysis job, this node needs to be connected to the `GetMGXJob` node (3.5). Figure 3.6 shows a minimal example of a Conveyor-based pipeline for use within the MGX framework. Once executed, the pipeline would set the discard flag for all sequences.

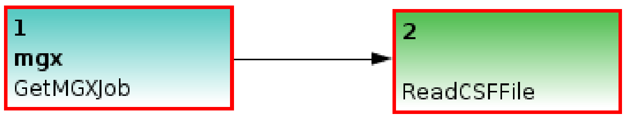


Figure 3.5: The `ReadCSF` node is used to obtained metagenome sequence data from within MGX; it has one input and needs to be connected to the `GetMGXJob` node.

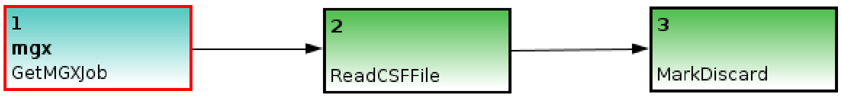


Figure 3.6: A minimal working example of a pipeline developed for MGX, which would set the discard flag for all sequences.

3.3 Annotating metagenome sequences

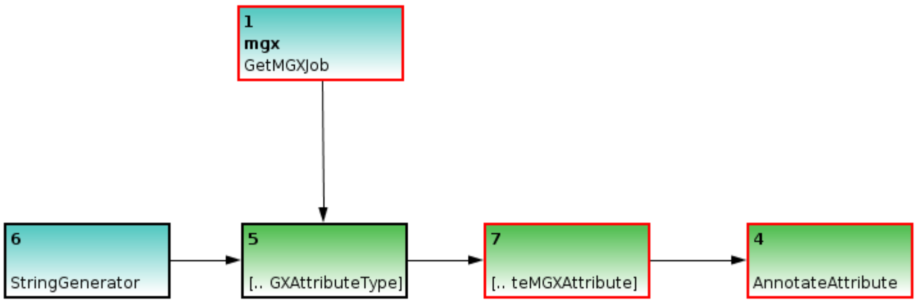
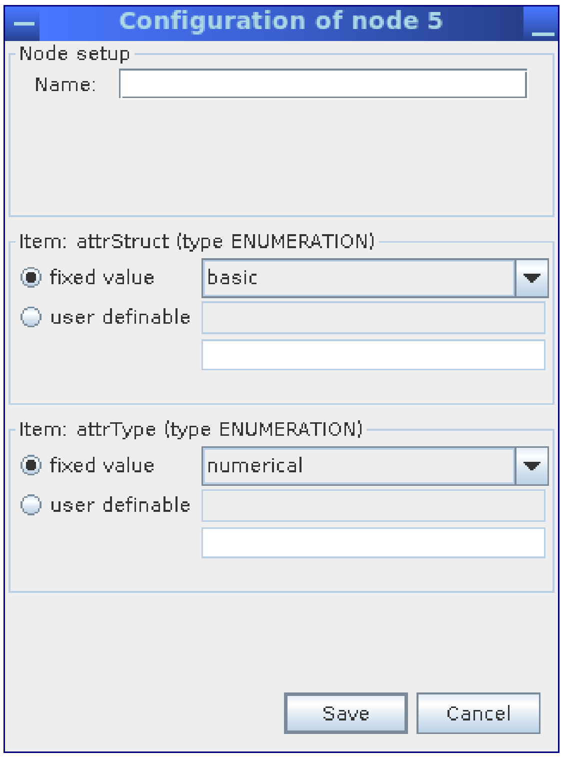


Figure 3.7: Basic template to illustrate sequence annotation. A `StringGenerator` is used to generate a label for the attribute type (`CreateMGXAttributeType`), which also requires job context information. The attribute type is required to create attributes, thus the node is connected to the `CreateMGXAttribute` node. Finally, the annotation can be saved to the project database (`AnnotateAttribute` node).

Annotation of metagenome sequences requires an “attribute type” and an “attribute”. As an example, we will illustrate the implementation of a pipeline for the analysis of GC content within metagenome sequences. We use a `StringGenerator` node configured to generate the string "GC" to create a label for the attribute type. As GC content is indicated by a number, we appropriately configure the `CreateMGXAttributeType` node to emit a basic (i.e. not hierarchical) as well as numerical "attribute type" (3.8).


