

**User Guide**

Table of Contents

[License 4](#_Toc414878523)

[Requirements 4](#_Toc414878524)

[For running ResCon software: 4](#_Toc414878525)

[For running ResCon script: 4](#_Toc414878526)

[Installation Instructions 5](#_Toc414878527)

[For Windows: 5](#_Toc414878528)

[For Mac: 5](#_Toc414878529)

[For Ubuntu: 5](#_Toc414878530)

[What the hell is ResCon? 6](#_Toc414878531)

[Mismatch analyzer 8](#_Toc414878532)

[Demo 1: Mismatch analysis (*Clustal* Omega *required*): 12](#_Toc414878533)

[Demo 2: Mismatch analysis (*Clustal Omega not required*): 13](#_Toc414878534)

[Overview of output files: 14](#_Toc414878535)

[1. Formatted\_Alignment.html 14](#_Toc414878536)

[2. Mismatches\_Tabulated.csv 16](#_Toc414878537)

[3. Mismatches\_Detailed.txt: 17](#_Toc414878538)

[4. log.txt: 17](#_Toc414878539)

[ResCon – More tools 18](#_Toc414878540)

[1. Subtree Sequences Extractor: 18](#_Toc414878541)

[Demo: 19](#_Toc414878542)

[Tips and Tricks / Troubleshooting: 21](#_Toc414878543)

[2. GenBank/GenPept to fasta converter: 23](#_Toc414878544)

[Demo: 25](#_Toc414878545)

[3. Filter fasta by Sequences’ ID 26](#_Toc414878546)

[Demo - 1 (Using *Complete* sequence IDs to filter fasta sequences): 27](#_Toc414878547)

[Demo – 2 (Using *Partial* sequence IDs to filter fasta sequences): 28](#_Toc414878548)

[4. Filter fasta by Blast’s E-value: 29](#_Toc414878549)

[Demo 30](#_Toc414878550)

[Tips and Tricks / Troubleshooting: 31](#_Toc414878551)

[5. Filter fasta by Sequences’ Description: 32](#_Toc414878552)

[Demo 1 – Extracting complete Identifiers: 38](#_Toc414878553)

[Demo 2 – Extracting Partial Descriptions: 39](#_Toc414878554)

[6. Fasta Description/ID Extractor: 36](#_Toc414878555)

[Demo 1 - Using Complete Descriptions from a text file: 34](#_Toc414878556)

[Demo 2 - Using Partial Descriptions as a list: 35](#_Toc414878557)

[Appendix A: Understanding fasta format 40](#_Toc414878558)

[Appendix B: Log file 41](#_Toc414878559)

[Appendix C: Clustal Omega Parameters 43](#_Toc414878560)

[Appendix D: Installing Clustal Omega 47](#_Toc414878561)

[For Windows: 47](#_Toc414878562)

[For UNIX based OS (Mac & Ubuntu): 48](#_Toc414878563)

License

ResCon is licensed under GNU Lesser General Public License.

Authors: Manavalan Gajapathy, Joseph D. Ng

Copyright (C) <2015>

# Requirements

## For running ResCon software:

* Choose ResCon specific to your platform.
* Clustal Omega installation is optional but it is required for complete use of ResCon. See [Appendix D](#_Appendix_D:_Installing) for Clustal Omega installation instructions.
* The following are usually already installed in your computer. But if installation of ResCon results in an error, you may have to download the following:
  + Mac and Linux OS usually would have Tcl/Tk environment pre-installed and most Windows OS would have it already installed; if not, you may download it from <http://www.activestate.com/activetcl/downloads>.
  + *Windows OS only:* Many Windows OS computers may already have ‘*Microsoft Visual C++ 2008 Redistributable Package*’ installed. If it is not installed already, you may have to install it from <http://www.microsoft.com/en-us/download/details.aspx?id=29> for 32 bit Windows OS.  
    ~~or~~ [~~http://www.microsoft.com/en-us/download/details.aspx?id=15336~~](http://www.microsoft.com/en-us/download/details.aspx?id=15336) ~~for 64 bit Windows OS.~~

## For running ResCon script:

* Python 2.7

Download source: <https://www.python.org/download/releases/2.7/>

* BioPython library is required.

Download source: <http://biopython.org/wiki/Download>

* Though installation of Clustal omega is required for complete experience of ResCon, *it is not necessary if clustal alignment will not be run through ResCon*. See [Appendix D](#_Appendix_D:_Installing) for Clustal Omega installation instructions.

# Installation Instructions

## For Windows:

Run the setup file for Windows and install it. You may need administrator password for installation.

Note: You may need the following if installation fails or if ResCon would not open after installation.

1. Many Windows OS computers may already have ‘*Microsoft Visual C++ 2008 Redistributable Package*’ installed. If it is not installed already, you may have to install it from <http://www.microsoft.com/en-us/download/details.aspx?id=29> for 32 bit Windows or <http://www.microsoft.com/en-us/download/details.aspx?id=15336> for 64 Windows.
2. Most Windows OS would have Tcl/Tk already installed; if not, you may download it from <http://www.activestate.com/activetcl/downloads>.

## For Mac:

Unzip the file provided for Mac and move the ResCon.app file in to 'Applications' folder or save it at a location of your choice. If saved at ‘Applications’ folder, when you open ResCon for the first time, right click on ResCon.app and click open

*Important note for Mac users*:

1. When you start ResCon, it will start minimized. Go to the dock, double click ResCon icon and it should work just fine now.
2. If ResCon says that ‘Clustal omega is not installed on your Mac’ when it is actually properly installed, then you have to open ResCon app from terminal. Open ‘Terminal’. Type open and then drag and drop ResCon.app file into the terminal and then press Enter. It should work just fine now.
3. When you open ResCon for the first time, your Mac may say ‘ResCon can’t be opened because it is from an unidentified developer”. In that case, follow these steps:
   1. Right-click (or control-click) on ResCon.app and click ‘open’.
   2. Click ‘open’ on the pop-up window.

You would have to do this only the first-ever time you run ResCon.

## For Ubuntu:

Unzip installation file provided for Ubuntu OS and then move the folder to location of your choice. Double click the executable file to run ResCon.

# What the hell is ResCon?

ResCon houses tools as many as the number of continents. It was built to analyze residue conservation in protein sequences at positions of your choice. Following provides a brief introduction for each of those tools.

1. [**Mismatch analyzer**](#_Mismatch_analyzer)**:**

This tool compares your protein sequence with other bunch of sequences and assists in analyzing conservation of residues at the positions you are interested in, based on multiple sequence alignment. Further, this tool helps you introducing taxonomy in to this analysis, provided you have taxonomy information in the sequence identifier.

1. [**Subtree Sequences Extractor**](#_1._Subtree_Sequences)**:**

This tool can read phylogenetic tree and then extract only the sequences corresponding to a subtree of your interest, based on that subtree’s branch length. This tool extracts sequence IDs corresponding to subtree of your interest and then filters sequences of that subtree from a fasta file containing sequence data.

1. [**GenBank/GenPept to fasta converter**](#_2._GenBank/GenPept_to)**:**

This tool converts GenBank or GenPept files to fasta format while conserving taxonomy information in fasta sequences’ header.

1. [**Filter fasta by Sequences’ ID**](#_3._Filter_fasta)**:**

This tool extracts sequences from a fasta file based on the sequence IDs provided by you. You may provide either complete or partial sequence ID.

1. [**Filter fasta by Blast’s E-value**](#_4._Filter_fasta)**:**

This tool reads a file of sequences in fasta format and a XML file containing BLAST results and then extract the sequences below or above E-value threshold of your choice.

1. [**Filter fasta by Sequences’ Description**](#_5.__Filter)**:**

This tool extracts sequences from a fasta file based on the sequence descriptions provided by you. You may provide either complete or partial sequence description.

1. [**Fasta Description/ID Extractor**](#_6.__Fasta)**:**

This tool enables you to extract descriptions or identifiers, complete or partial, from sequence headers of fasta sequences.

