



FasParser Manual

[For version [2.3.0](#)]



Sun Yan-Bo (edit at 2018-01-31)

Kunming Institute of Zoology, Chinese Academy of Science

Contents

Introduction.....	3
Background.....	3
Citations	3
Installation	4
Package Usage.....	5
[Select multiple files]	5
[Align].....	5
[Trim].....	6
[Sim].....	7
[Format].....	7
[Merge].....	8
Concatenate with same ID	8
Combine FASTA files together (append).....	9
[Sort].....	9
Sort Sequences	9
Rename sequences.....	10
[Filter]	11
Extraction & Filtration based on IDs.....	11
Extraction based on keywords.....	11
Extract 3-site columns for Codon sequences	12
Filtration of columns based on Gap frequency.....	13
Filtration of sequences based on sequence length	13
Filtration based on similarity with a reference sequence	14
[DNA2AA]	15
[AA2DNA]	15
[ORF]	16
[2Seq].....	17
[2Align]	17
[Primer].....	18

[PAML]	19
[GeneDrawer]	19
Contact.....	20

Introduction

Background

With the development of sequencing technology in recent times, a great number of molecular sequences (DNA and RNA) have been generated. Molecular analyses based on these sequences have become one of the most important measures for assessing their potential biological significance. The increase in the amount of available sequence data has made its manipulation tricky, especially for those without programming experience. Hence, it has now become necessary to develop one or more user-friendly software to perform such analyses in a [batch mode](#).

Common manipulations may include extracting/removing subsequences or subalignments, estimating the sequence similarity and identifying the non-homologous sequence(s), converting between file formats, and/or alignment trimming to exclude poorly aligned regions from a multiple sequence alignment. While, with the increase in the amount of available sequence data, it made sequence manipulation tricky, especially for those without programming experience.

To achieve the objective of simplifying many common tasks in sequence manipulation for biologists, we provide a new program package named '[FasParser](#)' for manipulating sequence files. It is designed with a user-friendly GUI and batch processing modes, which allows users to handle multiple sequence files in a simple way. Presently, the main functions of FasParser2 involve: (1) batch performing alignment construction with Muscle (Edgar, 2004), Mafft (Katoh, et al., 2002), or Prank (Loytynoja and Goldman, 2005), (2) alignment trimming on poorly-aligned regions, (3) alignment format conversion between FASTA and either PHYLIP, PAML, or NEXUS, (4) sorting & classifying sequences according to accession numbers or sequence length, which can also rename sequences, (5) concatenating & merging sequences for a particular set of samples from multiple sequence files, (6) translation and ORF prediction, (7) extracting and filtering sequences according to ID, sequence length, or sequence similarity, (8) detecting positive selection with CodeML (Yang, 2007), (9) designing PCR primers with Primer3 (Rozen and Skaletsky, 2000), and (10) extracting consensus-aligned regions between different aligner results (for a same gene). Moreover, there has been designed a specialized [Editor](#) for FasParser (v2.0+) to easily viewing and editing sequences.

Citations

If you use this package in your work, please cite these references:

1. Yan-Bo Sun. *FasParser: a package for manipulating sequence data*, *Zoological*

Research 38(2): 110-112, 2017

2. Yan-Bo Sun. *FasParser2: A Graphical Platform for Batch Manipulating Biological Sequence* unpublished.

Below is a non-exhaustive list of publications related to the programs integrated:

1. Edgar RC. 2004. *MUSCLE: a multiple sequence alignment method with reduced time and space complexity*. *BMC Bioinformatics*, 5: 113.
2. Katoh K, Misawa K, Kuma K, Miyata T. 2002. *MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform*. *Nucleic Acids Research*, 30: 3059-3066.
3. Loytynoja A & Goldman N. 2005. *An algorithm for progressive multiple alignment of sequences with insertions*. *Proceedings of the National Academy of Sciences of the United States of America*, 102: 10557-10562.
4. Nei M & Gojobori T. 1986. *Simple methods for estimating the numbers of synonymous and nonsynonymous nucleotide substitutions*. *Molecular Biology and Evolution*, 3: 418-426.
5. Untergasser A, et al. (2012) *Primer3--new capabilities and interfaces*. *Nucleic Acids Res* 40(15):e115.
6. Yang Z (2007) *PAML 4: phylogenetic analysis by maximum likelihood*. *Molecular Biology and Evolution* 24(8):1586-1591.

Installation

The 'FasParser' has been developed into a standalone **Windows System Application** (compiled and tested on Windows 7/10). It can run on most Windows systems with no installation of other programs.

Download the **newest setup program** (i.e. 'FasParserX.X_setup.exe') from <https://github.com/Sun-Yanbo/FasParser> to your disk, and then double click it to install the whole package. Normally, it is well using the default installation parameters (by clicking the Next button to end). After successful installation, you would get a screen of the Home page of this package (**Figure 1**).

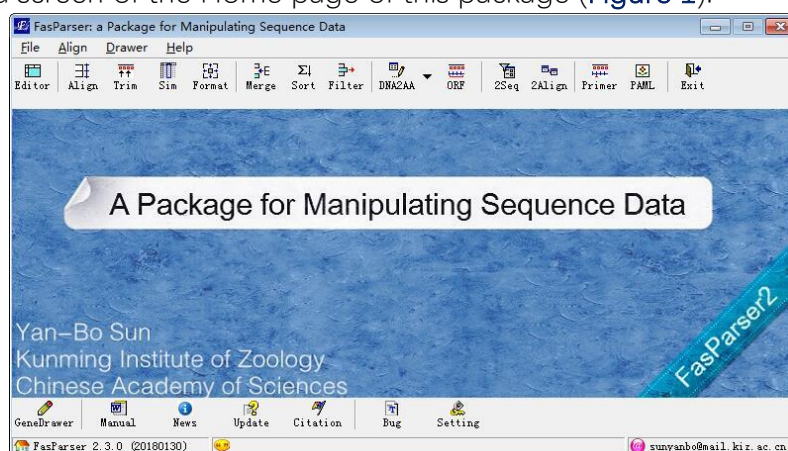


Figure 1. The Home page of the FasParser package.

Package Usage

[Select multiple files]

There are [3 different ways to select multiple files](#) into the checking list box:

1. Using the button “upload”;
2. Drag the files into the checking list box;
3. Drag the folder which contains the sequence files into the checking list box;

[Align]

This program is designed to **construct alignments for multiple FASTA files**. There are 3 aligners ([MUSCLE](#), [MAFFT](#), and [PRANK](#), see below reference) have been integrated into FasParser. The first two aligners are always faster than Prank, but Prank could generate more accurate results. In addition, the first two aligners can auto recognize the input type (nuclear DNA or protein sequence), while, the Prank should be told the sequence type; user should select the corresponding aligner by yourself (including [Prank-DNA](#), [Prank-Codon](#), and [Prank-AA](#), [Figure 2](#)). The running process information can be viewed in the bottom text box. User should use ‘*Scan*’ button to select the Fasta files to analyze. The output files will be saved in the folder which contains the input files, with ‘*. <aligner>.fas’. Please note all the default parameters of the aligners are used by FasParser.