Configuration of node 5

Node setup
Name:

Item: attrStruct (type ENUMERATION)
(● fixed value) basic
user definable

Item: attrType (type ENUMERATION)
(● fixed value) numerical
user definable

Figure 3.8: Within the configuration dialog for the `CreateMGXAttributeType` node, structure and type of the generated attribute values are defined.

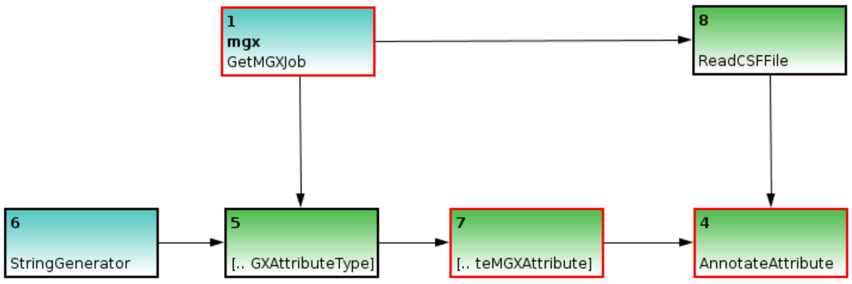


Figure 3.9: Incomplete example; extending upon 3.7, the `ReadCSF` node will provide the necessary metagenome sequences to be annotated. Still, there is no actual analysis specified.

In a second step, we use the `ReadCSF` node to obtain access to the individual metagenome sequences; as MGX annotates sequences individually, a connection between `ReadCSF` and `AnnotateAttribute` is required (3.9). Subsequently, we implement the actual analysis, which is provided by the `GCContent` node. It will process all sequences and emit the corresponding GC content for each of them. To convert these values to appropriate “attributes”, an “attribute type” is required for each value; therefore, a `Repeat` node is inserted between nodes 5 and 7 (3.10).

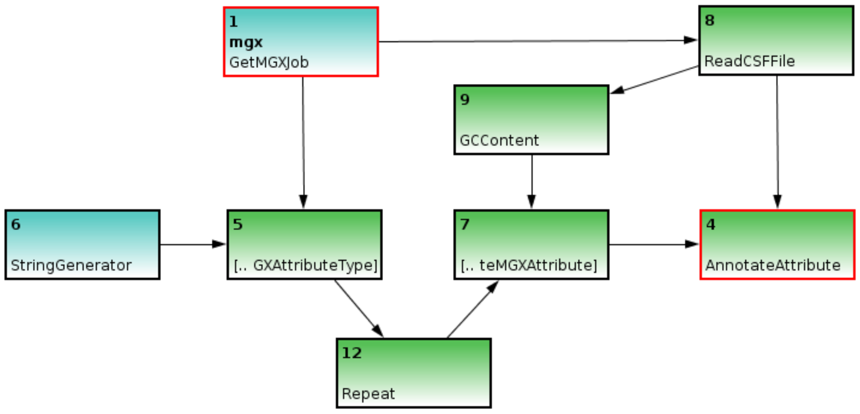


Figure 3.10: Step 2: The `GCContent` node represents the actual analysis step; it is used to determine the GC content of a DNA sequence, which will then be converted into an “attribute”. Since an “attribute type” is required for each “attribute”, a `Repeat` node is inserted between nodes 5 and 7.

Finally, as an annotation always refers to only a part of a sequence, we will need to generate the corresponding start and end coordinates; since GC content refers to the full sequence, we can use an `ULongGenerator` node configured to emit 0 (MGX uses 0-based coordinates) to generate the start coordinate; this node needs to be connected to a `Repeat` node to generate a series of 0s.
The end coordinate can be created based on the sequences’ length, with 1 subtracted, obtained through the `GetLength` and `MinusOne` nodes (3.11).
The `GetMGXJob` node will retain its red border due to missing configuration; this, however, can be ignored, as appropriate configuration will be provided by the MGX framework automatically.