# Mismatch analyzer

ResCon’s ‘Mismatch analyzer’ is a tool that compares your protein sequence with other bunch of sequences and assists in analyzing conservation of residues at the positions you are interested in, based on multiple sequence alignment. Further, this tool helps you introducing taxonomy in to this analysis, provided you have taxonomy information in the sequence identifier.

**Sequences file:**

This file should have all your target sequences in fasta format. Each sequence record must have a title. If title has pipe or vertical bar symbol ‘|’, ResCon will treat it as delimiter and divide it into individual elements when written in to output files.

Sequences file may or may not have reference sequence included in it. If reference sequence is already present in ‘Sequences file’, its sequence and title must be identical in both files. Number of sequences allowed is not restricted but larger number of sequences may result in longer processing time for clustal alignment.

**Reference file:**

This file should have your reference sequence in fasta format. Only one sequence is allowed in this file.

**Output folder:**

This indicates where output files will be saved. By default, ResCon will select to save output files in to a folder called ‘Output’ where ‘Sequences file’ is located. You may choose to change this to different folder, though.

**Residue positions:**

This refers to the residue positions, based on reference sequence, of your interest for which you would like to perform mismatch/match analysis. If you are interested in more than one residue position, enter them as comma separated numbers.

**Clustal alignment required - Checkbox:**

* ***If checked:***

This indicates that you need ResCon to perform multiple sequence alignment using Clustal Omega. Note that, in this case, you need to have Clustal omega installed in your computer. See [Appendix D](#_Appendix_D:_Installing) for Clustal Omega installation instructions.

**Clustal-O command:**

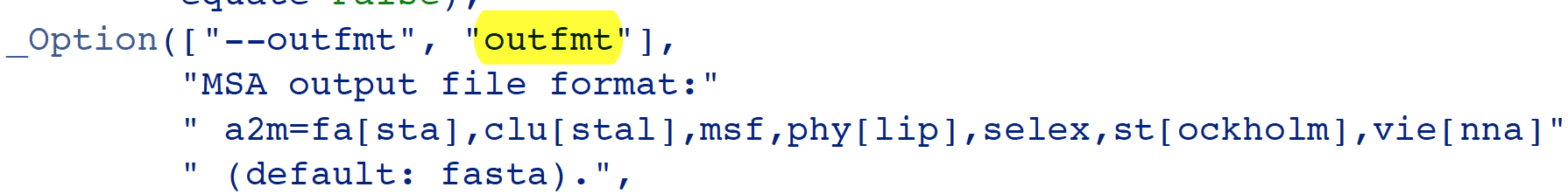
This is where you enter the parameters required to execute clustal omega. Clustal Omega is a command-based program and it allows changing parameters to build multiple sequence alignment. By default, ResCon uses following command:

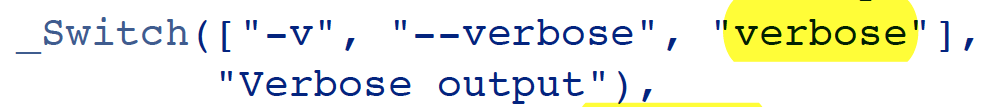
{'outfmt': 'clu', 'auto':'True', 'force': 'True', 'outfile': 'default.aln'}

You may change this to fit your requirements. If you have to edit this default command, follow these instructions:

1. Command must open and close with curly braces.
2. Each set of parameter and value must be separated by comma.
3. Parameters and their values, each, must be inside single or double quotes and separated by colon (:) symbol.

Parameters available for Clustal omega are shown in [appendix C](#_Appendix_C:_Clustal) and the instructions below explain its structure.





Yellow highlighted text signifies the parameters that you may use for command line in ResCon. If a parameter belongs to ‘Option’ category (as marked with red arrow in above figure), then you should provide value for that parameter inside single or double quote (for example, 'outfmt': 'clu'). However if that value is a number, enter only the number without single or double quotes (for example, ‘iterations’: 3). If they belong to ‘Switch’ category (blue arrow marked) instead, their value is provided as either ‘True’ or ‘False’ (for example, 'auto': 'True').

Tips and tricks / Troubleshooting:

1. Parameter ‘outfile’ *must always be included* as part of command line. By default, value for ‘outfile’ is set to ‘default’, meaning that output alignment file will be named by ResCon. If you like to provide your own filename instead, replace ‘default’ with filename of your choice along with its file path. For example: ‘outfile’: ‘C:/phys/chem/new\_name.clustal’.
2. If you need Clustal omega to build phylogenetic tree as output, include

'guidetree\_out': 'default.newick'

as part of the command and this will instruct ResCon to name and save the phylogenetic tree in the output folder chosen. If you like to provide your own filename instead, replace ‘default. newick’ with filename of your choice along with its file path. For example: ‘guidetree\_out’: ‘C:/phys/chem/new\_name. newick'.

1. ‘Log file’ stored in output folder shows the command directed to your computer terminal. If you run into error with clustal omega command line, see log file for clues on what is going wrong. See [Appendix B](#_Appendix_B:_Log) to read about log file.

* ***If unchecked:***

This indicates that you need ResCon to use a pre-aligned file for mismatch analysis. Note that, in this case, your multiple sequence alignment file must have reference sequence already as a part of it and ID of that sequence should be same as that of sequence in ‘Reference file’.

**~~Clustal Aligned file:~~**

~~This file should have multiple sequence alignment data in clustal ‘.aln’ format (with or without numbers).~~ If your alignment is in any other format, alignment format converters are available online:

1. ALTER:

<http://sing.ei.uvigo.es/ALTER/>

1. Format Converter v2.3.5:

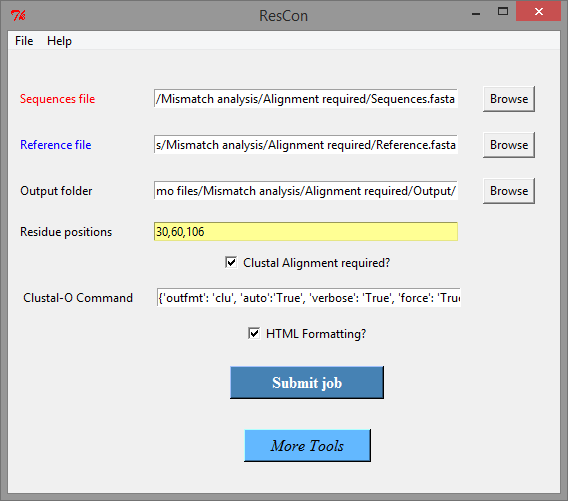
<http://www.hiv.lanl.gov/content/sequence/FORMAT_CONVERSION/form.html>

**HTML Color-Coding required? - Checkbox:**

If checked, ResCon will color-code multiple sequence alignment at the residue positions requested for mismatch analysis and this color-coded html file will be saved in the folder selected at ‘Output folder’ box. ~~Note that, your multiple sequence alignment must in clustal ‘.aln’ format (with or without numbers) for this to work.~~

## Demo 1: Mismatch analysis (*Clustal* Omega *required*):

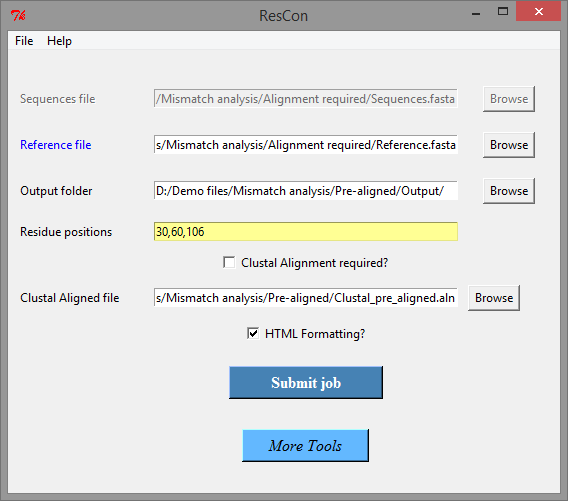
Use the demo files from folder ‘Demo files\Mismatch analyzer\Alignment reuired\’ to get a feel of how ResCon works. You will need to have Clustal Omega installed in your computer. Here you will analyze whether residues at positions 30, 60, and 106 in reference sequence are conserved or not in target sequences.