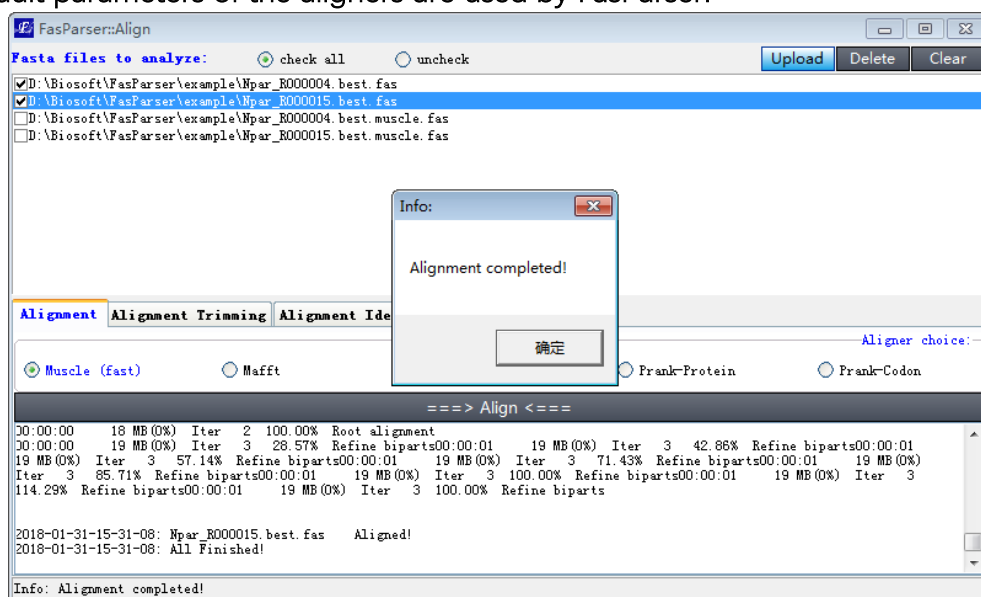


Figure 2. Overview of alignment construction

In versions $\geq 2.3.0$, the aligned files will be displayed in the checking list box and can be viewed by double clicks.

[Trim]

Alignment quality is of great importance for downstream analyses, like phylogenetic inference and positive selection detection. The low-quality alignment always produces false positive results, like wrong phylogenetic positions of species. So, it is much better to strip the poorly-aligned regions in the raw alignments before conducting downstream evolutionary analyses. The 'Trim' function provides a simple and efficient trimming method to filter the low-quality regions in each alignment. [\[It is more suitable for preparing high-quality alignments for positive selection detection\]](#)

'Trim' will apply a sliding window method to identify which regions hold low-similarity sequences based on a dynamic cutoff that will be estimated from the total alignment (1000 times). There are two steps to trim an alignment. The 1st step is for the gap-near regions, and the window will be sliding from the gap "-" to both left and right ends. The 2nd step is for the block regions. After the 1st step trimming, the 2nd will trim again from the left end to right.

User should tell which type of seqs you provided: [codon](#) or [DNA](#), and then you can click the "Start" button to start the trimming analyses for multiple files ([Figure 3](#)). The output files will be saved in the folder which contains the input files, with ['*.trimC.fas'](#) or ['*.trimD.fas'](#) for codon or DNA, respectively.

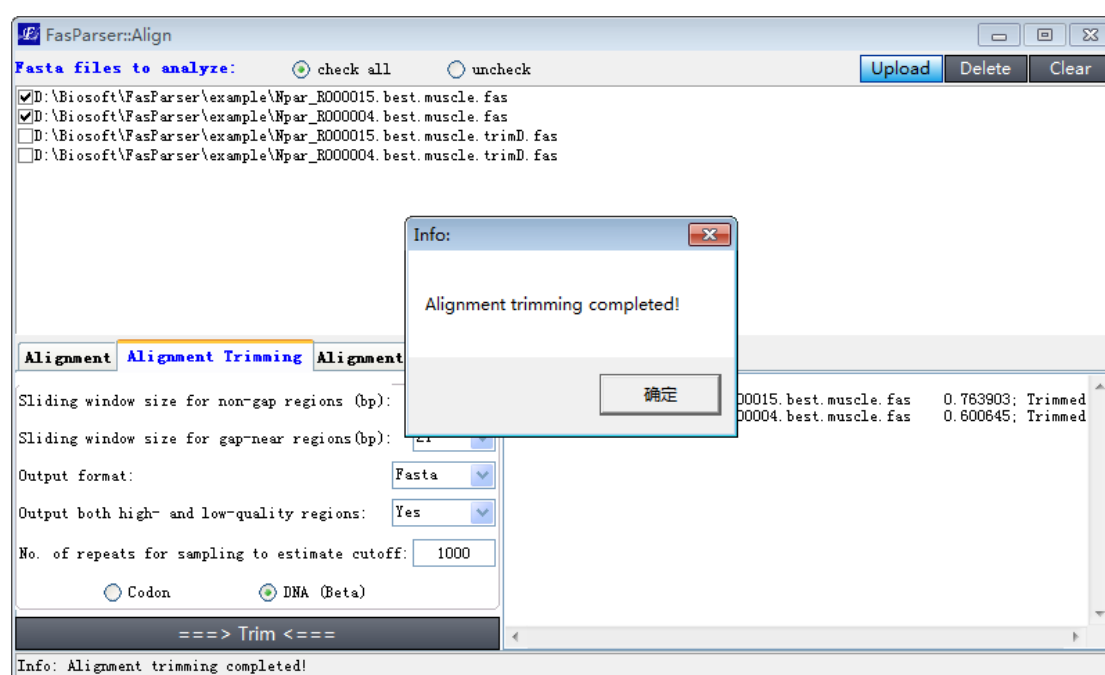


Figure 3. Overview of alignment trimming.

[Sim]

It is important to know the conservation level for a gene or a segment among different species (always refer to the species you studied). If the mean identity of an alignment is too low, it might be due to rapid evolution of this gene or non-homologous prediction for one or more species in this alignment. For each alignment, 'Sim' (Similarity) provides three identity calculations: the minimum pair-wise identity within the alignment ([min.](#)); the maximum pair-wise identity ([max.](#)); and the mean identity ([mean](#)).

Similarity, user should select files to analyze through clicking the '*Scan*' button, and then click the "*Start*" button to start the estimates ([Figure 4](#)).

Please note that during the similarity estimates, the gaps are also taken into account according to "Score = Score -1" if one base match a gap "-".

