

MUSCLE Help and Documentation

[Input form](#) | [Other MSA Tools](#) | [Bioinformatics Tools FAQ](#) | [Help & Support](#)

Contents

- [Introduction](#)
- [How to Access MUSCLE](#)
 - [Web Form](#)
 - [Web Services](#)
 - [REST API](#)
 - [SOAP API](#)
 - [Open API Interface](#)
 - [Common Workflow Language](#)
- [Reference](#)

Introduction

MUSCLE stands for **M**ultiple **S**equence **C**omparison by **L**og-**E**xpectation. MUSCLE is claimed to achieve both better average accuracy and better speed than ClustalW2 or **T-Coffee**, depending on the chosen options. MUSCLE enables high-throughput applications to achieve average accuracy comparable to the most accurate tools previously available, which is expected to be increasingly important in view of the continuing rapid growth in sequence data. Multiple alignments of protein sequences are important in many applications, including phylogenetic tree estimation, secondary structure prediction and critical residue identification.

Official Website

- [MUSCLE website](#)

Download Software

- [HTTP Download](#)



Important Note

- This is **NOT a pairwise alignment tool**. To align two sequences please select a service from the [pairwise alignment tools](#) section.
- MSA tool algorithms are **NOT intended to produce genome synteny maps**.
- There is currently a limit of **500 sequences** or a maximum file size of **1MB of data**.
- We kindly ask all users of EMBL-EBI Web Services to submit tool jobs in batches of **no more than 30 at a time** and not to submit more until the results and processing is complete. Please ensure that a valid email address is provided. Excessive usage of a particular resource will be dealt with in accordance with [EMBL-EBI's Terms of Use](#). Please [contact us](#) if you need further information.

How to Access MUSCLE

You can access the tool via

[[Web Form](#)] [[REST API](#)] [[SOAP API](#)] [[Open API Interface](#)] [[Common Workflow Language](#)]

Web Form

Web interface for Muscle is available at : <https://www.ebi.ac.uk/Tools/msa/muscle/>

How to use this tool

Running a tool from the web form is a simple multiple steps process, starting at the top of the page and following the steps to the bottom.

Each tool has at least 2 steps, but most of them have more:

- The first steps are usually where the user sets the tool input (e.g. sequences, databases...)
- In the following steps, the user has the possibility to change the default tool parameters
- And finally, the last step is always the tool submission step, where the user can specify a title to be associated with the results and an email address for email notification. Using the submit button will effectively submit the information specified previously in the form to launch the tool on the server

Note that the parameters are validated prior to launching the tool on the server and in the event of a missing or wrong combination of parameters, the user will be notified directly in the form.

Step 1 - Sequence

Sequence Input Window

Three or more sequences to be aligned can be entered directly into this form. Sequences can be in GCG, FASTA, EMBL, GenBank, PIR, NBRF, PHYLIP or UniProtKB/Swiss-Prot format. (See [example input formats](#)). Partially formatted sequences are not accepted. Adding a return to the end of the sequence may help certain applications understand the input. Note that directly using data from word processors may yield unpredictable results as hidden/control characters may be present.

Sequence File Upload

A file containing three or more valid sequences in any format (GCG, FASTA, EMBL, GenBank, PIR, NBRF, PHYLIP or UniProtKB/Swiss-Prot) can be uploaded and used as input for the multiple sequence alignment. (See [example input formats](#)). Word processor files may yield unpredictable results as hidden/control characters may be present in the files. It is best to save files with the Unix format option to avoid hidden Windows characters.

STEP 2 - Set your Parameters

Output Format

Format for generated multiple sequence alignment. (See [example output formats](#)).

Output Format	Description	Abbreviation
Pearson/FASTA	Pearson or FASTA sequence format	fasta
ClustalW	ClustalW alignment format without base/residue numbering	clw
ClustalW (strict)	Strict ClustalW alignment format without base/residue numbering	clwstrict
HTML	HTML format colored alignment	html
GCG MSF	GCG Multiple Sequence File (MSF) alignment format	msf
Phylip interleaved	PHYLIP interleaved alignment format	phyi
Phylip sequential	PHYLIP sequential alignment format	phys

Default value is: ClustalW [clw]

Output Tree

The guide tree to output

Matrix (Protein Only)	Abbreviation
none	none
From first iteration	tree1
From second iteration	tree2

Default value is: none

Step 3 - Submission

Job title

It's possible to identify the tool result by giving it a name. This name will be associated to the results and might appear in some of the graphical representations of the results.

Email Notification

Running a tool is usually an interactive process, the results are delivered directly to the browser when they become available. Depending on the tool and its input parameters, this may take quite a long time. It's possible to be notified by email when the job is finished by simply ticking the box "Be notified by email". An email with a link to the results will be sent to the email address specified in the corresponding text box. Email notifications require valid email addresses.