Click ‘Submit job’ after filling in the required fields as shown in above figure. When job is done, files will be written in to the output folder selected. See ‘[Overview of output files](#_Overview_of_output)’ to understand how to use output files.

## Demo 2: Mismatch analysis (*Clustal Omega not required*):

Use the demo files from folder ‘Demo files\Mismatch analyzer\Pre-aligned\’. You do not need to have Clustal omega installed in your computer for this to work. Note that you are using pre-aligned multiple sequence alignment in clustal format as input and this alignment already has reference sequence as part of it. Here you will analyze whether residues at positions 30, 60, and 106 in reference sequence are conserved or not in target sequences used in multiple sequence alignment.

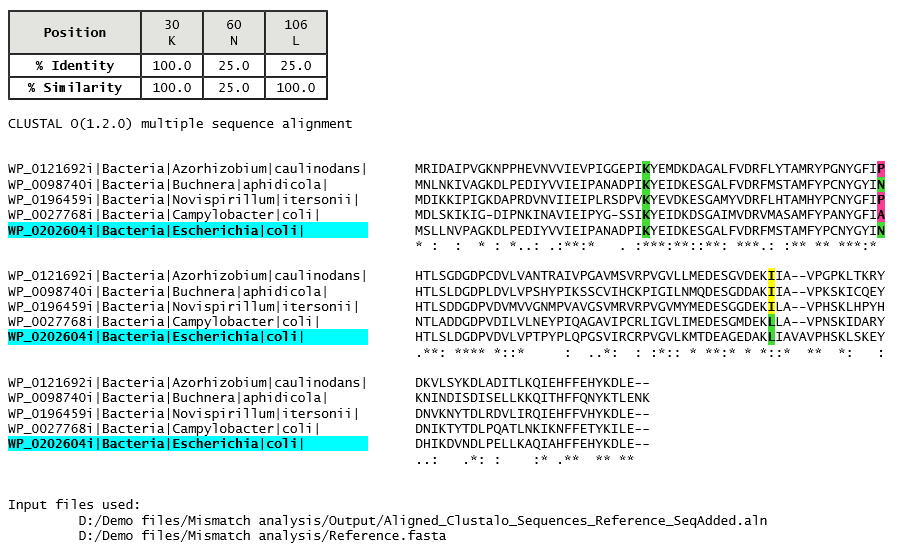


Click ‘Submit job’ after filling in the required fields as shown in above figure. When job is done, files will be written in to the output folder selected. See ‘[Overview of output files](#_Overview_of_output)’ to understand how to use output files.

## Overview of output files:

### Formatted\_Alignment.html

* This html file contains multiple sequence alignment, color-coded at the sites requested for mismatch analysis. Use any internet browser to view this file.



* This html file has three parts (shown in yellow circles):

1 - Table shows requested sites and their % identity and % similarity.

2 - Cyan highlighted ID refers to reference sequence

3 - Shows input files used to get this output

* Residue Sites under mismatch analysis are highlighted with three different colors as follows:

Green : Matching residue

Yellow : *Mismatch* but a *similar* residue

Pink : *Mismatch* and a *non-similar* residue

Essentially ‘green’ highlighted are matching residues whereas both ‘yellow’ and ‘pink’ highlighted are mismatching residues. Similarity of residues is defined on the basis of their physiochemical properties (same as Clustal Omega’s coloring scheme) as shown below.

|  |  |
| --- | --- |
| **Residue** | **Property** |
| AVFPMILW | Small (small + hydrophobic (including aromatic -Y)) |
| DE | Acidic |
| RK | Basic - H |
| STYHCNGQ | Hydroxyl + sulfhydryl + amine + G |

### 2. Mismatches\_Tabulated.csv

* This is a csv (comma separated values) file that can be opened using any spreadsheet software. If your spreadsheet software asks for delimiter information, use comma as delimiter.
* This has 3 sections:

1. Shows details of sequence records that have at least one mismatch at the residue positions requested.
2. Shows unique residues present at each site along with their count and fraction at the requested sites in the alignment. Also, % identity and % similarity are shown*. Note that Reference sequence is not included in this calculation.*
3. Shows details of sequence records that does not have mismatch at any of the residue positions requested.



Details shown for sequence record:

* In section 1 and 3, result for each sequence record is shown. Pipe symbol “|” in sequence ID is a delimiting character and is used to divide the sequence id into parts and are then labeled as Title\_1, Title\_2, etc.. For example:

*For ID:* “WP\_0121692i|Bacteria|Azorhizobium|caulinodans|”, it is shown as:



* In cases where all sequences do not have same number of pipe symbols, missing sections will be mentioned as “-na-” in output csv and txt files.
* *Symbols* used here:

Equal sign = Matching residue

Star sign \* Gap in alignment (star symbol used instead of hyphen symbol to improve visibility in spreadsheet software)

### 3. **Mismatches\_Detailed.txt:**

Data presented in this file is same as that shown in csv file except that, here, it is presented in text format and details for each requested sites are provided separately for each site.

### 4. log.txt:

See [Appendix B](#_Appendix_B:_Log_1).

# ResCon – More tools

# 1. Subtree Sequences Extractor:

This tool can read phylogenetic tree (in newick format) and then extract only the sequences corresponding to a subtree of your interest, based on that subtree’s branch length. Newick phylogenetic tree contains only the sequence IDs but not sequence data itself. This tool ‘Subtree Sequences Extractor’ extracts sequence IDs corresponding to subtree of your interest and then filters sequences of that subtree from a fasta file containing sequence data.

**Newick file:**

This must be a phylogenetic tree in newick format. Software and online converters are available elsewhere to aid in conversion of other formats to newick format.

**FASTA file:**

This must be in fasta file format (preferably the same fasta file used towards making the phylogenetic tree used in ‘Newick file’).

**Output folder:**

This indicates where output files will be saved. By default, ResCon will select to save output files in to a folder called ‘Output’ where ‘Newick file’ is located. You may choose to change this to different folder, though.

**Clade’s Branch Length:**

This may contain one or more branch length values. *If more than one, they should be comma separated*. If your branch length is 0.123456, you may enter this in one of following formats:

1) 0.123456 2) 123456E-6 3) 123456e-6

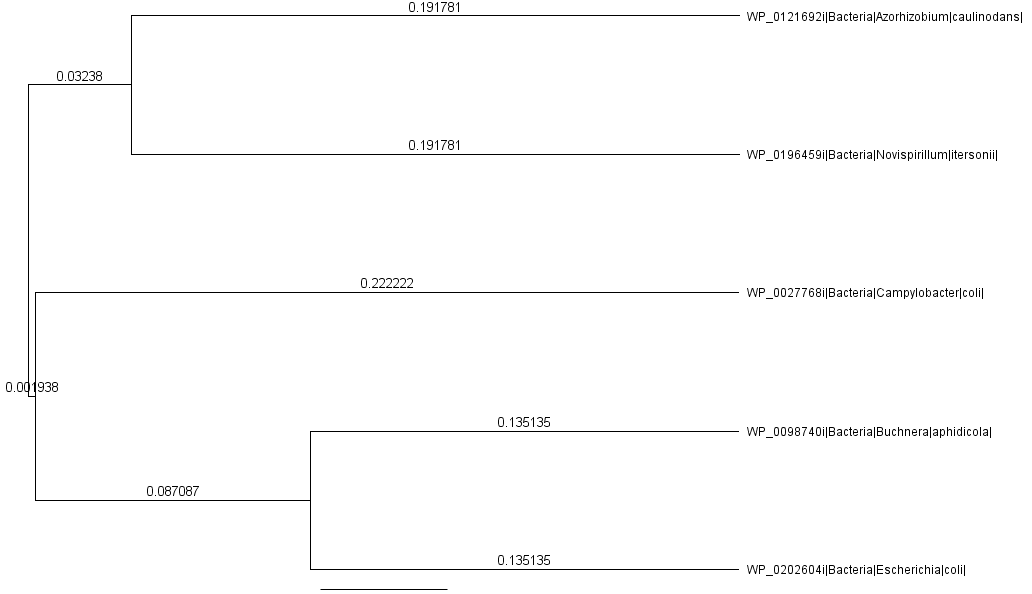
*Obtaining correct branch length is little tricky;* so read the following before you proceed further with this tool:

This tool requires using complete branch length number, not a rounded number. Unfortunately, many phylogenetic tree viewers do not show complete branch length but instead show them rounded to three/four decimal points. For example, in the demo files used, one of the branch lengths is ‘0.0323797’ but a tree viewer may show this as ‘0.03238’ or ‘0.0324’. Subtree Sequences Extractor will process properly only if ‘0.0323797’ is used. In such instances, following work-around helps.