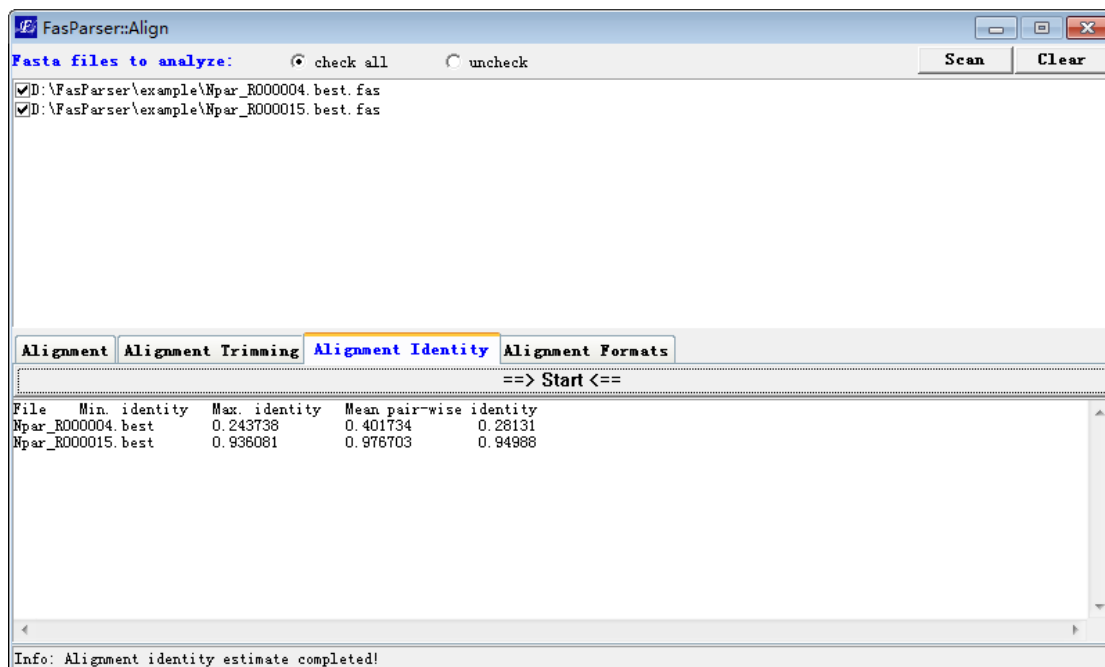


Figure 4. Overview of alignment similarity estimates.

[Format]

This function is used to convert the FASTA files to other formatted ones. Presently, there are 4 different formats available: FASTA, PHYLIP (Phylipi and Phylips), PAML, and NEXUS ([Figure 5](#)). The output files will be saved in the folder which contains the input files, with '**. <format>.fas*'. [The result files are also displayed in the window and can be viewed by double clicks.](#)

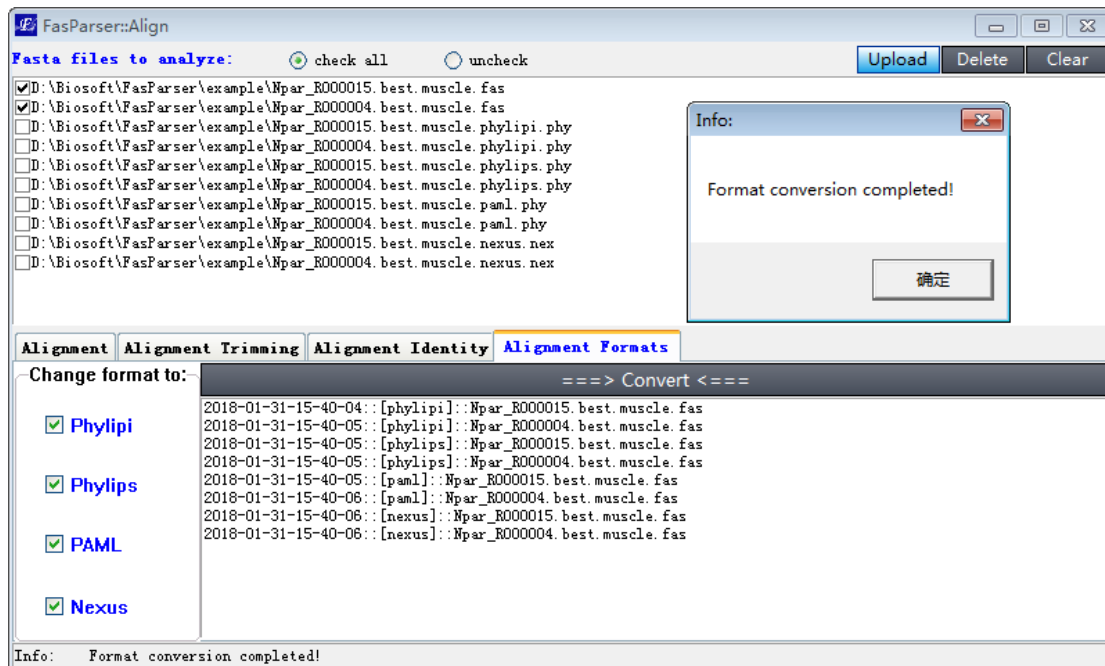


Figure 5. Overview of alignment format conversion.

[Merge]

Concatenate with same ID

This program is used to 'concatenate' sequences of the same IDs from multiple FASTA files. It is much useful in phylogenetic or other analyses, especially when users generated multiple loci sequences for a set of samples and want to generate a "super" sequence by concatenating all the loci sequences together for each sample.

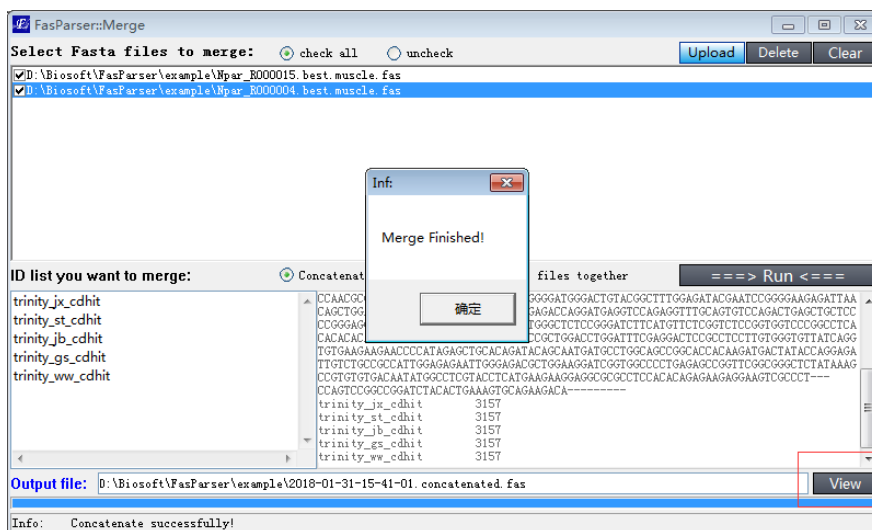


Figure 6.1. Overview of sequence concatenation manipulation.

Please note that for **concatenate**, if the raw FASTA files are not aligned before,

the final concatenated FASTA file should be aligned first before conducting some other analyses. (We recommend user to construct alignment for each file before, and each file must have a same ID list).

To start this analysis, user should define an output file name and its location (can be automated created by FasParser). User can define which IDs will be concatenated by editing the ID list in the textbox (left-below, [Figure 6.1](#)). The [View](#) button can perform the user default execute program to open the result file.