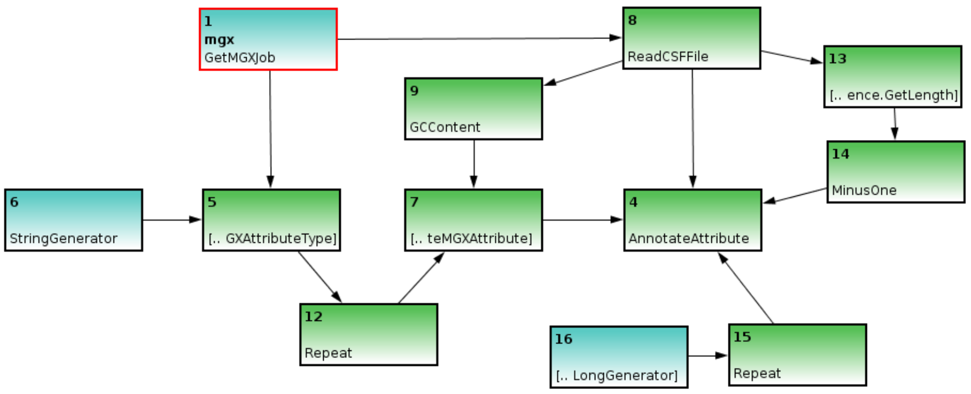


Figure 3.11: Completing the workflow: the `ULongGenerator` and `GetLength` nodes are added to specify coordinates for the subregion of the DNA sequence described by the “attribute”; the start coordinate is simply repeated, while 1 is subtracted from the sequences length due to 0-based coordinates.

3.3.1 Creating hierarchical attributes

Annotation of hierarchical attributes requires a little more effort. The `CreateHierarchicalMGXAttribute` node is used to obtain the inner structure of the hierarchy in a bottom-up approach; It contains several loops which will be explained in more detail.

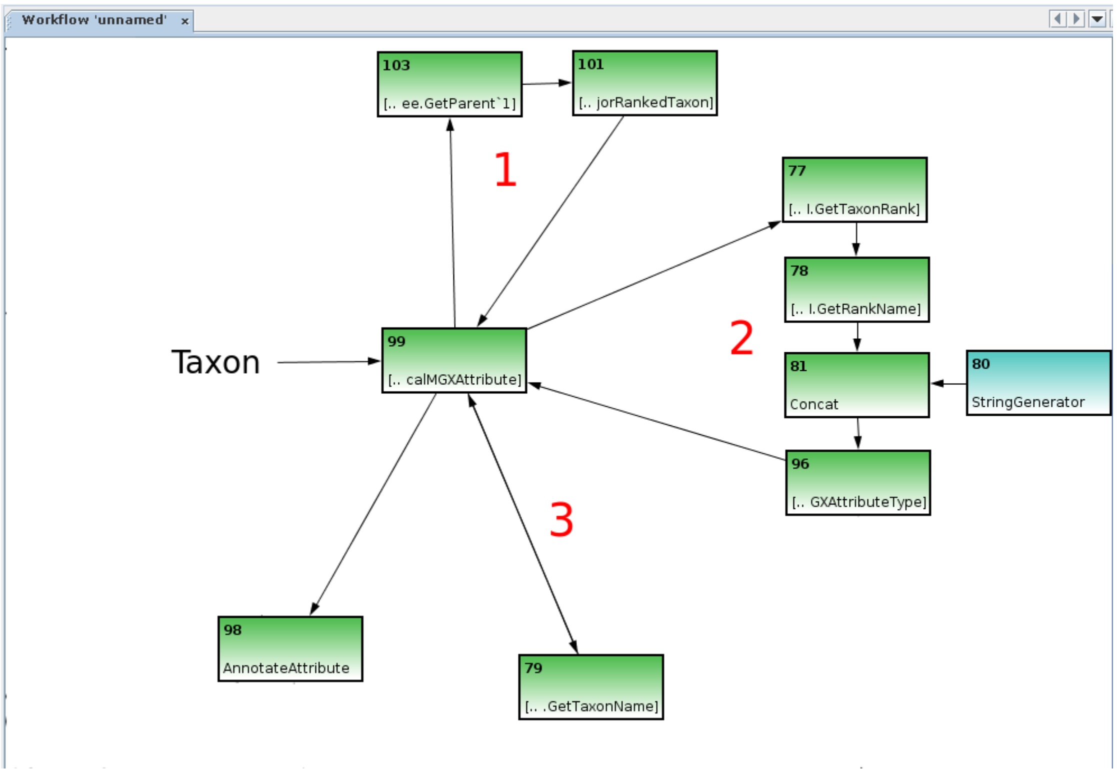


Figure 3.12: The `CreateHierarchicalMGXAttribute` node requires three loops (note double-ended arrow on third loop between nodes 99 and 79) to create the internal structure of the hierarchy. Several connections were removed from the figure for illustrative purposes.