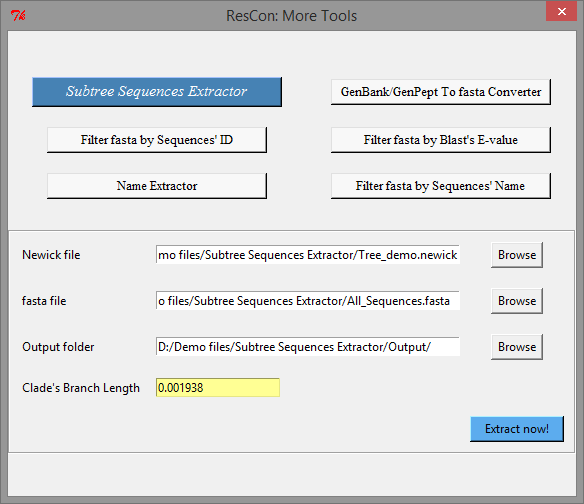
*Work-around:* Open newick tree file in a internet browser or in a text editor and search for ‘0.0323’ using search feature (Ctrl + F). Go through all the hits and now you will be able to decipher which number you have to use.

## Demo:

Use the demo files from folder ‘Demo files\1. Subtree Sequences Extractor\’ to get a feel of how tool ‘Subtree Sequences Extractor’ works.



Above figure shows the phylogenetic tree from demo file ‘*Tree\_demo.newick*’. Arrow marks show branch lengths of subtrees. In the demo here, we will extract sequences corresponding to subtree with branch length ‘0.001938’.



Click ‘Extract now’ after filling in the required fields as shown in above figure. When the job is done, files will be written in to the selected output folder.

### Tips and Tricks / Troubleshooting:

1. *Error: Verify branch length entered. No such branch length is found in   
   newick tree file:*

See the solution provided as ‘*Work-around*’ under ‘Clade’s Branch Length’.

1. *Error: More than one instances of such branch length found:*

In some extreme cases, you may have same branch length for more than one subtree. You may choose to use following work-around to edit branch lengths, assuming that proper care is taken not to use that edited file elsewhere where branch length data is necessary for further studies. We recommend to work on a duplicate file to avoid any potential issues.

Solution: Open the newick file in a text editor, search for that branch length using inbuilt search feature and edit one of the branch lengths to a different number. Save it as same file or a new file and then use ‘ResCon – Subtree Sequences Extractor’ to extract the sequences and it should work now.

1. *Number of sequences extracted is less than the expected number of sequences:*

One possible culprit here in such case could be that your newick file has some branch lengths with negative values. Negative branch lengths (even negative zero ie. -0) spell trouble when using ‘ResCon – Subtree Sequences Extractor’. The work-around for this problem involves editing the newick file and hence care must be taken not to use edited file where branch lengths are necessary. We recommend working on a duplicate file to avoid any potential issues.

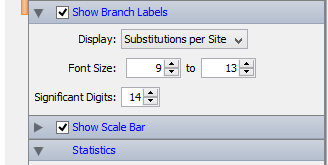
Solution: Open the newick file in a text editor, search for ‘:-’ (i.e. colon followed by dash) using inbuilt find and replace feature, and then use ‘replace all’ feature to replace all ‘:-’ with just a colon ‘:’. Now save this file and try extracting sequences using this edited file in ‘ResCon – Subtree Sequences Extractor’.

1. ‘ResCon – Subtree Sequences Extractor’ cannot extract if the clade has only one sequence ID in it. Use search feature in a text editor and copy it manually.
2. Sequence IDs used to build phylogenetic tree in newick format cannot have following sequences:

* Round braces (parentheses)
* Square braces
* Comma
* Single quote
* Colon
* Semi-colon

Any phylogenetic tree viewer or processer will most likely have trouble reading newick file with above symbols. You may use any text editor’s find and replace feature to replace above symbols with some other character (for example, underscore).

1. Unfortunately not all phylogenetic tree viewers can show branch lengths. We recommend software ‘Geneious’ to visualize phylogenetic tree files as it offers an option to show complete branch length with all digits. This doesn’t work all the time but it is better than most other software we have tried. If you use Geneious, change ‘significant digits’ under ‘show branch labels’ to a number greater than 10 as shown in the figure below and this will work for most of the time. This feature works even after trial period of Geneious is expired.



Following online phylogenetic tree viewers show branch lengths but number of digits they show are limited:

1. [http://evolgenius.info/evolview/#mytrees/qq/q](http://evolgenius.info/evolview/%23mytrees/qq/q)
2. <http://phylogeny.lirmm.fr/phylo_cgi/index.cgi>

# 2. GenBank/GenPept to fasta converter:

This tool converts GenBank or GenPept files to fasta format while conserving taxonomy information in fasta sequences’ header. Fasta files available from ncbi database do not offer any taxonomy data except genus and species information. GenPept or GenBank files have such taxonomy information and this tool aids in creating fasta data while conserving taxonomy information.

**GenBank/GenPept file:**

This must be in either GenBank or GenPept format. Number of sequence records data they may have are not limited.

**Output folder:**

This indicates where output files will be saved. By default, ResCon will select to save output files in to a folder called ‘Output’ where ‘GenBank/GenPept file’ is located. You may choose to change this to different folder, though.

**Fasta ID length:**

This indicates the maximum length of identifier in fasta header (see [Appendix A](#_Appendix_A:_Understanding_1)). Certain applications may restrict length of fasta identifier to certain length and this feature is helpful in those cases. Default value is 127 as Clustal omega can handle up to 127 characters in fasta header. You may change this to suit your needs.

**Header options:**

You may check the fields you may like to be included in fasta identifier (see   
[Appendix A](#_Appendix_A:_Understanding_1)). We recommend selecting at least one option that is unique to a sequence (for example, Locus name and GI are unique values). Selected fields appear in the order shown (from left to right).

**Retrieval options:**

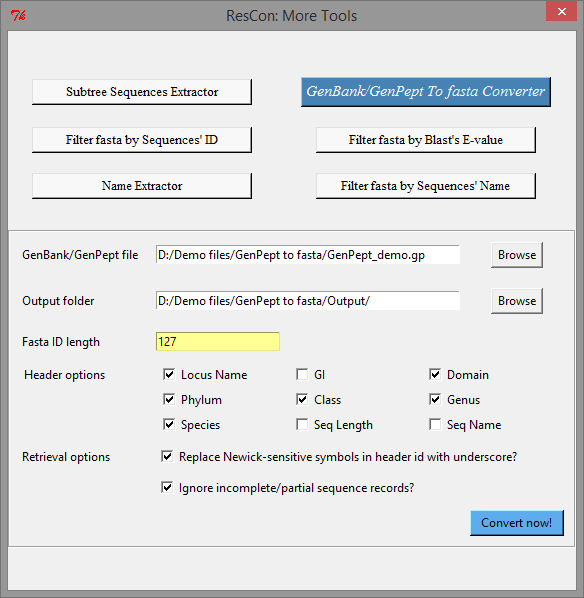
1. *Replace Newick-sensitive symbols in header id with underscore symbol?*

Check this box if the converted output fasta sequences will be used to build a phylogenetic tree in newick format. Newick phylogenetic tree format will result in error if sequence ID contains symbols such as colon, semicolon, comma, single quote, parenthesis and square brackets as part of sequence’s ID.