Combine FASTA files together (append)

In addition to the concatenate, user may just want to 'combine' all the selected files together. It is easy to achieve this by using the Merge function in FasParser ([Figure 6.2](#)):

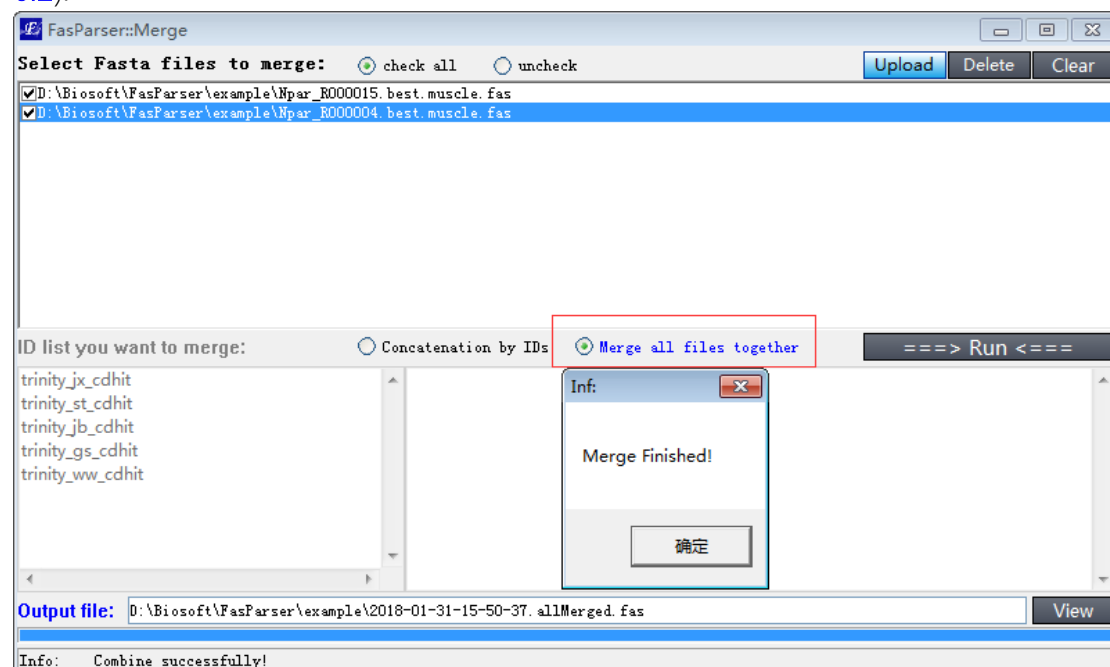


Figure 6.2. Overview of sequence combine manipulation

[Sort]

Sort Sequences

This program will allow users to sort their FASTA files according to either the **ID names**, **sequence lengths**, or a **provided list of IDs**. Part of this function (with the ID list provided by users) is much like the [Extraction](#) analysis in [Filter] module.

Please note that the ID is automatically recognized from the raw ID, which is [the first continuous string before symbols space, ".", and "|](#)". For example, if the raw ID

in a FASTA file is:

```
`>Uma_R000001.2 locus=scaffold79:384179:406202:-'
```

You can use the ID 'Uma_R000001.2' to search its sequence. Sometimes, the ID 'Uma_R000001' is also ok, if there is only one targeted ID. If the provided IDs cannot be recognized, there will be no sequence reported. Another example. You can use 'gi|947195581' to represent the sequence ID :

```
`>gi|947195581|ref|XP_006139827.2| PREDICTED...'
```

The output files will be saved in the folder which contains the input files, with '*.sort.1.fas' for alphabetically, '*.sort.2.1.fas' and '*.sort.2.2.fas' for length-descend and length-ascend, respectively.

Rename sequences

With the provided ID list (left-below textbox), the program can also rename the raw sequence IDs. To achieve that, the ID list should have two columns, the first column refers to the raw ID, and the second column refers to the new ID, the separator between the two columns must be "=", as below:

```
Uma_R000001.2=>R000001.2
Uma_R000003.2
Uma_R000002.2=>R000002
...
```

Under the above query IDs, program will use R000001.2 to replace Uma_R000001.2, and use R000002 to replace Uma_R000002.2. If there is only one column, the raw IDs will be retained (Figure 7). The output files will be saved in the folder which contains the input files, with '*.sort.3.fas'.

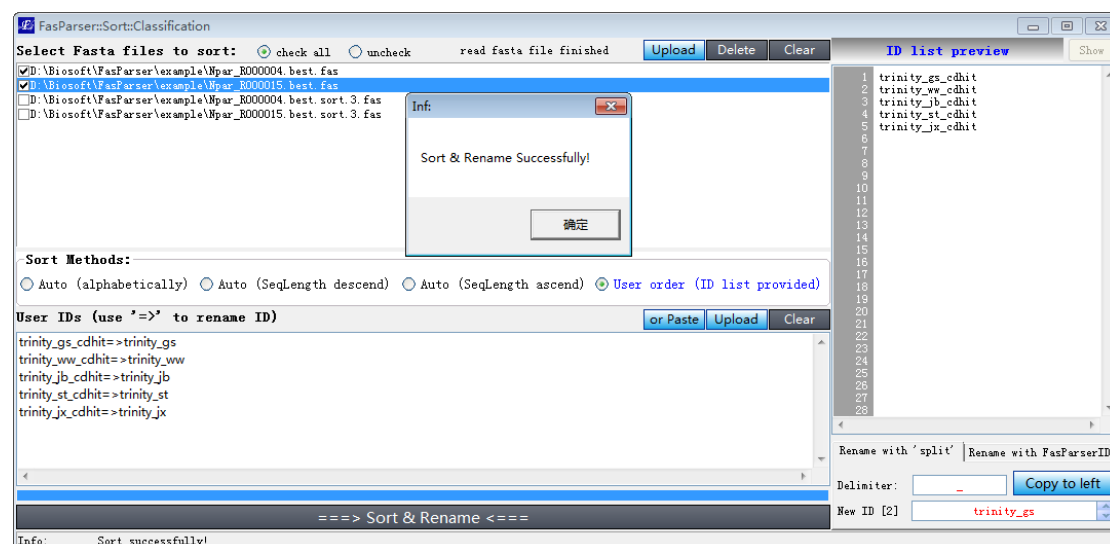


Figure 7. Overview of Sort & Rename sequences.

[Filter]

This program is designed to perform some [extraction and filtration analyses](#) within a particular FASTA file. It is a much common sequence manipulation. With this program, users can [extract or remove a set of sequences](#) from the raw FASTA file based on query IDs, the positions per codon, and can also filter the sequences based on the sequence length, or similarity with a provided reference sequence.

Extraction & Filtration based on IDs

Based on this function, user can extract or remove some sequences from one or more FASTA files. Just select the files you want to analyze, and then input the IDs (you want to query) in the left-below textbox. [Select the appropriated operation: *extract* or *remove*, whether or not with reverse complement operation](#), and then click the “*Run*” button ([Figure. 8.1](#)). The output files will be saved in the folder which contains the input files, with ‘**.extract.fas*’ or ‘**.removed.fas*’.