Email Address

If email notification is requested, then a valid Internet email address in the form joe@example.org must be provided. This is not required when running the tool interactively (The results will be delivered to the browser window when they are ready).

[Top](#)

Web Services

Web Services are available using REST and SOAP protocols that enable programmatic access and allow their integration into other applications and analytical workflows and pipelines.

For an introduction on how to run these clients and use them in workflows please see the [webinar series](#).

REST API

The [Representational State Transfer \(REST\)](#) sample clients are provided for a number of programming languages. For details of how to use these clients, [download](#) the client and run the program without any arguments.

Language	Download	Requirements
Perl	muscle.pl	LWP and XML::Simple
Python	muscle.py	xmltramp2

For details see [Environment setup for REST Web Services](#) and [Examples for Perl REST Web Services Clients](#) pages.

WADL

The [WADL](#) for the MUSCLE (REST) service: <http://www.ebi.ac.uk/Tools/services/rest/muscle?wadl>

Tool Parameters

Parameters	Command line parameter
OUTPUT TREE	--tree
OUTPUT ORDER	--order

[Top](#)

SOAP API

The [Simple Object Access Protocol \(SOAP\)](#) sample clients are provided for a number of programming languages. For details of how to use these clients, [download](#) the client and run the program without any arguments.

WSDL

The [WSDL](#) for the MUSCLE (SOAP) service: <http://www.ebi.ac.uk/Tools/services/soap/muscle?wsdl>

The [Kissinger Research Group, University of Georgia](#) provide a SAWSDL for the MUSCLE (SOAP) web service, which is annotated using [OBLws](#): <http://mango.ctegd.uga.edu/jkissingLab/SWS/Wsannotation/resources/muscle.sawSDL>

Methods	Description	Arguments	Returns
getParameters()	Get a list of the parameter names	<i>none</i>	a list of strings giving the names of the parameters.
getParameterDetails(parameterId)	Get details of a specific parameter	parameterId: identifier/name of the parameter to fetch details of	a wsParameterDetails describing the parameter and its values.

run(email, title, params)	Submit a job to the service	<p>email: (required) user e-mail address. See Why do you need my e-mail address?</p> <p>title: job title. Default: ""</p> <p>params: (required) parameters for the tool. These are described by the InputParameters data structure.</p>	a string containing the job identifier (jobId).
getStatus(jobId)	Get the status of a submitted job.	jobId: (required) job identifier.	<p>a string containing the status.</p> <p>The values for the status are:</p> <p>RUNNING: the job is currently being processed.</p> <p>FINISHED: job has finished, and the results can then be retrieved.</p> <p>ERROR: an error occurred attempting to get the job status.</p> <p>FAILURE: the job failed.</p> <p>NOT_FOUND: the job cannot be found.</p> <p>Further details can be found in Synchronous and Asynchronous Access: Job Dispatcher.</p>
getResultTypes(jobId)	Get the available result types for a finished job	jobId: (required) job identifier.	a list of wsResultType data structures describing the available result types.
getResult(jobId, type, parameters)	Get the result of a job of the specified type.	<p>jobId: (required) job identifier.</p> <p>type: (required) string specifying the result type to fetch. See getResultTypes(jobId) for details of the available types.</p> <p>parameters: optional list of wsRawOutputParameter used to provide additional parameters for derived result types.</p>	the result data for the specified type, base64 encoded. Depending on the SOAP library and programming language used the result may be returned in decoded form. For some result types (e.g. images) this will be binary data rather than a text string.

InputParameters

The input parameters for the job:

Attribute	Type	Description
format	string	alignment format
tree	string	tree type
order	string	order of sequences in the alignment
sequence	string	sequences to align

More detailed information about each parameter, including valid values can be obtained using the getParameterDetails(parameterId) operation.

wsParameterDetails

Descriptive information about a tool parameter. Returned by getParameterDetails(parameterId).

Attribute	Type	Description
name	string	Name of the parameter.
description	string	Description of the parameter, suitable for use in option help interfaces.
type	string	Data type of the parameter.
values	list of wsParameterValue	List of valid values for the option.

wsParameterValue

Description of a tool parameter value. Used in wsParameterDetails.

Attribute	Type	Description
label	string	Display name of the value, for use in interfaces.
value	string	String representation of the value to be passed to the tool parameter.
defaultValue	boolean	Flag indicating if this value is the default.
properties	list of wsProperty	List of key/value pairs providing further information.

wsProperty

Properties of a tool parameter value. Used in wsParameterValue.