1. *Ignore incomplete/partial sequence records?*

Check this option if you prefer not to extract sequences annotated as partial/incomplete in to output fasta file. Note that not all incomplete or partial sequences are annotated as such.

## Demo:



Use the demo files from folder ‘Demo files\2. GenPept to fasta\’ to get a feel of how tool ‘ResCon - GenBank/GenPept to fasta converter’ works. Click ‘Convert now’ after filling in the required fields as shown in above figure. When the job is done, files will be written in to the selected output folder.

# 3. Filter fasta by Sequences’ ID

This tool reads a file of fasta sequences and then extracts the sequences based on sequence IDs provided by you. You may provide either complete sequence ID or part of the sequence ID. See [Appendix A](#_Appendix_A:_Understanding) to learn more about fasta sequence ID.

**FASTA file:**

This must be a fasta file. Number of sequences in it are not limited. Of course, sequences must have identifiers (IDs) in their header.

**File with IDs:**

This should be a text file containing only one sequence ID per line. If ‘IDs are partial’ checkbox is selected, each line in text file should have a partial ID. Number of IDs allowed in this file is not restricted.

**Output folder:**

This indicates where output files will be saved. By default, ResCon will select to save output files in to a folder called ‘Output’ where ‘FASTA file’ is located. You may choose to change this to different folder, though.

**Checkboxes – ‘Include IDs in list’ / ‘Exclude IDs in list’:**

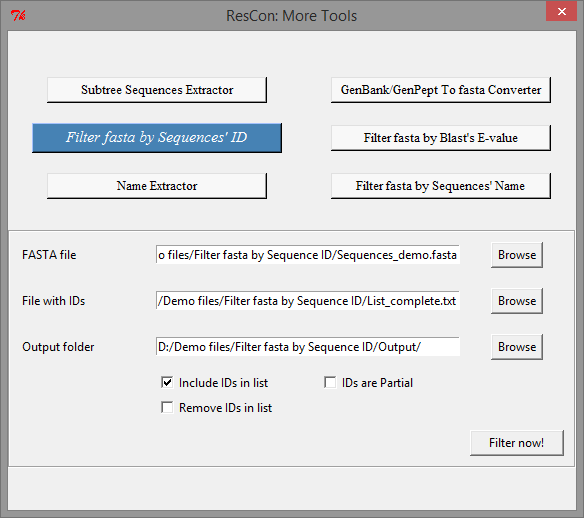
Check one of these boxes depending on if you would like to include or exclude sequences in fasta file based on sequence IDs present in ‘File with IDs’.

**IDs are partial:**

Check this box if ‘File with IDs’ has partial IDs instead of complete ID. If partial IDs are to be used, it is a good idea to choose partial IDs that are unique (i.e. one partial ID corresponds to only one sequence in FASTA file).

## Demo - 1 (Using *Complete* sequence IDs to filter fasta sequences):

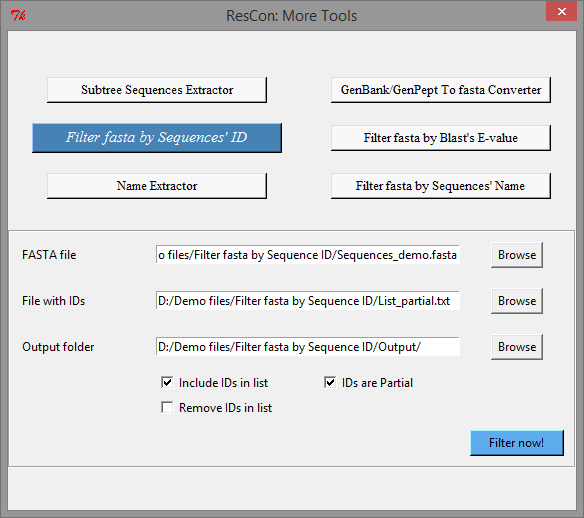
Choose demo files ‘Sequences\_demo.fasta’ and ‘List\_complete.txt’ from folder ‘Demo files\ 3. Filter fasta by Sequence ID\’ to get a feel of how the tool “ResCon - Filter fasta by Sequences’ ID” works. Make sure ‘IDs are Partial’ checkbox is unchecked.



Click ‘Filter now’ after populating the required fields as shown in above figure. When the job is done, files will be written in to the selected output folder.

## Demo – 2 (Using *Partial* sequence IDs to filter fasta sequences):

Now choose demo files ‘*Sequences\_demo.fasta*’ and ‘*List\_partial.txt*’ from folder ‘Demo files\3. Filter fasta by Sequence ID\’ and check ‘IDs are Partial’ checkbox.



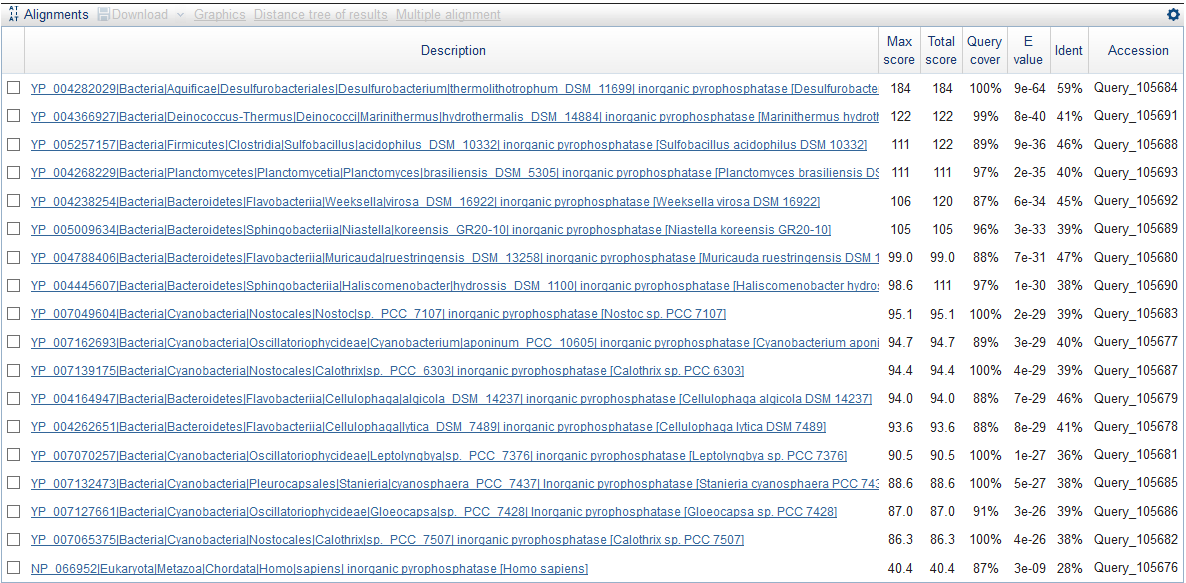
Click ‘Filter now’ after populating the required fields as shown in above figure. When the job is done, files will be written in to the selected output folder.

Note: You may extract sequences using either complete sequence IDs or partial sequence IDs when ‘IDs are partial’ checkbox is selected. But be aware that if you are extracting sequences using complete sequence IDs, keeping ‘IDs are partial’ checkbox unchecked will work faster than having it checked.

# 4. Filter fasta by Blast’s E-value:

This tool reads a file of sequences in fasta format and a XML file containing BLAST results and then extract the sequences based on user provided E-value.

Consider this scenario – Sequences in file *‘Sequences\_demo.fasta’* was BLASTed against *E. coli’s* IPPase and results are shown in the figure below. Now let’s say you want to filter sequences whose E-value is below or above certain threshold value. Bingo! This is where this tool is helpful.