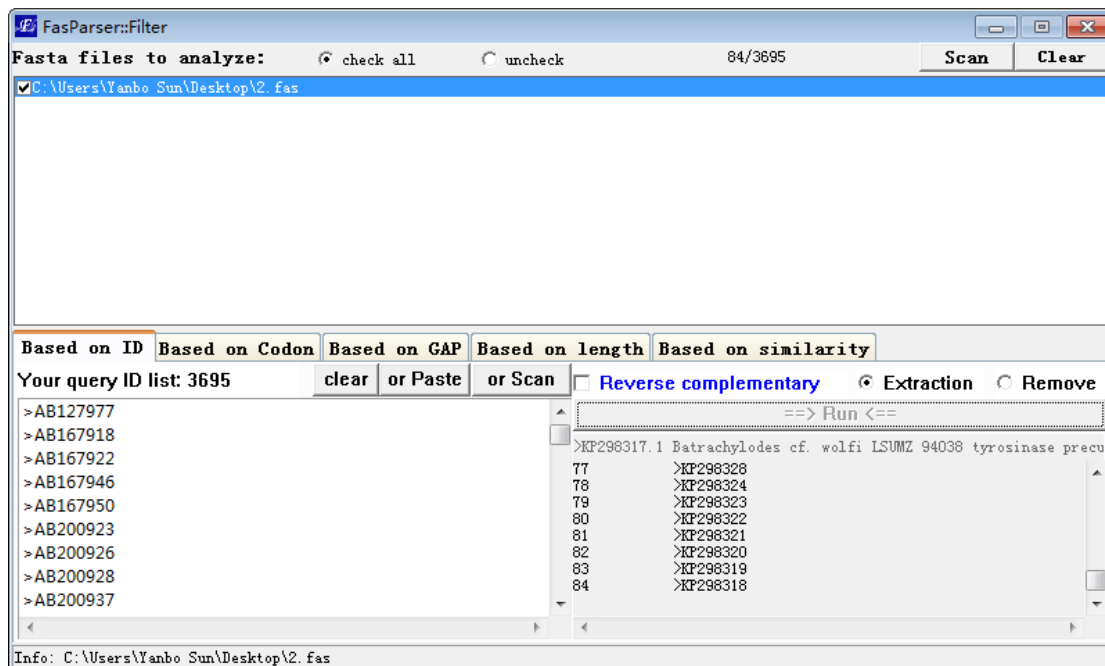


Figure 8.1. Overview of extraction & filtration based on query IDs.

Extraction based on keywords

In addition, this section also provides a [Classification](#) function, which means that you can use [one or more keywords to extract sequences](#) from a raw FASTA file and save them into separate FASTA files. The keywords can be the genus name, species name, or some other words, which should be part of the raw IDs ([Figure](#)

7.2).

The user should input the **keywords** in the right-below textbox, and then click the “*Classification*” button to run this analysis. The output files will be saved in the folder which contains the input files, with ‘*.class.[keyword].fas’.

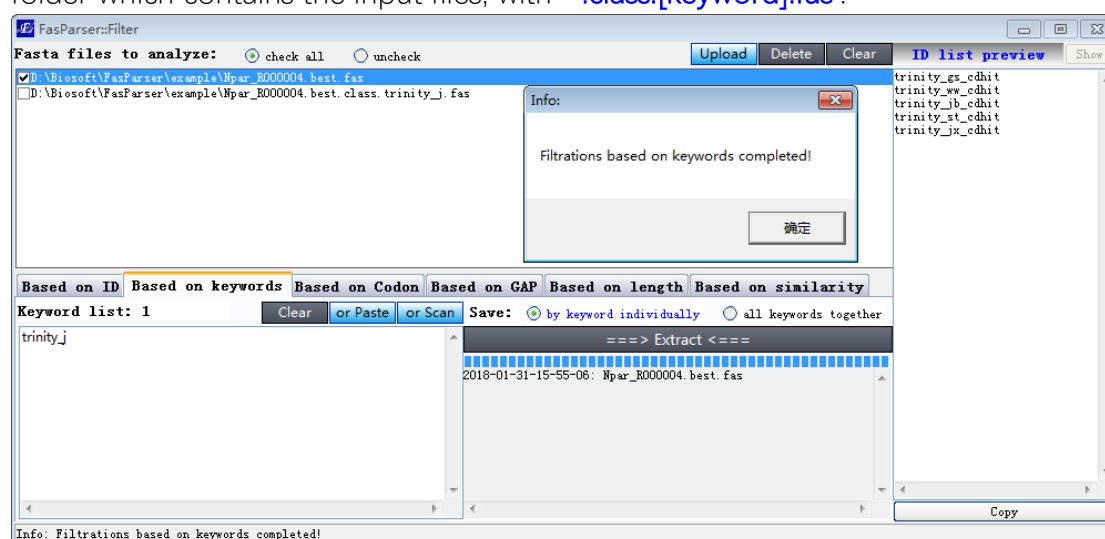


Figure 7.2. Overview of extraction & filtration based on ID keywords.

Extract 3-site columns for Codon sequences

Based on this function, user can extract each-site columns of codon sequences for multiple FASTA files. Just select the files you want to analyze, and then click the “*Run*” button (Figure. 8.2). The output files will be saved in the folder which contains the input files, with ‘*.codon1.fas’, ‘*.codon2.fas’, and ‘*.codon3.fas’.

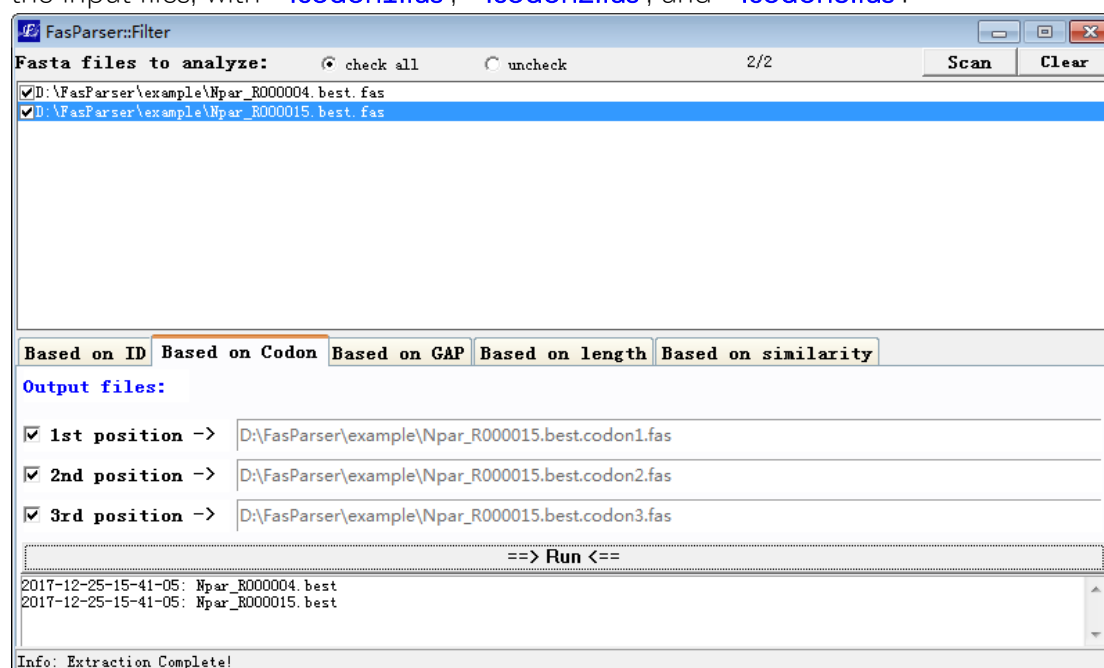


Figure 8.2. Overview of extraction of site columns for codon sequences.

Filtration of columns based on Gap frequency

This is a common function to cut the raw alignment by deleting columns with many gaps ('-'). This function can also fill up the missing data (with 'N') at the beginning and ending of an alignment, which is useful for some phylogenetic analyses. To perform this analysis, user should provide a gap-frequency [cutoff value](#), and then select appropriated operation, like analyzing which region, and how to deal with the columns with a lower value (than the cutoff). Click the '*Run*' button to start this analysis ([Figure 8.3](#)).

The output files will be save in the folder which contains the raw input files, with a name format of "**.cut.end.fas*" or "**.cut.all.fas*".