Attribute	Type	Description
key	string	
value	string	

wsRawOutputParameter

Additional parameters passed when requesting a result. See getResult(jobId, type, parameters).

Attribute	Type	Description
name	string	
value	list of string	

wsResultType

Description of a result type. Returned by getResultTypes(jobId).

Attribute	Type	Description
identifier	string	Identifier for the result type. Passed as type to getResult(jobId, type, parameters).
label	string	Display name for use in user interfaces.
description	string	Description of the result type, for use in help interfaces.
mediaType	string	MIME type of the returned data.
fileSuffix	string	Suggested suffix for file name, if writing data to disk

[Top](#)

Open API Interface

<https://www.ebi.ac.uk/Tools/common/tools/help/index.html?tool=muscle>

Endpoints	Description	Arguments	Returns
https://www.ebi.ac.uk/Tools/services/rest/muscle/parameters	Get a list of the parameter names.	<i>none</i>	A list of strings giving the names of the parameters.

<p>https://www.ebi.ac.uk/Tools/services/rest/muscle/parameterdetails/format</p> <p>https://www.ebi.ac.uk/Tools/services/rest/muscle/parameterdetails/tree</p> <p>https://www.ebi.ac.uk/Tools/services/rest/muscle/parameterdetails/sequence</p>	Get details of a specific parameter.	parameterdetail: identifier /name of the parameter to fetch details of.	A wsParameterDetails describing the parameter and its values.
<p>https://www.ebi.ac.uk/Tools/services/rest/muscle/run</p>	Submit a job to the service.	<p>email: (required) user e-mail address. See Why do you need my e-mail address?</p> <p>title: job title. Default: "".</p> <p>params: (required) parameters for the tool. These are described by the InputParameters data structure.</p>	A string containing the job identifier (jobId).
<p><a href="https://www.ebi.ac.uk/Tools/services/rest/muscle/status/<jobid>">https://www.ebi.ac.uk/Tools/services/rest/muscle/status/<jobid></p>	Get the status of a submitted job.	jobId: (required) job identifier.	<p>A string containing the status.</p> <p>The values for the status are:</p> <ul style="list-style-type: none"> • RUNNING: the job is currently being processed. • FINISHED: job has finished, and the results can then be retrieved. • ERROR: an error occurred attempting to get the job status. • FAILURE: the job failed. • NOT_FOUND: the job cannot be found. <p>Further details can be found in Synchronous and Asynchronous Access: Job Dispatcher.</p>
<p><a href="https://www.ebi.ac.uk/Tools/services/rest/muscle/resulttypes/<jobid>">https://www.ebi.ac.uk/Tools/services/rest/muscle/resulttypes/<jobid></p>	Get the available result types for a finished job.	jobId: (required) job identifier.	Returns: a list of wsResultType data structures describing the available result types.
<p><a href="https://www.ebi.ac.uk/Tools/services/rest/muscle/result/<jobid>/out">https://www.ebi.ac.uk/Tools/services/rest/muscle/result/<jobid>/out</p> <p><a href="https://www.ebi.ac.uk/Tools/services/rest/muscle/result/<jobid>/error">https://www.ebi.ac.uk/Tools/services/rest/muscle/result/<jobid>/error</p> <p><a href="https://www.ebi.ac.uk/Tools/services/rest/muscle/result/<jobid>/sequence">https://www.ebi.ac.uk/Tools/services/rest/muscle/result/<jobid>/sequence</p> <p><a href="https://www.ebi.ac.uk/Tools/services/rest/muscle/result/<jobid>/phylo tree">https://www.ebi.ac.uk/Tools/services/rest/muscle/result/<jobid>/phylo tree</p> <p><a href="https://www.ebi.ac.uk/Tools/services/rest/muscle/result/<jobid>/pim">https://www.ebi.ac.uk/Tools/services/rest/muscle/result/<jobid>/pim</p>	Get the result of a job of the specified type.	<p>jobId: (required) job identifier.</p> <p>type: (required) string specifying the result type to fetch. See getResultTypes(jobId) for details of the available types.</p> <p>parameters: optional list of wsRawOutputParameter used to provide additional parameters for derived result types.</p>	Returns: the result data for the specified type, base64 encoded. Depending on the SOAP library and programming language used the result may be returned in decoded form. For some result types (e.g. images) this will be binary data rather than a text string.

[Top](#)

Common Workflow Language

CWL (Common Workflow Language) implementation for consuming [EMBL-EBI Bioinformatics Web Services tools'](#) clients are available at <https://github.com/ebi-wp/webservice-cwl>

For details, see [CWL Workflows](#) page.

[Top](#)

Reference

[MUSCLE](#)

[EBI Web Services](#)

[Top](#)