**XML BLAST file:**

This field should be filled with BLAST data in XML file format. If you use NCBI for BLAST, there is an option to download data in XML format.

**FASTA file:**

This field should be filled with a fasta file preferably the same fasta file used to obtain BLAST data.

**Output folder:**

This indicates where output files will be saved. By default, ResCon will select to save output files in to a folder called ‘Output’ where ‘XML BLAST file’ is located. You may choose to change this to different folder, though.

**E-value Threshold:**

The threshold E-value you would like to use can be specified as an integer value or as a decimal value. If it is a decimal number, for example 0.0023, it could be entered as 0.0023 or as 2.3E-4 or as 2.3e-4. See [Tips and Tricks / Troubleshooting](#_Tips_and_Tricks) to avoid potential mistakes.

**Lower than E-threshold:**

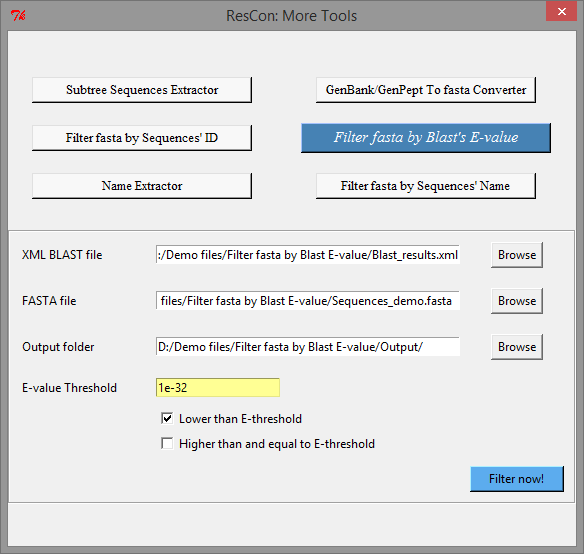
Check this box if you need to extract sequences whose E-values are less than the E-value threshold entered.

**Higher than and equal to E-threshold:**

Check this if you need to extract sequences whose E-values are higher than and equal to the provided E-value threshold. Note that this will result in extracting sequences that are not in blast XML file but are present in provided fasta file that do not have any E-value listed in XML file. Such instances arise when a sequence does not share any significant similarity to the query sequence.

Demo:

Use the demo files from folder ‘Demo files\4. Filter fasta by Blast E-value\’ to get a feel of how the tool ‘ResCon - Filter fasta by Blast’s E-value’ works. Here, we will extract sequences whose E-value is less than 1e-32. Click ‘Convert now’ after filling in the required fields as shown in figure below. When the job is done, files will be written in to the selected output folder.



### Tips and Tricks / Troubleshooting:

1. *Error: Number of sequences filtered is not same as expected number of sequences:*

Possibly this is due to E-value threshold entered. Even though E-values appear as numbers with only one decimal place in NCBI blast table, they actually are not. They are actually numbers rounded off to only one decimal place. For example, be it 2.3, 2.30116 or 2.2844, all these will show up as just 2.3.

This is important to know because if E-value threshold is provided as 2.3 and if certain sequence’s E-value is 2.30116, then this sequence record would not be written into output file when 'lower than E value’ is selected.

# 5. Filter fasta by Sequences’ Description:

This tool reads a file of fasta sequences and then extracts the sequences based on sequence descriptions provided by you. You may provide either complete sequence description or part of the sequence description. See [Appendix A](#_Appendix_A:_Understanding) to learn more about fasta sequence description.

**FASTA file:**

This must be a fasta file. Number of sequences in it are not limited. Of course, sequences must have descriptions in their header.

**Output folder:**

This indicates where output files will be saved. By default, ResCon will select to save output files in to a folder called ‘Output’ where ‘FASTA file’ is located. You may choose to change this to a different folder, though.

**Get Descriptions from text file - Checkbox:**

* ***If unchecked:***

This indicates that you need to provide descriptions (or partial descriptions) as a list, not from a file.

**List of Descriptions:**

This is the field where you will provide descriptions as double comma separated terms. For example: “*DNA polymerase,,Helicase*”. Double commas are used as delimiters, instead of single commas, because single comma may sometimes inherently be a part of the description. Note that you may not add additional space character if they are not part of the sequence description itself, or else you may not extract the intended sequence records.

* ***If checked:***

This indicates that you need to provide descriptions (complete or partial) in a text file.

**File with Descriptions:**

This should be a text file containing one sequence description per line. If ‘Filter using partial Descriptions’ checkbox is selected, each line in text file should have a partial description. Number of descriptions allowed in this file is not restricted. Note that you may not include additional space character if they are not part of the sequence description itself, or else you may not extract the intended sequence records.

**Filter using Partial Descriptions:**

Check this box if you need to match only certain part of the sequence descriptions from fasta sequence headers. Assuming you have certain regularity in the structure of descriptions of all sequences present in your fasta file, you may use regular expression so as to filter fasta sequences on the basis of partial descriptions.

**Regular Expression:**

This field should contain the regular expression that matches to your intended part of sequence descriptions. By default, the tool uses (.+) \[.+ as the regular expression. This means that the part of sequence description from its beginning up to character before space followed by a open square brace will be used for matching. For example, if sequence description is

inorganic pyrophosphatase [Escherichia coli]

then use of default regular expression will result in

inorganic pyrophosphatase

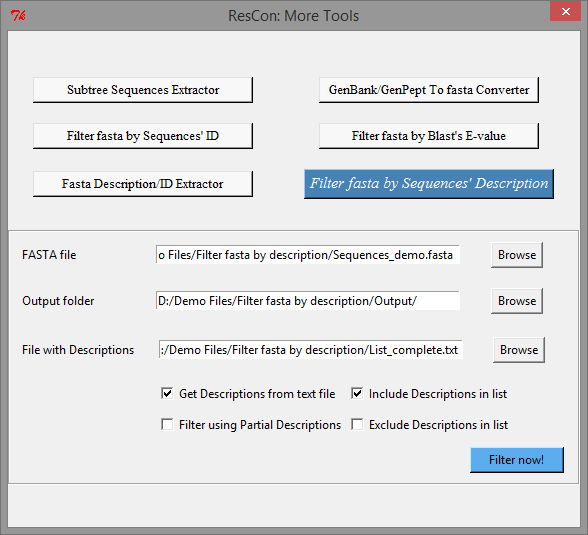
While regular expression seems intimidating at first, trust us it does not take long to learn using it. This web site could be a good start- <http://www.proftpd.org/docs/howto/Regex.html>. You may use any text editor, which allows use of regular expressions or wild card as part of their inbuilt search (ctrl + F) feature, to test and practice them. For example, notepad++ (not notepad), text wrangler, gedit, Microsoft word, etc. have this feature.

**Include / Exclude Descriptions in list:**

Check one of these boxes depending on if you need to include or exclude the sequences that match to the provided complete or partial descriptions.

## Demo 1 - Using Complete Descriptions from a text file:

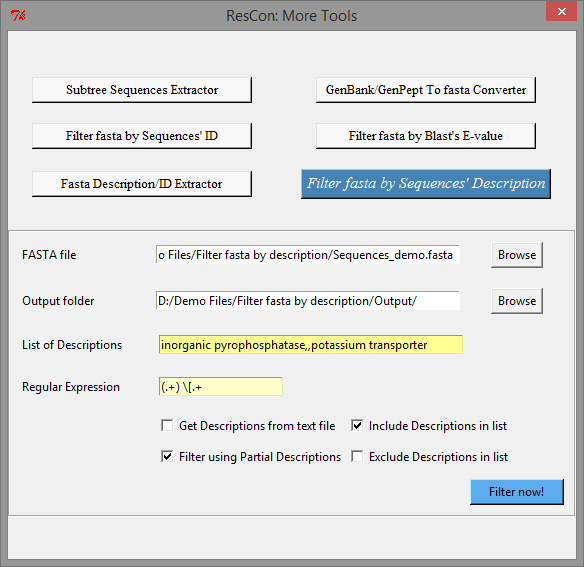
Use the demo files ‘Sequences\_demo.fasta’ and ‘List\_complete.txt’ from folder ‘Demo Files\5. Filter fasta by description\’ to get a feel of how the tool ‘ResCon - Filter fasta by Sequences’ Description’ works. Here, we will extract fasta sequences that have matching descriptions provided in a text file.