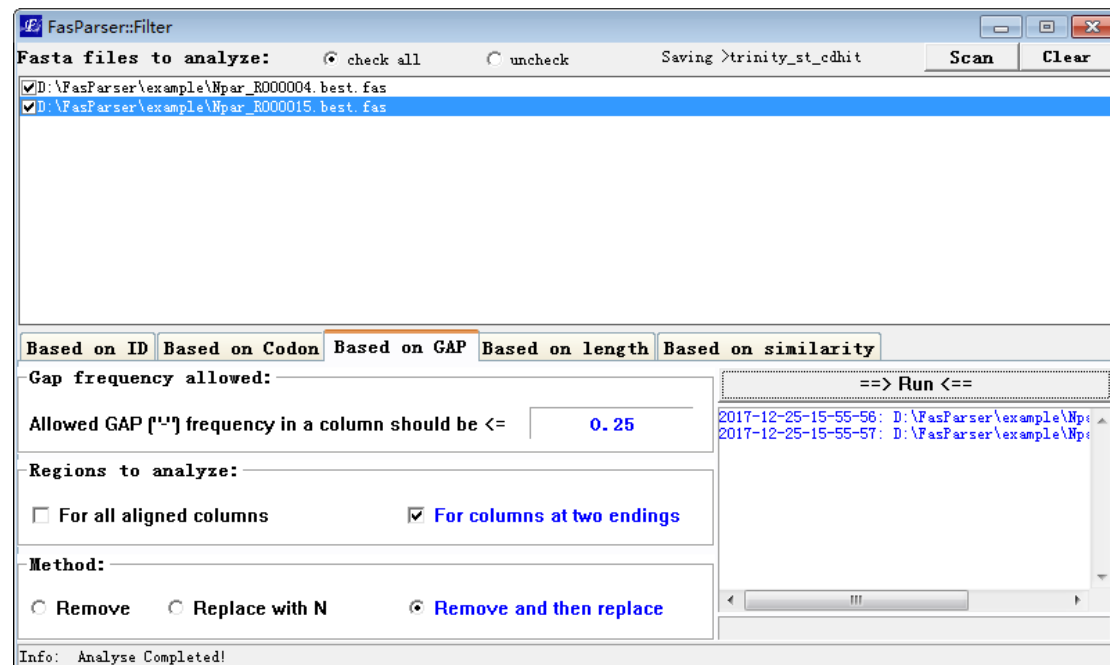


Figure 8.3. Overview of the column-filtration based on Gap frequency.

Filtration of sequences based on sequence length

Maybe there are some too short sequences in a file, which would significantly influence the final available sequence length. So, it is usually useful to identify and remove the too short sequences. To do this, FasParser provides some different methods.

The [1st method is designed for alignment files](#) (slower). The program will auto-estimate the block (no-gap regions) length under removal of each single sequence, and then identify the bad sequences whose removal can increase significantly the final block lengths.

The 2nd method is designed for non-aligned files (faster). This method will identify directly the too short sequences according to their sequence lengths. The formula used is "mean - 3*std" if there are more than 100 sequences. If there are less than 100 sequences, Grubbs method will be used.

Filtration based on similarity with a reference sequence

This function is designed for manipulating multiple sequences, which are always collected from different sources without few sequence checking tasks. There may be some non-homologous sequences in the raw inputted sequences, or the sequences come from different parts of a same gene. Before downstream analyses, it is much better to do some filtration task on these sequences to improve their availability in other analyses.

To do this work, user should provide a reference sequence that is from your previous or other peoples' work to identify the non-homologous sequences and cut the input sequences to a same length based on the alignment result with the reference sequence.

After setting appropriated parameters, just click the "Run" button to run this analysis. There would generate 2 or 4 output files for each input file, which contain the final alignment (cut to a length based on the reference sequence), as well as the homologous identification results (Figure. 8.4).

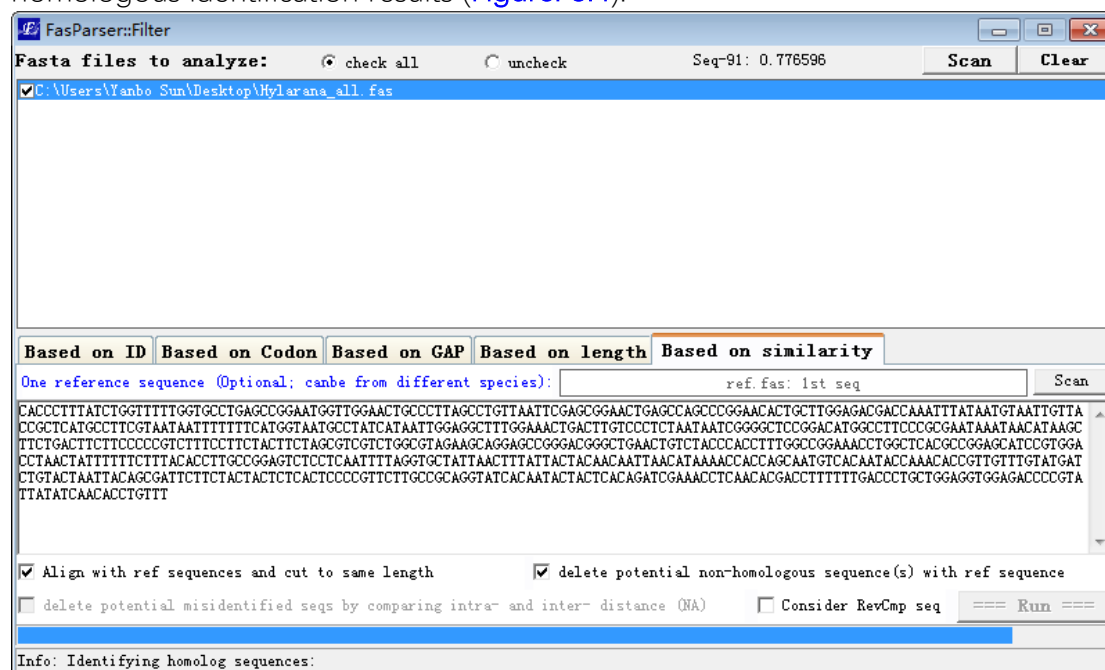
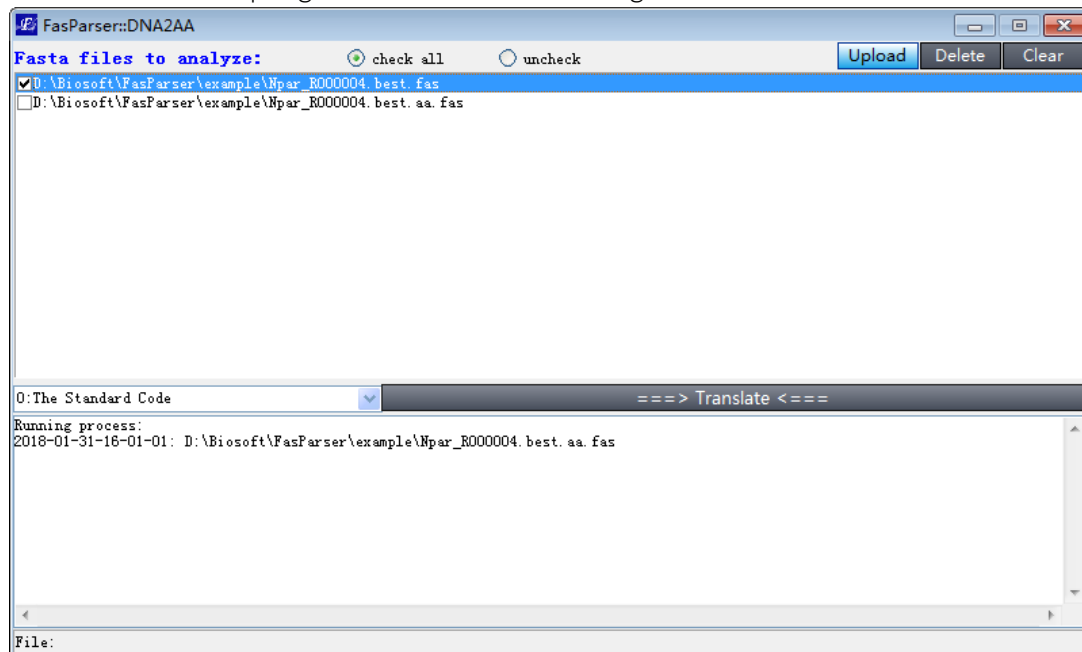