A single object, e.g. a NCBI taxon generated by the Kraken[Wood and Salzberg, 2014] classifier, is provided as an input into the node (3.12). The first loop is required to obtain the objects parent object, thus defining the hierarchy. In this example it is implemented using the `GetParent` and `GetMajorRankedTaxon` nodes, thus making sure only the major taxonomic ranks (superkingdom, phylum, class, . . .) are included.
The second loop is used to obtain the corresponding attribute type for an object: it operates on the initial taxon as well as its parents obtained by the first loop. `GetTaxonRank` and `GetRankName` nodes provide the corresponding ranks' name, e.g. "class"; The `StringGenerator` and `Concat` nodes are then used to create the attribute type: "NCBI_class". This value is used to create the corresponding attribute type employing the `CreateMGXAttributeType` node, which is returned into the `CreateHierarchicalMGXAttribute` node.
The third and final loop is used to map a data object to its name, which is used to create the attributes value; it is built up using the `GetTaxonName` node, which delivers its output back into the node.

Thus, the three loops might be termed as Get parent, Get AttributeType for object and Generate value.

The `CreateHierarchicalMGXAttribute` node emits a hierarchical MGXAttribute for the initial data object, with the corresponding AttributeType provided by loop 2 and the MGXAttribute's value obtained using loop 3. Internally, loop 1 is used repetitively until the root node is reached, with all intermediary results passing through loops 2 and 3, thus generating a single path of hierarchical attributes within the taxonomic tree. The output of the `CreateHierarchicalMGXAttribute` is connected to the `AnnotateAttribute` node as in the previous example.

For brevity's sake, several connections are hidden within the image, which have already been explained in the previous section; the `CreateMGXAttributeType` node needs an incoming connection providing a MGXJob, and the `AnnotateAttribute` node requires additional connections providing the sequence to be annotated and start/stop coordinates for the subregion which is described by the annotation.

Bibliography

[Altschul et al., 1990] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. Journal of molecular biology, 215(3):403–410.

[Hughes et al., 2002] Hughes, J. B., Hellmann, J. J., Ricketts, T. H., and Bohannan, B. J. (2002). Counting the uncountable: Statistical approaches to estimating microbial diversity. Applied and environmental microbiology, 68(1):448.

[Langmead and Salzberg, 2012] Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. Nature Methods, 9(4):357–359.

[Linke et al., 2011] Linke, B., Giegerich, R., and Goesmann, A. (2011). Conveyor: a workflow engine for bioinformatic analyses. Bioinformatics, 27(7):903–911.

[Liu et al., 2012] Liu, J., Wang, H., Yang, H., Zhang, Y., Wang, J., Zhao, F., and Qi, J. (2012). Composition-based classification of short metagenomic sequences elucidates the landscapes of taxonomic and functional enrichment of microorganisms. Nucleic Acids Research, page gks828.

[Mavromatis et al., 2007] Mavromatis, K., Ivanova, N., Barry, K., Shapiro, H., Goltsman, E., McHardy, A. C., Rigoutsos, I., Salamov, A., Korzeniewski, F., Land, M., et al. (2007). Use of simulated data sets to evaluate the fidelity of metagenomic processing methods. Nature Methods, 4(6):495–500.

[Niu et al., 2011] Niu, B., Zhu, Z., Fu, L., Wu, S., and Li, W. (2011). FR-HIT, a very fast program to recruit metagenomic reads to homologous reference genomes. Bioinformatics, 27(12):1704–1705.

[Simpson, 1949] Simpson, E. H. (1949). Measurement of diversity. Nature.

[Spellerberg and Fedor, 2003] Spellerberg, I. F. and Fedor, P. J. (2003). A tribute to Claude Shannon (1916–2001) and a plea for more rigorous use of species richness, species diversity and the Shannon–Wiener Index. Global ecology and biogeography, 12(3):177–179.

[Wood and Salzberg, 2014] Wood, D. and Salzberg, S. (2014). Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biol, 15(3):R46.

List of Figures

List of Tables