Click ‘Filter now’ after filling in the required fields as shown in above figure. When the job is done, files will be written in to the selected output folder. The output file should have 35 sequences written in to it.

## Demo 2 - Using Partial Descriptions as a list:

Use file ‘Sequences\_demo.fasta’ from folder ‘Demo Files\5. Filter fasta by description\’ for this demo. Here we will extract sequences whose partial descriptions (based on regular expression) match with partial descriptions we provide as a list. For ‘List of Descriptions’ box, enter ‘inorganic pyrophosphatase,,potassium transporter’. Note that partial descriptions are separated by double comma and no extra space in the end of partial descriptions.



Click ‘Filter now’ after filling in the required fields as shown in above figure. When the job is done, files will be written in to the selected output folder. The output file should have 326 sequences written in to it.

# 6. Fasta Description/ID Extractor:

This tool enables you to extract descriptions or identifiers, complete or partial, from sequence headers in a fasta file. See [Appendix A](#_Appendix_A:_Understanding) to learn more about fasta sequence description and identifier. You may choose between **1. Description Extractor** and **2. Identifier Extractor.**  Here, we explain only the features of ‘Description Extractor’ as ‘Identifier Extractor’ has the same features except that the latter will extract sequence identifiers.

**FASTA file:**

This must be a fasta file. Number of sequences in it are not limited. Of course, sequences must have descriptions in their header.

**Output folder:**

This indicates where output files will be saved. By default, ResCon will select to save output files in to a folder called ‘Output’ where ‘FASTA file’ is located. You may choose to change this to a different folder, though.

**Extract Partial Description:**

Check this box if you need to extract only certain part of the sequence descriptions from fasta sequence headers. Assuming you have certain regularity in the structure of descriptions of all sequences present in your fasta file, you may use regular expression so as to extract only the partial descriptions.

**Regular Expression:**

This field should contain the regular expression that matches to your intended part of sequence descriptions. By default, the tool uses ‘(.+) \[.+’ as the regular expression. This means that the part of sequence description from its beginning up to character before space followed by a open square brace will be extracted. For example, if sequence description is

inorganic pyrophosphatase [Escherichia coli]

then use of default regular expression will result in

inorganic pyrophosphatase

While regular expression seems intimidating at first, trust us it does not take long to learn using it. This web site could be a good start- <http://www.proftpd.org/docs/howto/Regex.html>. You may use any text editor, which allows use of regular expressions or wild card as part of their inbuilt search (ctrl + F) feature, to test and practice them. For example, notepad++ (not notepad), text wrangler, gedit, Microsoft word, etc. have this feature.

**Note for ‘Identifier Extractor’:**

The default regular expression used in Identifier Extractor is ‘(\w+)\|.+’. This will extract the part of the identifier up to the character before the first pipe ‘|’ character. For example, for sequence identifier

WP\_123890|Bacteria|Escherichia|coli

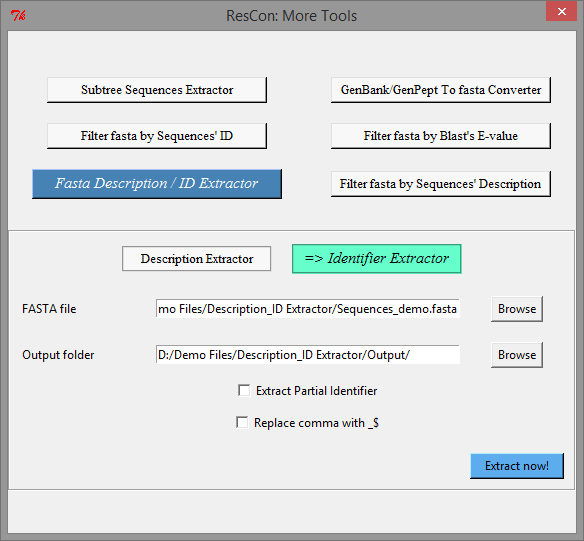
the part that will be extracted is ‘WP\_123890’.

**Replace comma with \_$:**

Check this box if you would like to replace comma in sequence descriptions with symbols ‘\_$’. This will make your life easier if descriptions have comma symbol in them but you do not want them to be split in multiple cells in output csv (comma separated values) file. If needed, you may later replace \_$ in the output file with a comma symbol using ‘find and replace’ feature in the spreadsheet software.

## Demo 1 – Extracting complete Identifiers:

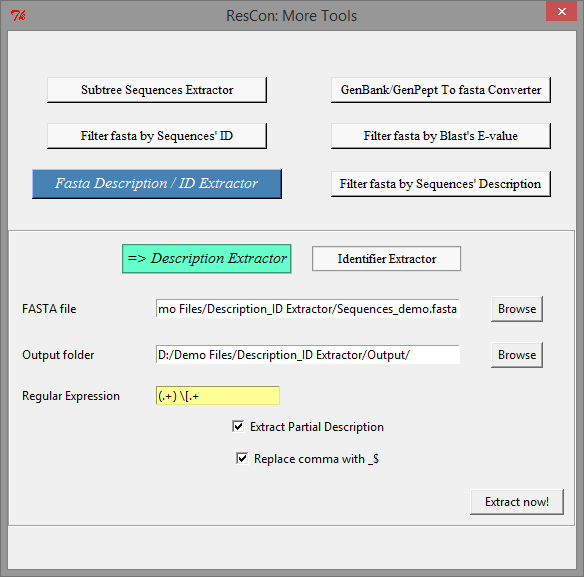
Use demo file from folder ‘Demo Files\6. Description\_ID Extractor\’ to get a feel of how the tool ‘ResCon – Fasta Description/ID Extractor’ works. Here, we will extract complete identifiers from a fasta file.



Click ‘Extract now’ after filling in the required fields as shown in above figure. When the job is done, files will be written in to the selected output folder. The output csv file will show the list of complete identifiers and will also show their count.

## Demo 2 – Extracting Partial Descriptions:

Use demo file from folder ‘Demo Files\6. Description\_ID Extractor\’ to get a feel of how the tool ‘ResCon – Fasta Description / ID Extractor’ works. Here, we will extract partial descriptions from a fasta file by using the default regular expression. Also, we will choose to replace commas in the description with ‘\_$’.



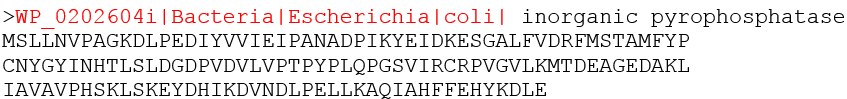
Click ‘Extract now’ after filling in the required fields as shown in above figure. When the job is done, files will be written in to the selected output folder. The output csv file will show the list of partial descriptions and will also show their count.

# Appendix A: Understanding fasta format

Complete understanding of fasta format will enable easy understanding and proper use of ResCon. Fasta sequence has two sections: a header line and one or multiple lines of sequence data. More than one sequence records may be present in fasta file.

*Description*

*Identifier*



First line is the header line and it has three parts.

1. First character in header line is always “>” (greater than symbol). This signifies the beginning of a new sequence record. This symbol “>” should always be present for each sequence record for a valid fasta format.
2. The section that follows “>” until the first empty empty space is called as *identifier* or *ID*. Obviously identifier cannot have empty space in them. In the figure above, red highlighted part is the identifier. This part is optional in fasta format. However, this is the part showing id of sequences and hence *identifier* is necessary to use ResCon.

Certain programs may limit the length of identifier to certain number of characters. For example, Clustal omega allows up to 127 characters and hence ResCon recommends maximum length of identifier to 127.