Figure. 8.4. Overview of the sequence filtration based on similarity with a reference sequence.

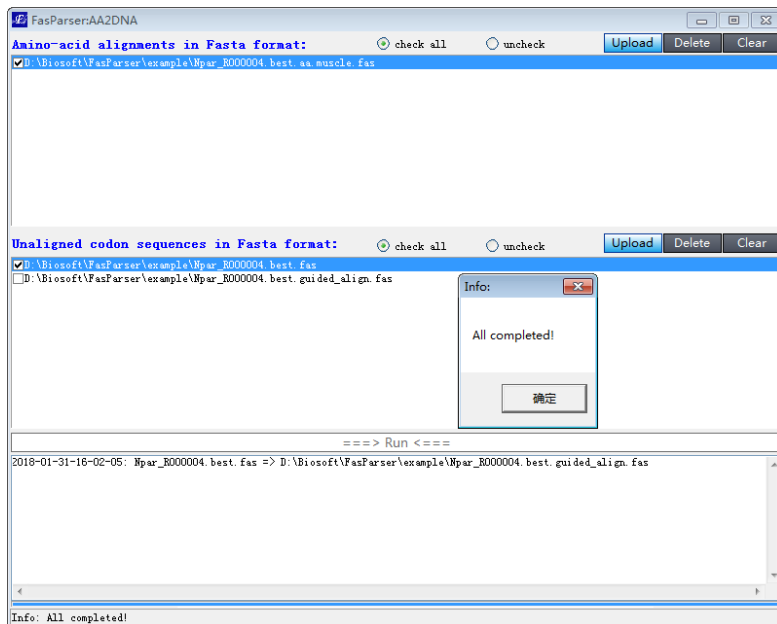
[DNA2AA]

This function is used to [translate the DNA sequences to protein sequences](#). There are two modes available. If users have many FASTA files, you can use **the batch mode** to conduct such analyses, with genetic code selected. The output files will be saved into the folder which contains the raw input files, with a filename as **"*.aa.fas"**. You can also input a pure DNA sequence or a FASTA file under the manual mode, the program can automate recognize them.



[AA2DNA]

This function is used to [generate codon alignments guided by the amino-acid alignments](#). The raw DNA sequence files (unaligned) and the amino-acid alignment files (aligned) are needed for this function. It is always used in transcriptome analyses. For example, you have 1000 genes with 5 species. Each gene has been translated to amino-acid sequences and aligned by Muscle or others. You can use this function to generate codon alignments based on the amino-acid sequences.



[ORF]

This program is designed to **identify the ORF for cDNA sequence(s)**. The cDNA sequences can be organized into a FASTA file or as single sequence. There are 2 choices to obtain the ORFs: a) **get the longest one** if there are present more than one potential ORF in the input cDNA sequence, and b) **get all the potential ORF sequences** (Figure 9).

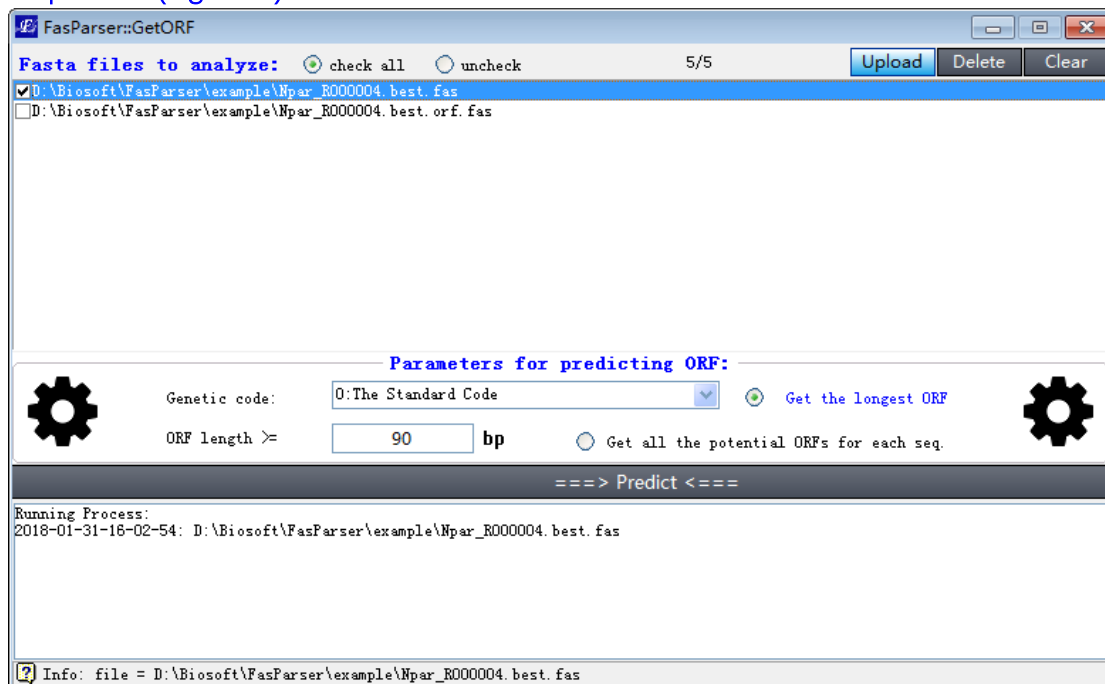


Figure 9. overview of identification of ORFs for sequences.

[2Seq]

The program “2Seq” is designed to easily **count and view differences between two DNA sequences at both DNA and codon levels**. Under the codon level, the program can also estimate the total number of synonymous (S) and non-synonymous (N) sites for the first sequence and then calculate the number of synonymous and non-synonymous substitutions between the two sequences. To do that, users just need to **put two sequences into the two above textboxes** and **click the Run button**. The program could then provide a view the differences between the two sequences in colors and the summary of identified mutations or substitutions (**Figure 10**).

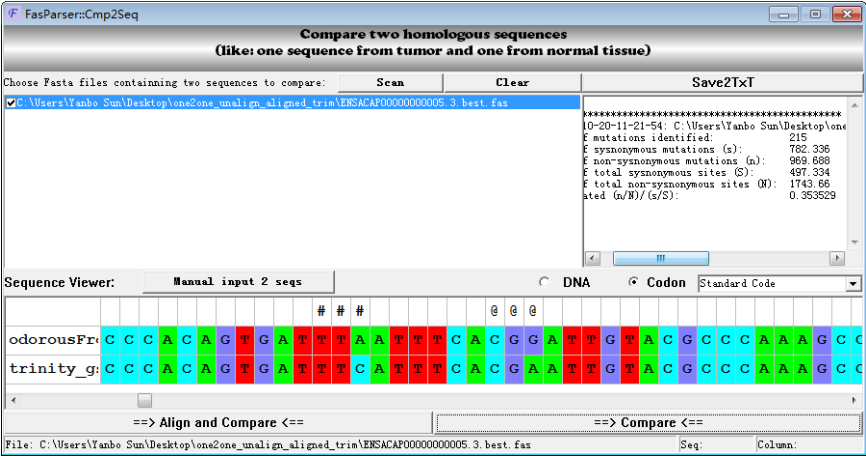


Figure 10. Overview of comparison between two sequences.

[2Align]

This program “2Align” is designed to **compare different alignments of a same gene that could be generated by different aligners**. It is well known that there is almost no current method correctly aligns the entire sequence, and different aligners always correctly align different regions. One simple method is to identify the overlapped regions between different aligner-generated alignments, which might be useful for some other analyses, like phylogenetic reference and/or positive selection detection. This function needs users to **input two alignments through the above scan buttons** and **click the “Run” button** to view their overlaps (**Figure 11**). ***This is the only function currently cannot run at the batch mode.**

[PAML]

FasParser provided a batch mode to run codeml in PAML package (Yang et al. 2007). It is much better for user to read the manual of PAML to understand the parameter settings and add proper symbols to the phylogenetic tree (like below, [Figure 13](#)).

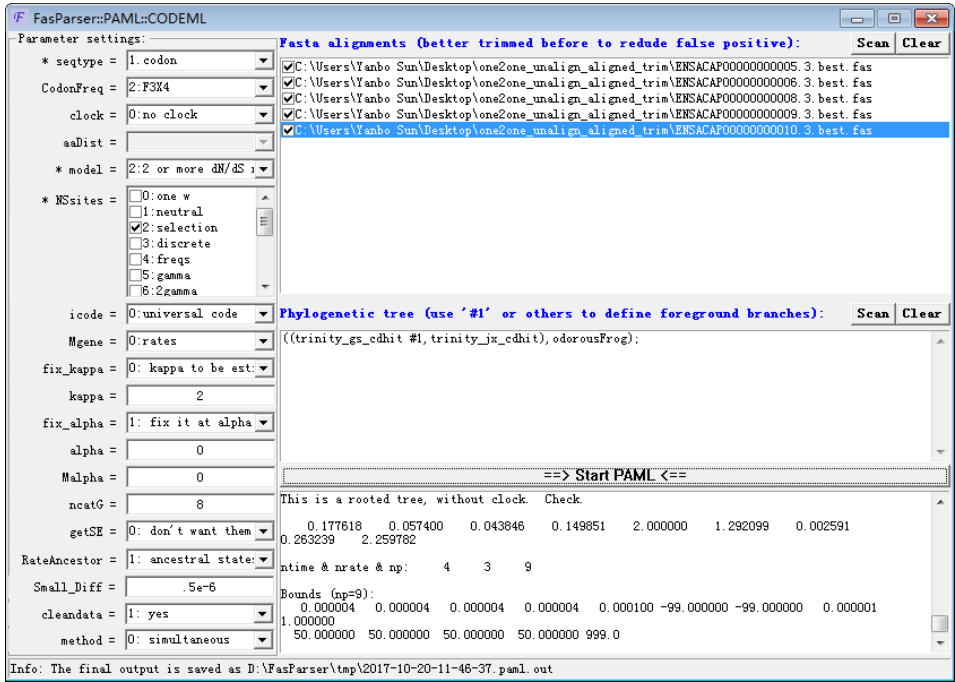


Figure 13. Positive selection detection.

Below is an example of running output:

GeneID	Ln Alternative	Ln Null	Data	Sig.
ENSACAP000000000005.3.best.fas	-4271.375567	-4271.375567	0	
ENSACAP000000000006.3.best.fas	-6538.150389	-6538.150389	0	
ENSACAP000000000008.3.best.fas	-1092.144468	-1092.144468	0	
ENSACAP000000000009.3.best.fas	-2101.001357	-2094.010118	13.9829	P < 0.05
ENSACAP000000000010.3.best.fas	-1165.844923	-1165.844923	0	

[GeneDrawer]

FasParser (since version 2) provides a gene structure drawing fiction. It is very easy to use. It can be run in two different mode, based on the data type of inputs. It now can read either gene annotation or gene sequences ([Figure 14](#)). Presently, the drawer function has litter photo-editing interfaces, which will be modified in the future versions.

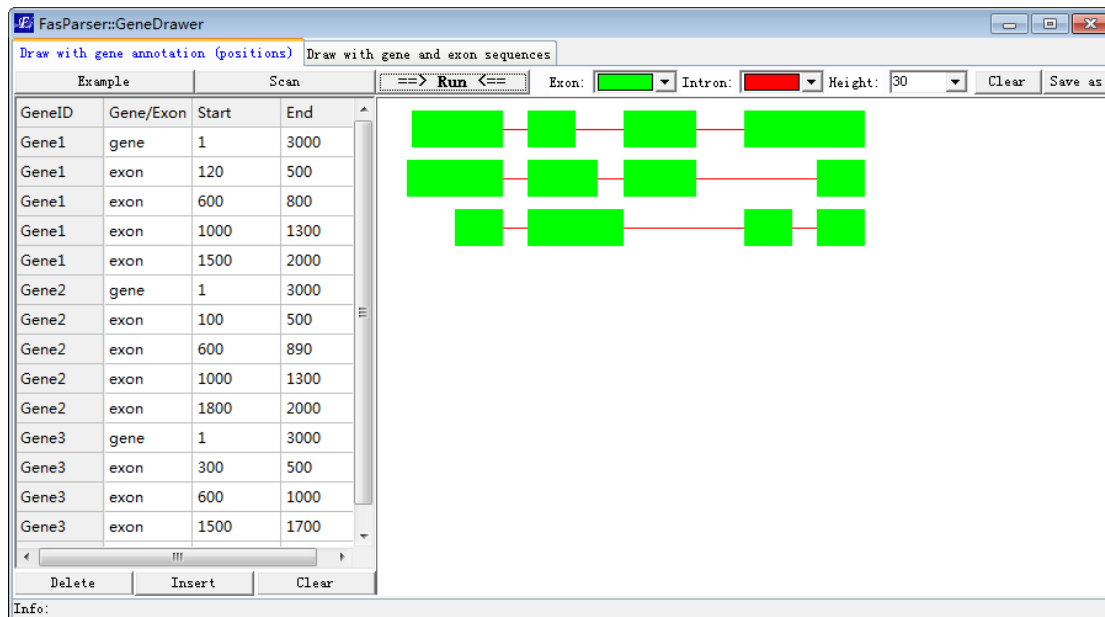


Figure 14. Overview of gene structure drawing function.

Contact

If you have any question, or program bug, please feel free to contact the me (Yan-Bo Sun). My email address is sunyanbo@mail.kiz.ac.cn

FasParser also has a bug report interface ([Bug]), through which you can also contact me and report the errors of this program. I believe FasParser will grow better under your suggestions.