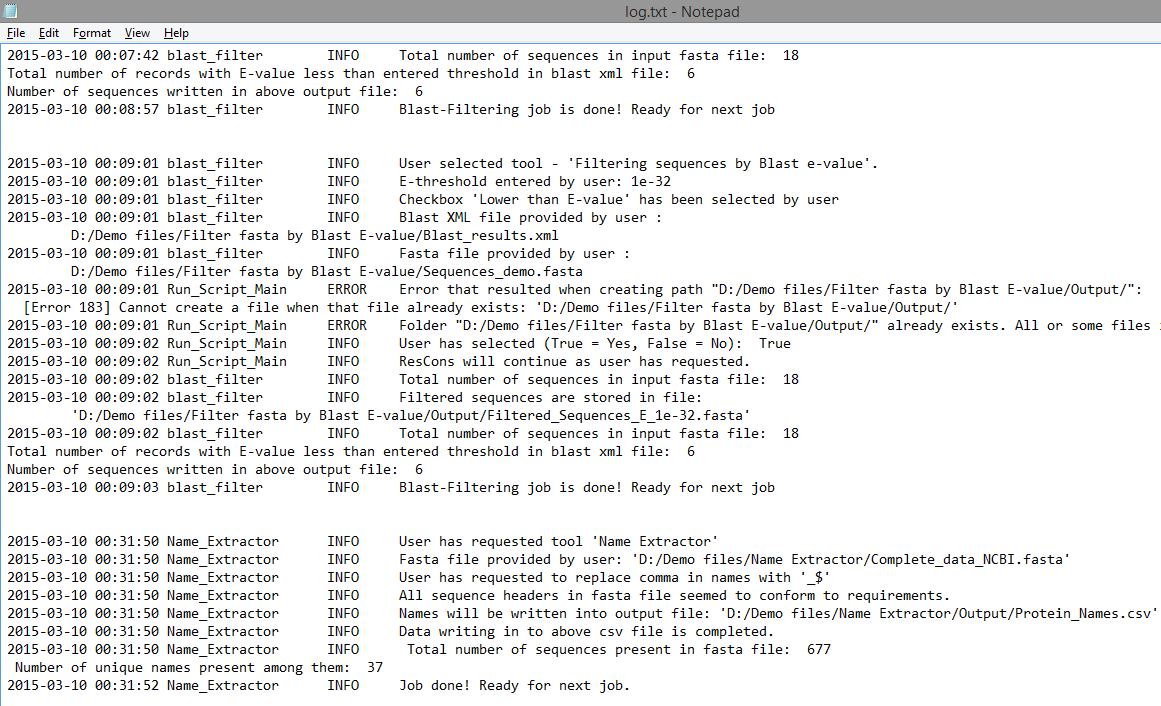
1. Rest of the header that follows identifier is called *description*. This part is optional and space character is allowed here.

# Appendix B: Log file

ResCon writes log files along with output files and they serve two purposes:

1) It serves as a notebook (history) storing all the input details you have used.

2) It tracks the processes of ResCon and also stores any error that may come up.



In the screenshot above showing a log file, you will notice multiple blocks and each block corresponds to a process request. Final block of data in the log file corresponds to the most recent process request. *So, in short, to find logs for a particular process request, open log file from corresponding output folder, scroll all the way down and the final block of logs is what you are looking for.*

If you happen to run into any error (hopefully you will not!) using ResCon, you will want to see ‘native log file’ to know the nature of error. Location of this native log file in your computer differs depending on your operating system. See the instructions below to find this log file:

*For windows:*

Located at ‘C:\temp’ under title ‘Logs\_ResCon.txt’. If your computer does not have C drive, look for temp folder in any other drive it has.

*For Mac:*

Right click on ResCon.app and click ‘Show Package Contents’. You will find log file titled “Logs\_ResCon.txt” under ‘Contents\Resources\’.

*For Ubuntu:*

Log file titled “Logs\_ResCon.txt” is present in the same folder that contains ResCon executable program.

# Appendix C: Clustal Omega Parameters







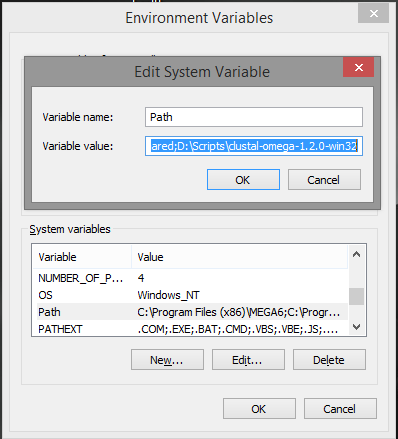


# Appendix D: Installing Clustal Omega

Clustal omega is available for download from <http://www.clustal.org/omega/>. Here we provide basic guide on how to get Clustal Omega working on your computer. If this doesn’t work for you, you may have to refer to installation instructions available from Clustal Omega website.

## For Windows:

Download precompiled binary file for Windows from above link and then extract and save it anywhere in your computer. Next you have to set PATH so that Windows will always recognize where Clustal Omega is installed (stored). Follow the instructions below to set path (you will need administrator password). If this doesn’t work for your windows version, google ‘how to set path in windows (your version)’.



* 1. Open ‘My Computer’ or ‘This PC’. Right click on an empty space and click ‘Properties’.
  2. Click ‘Advanced system settings’ and then ‘Environment variables’.
  3. Select ‘Path’ under System variables.
  4. Now click ‘Edit’ and add file path of the Clustal omega folder as shown in the image here. Use semi-colon to separate it from other file paths.
  5. That’s it. Now open command prompt, type ‘clustalo’ and press Enter. It should say something like this - ‘FATAL: No sequence input was provided’. This means installation has been successful.

## For UNIX based OS (Mac & Ubuntu):

You may download and use pre-compiled standalone binary file available for your OS or compile from the source code available from <http://www.clustal.org/omega/>. Either way, chances are you need to install ‘argtable2’ to install Clustal omega successfully. Follow these steps:

1. Download ‘argtable2’ source code (file name: argtable2-13.tar.gz, at the time of writing) from <http://argtable.sourceforge.net/>.
2. Extract or decompress the downloaded file.
3. Open ‘Terminal’ and execute these commands.

cd argtable2-13 (this is to take you inside the extracted argtable2 folder. If this doesn’t work, type ‘cd ’ and drag and drop extracted argtable2 folder into the terminal)

./configure

make

make check (this step is optional)

sudo make install (you will need password)

make clean

1. Now ‘argtable2’ is installed and next is to install Clustal Omega. Follow these steps to install using its source code.
2. Download file ‘Source code .tar.gz (1.2.1)’ from <http://www.clustal.org/omega/>. and then extract it.
3. Open Terminal and change directory to extracted folder using ‘cd’ command. (See step 3 for more details on this).
4. Once you are inside the extracted folder (or directory), execute these commands:

./configure

make

sudo make install (you will need password)

make clean

1. Now open a new Terminal, type ‘clustalo’ and press Enter. It should say something like this - ‘FATAL: No sequence input was provided’. This means installation has been successful.

If you rather install using precompiled standalone library, download the file for your OS under section called precompiled binaries from this link - <http://www.clustal.org/omega/#Download>. Move this downloaded file to location of your choice and then rename it to ‘clustalo’. Next, use following command in Terminal to make it executable:

chmod u+x clustalo

Above command assumes you are already in the directory/folder that contains clustalo file. If you don’t know how to do this, type ‘chmod u+x ’ in the terminal and then drag and drop ‘clustalo’ file to that terminal and then press Enter. Installation should be successful now. Follow step 8 to verify Clustal Omega behaves properly.

**Alternate method for Ubuntu:**

Since Clustal Omega is available through Ubuntu’s Advanced Packaging Tool (APT), it might be just easy to install it using apt-get. This has added advantage of installing dependancies (including argtable2) automatically. In such case, use the following command:

sudo apt-get install clustalo

**முற்றும்**

(The End)